

# **Fach Informatik**

## **Fach- und bildungswissenschaftliche Grundlagen für den Informatikunterricht in der Sekundarstufe I**

Inaugural-Dissertation  
zur Erlangung des Doktorgrades  
der Naturwissenschaften im Fachbereich  
Mathematik und Informatik  
der Mathematisch-Naturwissenschaftlichen Fakultät  
der Westfälischen Wilhelms-Universität Münster

vorgelegt von

**Arno Pasternak**  
aus Sandhorst (Ostfriesland)  
2013

Dekan:	Prof. Dr. Martin Stein
Erster Gutachter:	Prof. Dr. Jan Vahrenhold
Zweiter Gutachter:	Prof. Dr. Carsten Schulte
Tag der mündlichen Prüfung:	17. Mai 2013
Tag der Promotion:	17. Mai 2013

# Vorwort

Im Oktober 1982 habe ich die Ausbildung zum Lehrer mit dem Ende des Referendariats am Studienseminar in Leer/Ostfriesland abgeschlossen und konnte nach dem Sommer 1983 in Hagen an der Fritz-Steinhoff-Gesamtschule als Lehrer für Mathematik, Physik und Informatik unterrichten. In dieser Schule existierte entsprechend der Aufbruchstimmung in den Gesamtschulen ein innovatives Klima, in dem jeder interessierte Kollege seine inhaltlichen Ideen einbringen und teilweise verwirklichen konnte. Dies galt nicht nur für das Schulfach Informatik, sondern für alle Fächer.

Die grosse Aufbruchstimmung aus den damaligen Jahren ist zwar nicht mehr in dieser Intensität vorhanden, in vielen Fächern sind zudem manche Veränderungen von den Kollegen und durch curriculare Vorgaben wieder revidiert worden, aber viele inhaltliche Änderungen haben sich bis heute erhalten. Und immer noch ist ein nicht unerheblicher Teil des Kollegiums an innovativen Prozessen interessiert und beteiligt.

In einer solchen Schule fühlt man sich als an curricularen Prozessen interessierter Kollege wohl. Dass diese curriculare Arbeit für mich wesentlich im relativ neuen Fach Informatik stattfinden konnte und kann, ist nicht weiter verwunderlich. Informatik gab es an der FSG seit den 80er Jahren in der Sekundarstufe I in Form eines Wahlpflicht-II-Angebotes und in der Sekundarstufe II in Form von Grundkursen und später auch Leistungskursen. Regelmässig sind in den letzten 30 Jahren die Curricula diskutiert und weiterentwickelt worden. Diese intensiven Diskussionen vor allem mit den Kollegen *Karl-Heinz Behmer*, *Heinz Ziegeldorf* und später auch mit *Michael Mayer* in der Fachkonferenz Informatik haben diesen Innovationsgeist über viele Jahre erhalten.

Im Jahre 2008 wurden an der *TU Dortmund* von *Jan Vahrenhold* für die im Aufbau begriffene Arbeitsgruppe *Didaktik der Informatik* auch Lehrer gesucht. So konnte ich meine didaktische Arbeit in einer neuen Form weiterführen und dabei auch die eigene pädagogische Arbeit der letzten Jahrzehnte reflektieren und auswerten. Im Laufe der letzten fünf Jahre hat sich eine hervorragende Arbeitsgruppe unter der Leitung von *Jan Vahrenhold* etabliert. Vielfältige Diskussionen mit ihm und den anderen Mitgliedern der Arbeitsgruppe wie *Renate Thies*, *Holger Danielsiek* und *Wolfgang Paul* haben entscheidend zur Weiterentwicklung didaktischer Positionen geführt. Im Rahmen dieser Arbeit wurde die Struktur der *roten Fäden* herausgearbeitet, die dann später auf der Tagung *Informatik und Schule 2009* [PV09] und in den USA auf den Tagungen *SIGCSE 2010* [PV10] und *SIGCSE 2012* [PV12] präsentiert und diskutiert wurden. Diese Arbeiten finden ihren Niederschlag in dieser Dissertation. Diese Arbeitsgruppe stellt die bisher kreativste und intensivste intellektuelle Herausforderung dar, die ich in den vielen Jahren meiner Tätigkeit als Lehrer erlebt habe. Das eine solche Arbeit nur möglich ist, wenn man sich darüber hinaus auch gut persönlich versteht, ist naheliegend. Ich hoffe und gehe davon aus, dass diese Zusammenarbeit im beruflichen und privaten Bereich mit der Promotion nicht beendet ist.

Das in dieser Arbeit beschriebene Unterrichtsprojekt hat weitere vielseitige Unterstützung gefunden. Viele Unterstützer kommen aus der Fritz-Steinhoff-Gesamtschule Hagen. Die Schulleitung unter dem damaligen Schulleiter *Werner Kerski* hat die Idee einer Wahlpflicht-(I)-Gruppe Informatik/Physik sofort positiv aufgenommen und umgesetzt. Dies belegt die bereits erwähnte Innovationsbereitschaft an dieser Schule. Die Kolleginnen und Kollegen haben

ausnahmslos dieses Projekt akzeptiert und unterstützt, auch wenn sicher der eine oder die andere nicht ganz mit meinen Positionen übereinstimmt. Vor allem muss ich ‚meiner‘ langjährigen 1. Klassenlehrerin *Marion Geyer* danken, die in dieser Zeit meine nur reduziert mögliche Tätigkeit als 2. Klassenlehrer aufgefangen hat und nicht unwesentlich an der Idee und dem Zustandekommen des Wahlpflicht-Kurses beteiligt war.

Eine empirische Forschung im Rahmen der Didaktik verlangt Kontroll- oder Vergleichsgruppen. Für diese Arbeit konnte ich Kurse und Klassen an der Hauptschule Vorhalle und Realschule Halden einbeziehen. Die dort unterrichtenden Kollegen haben jederzeit und mit grossem Verständnis meine Untersuchungen unterstützt. Besonders danken muss ich den ‚Kontaktpersonen‘ aus und für diese Schulen *Gorden Storkmann* und *Susanne Ruhkamp* für die gute Zusammenarbeit.

Die Inhalte von Unterrichtseinheiten wurden teilweise in Übungs- und Seminarveranstaltungen mit Dortmunder Lehramtsstudenten vorbereitet, diskutiert und analysiert. Fragebogen mussten erfasst werden, Interviews durchgeführt und transkribiert werden. Neben bereits erwähnten ‚Mitreitern‘ sind hier *Rebecca Doherty*, *Holger Kalinowski*, *Nico Gryzan*, *Andreas Thom*, *Matthias Völker* und *Jörn Gödel* zu nennen. ‚Last, but not least‘ half meine Englischkollegin an der FSG *Ararita Neuhaus*, dass meine Texte auch in englischer Sprache lesbar sind. Dabei hat sie nicht nur in vielen Stunden die Texte mit mir sprachlich diskutiert und optimiert, sondern auch als Lehrerin die Inhalte aus einer Sicht jenseits der Informatik kritisch hinterfragt.

Eine weitere ‚Quelle‘ meiner Arbeit sind neben der Fritz-Steinhoff-Schule und der TU Dortmund die *fachdidaktischen Gespräche zur Informatik in Königstein* in der Sächsischen Schweiz. Dieser Kreis existiert seit 1994 und gab mir seitdem die Möglichkeit, immer an den aktuellen Diskussionen in der Fachdidaktik teilzunehmen. Die in diesem Kreis entwickelten *Bildungsstandards Informatik* sind das beredete Zeugnis der dort geleisteten Arbeit. Es hat mich daher sehr gefreut, dass sich aus diesem Kreis *Carsten Schulte* als zweiter Gutachter für diese Arbeit zur Verfügung gestellt hat. Ebenso hat mich gefreut, dass mit *Maria Knobelsdorf* im WS 2012/2013 ein weiteres Mitglied dieses Arbeitskreises in unserer Arbeitsgruppe an der TU Dortmund in unsere und damit auch meine Arbeit und Diskussionen einbezogen war. Mit *Dieter Engbring* und *Ludger Humbert* aus dem Königsteiner Gesprächskreis verbindet mich eine noch längere Zusammenarbeit fast seit Beginn meiner Lehrertätigkeit in Hagen. Immer wiederkehrende intensive Diskussionen über Themen der Fachdidaktik Informatik haben unter anderem zu mehreren gemeinsamen Beiträgen in der Informatikdidaktik geführt.

Noch viel weiter zurück reichen die eigentlichen Wurzeln. Diese liegen in meiner Familie und damit natürlich bei meinen Eltern *Gertrud* und *Hans Pasternak*. Es verwundert nicht, dass in einer Familie, in der beide Eltern Lehrer waren und in der sehr häufig über Schule und Pädagogik teilweise auch sehr heftig diskutiert und gestritten wurde, von vier Kindern zwei auch Lehrer geworden sind. Besonders geprägt hat mich die kritische Distanz zu Personen und Institutionen, wie sie beispielsweise durch viele Anmerkungen in den von meinem Vater gelesenen Büchern bezeugt wird. Die umfangreiche Bibliothek meiner Eltern war zudem eine Fundgruppe für die literarische Arbeit. Ganz wesentlich für mein Bewusstsein als Lehrer ist die Einstellung meiner Eltern, die in der Bildung und damit auch in der Schulbildung nicht das Privileg einer Minderheit in der Gesellschaft, sondern das Recht einer jeder Person gesehen haben, um im zukünftigen Leben im Beruf und im privaten Umfeld entsprechend seiner



Fähigkeiten, Leistungen und Interessen bestehen zu können.

Eine sicherlich sehr seltene Besonderheit war es, dass sich meine drei Kinder *Aike Jan*, *Mareke* und *Insa* zur selben Zeit im Studium befanden, in der ich als ihr Vater an meiner Dissertation ‚gewerkelt‘ habe. So gab es immer wieder intensive Diskussionen über Themen in den Fächern Mathematik, Physik, Biologie und Informatik sowie teilweise auch deren Umsetzung in der Schule, aber auch um (technische) Randprobleme wie die Nutzung von (Software-)Werkzeugen wie beispielsweise der Umgang mit  $\text{\LaTeX}$ . So entstanden bzw. entstehen innerhalb etwa eines Jahres eine Bachelor-, eine Staatsexamens-, eine Diplom- und eine Doktorarbeit von vier Personen aus einer Familie.

Gut, dass es da mit dem Münsterländer *Fiene* jemand gab, die immer wieder deutlich machte, dass beispielsweise gutes Wetter nicht nur zum Schreiben, sondern auch noch anders genutzt werden kann. Mit ihrer Schnauze hob sie dann immer wieder meine Hand sehr eindeutig von der Tastatur, womit sie ein Weiterschreiben verunmöglichte und damit unmissverständlich einforderte, mit ihr eine Tour zu Fuss oder mit dem Rad durchzuführen.



## Kurzfassung

Vor 30 Jahren entstand das Fach Informatik in der Schule [Del80, S.5ff], [Der81a]. In dieser doch kurzen Zeit hat es sehr unterschiedliche Herangehensweisen gegeben, wie Informatik in der Schule unterrichtet werden soll.

Nach dem *hardwareorientierte Ansatz* und dem Aufkommen der Mikrocomputer in der Schule setzte sich der *algorithmorientierte Ansatz* durch. Dieser wurde in der Sekundarstufe I durch den *anwendungsorientierten Ansatz* weitgehend abgelöst, der mit der Zeit immer mehr zu einem *benutzungsorientierten Ansatz* mutierte. In der Sekundarstufe II wurde inzwischen der algorithmorientierte Ansatz zu einem *objektorientierten Ansatz* erweitert. Hinzu kommen noch weitere Aspekte wie beispielsweise *Gesellschaft und Informatik*, die entweder parallel oder konträr zu diesen didaktischen Konzepten realisiert werden sollen.

Viele ausserhalb der Informatik stehende Personen bezweifeln zudem generell, dass Informatik ein *allgemeinbildendes Fach* ist. Sie sehen es eher wie das Fach Maschinenbau, das als Ingenieurwissenschaft die Umsetzung physikalischer Ideen darstellt und aus dieser Sicht nicht in die allgemeinbildende Schule gehört.

Im Jahre 2006 hat *Jeannette M. Wing* postuliert, dass es ein *Computational Thinking* [Win06] gibt, das als Kernpunkte *Automation* und *Abstraktion* beinhaltet. Will das Schulfach Informatik sich als allgemeinbildendes Fach behaupten, ist es notwendig, herauszuarbeiten, dass dieses *informatische Denken* eine grundsätzliche Denkstruktur wie beispielsweise das mathematische oder physikalische Denken darstellt, die unabhängig von der Anwendung konkreter Werkzeuge für das heutige Denken in Wissenschaft und Gesellschaft benötigt wird.

In dieser Arbeit wird gezeigt, dass genau diese grundsätzliche Denkstruktur vorliegt. Damit ist aus fachwissenschaftlicher Sicht Informatik als *Pflichtfach* in der Schule notwendig. Eine fachwissenschaftliche Begründung reicht für ein Schulfach nicht aus. Es muss überprüft werden, inwieweit Informatik auch den bildungswissenschaftlichen Bedingungen entspricht. Es zeigt sich, dass Informatik diesen Ansprüchen eines Schulfaches wie auch lernpsychologischen Ansprüchen gerecht wird.

Ausgehend von diesem Begründungszusammenhang wird ein Unterrichtsprojekt für einen *Informatikunterricht in der Sekundarstufe I* beschrieben, das in einem Wahlpflichtkurs mit Hauptfachcharakter an der Fritz-Steinhoff-Gesamtschule in Hagen exemplarisch in einem 6. Jahrgang durchgeführt und evaluiert wurde. Mit diesem Kurs wird gezeigt, dass ein auf *informatischen Denken* basierender Unterricht in der Sekundarstufe I etabliert werden kann. Die Zusammensetzung dieses Kurses entspricht leistungsmässig der Schulpopulation und verdeutlicht damit, dass Informatik als Fach für alle Schüler etabliert werden kann. Curriculare Basis für diesen Kurs sind die *Bildungsstandards Informatik* [Arb08].

Gerade für die Sekundarstufe I ist ein Unterricht notwendig, der ausgehend von den Vorkenntnissen der Schüler und ihrer Lebenswelt einige exemplarische und begründete Ideen der Informatik zum Thema hat. Dazu reicht es nicht aus, in Form einer Abbilddidaktik die Informatik der Hochschule oder der Sekundarstufe II auf die unteren Jahrgänge abzubilden. Es ist auch nicht zielführend, relativ beliebige Kontexte aus der Lebenswirklichkeit der Schülerinnen und Schüler auszuwählen und aus ihnen die Informatik zu extrahieren. Dagegen bietet es sich

an, die fachlich ausgewählten Gegenstände in *roten Fäden* zu organisieren und die unterrichtlichen Probleme, Kontexte und Projekte derart auszuwählen, dass möglichst oft verschiedene Fäden im konkreten Unterricht zeitnah *verwoben* werden.

Die Evaluation des vorgestellten Projektes zeigt, dass ein derartig an informatischen Standards ausgerichteter Unterricht einen speziellen ITG-Kurs, wie er heute häufig anzutreffen ist, überflüssig macht.

## Abstract

About 30 years ago *computer science* became a subject at school [Del80, S.5ff], [Der81a]. 30 years are little time considering the history of school. During this period different trends in teaching computer science have been recognized..

In the beginning computer science was taught *hardware orientated*. After the emergence of the microcomputer in school computer science was taught *algorithms oriented*. In the 1980s, in the lower secondary education the *application oriented approach* followed, which mutated to an *user orientated approach* in the time following. In the higher secondary education the algorithms orientated approach extended to an *object orientated approach*. Further aspects like *computer science and society* are discussed parallelly or contrarily to these didactical concepts.

Many people outside computer science generally doubt that computer science provides *general education*. They feel computer science is more like *engineering*, as the implementation of technical, physical or mathematical ideas, and therefore computer science doesn't fulfill the imperative of holistic education.

2006 Jeannette M. Wing postulated a *computational thinking* [Win06] consisting of *automation* and *abstraction*. To constitute computer science as a general education subject in schools, it is necessary to prove the *computational thinking* a basic structure of thinking in science and society comparable to mathematical or physical thinking regardless of concrete realisation of any specific (software-)tool.

This dissertation shows the existence and importance of this mental structure of thinking. Hence computer science is necessary as a *compulsory subject* in schools. However, this justification is not sufficient. We have to prove the necessity of computer science as a subject from the point of view of educational sciences. Furthermore computer science meets the demands of learning psychology.

Proceeding from this context an education project *computer science in the lower secondary education* will be outlined. This course took place in grade 6 with the character of a main subject at *Fritz-Steinhoff-Gesamtschule*. The procedure and outcome were evaluated. The composition of this course matched the school population with reference of social structure and thus it may be assumed, computer science can be established as a subject for all students. The curricular basis of this approach are the *Bildungsstandards Informatik* [Arb08].

Teaching in the lower secondary grades requires lessons based on the prior knowledge and every day life of students. It is not appropriate merely to project the principles of teaching computer science in university or higher secondary education to the lower secondary education. Neither does it pay to extract computer science items in unselected contexts from the student's life. A suitable approach to present the ideas of computer science in a coherent way is to organize the selected items in *strands taught in a braided way*. Selecting problems, contexts and projects in such lessons supports the topical coherence of different strands and should be practised as often as possible.

The evaluation additionally shows that there is no need for a special ICT course as commonly found, if there is an offer of lessons orientated on standards of computer science education.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
<b>2</b>	<b>Fachwissenschaftliche Legitimation</b>	<b>13</b>
2.1	Geistiges Rückgrad der Informatik: Informatisches Denken . . . . .	13
2.2	Sprache und Sprachwissenschaften . . . . .	15
2.2.1	Programmiersprachen . . . . .	15
2.2.2	Formale Sprachen . . . . .	17
2.2.3	Natürliche Sprachen . . . . .	17
2.2.4	Fazit Sprache und Sprachwissenschaften . . . . .	20
2.3	Algorithmik . . . . .	20
2.3.1	Algorithmik und Sprache . . . . .	20
2.3.2	Algorithmik und Mathematik . . . . .	22
2.3.3	Fazit Algorithmik . . . . .	25
2.4	Mathematik . . . . .	26
2.4.1	Numerik . . . . .	26
2.4.2	Statistik . . . . .	27
2.4.3	Diskrete Mathematik . . . . .	27
2.4.4	Fazit Mathematik . . . . .	30
2.4.5	Ausflug: Der Streit der Fakultäten . . . . .	31
2.5	Physik . . . . .	32
2.5.1	Die Krise der Physik . . . . .	32
2.5.2	Elektromagnetische Wellen . . . . .	32
2.5.3	Telefon . . . . .	33
2.5.4	Ausbildung in der Physik . . . . .	34
2.5.5	Fazit Physik . . . . .	36
2.6	Philosophie . . . . .	36
2.6.1	Das Gehirn als informationsverarbeitende Maschine . . . . .	37
2.6.2	Der Begriff <i>Information</i> aus philosophischer Sicht . . . . .	38
2.6.3	Künstliche Intelligenz . . . . .	40
2.6.4	Ethik . . . . .	42
2.6.5	Fazit Philosophie . . . . .	42
2.7	Biologie . . . . .	43
2.7.1	Neuronale Netzwerke . . . . .	43
2.7.2	Genetik . . . . .	44
2.7.3	Fazit Biologie . . . . .	46

2.8	Zusammenfassung und Fazit . . . . .	46
2.8.1	Bedeutung der Informatik für andere Disziplinen . . . . .	46
2.8.2	Geschichte des Begriffs Informatik . . . . .	47
2.8.3	Informatik und informatisches Denken . . . . .	49
2.8.4	Curriculare Konsequenzen . . . . .	49
<b>3</b>	<b>Bildungswissenschaftliche Legitimation</b>	<b>51</b>
3.1	Das Bildungssystem der Bundesrepublik Deutschland . . . . .	52
3.1.1	Die deutsche Bildungskatastrophe . . . . .	53
3.1.2	Bildung ist Bürgerrecht . . . . .	54
3.1.3	Der Bildungsgesamtplan . . . . .	56
3.1.4	Auswirkungen . . . . .	57
3.2	Definition des Bildungsbegriffes . . . . .	58
3.2.1	Bildung aus der Vergangenheit, der Gegenwart oder für die Zukunft? . . . . .	59
3.2.2	Kategoriale Bildung . . . . .	60
3.2.3	Die lerntheoretische Didaktik . . . . .	60
3.2.4	Die curriculare Didaktik . . . . .	61
3.2.5	Die lernzielorientierte Didaktik . . . . .	63
3.2.6	Didaktische Analyse . . . . .	63
3.3	Der (Allgemein-)Bildungsbegriff nach Heymann . . . . .	64
3.3.1	Der Begriff <i>Bildung</i> . . . . .	64
3.3.2	Der Begriff <i>Allgemeinbildung</i> . . . . .	65
3.3.3	Aufgaben der allgemeinbildenden Schule . . . . .	65
3.4	Allgemeinbildung und Informatik . . . . .	67
3.4.1	Lebensvorbereitung . . . . .	67
3.4.2	Berufswelt . . . . .	68
3.4.3	Privat- und Freizeitwelt . . . . .	69
3.4.4	Demokratischer Bürger . . . . .	69
3.4.5	Stiftung kultureller Kohärenz . . . . .	70
3.4.6	Weltorientierung . . . . .	71
3.4.7	Anleitung zum kritischen Vernunftgebrauch Entfaltung von Verantwortungsbereitschaft Einübung in Verständigung und Kooperation . . . . .	71
3.4.8	Individueller Bürger . . . . .	72
3.4.9	Fazit Allgemeinbildung und Informatik . . . . .	72
3.5	Pisa und die Informatik . . . . .	73
3.5.1	Bildungsstandards . . . . .	74
3.5.2	Der Kompetenzbegriff . . . . .	75
3.5.3	Die Bildungsstandards Informatik . . . . .	76
3.6	Curriculare Konsequenzen . . . . .	78
3.7	Zusammenfassung . . . . .	79
<b>4</b>	<b>Lernpsychologie</b>	<b>81</b>
4.1	Lerntheorien . . . . .	82



4.1.1	Begriff des Lernens . . . . .	82
4.1.2	Behaviorismus . . . . .	82
4.1.3	Kognitivismus . . . . .	83
4.1.4	Konstruktivismus . . . . .	85
4.1.5	Zusammenfassung Lerntheorien . . . . .	87
4.2	Psychologie des Kindes . . . . .	88
4.2.1	Die Psychologie der Intelligenz nach Piaget . . . . .	88
4.2.2	Die Erziehung nach Bruner . . . . .	89
4.2.3	Die Lernstufen nach van Hiele . . . . .	90
4.2.4	Zusammenfassung Psychologie des Kindes . . . . .	92

**5 Das Schulfach Informatik 93**

5.1	Didaktisches Dreieck der Informatik . . . . .	93
5.1.1	Umsetzung der informatischen Bildung in den 80er Jahren . . . . .	94
5.1.2	Problematik integrativer Ansätze . . . . .	95
5.1.3	Das didaktische Dreieck der Informatik . . . . .	96
5.1.4	Fazit didaktisches Dreieck der Informatik . . . . .	100
5.2	Visuelle und textuelle Programmiersprachen . . . . .	101
5.2.1	Sprachenlernen . . . . .	101
5.2.2	Übertrag auf den Informatikunterricht . . . . .	102
5.2.3	Visuelle Programmiersprachen ideal für Anfänger? . . . . .	105
5.2.4	Visualisieren von Algorithmen . . . . .	106
5.2.5	Fazit visuelle und textuelle Programmiersprachen . . . . .	106
5.3	Leitlinien der didaktischen Orientierung . . . . .	107
5.3.1	Exemplarisches Lernen . . . . .	107
5.3.2	Fundamentale Ideen . . . . .	108
5.3.3	Basiskonzepte . . . . .	110
5.3.4	Fazit Leitlinien der didaktischen Orientierung . . . . .	111
5.4	Auswahl von Inhalten . . . . .	111
5.4.1	Great Principles . . . . .	111
5.4.2	Didaktische Linsen . . . . .	112
5.4.3	Fazit Auswahl von Inhalten . . . . .	114
5.5	Informatik und Kontext . . . . .	114
5.5.1	Informatik im Kontext . . . . .	117
5.5.2	Grundbildung Informatik . . . . .	119
5.5.3	Kontextorientierter Unterricht . . . . .	122
5.5.4	Fazit Informatik und Kontext . . . . .	124
5.6	Ein Blick über die Grenzen . . . . .	125
5.6.1	USA . . . . .	125
5.6.2	Grossbritannien . . . . .	127
5.6.3	Niederlande . . . . .	128
5.6.4	Weitere Länder . . . . .	129
5.6.5	Fazit Ein Blick über die Grenzen . . . . .	130
5.7	Die Bildungsstandards Informatik . . . . .	130

5.7.1	Die Bildungsmatrix . . . . .	132
5.7.2	Fazit Bildungsstandards . . . . .	134
5.8	Zusammenfassung und Konsequenzen . . . . .	134
5.8.1	Konsequenzen für das Schulfach Informatik . . . . .	135
<b>6</b>	<b>Das Konzept der Roten Fäden</b>	<b>137</b>
6.0.2	Definition der <i>Roten Fäden</i> . . . . .	137
6.1	Beispielhaftes Unterrichten mit <i>Roten Fäden</i> . . . . .	139
6.1.1	Unterrichtliche Gestaltung . . . . .	139
6.1.2	Einbettung in die Bildungsstandards . . . . .	140
6.1.3	Verwobenes Unterrichten . . . . .	141
6.2	Programmiersprache als <i>Roter Faden</i> . . . . .	142
6.3	<i>Rote Fäden</i> im Vergleich gegenüber anderen Konzepten . . . . .	143
6.3.1	Fundamentale Ideen . . . . .	144
6.3.2	Bildungsstandards . . . . .	144
6.3.3	Great Principles . . . . .	145
6.3.4	Didaktische Linsen . . . . .	145
6.3.5	Informatikturm . . . . .	145
6.4	Zusammenfassung . . . . .	146
<b>7</b>	<b>Unterrichtsprojekt an der FSG</b>	<b>149</b>
7.1	Bildung des WP-Faches NW (If/PH) . . . . .	150
7.1.1	Informatik als Wahlpflicht II - Fach . . . . .	150
7.1.2	Informatik als Wahlpflicht (I) - Fach . . . . .	151
7.1.3	Hypothesen . . . . .	152
7.1.4	Vorstellung . . . . .	153
7.2	Darstellung der gebildeten Gruppe . . . . .	154
7.2.1	Eingangspopulation . . . . .	156
7.2.2	Unterrichtliche Voraussetzungen . . . . .	156
7.3	Planung des Unterrichts für das erste Jahr (Jahrgangsstufe 6) . . . . .	157
7.3.1	Verwobenes Unterrichten . . . . .	158
7.4	Der rote Faden <i>Programmiersprache</i> . . . . .	158
7.4.1	Auswahl einer Programmiersprache . . . . .	159
7.4.2	Werkzeug Programmiersprache Tcl/Tk . . . . .	161
7.4.3	Zwei kleine Beispiele in Tcl/Tk . . . . .	162
7.5	Unterrichtsverlauf Informatik Jahrgangsstufe 6 . . . . .	164
7.5.1	Einheit 1: Buttons und Labels <i>Dauer: 5 Wochen</i> . . . . .	164
7.5.2	Einheit 2: Bilder und Webseiten <i>Dauer: 5 Wochen</i> . . . . .	166
7.5.3	Einheit 3: Dateiorganisation <i>Dauer: 3 Wochen</i> . . . . .	167
7.5.4	Einheit 4: Zeichnungen in Tcl/Tk <i>Dauer: 8 Wochen</i> . . . . .	168
7.5.5	Einheit 5: Vektor- und Pixelgrafik <i>Dauer: 6 Wochen</i> . . . . .	169
7.5.6	Einheit 6: Variablen <i>Dauer: 5 Wochen</i> . . . . .	171
7.5.7	Zusammenfassung . . . . .	173
7.6	Kursarbeiten . . . . .	174

7.6.1	Aufgabenbeispiele . . . . .	176
7.6.2	Zusammenfassung . . . . .	182
<b>8</b>	<b>Evaluation</b>	<b>183</b>
8.1	Ergebnisse der Kursarbeiten . . . . .	183
8.1.1	Zeugnisnoten . . . . .	185
8.1.2	Lieblingsfach . . . . .	186
8.1.3	Zusammenfassung Ergebnisse der Kursarbeiten . . . . .	187
8.2	Empirische Untersuchung des Unterrichtsprojekts . . . . .	188
8.3	Beteiligte Schüler-Gruppen . . . . .	189
8.3.1	Gruppen an der Fritz-Steinhoff-Gesamtschule . . . . .	189
8.3.2	Gruppen an der Hauptschule Vorhalle . . . . .	189
8.3.3	Gruppen an der Realschule Halden . . . . .	190
8.4	Untersuchungsmethodik . . . . .	191
8.5	Untersuchungsverfahren . . . . .	192
8.5.1	Schülerfragebogen . . . . .	192
8.5.2	Überprüfung mit <i>Think Aloud</i> -Interviews . . . . .	193
8.5.3	Aufgaben aus dem <i>Biber</i> -Wettbewerb . . . . .	193
8.6	Auswertung der Fragen aus dem <i>Biber</i> -Wettbewerb . . . . .	194
8.7	Auswertung des Fragebogens ‚Wissen‘ . . . . .	196
8.7.1	Programmiersprachliche Konzepte . . . . .	196
8.7.2	Graphik . . . . .	203
8.7.3	Netze . . . . .	205
8.7.4	Betriebssysteme . . . . .	206
8.8	Auswertung des Fragebogens zu Aussagen über die Informatik . . . . .	213
8.8.1	Zusammenfassung . . . . .	224
8.9	Auswertung des Fragebogens zu Einstellungen, Meinungen und Verhalten . . . . .	225
8.9.1	Beschäftigung mit dem Computer . . . . .	225
8.9.2	Zusammenfassung . . . . .	240
8.10	Follow-Up – Untersuchungen . . . . .	240
8.10.1	Ausgewählte Beispiele aus der Follow-Up-Untersuchung . . . . .	241
8.10.2	Zusammenfassung der Aussagen zum Follow-Up – Test . . . . .	251
8.11	Auswertung der <i>Think-Aloud</i> -Interviews . . . . .	252
8.11.1	Interviews nach dem Posttest . . . . .	252
8.11.2	Interviews nach dem Follow-Up-Text . . . . .	256
8.11.3	Interviews am Ende des Schuljahres 2011/2012 . . . . .	258
8.12	Quantitative Untersuchungen . . . . .	259
8.12.1	Fragebogen . . . . .	259
8.12.2	Think-Aloud . . . . .	263
8.12.3	Gender-Problematik . . . . .	263
8.13	Informatik versus ITG . . . . .	264
<b>9</b>	<b>Fazit und weitere Untersuchungen</b>	<b>269</b>
9.1	Fazit Informatikunterricht nach 1 Jahr . . . . .	269

9.1.1	Unterrichtsfach Informatik . . . . .	269
9.1.2	Beschränkung des Curriculums . . . . .	272
9.2	Weitere Untersuchungen . . . . .	273
<b>10</b>	<b>Schlussbemerkung</b>	<b>275</b>
<b>11</b>	<b>Anhang</b>	<b>277</b>
11.1	Eltern - Informationsbroschüre: WP-Wahl 3/2010 . . . . .	278
11.2	Kursmappe . . . . .	282
11.3	Materialien . . . . .	286
11.4	Übungen . . . . .	296
11.5	Teste . . . . .	332
11.6	Python . . . . .	360
11.7	Fragebögen . . . . .	368
11.8	Auswertung Biber-Fragen . . . . .	380
11.9	Follow Up . . . . .	388
11.10	Think Aloud . . . . .	398
11.10.1	Beispielhafte Interviews . . . . .	398
11.10.2	Zusammenfassung 'Think Aloud' nach dem Posttest . . . . .	415
11.10.3	Zusammenfassung 'Think Aloud' nach dem Posttest und Follow-Up- Test . . . . .	419
11.10.4	Zusammenfassung 'Think Aloud' am Ende des Schuljahres 2011/2012	422
11.11	Exemplarische Daten . . . . .	424
11.12	Skripte in R . . . . .	431
11.12.1	Beispiel einer Auswertung der 'Halbzeit 1' . . . . .	431
11.12.2	Beispiel einer Auswertung der 'Halbzeit 2' . . . . .	433
11.12.3	Beispiel einer Auswertung der 'Nachspielzeit' . . . . .	434
11.12.4	Beispiel einer Auswertung des 'Elfmeterschiessens' . . . . .	436
11.12.5	Beispiel einer Vergleichsauswertung . . . . .	438
11.12.6	Variablen-Auswertung . . . . .	441
11.12.7	Jungen/Mädchen-Auswertung . . . . .	444
11.13	Lehrpläne an der HS Vorhalle und der RS Halden . . . . .	448
11.13.1	Lehrplan Informatik an der Hauptschule Vorhalle (Auszug) . . . . .	448
11.13.2	Lehrplan Informatik an der Realschule Halden (Auszug) . . . . .	451
	<b>Literaturverzeichnis</b>	<b>453</b>

# 1 Einleitung

Das Studienfach *Informatik* gibt es seit ca. 50 Jahren<sup>1</sup> an deutschen Hochschulen und Universitäten. Ein der Rolle des Universitätsfaches Informatik an den Hochschulen vergleichbares Schulfach gibt es nicht in der Sekundarstufe II und erst recht nicht in der Sekundarstufe I. Ein Pflichtfach oder sogar ein Hauptfach Informatik in der Sekundarstufe I ist zur Zeit überhaupt nicht in Sicht. Nun gibt es viele Studienfächer wie beispielsweise Maschinenbau oder Jura, denen ebenfalls kein (obligatorisches) Schulfach vorgeschaltet ist. Eine Disziplin als Schulfach zu etablieren, verlangt daher eine Begründung, warum die in diesem Fach erwerbbaaren Kompetenzen über das reine Fachwissen hinaus zur Bildung der Lernenden beiträgt und eventuell sogar notwendig ist.

Diese Arbeit begründet und zeigt aus fach- und bildungswissenschaftlichen Anforderungen, dass eine derartige *informatische Bildung* als Bestandteil einer Allgemeinbildung existieren sollte und in der Schule durch ein Pflicht- und auch Hauptfach Informatik ab der Sekundarstufe I erfolgreich vermittelbar ist.

## Bestrebungen um ein Schulfach Informatik

Das Schulfach Informatik ist heute im Kanon der Schulfächer relativ unbedeutend. Das verwundert, weil bereits im Jahrzehnt nach der Gründung der Hochschulinformatik in einigen deutschen Bundesländern vielfältige Schulversuche durchgeführt wurden [Del80, S.5ff]. In Nordrhein-Westfalen mündeten diese Bestrebungen 1981 in den Richtlinien eines anerkannten Schulfaches Informatik in der Sekundarstufe II [Der81a]. Im Vergleich dazu war die Etablierung der naturwissenschaftlichen Fächer in den Gymnasien Ende des 19. Jahrhunderts und Anfang des 20. Jahrhunderts [Gut08] ein langsamer Prozess.

Viele Menschen fühlten sich regelrecht *überrollt* von den Ideen, den Produkten und den Werkzeugen der Informatik. Je nach Stand dieser Technik wurden entsprechend die Curricula in Informatik angepasst. In den 60er und 70er Jahren waren die Computer noch grosse, unförmige Maschinen, die durch die Kunst der Programmierung ihr Werk verrichteten. Ein Unterricht in der Sekundarstufe II, in der die Beschäftigung mit Algorithmik und Programmierung im Vordergrund stand, erschien als die adäquate Antwort auf diese ‚Revolution durch die Computer‘. Es war nur scheinbar eine Frage der Zeit, wann das Schulfach Informatik eine entscheidende Rolle in der Laufbahn der Schülerinnen und Schüler erhalten sollte.

Diese intensiven Bemühungen um das neue Fach kamen sehr schnell zum Erliegen, nachdem ergänzend zum Schulfach Informatik in der Sekundarstufe I in den 80er Jahren eine

---

<sup>1</sup>Als ‚Gründungstermin‘ der Universitätsinformatik wird hier das Datum des Schreibens der *Westdeutschen Rektorenkonferenz (WRK)* vom 12. Dezember 1969 an einige Fakultätentage mit der Aufforderung zur Formulierung einer *Rahmenprüfungsordnung Informatik* angenommen [Gör98, S.11].

intensive Diskussion um das Fach Informatik in der Sekundarstufe I begann. Diese Diskussion führte eigentlich überraschend im Gegensatz zur Situation in der Sekundarstufe II nicht in der Einrichtung eines neuen Faches. Die Folge dieser Diskussion waren unterschiedlichste Modelle einer ursprünglich so genannten *Grundbildung Informatik* [BHW85] *im Rahmen der etablierten Schulfächer* und entsprechend in NRW die Aufstellung eines fachübergreifenden *Rahmenkonzept Neue Informations- und Kommunikationstechnologien in der Schule* [NRW85] für eine informations- und kommunikationstechnologische Grundbildung (ITG). Im Unterschied zu einem neuen Fach in der Sekundarstufe II hätte die Einrichtung eines eigenständigen Faches in der Sekundarstufe I die Veränderung der Stundentafel führen müssen, die alle etablierten Fächer zu verhindern wussten. Mit dieser Entscheidung waren die Bemühungen um ein verbindliches Schulfach Informatik in der Sekundarstufe I für mehr als ein Jahrzehnt beendet.

Trotzdem gilt für die Entscheidungen in den beiden Sekundarstufen – die Entwicklung eines Schulfaches Informatik in der Sekundarstufe II als auch die Einführung einer ITG anstelle eines eigenen Schulfaches Informatik in der Sekundarstufe I – ist Folge einer allgemein anerkannten Position, die 1981 im Vorwort eines Bandes für die didaktische Handreichung für das Schulfach Informatik von *Wolfgang Arlt* folgendermassen charakterisiert wird:

„Die ständig zunehmende Integration von EDV-Anlagen, der steigende Einsatz von Mikroelektronik und Automatisierung oder der Einsatz von ‚Robotern‘ bedeutet eine neue Herausforderung an die Gesellschaft, besonders aber an das Ausbildungssystem, um ihre Schüler befähigen zu können, mit diesen ‚modernen‘ Entwicklungen fertig zu werden.“

## Deformation des Informatikunterrichtes

Es waren aber nicht nur die Ängste um die Anteile an der Stundentafel, die ein Schulfach Informatik in der Sekundarstufe I verhinderten: Mit dem Aufkommen der kleinen Mikrocomputer, den PCs, wie sie später genannt wurden, veränderte sich die Sichtweise vieler Menschen: Nicht mehr die Programmierung, sondern die *Anwendung* stand im Vordergrund. *Anwendungsorientierung* und schliesslich die *Benutzungsorientierung* prägte nun das Bewusstsein breiter Kreise in der Öffentlichkeit, der Schuladministration und auch vieler (Informatik-)Lehrer. Mit der Vernetzung der Computer ab Mitte der 90er Jahre spätestens durch das Internet trat neben diese Sichtweise zusätzlich die *Medienorientierung* durch und mit den Computern. In diesem Lichte reduziert sich die Informatik auf die Anwendung als Werkzeug und Medien, genauso wie beispielsweise der Maschinenbau physikalische Erkenntnisse in technischen Produkten zum Nutzen der Menschen erforscht, entwickelt und produziert.

Maschinenbau und vergleichbare Fächer wie Elektrotechnik oder Nachrichtentechnik sind daher zu Recht keine allgemeinbildenden Schulfächer. Sollte der Charakter der Informatik allerdings diesen Fächern ähnlich sein, ist die Gefahr gross, dass die Informatik wieder aus dem Kanon der Schule gänzlich verdrängt wird. Die Entwicklung beispielsweise eines *Europäischen Computer Führerschein (ECDL)*<sup>2</sup> in teilweiser Konkurrenz zum Schulfach Informatik [BFM<sup>+</sup>10, S.219-227] macht deutlich, dass nicht gesichert ist, dass in Zukunft Informatik als Schulfach Bestand hat.

<sup>2</sup><http://www.ecdl.de>, letzter Zugriff: 29.11.2012

Die Durchdringung der Gesellschaft in Beruf und Freizeit mit informationstechnologischen Geräten reicht nicht aus, ein Schulfach zu begründen. Nur dann, wenn jenseits der Gerätenutzung als Werkzeug und Medien eine Bedeutung der Informatik für praktisch jeden Einzelnen vorhanden, sichtbar und anerkannt ist, ist das Schulfach Informatik zu sichern. Mathematik, die naturwissenschaftlichen Fächer usw. proklamieren ein fachspezifisches Denken – mathematisches Denken, physikalisches Denken etc. – mit für den Bildungsprozess des Einzelnen wichtiger Bedeutung und dieses rechtfertigt sie im Gegensatz zu anderen Wissensbereichen als Schulfach.

## **Informatisches Denken**

Ein dem ‚mathematischen‘ oder dem ‚physikalischen‘ vergleichbares *informatisches Denken* proklamiert *Jeannette Wing* 2006:

*“Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone“. “Computational thinking is a fundamental skill for everyone, not just for computer scientists.*

[...]

*Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. ... “ [Win06, S.33]*

Dieses *informatische Denken* besteht demnach im Wesentlichen aus *Abstrahieren* und *Automatisieren* und ist ebenso fundamental wie beispielsweise das mathematische Denken. Und genau wie das mathematische Denken nicht nur vom Mathematiker oder von einigen weiteren Naturwissenschaftlern und Technikern benötigt wird, sondern in vielfältigster Form von den meisten Menschen im Berufsleben und Freizeit abverlangt wird, ist das informatische Denken nicht nur nötig für den relativ kleinen Teil der Informatiker an der Bevölkerung, sondern für viele Menschen in unterschiedlichster Zusammenhängen.

## **Informatisches Denken in anderen Disziplinen**

Aus dieser Bedeutung des informatischen Denkens in anderen Fächern kann sich das Schulfach Informatik in vergleichbarer Weise wie die klassischen Schulfächer Mathematik, Sprachen etc. begründen.

Im ersten Teil dieser Arbeit wird daher in Kapitel 2 *Fachwissenschaftliche Legitimation* gezeigt, wie dieses informatische Denken die anderen Disziplinen beeinflusst bzw. beeinflusst hat. Dabei wird ein historischer Zugang gewählt. Fragestellungen, die in den verschiedensten Disziplinen entstanden sind, haben entweder zur Informatik und informatischen Denken hingeführt bzw. nutzten zunehmend die Erkenntnisse und das Denken der Informatik für die eigenen Disziplin.

Dieser Prozess ist in den verschiedenen Fächern unterschiedlich verlaufen. In Fächern wie Mathematik und Physik hat sich die Informatik ‚herausentwickelt‘, in anderen Fächern beispielsweise in den Sprachen haben Erkenntnisse aus diesen Fächern mit Erkenntnissen der In-

formatik zu Fortschritten geführt. Neben den mathematisch-naturwissenschaftlichen und den sprachlichen Fächern stellen die gesellschaftswissenschaftlichen Fächern die dritte Säule der heutigen Schulbildung dar. Am Beispiel des Faches Philosophie wird gezeigt, dass auch in diesen Fächern informatisches Denken förderlich ist. Konsequenz ist, dass mit informatischen Denken und Kenntnissen nicht nur technische Produkte produziert werden können, deren Nutzung ohne informatisches Basiswissen möglich ist, sondern dass im Kernbereich vieler Disziplinen, die sich selbst teilweise weit weg von der Informatik fühlen, dieses informatische Denken notwendig ist.

## **Informatik als Bildungsgut in der Schule**

Informatik, vor allem informatisches Denken, ist ein sogenanntes Bildungsgut, das für sehr viele Menschen sinnvoll und teilweise notwendig ist. Das bedeutet allerdings nicht, dass dies zwingend in der Schule erworben werden muss. Da die Schule nur ein beschränktes Kontingent an Lernzeit zur Verfügung hat, muss eventuell eine begründete Auswahl an Inhalten getroffen werden. Zudem gibt es neben dem Argument, dass ein Fach ein Bildungsgut auch für andere Fächer darstellt noch weitere berechtigte Anliegen, als Schulfach etabliert zu werden. Beispielsweise können in einer demokratischen Gesellschaft Inhalte eines Faches für die *Welterkenntnis* so zentral sein, dass es im Unterricht der Schule zwingend vorkommen soll. Es ist daher zu prüfen, wie heute *Bildung* und *Allgemeinbildung* definiert werden. Aus dieser Definition ist dann abzuleiten, welche Facetten von Bildung durch Informatik abgedeckt werden.

Im Kapitel 3 *Bildungswissenschaftliche Legitimation* wird entsprechend herausgearbeitet, dass die bildungswissenschaftlichen Anforderungen Informatik in der Schule als (Pflicht-)Fach nahelegen. Dabei wird in dieser Arbeit in erster Linie auf die Sekundarstufe I Bezug genommen. Damit ergänzen diese Darstellungen die Untersuchungen anderer Autoren zum Schulfach Informatik vor allem in der Sekundarstufe II beispielsweise von *Hellmut Deller* [Del80], *Wolfgang Arlt* [Arl81], Anton Brenner und Rul Gunzenhäuser [Bre82], *Rüdeger Baumann* [Bau90] und *Ludger Humbert* [Hum05].

## **Lernpsychologie und das Schulfach Informatik**

Der Lernprozess des Menschen ist ein langer und langsamer Weg. Wir haben darauf zu achten, dass auf das Alter und den Entwicklungsstand der Kinder, Jugendlichen und jungen Erwachsenen geachtet wird. Dies verlangt zu prüfen, ob und wie ein Unterrichtsgegenstand oder Teile davon wann und wie in der Schule angeboten werden. Die Erkenntnisse der Lernpsychologie helfen, gerade ein relativ neues Unterrichtsfach wie Informatik entsprechend für die Schule aufzubereiten. Diese Bedingungen werden im Kapitel 4 über *Lernpsychologie* aus Sicht der Informatik betrachtet und analysiert.

Wir wissen aus den Erkenntnissen nicht nur von *Piaget* und *Bruner*, dass ein unterrichtlicher Inhalt sinnvoll in einem Spiralprozess immer wieder altersgerecht angeboten werden muss, wenn wir eine nachhaltige Wirkung erzielen wollen. Das bedeutet, dass Informatik nicht erst in der Sekundarstufe II angeboten werden sollte, sondern schon in der Sekundar-



stufe I als Fach vorhanden sein muss. Dies ist erst in Ansätzen realisiert und bedarf intensiver Forschungsarbeit, wie ein solcher Unterricht in der Sekundarstufe I aussehen kann.

## Unterrichtsprojekt an einer Gesamtschule

Im Kapitel 7 wird ein *Unterrichtsprojekt an der Fritz-Steinhoff-Gesamtschule in Hagen (FSG) in der Sekundarstufe I* vorgestellt und anschliessend ausgewertet, der die Realisierbarkeit eines allgemeinverbindlichen Informatikunterrichtes zeigt. Sowohl wichtig als auch bemerkenswert ist, dass dieses Unterrichtsprojekt nicht in einer ausgewählten Schülergruppe realisierbar wurde, sondern diese Gruppe aus Schülern aller Schichten und Begabungsstufen besteht. Es wurde darauf geachtet, dass die untersuchte Schülergruppe weitgehend in Grösse und Zusammensetzung dem gesamten Jahrgang an der FSG entspricht. Die Ergebnisse aus diesem Versuch lassen sich daher gut auf die Sekundarstufen I auch anderer Schulformen übertragen.

## Vision

Das in dieser Arbeit beschriebene Projekt zeigt: Unterricht in Informatik in der Sekundarstufe I bereits in den unteren Jahrgängen als Pflichtfach und sogar als Hauptfach ist möglich. Auch weitere Ansätze mit Unterricht in Informatik in der Sekundarstufe I in verschiedenen Bundesländern wie Bayern, Mecklenburg-Vorpommern und Sachsen [Sta10] zeigen, dass ein Schulfach Informatik in unteren Jahrgängen möglich und sinnvoll ist.

Eine Umsetzung in der Schule ist allerdings derzeit schwierig. Eine Diskussion wie in den 60er Jahren auch über die Fächerstruktur an der Schule [Rob67] findet nicht statt. Die aktuellen Untersuchungsprogramme wie PISA [KAH<sup>+</sup>10a, KAH<sup>+</sup>10b] und die daraus gefolgerten bundesdeutschen Antworten, die u.a. in den Bildungsstandards ihren Niederschlag finden [Kli07, JM07] fokussieren auf die anerkannten Schulfächer und verhindern weitergehende Reflektionen über Fächer und Inhalte in der Schule [Hey96, S.29]. Inhaltliche Defizite, die jenseits dieser Programme und der darauf basierenden Untersuchungen liegen, werden damit übersehen oder ignoriert. So richtet sich das Augenmerk vieler Bürger auf die Inhalte und Kompetenzerwerbung etablierter Fächer und die äussere Schulstruktur.

Der Mangel an informatischer Bildung kann aber nicht beliebig lange ignoriert werden. Die *Bildungsstandards Informatik* [Arb08], die als Folge dieses Mangels nicht von der Bildungsadministration in Bund und Ländern, sondern von interessierten Lehrern und Hochschullehrern entwickelt worden sind, verdeutlichen in Entstehungsgeschichte und Inhalt auf dieses Defizit.

In diesen Bildungsstandards Informatik wird zu Beginn für die Zukunft angenommen:

„Die Vision ist, dass informatisch gebildete Menschen alle informatischen Probleme, die ihnen in ihrem Leben begegnen werden, mit Selbstvertrauen anpacken und selbstständig allein oder im Team bewältigen können. Und die Lehrenden helfen den Lernenden dabei, ihre Kompetenz zum Lösen solcher Probleme einzusetzen, zu vertiefen und auszubauen. [Arb08, S.1]“

Der Raum, den Schülerinnen und Schüler für das Erwerben dieser informatischen Kompetenzen benötigen, sollte ihnen nicht lange verwehrt werden. Ein Pflichtfach Informatik ab der Sekundarstufe I fördert dieses Bildungsziel am besten.

# 2 Fachwissenschaftliche Legitimation

## Überblick

Das Schulfach Informatik ist seit den 70er Jahren in der Schule existent. In den ersten Jahren wurde es weitgehend entsprechend einer Abbilddidaktik wie die Hochschulinformatik unterrichtet. Die Inhalte des Faches wurden im Laufe der Jahrzehnte immer wieder kritisiert und wurden darauf hin entsprechend angepasst. Dies gilt vor allem für die Sekundarstufe I. Je mehr der PC in der Alltagswelt auftauchte, desto mehr wurde von vielen gerade Nicht-Informatikern erwartet, dass das Umgehen mit diesem Gerät und seinen Produkten im Fokus des Unterrichts steht. Diese Benutzerorientierung mit der Basis der Werkzeuge der Informatik bestimmt die Erwartungshaltung. Kein anderes Schulfach erfährt eine vergleichbare Forderung an seine Inhalte.

Diese Inhalte und Forderungen können keine allgemeinbildende Begründung für das Schulfach Informatik sein. In diesem Kapitel wird daher untersucht, wie das informatische Denken jenseits des Werkzeugcharakters in vielen Disziplinen förderlich, wenn nicht gar notwendig ist. Es wird gezeigt, dass genauso wie ein Physiker sich mit Mathematik beschäftigen muss, um seine eigene Disziplin zu verstehen, sich auch ein an der Informatik und seinen Produkten desinteressierter Mensch sich mit den Ideen der Informatik auseinandersetzen muss oder zumindest sollte, um sein eigenes Fach adäquat verstehen zu können.

In diesem Kapitel werden beispielhaft für mehrere Fachdisziplinen wie Mathematik, Physik, Sprachen oder auch Philosophie betrachtet, wie sich in den letzten Jahrzehnten Inhalte und Denkweisen des jeweiligen Faches und der Informatik gegenseitig beeinflusst haben und damit das informatische Denken heute weit in andere Disziplinen hineingewachsen ist.

Daraus folgt, dass das informatische Denken ein (Denk-)Werkzeug ist, das sich der junge Mensch aneignen muss. Darin liegt aus fachlicher Sicht in erster Linie der allgemeinbildende Charakter der Informatik.

## 2.1 Geistiges Rückgrad der Informatik: Informatisches Denken

Informatik ist eine junge Disziplin im Vergleich zu etablierten Wissenschaften wie Mathematik, Philosophie, Geschichte oder auch den neueren Wissenschaften im Bereich der Naturwissenschaften wie Physik, Chemie und Biologie. Die Informatik ist zweifelsohne auch eine ‚Technikwissenschaft‘ wie beispielsweise der Maschinenbau, der Bergbau oder die Elektrotechnik. Diese Seite der Informatik dient dazu, Produkte, Werkzeuge und Medien zu entwickeln und zu konstruieren. Viele sehen darin ausschliesslich die Informatik und begründen daraus ausschliesslich — wenn überhaupt — eine schulische Berechtigung für das Schulfach Informatik.

Aber Informatik ist mehr: In der Informatik und durch das Wirken der Informatik wird das *informatische Denken* [Win06] benötigt und entwickelt:

## Definition: Informatisches Denken

*“Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone“. “Computational thinking is a fundamental skill for everyone, not just for computer scientists.“ [Win06, S.33]*

Betrachtet man die informatischen Teildisziplinen genauer, lassen sich immer wieder dieselben gedanklichen Strukturen finden:

*“Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system’s design for simplicity and elegance. [Win06, S.33]*

Abstraktion und Automation sind nach dieser Definition zentrale Bestandteile dieses informatischen Denkens, das sich durch alle Teilgebiete der Informatik zieht.

*Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. ... “ [Win06, S.33]*

In diesem Kapitel wird dargelegt, wie diese neue Wissenschaft mit ihrem neuen *informatischen Denken* auf die Disziplinen, aus denen sie einst hervorgegangen ist, zurück wirkt und wie sie noch auf andere Fächer wirkt. Wir stellen fest, wieweit diese Ursprungsdisziplinen und die anderen Fächer mit ihrem jeweils eigenen Denken mit dem informatischen Denken verwooben sind, sodass man diese Wissenschaften ohne die Fähigkeit, informatisch denken zu können, nicht mehr betreiben kann. Ebenso wird in diesem Kapitel deutlich, dass die Techniken und Konzepte, auf die sich neben der Informatik andere Fächer beziehen wie beispielsweise die *Algorithmik*, die von der Mathematik beansprucht wird, ohne die Sichtweise der Informatik heute nicht mehr vollständig betrieben werden können. Damit ergibt sich allein aus der Sicht der heute üblichen Schulfächer die Notwendigkeit eines informatischen Unterrichtes in der Schule.

Die in diesem Kapitel dargestellten informatischen Inhalte stellen in der Summe nicht das gesamte Spektrum der Informatik dar. Sie sind ein Ausschnitt aus den Bereichen, bei denen eine nicht unwesentliche Schnittmenge mit anderen Fächern mehr oder weniger offensichtlich ist. Daneben hat die Informatik noch vielfältige eigenständige Strukturen entwickelt, die in vielfältiger Weise das Denken und Arbeiten der Welt heute beeinflussen. Erinnerung sei hier nur an die *Datenstrukturierung* beispielsweise in Form der semistrukturierten Daten, wie wir sie im Umfeld des Internets vorfinden (*HTML, XML*). Auch *Datenbanken, Betriebssysteme, Netze* zählen zu den originären informatischen Teildisziplinen. Neben den Einflüssen auf andere Fachdisziplinen, die in diesem Kapitel zur Diskussion stehen, verlangen sie ein informatisches Denken, das dementsprechend auch schulisch angelegt werden muss.

## 2.2 Sprache und Sprachwissenschaften

Die Ursprünge der Wissenschaft Informatik zu Beginn des 20. Jahrhunderts entstammen im Wesentlichen aus den Wissenschaften *Mathematik* und *Physik*. Das bedeutet aber nicht, dass diese Wissenschaften ausschliesslich den Nukleus des neuen Faches bildeten. Auch wenn zur Frühzeit der Computer und der beginnenden Informatik weitgehend rein numerische Berechnungen betrachtet und bearbeitet wurden, wurden sehr bald *sprachliche* Überlegungen angestellt.

### 2.2.1 Programmiersprachen

#### Sprachkonzepte aus der ‚Frühzeit‘ der Informatik

Allererste Gedanken über eine Sprache, die im Umfeld einer Maschine und nicht im Umfeld von lebenden Geschöpfen eine Bedeutung hat, wurden bereits lange vor der Existenz einer realen Maschine formuliert.

Eine erste wegen mechanischer Probleme damals nicht realisierbare Maschine hat *Charles Babbage* mit der *Analytic Engine* 1837 gedanklich konstruiert. Seine Mitarbeiterin *Ada Lovelace* und Dokumentarin der Arbeit von *Charles Babbage* fügte diesen Dokumentationen mit ihren Überlegungen zur Steuerung von solchen Maschinen einen eigenen wesentlichen Beitrag zur Informatik bei. Mit der nach ihr benannten Sprache *Ada* wird ihre Bedeutung unterstrichen und gewürdigt. Diese Beschreibungen der Steuerung einer Maschine stellen die Grundzüge einer Programmiersprache dar.

Einer der Urväter der Computer, *Konrad Zuse*, hat sich neben der technischen Entwicklung der Hardware eines Computers schon früh mit sprachlichen Konstrukten für Computer beschäftigt. So legte er Anfang der 40er Jahre mit einem Konzept einer Sprache namens *Plankalkül* [Zus72] den Grundstein für den Aufbau von Programmiersprachen, auch wenn zu seiner Zeit kaum einer die Bedeutung seiner Arbeit erkennen wollte: „Mir schwebte damals eine universelle Sprache vor, mit deren Hilfe man sich mit einem künstlichen Gehirn unterhalten konnte. Ich verfiel zunächst auf das Esperanto, fand aber schnell heraus, daß es sich dabei nur um eine von grammatikalischen Mängeln bereinigte Umgangssprache handelte. Danach studierte ich Carnaps ‚Logische Syntax der Sprache‘; aber Carnaps Gedanken hielten sich zu einseitig an die Mittel der mathematischen Logik. Eine brauchbare Lösung mußte irgendwo zwischen diesen beiden Möglichkeiten liegen.“ [Zus10, S.73] Des Weiteren resümiert Zuse: „Vielleicht wäre aber auch die Zeit noch nicht reif für solche Gedanken gewesen; eine Fachwelt, die dafür Interesse hätte haben können, entstand erst viel später.“ [Zus10, S.75]

Zu sehr wurden diese ersten Kolosse wirklich nur als grosse Rechenmaschinen gesehen. Da ihre Zahl weltweit kaum zweistellig war, war auch eine generelle Bedeutung dieser Überlegungen über einen äusserst kleinen Kreis von Eingeweihten nicht erkennbar.

Gedanken zur Konstruktion einer maschinellen Sprache wurden dann mit einigem zeitlichem Abstand an der ETH Zürich wieder aufgenommen. *Heinz Rutishauser* entwickelte dort Ansprüche an eine (Symbol-)Sprache [Rut56], die als Vorläufer der imperativen Sprachen wie *Algol* aufgefasst werden kann. Da Rutishauser sehr früh erkrankte, konnte er diese Arbeiten leider nicht fortsetzen und abschliessen.

## Maschinen- und Assemblersprachen

Die stärkere Verbreitung von elektronischen Rechnern der 1. Generation<sup>1</sup> führte an den verschiedensten Orten zu Gedanken über die Verbesserung der Steuerung von Computern jenseits der Maschinenebene. Aus dieser praktischen Anforderung entstanden die verschiedenen Formen der *Assemblersprachen*, die aber noch mehr an den Bedürfnissen der konkreten Maschine als an sprachlichen Anforderungen der sie nutzenden Programmierer orientiert waren. Diese praktikable Herangehensweise war also völlig anders entstanden als die prinzipiellen Überlegungen von Zuse und Rutishauser.

## Problemorientierte Programmiersprachen

In der Praxis zeigte es sich dann, dass die Arbeit mit Assemblersprachen das Problem der Programmierung nicht löste. Die Programmierung war an eine konkrete Maschine gebunden und damit auch an die physikalische Existenz konkreter technisch-physikalischer Konstrukte wie beispielsweise Register oder Speicher. Damit war die Programmierung auch immer eine Orientierung an der Hardware und der Umsetzung des konkreten Problems mit einer konkreten Hardware und nicht eine Umsetzung eines konkreten Problems in einer allgemein gültigen Beschreibung.

Mit einer solchen Beschreibung wird eine neue Sichtweise der Arbeit mit derartigen Maschinen angewandt: Die Beschreibung kann als eine Kommunikation des Programmierers mit der Maschine aufgefasst werden. Die Maschine muss dann in die Lage versetzt werden, aus dieser Kommunikation die konkreten Anweisungen für sich zu konstruieren. Mit dieser Sichtweise stellen sich damit der Informatik zumindest teilweise Fragestellungen, die bis dahin den Sprachwissenschaften vorbehalten waren: *Syntax*, *Semantik* und *Pragmatik* einer Sprache waren nun Themen, die die frühen Informatiker beschäftigten.

Mit dem Entstehen dieser sogenannten *Programmiersprachen der 3. Generation* verschob sich der Fokus von der Hardware zum Problem. Ziel bei allen Konstrukten bei den Assemblersprachen war es, die Hardware optimal anzusprechen. Dazu waren im Laufe der Zeit eine Reihe von Strukturen entwickelt worden, die diese Assemblersprachen schon auf den ersten Blick von den Maschinensprachen entfernt haben: symbolische Bezeichner, Mnemoniks, Makros und Prozeduren, jedoch waren alle in Assembler geschriebenen Programme durch einen oder mehrere Übersetzungsvorgänge in einen äquivalenten Code in Maschinensprache umsetzbar.

Sicher haben die dort entwickelten Konzepte die Entwicklung der Sprachen der 3. Generation beeinflusst. Aber erst jetzt konnte und wollte man sich bei der Konzeption immer mehr von der vorhandenen Hardware entfernen. Je mehr dies vollzogen wurde, desto mehr Konstrukte und Ansprüche wurden entwickelt, die das zu lösende Problem in den Vordergrund gestellt haben. Verdeutlicht wird dies beispielsweise durch die Ablösung der von der Hardware beeinflussten Programmierung mit bedingten und unbedingten Sprüngen durch die strukturierte Programmierung mit blockorientierten Wiederholungsanweisungen. Diese Entwicklung und Auseinandersetzung um syntaktische Strukturen kann an der Geschichte der Entwicklung der

---

<sup>1</sup>Die wenigen Relais- oder andersartig gesteuerten Rechner müssten eigentlich mit *0. Generation* bezeichnet werden.

Programmiersprachen *Algol 58*, *Algol 60*, *Algol 68*, *Algol W* und *Pascal* [Bau12] nachvollzogen werden. Nicht unwesentlich hat *Niklaus Wirth* von der ETH Zürich die Definition dieser Sprachfamilie beeinflusst. Ebenso sind Konstrukte wie der *Keller* bzw. *Stack* beschrieben von *Friedrich Ludwig Bauer* in München [Bau07, S.21] nur aus der Sicht des Problems und nicht aus der Sicht der Hardware möglich. Damit sind rekursive Strukturen einfach beschreibbar und stellen ein weiteres Beispiel dar, wie sich die Sprachen nicht hardwareorientiert, sondern problemorientiert entwickelt haben.

## 2.2.2 Formale Sprachen

Die Diskussion um Programmiersprachen und dann allgemeiner formaler Sprachen verlangte Strukturen der Darstellung und Analyse. Ebenso war es notwendig, darüber nachzudenken, ob es verschiedene Klassen von Programmiersprachen gibt und in wie weit Programmiersprachen und allgemeiner formale Sprachen von natürlichen Sprachen zu unterscheiden sind oder nicht. An dieser Stelle entwickelte sich ein gemeinsames Interesse von Linguisten und Informatikern hinsichtlich der Forschungen bezüglich dieser Strukturen.

Diese Schnittmenge in der Forschung wird in der Person *Noam Chomsky* deutlich, der nach seiner Ausbildung und in seiner Forschung Linguist war. Chomsky untersuchte die Eigenschaften von formalen Sprachen und entwickelte daraufhin die nach ihm benannte grundlegende *Chomsky-Hierarchie* [Cho56], Chomsky1959, [Weg05, S.128], [FB96, S.321] von Sprachen. Diese Charakterisierung von Sprachen klassifiziert die Sprachen in vier unterschiedliche Klassen:

**Chomsky 0:** allgemeine oder berechenbare Sprachen

**Chomsky 1:** kontextsensitive oder entscheidbare Sprachen

**Chomsky 2:** kontextfreie Sprachen

**Chomsky 3:** reguläre Sprachen

Diese Klassifizierung, den Bedürfnissen der Informatik entsprechend, von einem Linguisten mit auch informatischem Interesse herausgearbeitet, wirkt damit auf die linguistischen Forschungen in der formalen Linguistik zurück. Im Kapitel 4.1.3 über den *Kognitivismus* wird genauer beschrieben, dass einer der Ausgangspunkte Chomskys war, dass das Erlernen der Sprache kein Zufallsprodukt ist, sondern schon beim geborenen Kinde angelegt ist. Wenn dem so ist, mussten diese im Gehirn angelegten Strukturen auch in der Sprache wiederfindbar sein.

## 2.2.3 Natürliche Sprachen

Mit dem Aufkommen der Informatik wurde damit auch eine intensive Forschung in der Linguistik auf dem Gebiet der formalen Linguistik ausgelöst. Dazu gehört selbstverständlich auch die Darstellung der Syntax von formalen Sprachen. In der Linguistik wird bei der Beschreibung von Grammatiken unterschieden zwischen ...

- ... deskriptiver Grammatik und
- ... normativer oder präskriptiver Grammatik

Grammatiken beider Gruppen werden auch als *kodifizierte Grammatiken* zusammengefasst [Vol97, S.12/13]. Natürliche Sprachen entziehen sich zumindest einer vollständigen Beschrei-

bung. Dagegen ist die *innere Grammatik* ein „Regelsystem, das von allen Sprechern einer Sprache (in der Kindheit) erworben wird.“ [Vol97, S.13]

Mit dem Auftreten formaler Sprachen wie Programmiersprachen entstand auch ein Interesse nach Analyse der Grammatiken von natürlichen Sprachen. Die Weiterentwicklung führte bei Chomsky zur *Generativen Grammatik* [Cho73, S.29ff]. Mit dieser Grammatik behaupten die Befürworter, dass das Vermögen, eine Sprache zu lernen, nicht durch Training erfolgt, sondern durch kognitive Strukturen im Gehirn vererbt wird. Mit dieser Annahme bezieht Chomsky über die Linguistik hinaus eine bildungstheoretische Position: Er lehnt damit den *Behaviorismus* (siehe dazu Kapitel 4.1.2 als Grundstruktur des menschlichen Gehirns ab und befürwortet dadurch *kognitive Strukturen* und begründet damit den *Kognitivismus* (siehe dazu Kapitel 4.1.3). Diese Überlegungen in der Schnittmenge u.a. von *Sprachwissenschaften*, *Informatik* und *Lernpsychologie* zeigen beispielhaft die Verwobenheit verschiedener Disziplinen bei der Erkenntnisgewinnung.

Chomsky beruft sich auf Humboldt<sup>2</sup> und erläutert:

„Das heißt, die *Form einer Sprache*, das Schema für ihre Grammatik, ist zu einem sehr großen Teil vorgegeben, obwohl es nicht verfügbar ist für den Gebrauch ohne geeignete Erfahrung, die den Prozeß der Sprachformung in Gang setzt. Er wiederholt wie Leibniz die platonistische Auffassung, daß für das Individuum Lernen weitgehend eine Sache der *Wiedererzeugung* ist, also der Ausprägung dessen, was der Psyche schon eingeboren ist.

Diese Auffassung steht schroff der empiristischen Vorstellung (der vorherrschenden modernen Auffassung) gegenüber, daß die Sprache eine zufällige Konstruktion ist, die gelernt wird durch „Konditionierung“ [...].“ [Cho73, S.73]

Diese entscheidende Veränderung in der Sichtweise wird dementsprechend auch *kognitive Wende* [DFN03] genannt.

Diese Absage an empiristische Vorstellungen des Spracherwerbs durch die Anerkennung von kognitiven Strukturen besteht auch weiterhin, wenn auch die Erkenntnis des Einflusses der zwischenmenschlichen Interaktion beim Spracherwerb durch die Forschungen des Verhaltensforschers *Michael Tomasello* [Tom05] et al. die Bedeutung der kognitiven Strukturen eingeschränkt hat. So schreibt Tomasello:

*'This view is represented by a group of theories most often called cognitive-functional linguistics but sometimes also called usage-based linguistics to emphasize their central processing tenet that language structure emerges from language use [...]. Usage-based theories hold that the essence of language is its symbolic dimension, with grammar being derivative'. [Tom05, S.5]*

Nun wird niemand bestreiten wollen, dass der konkrete Erwerb einer Sprache in Interaktion mit anderen Menschen erfolgt, da ansonsten alle Menschen dieselbe Sprache sprechen müssten. Es bleibt damit aber noch nicht geklärt, ob und welche grammatikalischen Strukturen kognitiv angelegt sind. Der österreichische Germanist *Thomas Fritz* setzt sich im von

---

<sup>2</sup>„Unser Nachdenken entdeckt in ihr [*der Sprache*] eine unserer Geistesform zusagende *Gesetzmäßigkeit*“. [Hum36, S.60]



den Sprachwissenschaftlern *Ruth Eßer*, *Hans Jürgen Krumm* und *Hans Barkowski* herausgegebenen Buch *Bausteine für Babylon: Sprache, Kultur, Unterricht* mit den beiden scheinbar widersprechenden Positionen auseinander und schreibt: „Der Nativismus ist dem Universalismus verbunden, indem er allen Menschen die gleiche Ausstattung an kognitiver Fähigkeit und vor allem an Spracherwerbsfähigkeit zuspricht, der Konstruktivismus hingegen betont den Individualismus, die persönliche und im sozialen Geflecht entstehende Konstruktion von Bedeutung und in der Folge von Struktur, dem was Tomasello *emerging grammars* nennt.“ [EBK07, S.21]

## Darstellungen von Grammatiken

Wie bereits oben erwähnt, können ähnliche weitreichende Überlegungen mit der Entwicklung formaler Sprachen in der Informatik angestellt werden. Aus Sicht der Informatik war für die Übersetzung von einem Text in einer beliebigen Sprache in Maschinensprache mit Hilfe eines Compilers oder alternativ für die direkte Ausführung mit Hilfe eines Interpreters die Analyse und Beschreibung formaler Sprachen essentiell. Wesentliche Ideen mündeten dann in die *Backus-Naur-Form*, die dann 1959 in einer Vorform in der Beschreibung der Syntax und Semantik der Sprache Algol verwendet wurde [Bac59] (siehe Abbildung 2.1 und die dann endgültig im abschliessenden Bericht über Algol angewandt wurde [Nau63]. Für den Benutzer beziehungsweise einen Lernenden einer solchen Sprache haben sich die daraus abgeleiteten *Syntaxdiagramme* als sinnvolle Hilfe erwiesen. Diese wurden offensichtlich erstmals von Niklaus Wirth im revidierten Bericht der Sprache Pascal eingesetzt [Wir72, S.47ff]<sup>3</sup>

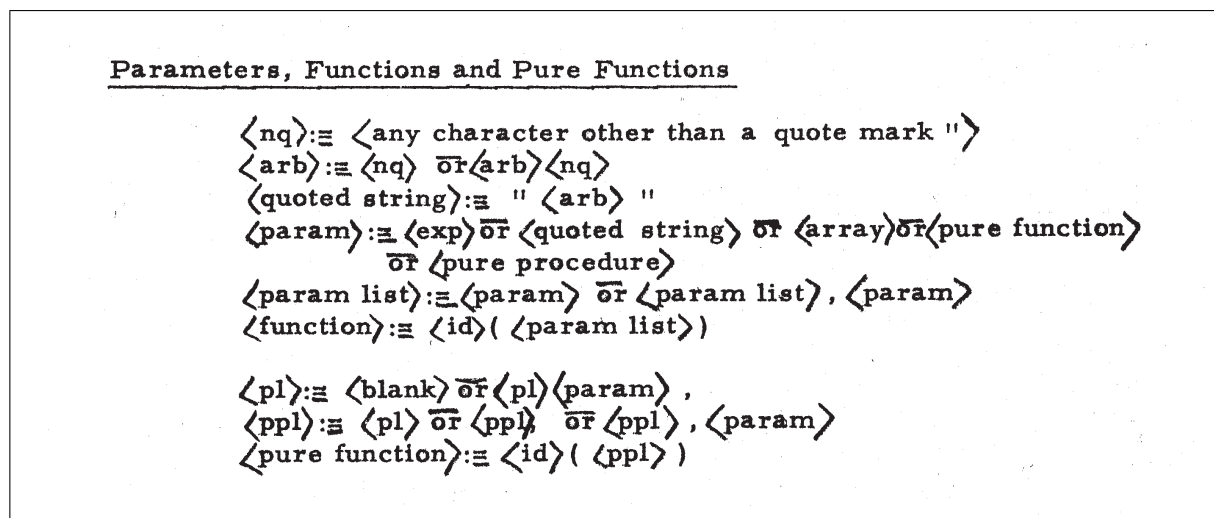


Abbildung 2.1: Auszug aus der Grammatik der Sprache *Algol* von Backus 1959

<sup>3</sup>Alle gefundenen Hinweise (Beispiel: [http://programonce.blogspot.com/2010\\_02\\_01\\_archive.html](http://programonce.blogspot.com/2010_02_01_archive.html) (aufgerufen: 15.3.2012)) auf den ersten Gebrauch von Syntaxdiagrammen beziehen sich immer auf die Sprachdefinitionen von Pascal [Wir72] und [JW74]. Es erhebt aber niemand die Urheberschaft auf die Idee der Syntaxdiagramme.

## 2.2.4 Fazit Sprache und Sprachwissenschaften

Das entstehende Fach Informatik hat neue, zuvor nicht bekannte bedeutsame Strukturen aus ihrer eigenen Notwendigkeit heraus entwickelt und damit das geistige Ideenbild nicht nur der eigenen, sondern auch das der Sprachwissenschaften beeinflusst. Sprachwissenschaftler benötigen *informatisches Denken*. Damit sind diese nicht nur für Informatiker von essentieller Bedeutung. Der Psychologe und zeitweilige Präsident der *American Psychological Association* George A. Miller beschreibt die Entwicklung in den unterschiedlichen Wissenschaften inklusive der entstehenden Informatik: “*Cognitive science is a child of the 1950s, the product of a time when psychology, anthropology and linguistics were redefining themselves and computer science and neuroscience as disciplines were coming into existence.*“ [Mil03]

## 2.3 Algorithmik

*Algorithmen* sind kein neues Thema, das von der Informatik zuerst untersucht wird. Sie stehen heute mit im Zentrum informatischer Betrachtungen und tragen einen nicht unwesentlichen Teil zum informatischen Denken bei. Daher ist es sinnvoll, zu hinterfragen, welchen Anteil informatische und nicht-informatische Anteile in der Algorithmik zu finden sind. Dazu bietet es sich an, auch hier einen historischen Blick in die Geschichte der Algorithmik aus verschiedenen Blickwinkeln zu werfen.

### 2.3.1 Algorithmik und Sprache

Mit den problemorientierten Programmiersprachen verschiebt sich die Beschreibung der Handlung als Steuerung eines Automaten hin zur Beschreibung eines Problems, das unabhängig vom konkreten Automaten gelöst werden soll. Genau wie die Sprache eng mit dem Denken über die betrachteten Objekte, die sie beschreibt, zusammenhängt, sind die Programmiersprachen mit den betrachteten Problemen verknüpft. Mit diesen Programmiersprachen konnten in den folgenden Jahrzehnten die verschiedensten Problemklassen analysiert werden. Die Geschwindigkeit, mit der dieser Prozess verlief, wird beispielsweise daran deutlich, dass das Suchverfahren *Quicksort*, das im Jahre 1962 von *Charles Antony Richard Hoare* wissenschaftlich veröffentlicht wurde [Hoa62], bereits wenig mehr als 15 Jahre später in einem Buch zum Selbststudium für Programmieranfänger enthalten ist [Sch79]. Wenn heute schon für Anfänger geeignetes Material 20 Jahre zuvor nur dem Wissenschaftler zugänglich und verständlich war, hat es ausgehend von einem sehr niedrigen Niveau einen gewaltigen Zuwachs an Erkenntnis und Ideen gegeben. Dabei ist die *Algorithmik* als Fachdisziplin nichts Neues. Allein die Abstammung des Wortes aus dem Namen *Muhammed al-Chwarizmi*, der etwa um das Jahr 800 gelebt hat, lässt erahnen, dass *Algorithmen* nicht eine Erfindung des 20. Jahrhunderts sind. Algorithmen sind früher in erster Linie Betrachtungsgebiet der Mathematik gewesen. So wird beispielsweise entsprechend in den *Richtlinien des Landes NRW für das Unterrichtsfach Mathematik in der Sekundarstufe II* die Idee des Algorithmus als eine von sieben wesentlichen Ideen der Mathematik dargestellt:

### „Idee des Algorithmus

Von der Grundschule bis in die gymnasiale Oberstufe hinein spielt das Arbeiten mit Algorithmen eine entscheidende Rolle. Ein mathematischer Algorithmus ist nichts anderes als eine standardisierte Problemlösung für eine Klasse strukturell verwandter Probleme, die man durch gedankliche Anstrengung und mit hinreichend viel Sachverstand auch jedes für sich, d. h. ohne diesen Algorithmus lösen könnte. Insofern spiegelt sich im Algorithmus das Prinzip der industriellen Fertigung.“ [Min99b, S.10]

Offensichtlich wird die Algorithmik aber doch in erster Linie als Verwertungsinstrument anderer mathematischer Ideen und Ergebnisse angesehen. So werden als Beispiele *Lösen von Gleichungssystemen* und *numerisches Integrieren* angegeben. Algorithmik versteht sich in der Mathematik in erster Linie als das Lösen mathematischer Terme und Gleichungen. Es wird im wahrsten Sinne des Wortes der Computer mit seinen Programmen als reine Rechenmaschine gesehen. Die Beschreibung des konkreten Algorithmus in einer konkreten Sprache spielt eigentlich keine Rolle.

Die Bedeutung der Algorithmik dagegen, wie Informatiker sie verstehen, ist vor allem die *Beschreibung* der Lösung von verschiedensten Problem(-Klassen) [CS86, S.22]. Es wird dabei keine Beschränkung der Problemklassen vorgenommen. Eine der zentralen Fragen aus Sicht der Informatik ist, ob es eine oder keine Lösbarkeit einer konkreten Problemklasse gibt. Zudem spielt die dafür aufzuwendende Zeit eine erhebliche Rolle.

Das Wort *Beschreibung* macht es deutlich: Zum *Beschreiben* wird eine *Sprache* benötigt. Mit den problemorientierten Programmiersprachen stehen erstmals in der Geschichte Sprachkonstrukte zur Verfügung, mit denen diese Aufgaben zufriedenstellend bewältigt werden können. Neben der *Syntax* und der *Semantik* wird nach dem Sinn, der *Pragmatik* von (Programmier-)Sprachen gefragt. Auch wenn die Mathematik zweifelsohne Algorithmen betrachtet, so wird erst durch die Informatik mit der Definition und Anwendung von Programmiersprachen und deren Anwendung bei der Darstellung von Algorithmen die Bedeutung der Algorithmik ausreichend gewürdigt.

Mit diesen Problembeschreibungssprachen wird zudem (teilweise) zum ersten Mal bewusst, wie der Mensch prinzipiell schon seit langer Zeit manche Problemstellung löst. Die Notwendigkeit, eine Handlung einschliesslich aller scheinbaren Nebensächlichkeiten korrekt in einem Programmtext aufzuschreiben, damit sie ein Automat tatsächlich lösen kann, hat manche unbewusste Tätigkeit durchschaubar gemacht. Als Beispiel diene erneut das ‚Sortieren‘: Das Verfahren, mit dem sinnvollerweise ein Stapel Karten sortiert wird, hängt von der Anzahl der Karten ab. Bei einer kleinen Kartenanzahl werden Sortieralgorithmen wie *Einfügsort* etc. verwendet, bei einer grossen Anzahl Verfahren, die als Anwendung der Idee des *Quicksort* angesehen werden können. Und trotzdem hat erst Hoare 1962 die Idee dieses Verfahrens formuliert. Wenn *Sprache* und *Denken* einen Zusammenhang haben, dann hat sich durch die Existenz und die Anwendung von *Programmiersprachen* der Wissenshorizont wesentlich erweitert. Gleichzeitig erwachsen der Sprachforschung mit den *Programmiersprachen* eine neue Gruppe von Sprachen, die es zu untersuchen gilt. Die Mathematik als ‚*Ursprungsmutter*‘ der *Algorithmik* hat keinen Zugang zu dieser Sprachebene und kann hier nicht mehr weiterhelfen. Hierzu ist das *informatische Denken* notwendig.

## 2.3.2 Algorithmik und Mathematik

Schon *Konrad Zuse* hatte in seinen Überlegungen zur Sprache festgestellt, dass eine Sprachebene zwischen mathematischer Logik und natürlicher Sprache notwendig ist, um algorithmische Probleme zu beschreiben [Zus10, S.72ff]. Gemeint sind damit algorithmische Beschreibungen jenseits von der Formulierung bestimmter konkreter Berechnungen wie beispielsweise das numerische Integrieren. Zuse erwähnt in seinen Aufzeichnung für seine geplante Dissertation: „In der Wirtschaft ist beispielsweise die Berechnung des Nettolohns keine reine Zahlenrechnung, da hier eine Reihe von Umständen (ist verheiratet, hat Kinder, ist kriegsbeschädigt, ist wehrsteuerpflichtig, hat Überstunden gemacht, usw.) mitsprechen, die erst nach bestimmter Kombination das Resultat ergeben.“ [Zus10, S.74]

Mit derartigen Fragestellungen beschäftigt sich die Mathematik nicht. Die zu behandelnden Probleme werden damit auch immer umfassender und verlangen entsprechend nach einer geeigneten Darstellung. Der Begriff des *Algorithmus* selbst gerät in den Fokus mit den zentralen Fragen: Sind alle Probleme überhaupt algorithmisch beschreibbar, gibt es einige grundlegenden Prinzipien, die für Algorithmen generell gelten?

Die Formulierung von Algorithmen hat sich mit den Programmiersprachen der 3. Generation von einer konkreten Maschine gelöst. Damit war es möglich, die adäquate Form der Beschreibung zu wählen, die dem Problem gerecht wird. Das können auf der einen Seite *iterative* oder *rekursive* Beschreibungen sein. Auch wenn sich oft für ein konkretes Problem eine Beschreibungsart aufzwingt, bleibt die Frage, ob bestimmte Algorithmen nur iterativ oder nur rekursiv dargestellt werden können.

Es ist nicht ganz selbstverständlich, dass es für jede iterative Beschreibung eine adäquate rekursive Beschreibung gibt. Diese Erkenntnis ist der ‚Schlusspunkt‘ einer langen Forschung, die in wesentlichen Zügen parallel zur theoretischen und praktischen Grundsteinlegung der Informatik in den ersten 40 Jahren des zwanzigsten Jahrhunderts stattgefunden hat. Sie findet im Umfeld der Diskussion um die Widerspruchsfreiheit der Mathematik und der Entwicklung der theoretischen Maschine von *Alan Mathison Turing* statt. Alan M. Turing [Tur36] und *Emil Leon Post* [Pos36] haben 1936 die *berechenbaren Funktionen* untersucht.

Die Möglichkeit, Algorithmen sowohl iterativ als auch rekursiv zu beschreiben, war nicht nur von theoretischem Interesse. Auch wenn beispielsweise eine konkrete Sprache keine Rekursion zulässt, ist es möglich, damit ein Problem zu lösen, für das zunächst nur eine rekursive Beschreibung vorliegt und umgekehrt.

Die Ergebnisse dieser Arbeiten flossen in die Entwicklung der Sprache *LISP* ein. Diese wurde hauptsächlich von *John McCarthy* [McC60, McC63] geleistet. Mit der Sprache *LISP* wurde zum ersten Mal eine Sprache definiert, die nicht wie alle Programmiersprachen zuvor ablauforientiert ist, sondern deklarativ ist. Dieses bedeutete eine weitere Stufe der Abstraktion.

Aus den Arbeiten mit *LISP* und den zugrunde liegenden *rekursiven Funktionen* wurde gezeigt, dass genauso eine Problem beschrieben werden kann wie mit den ablauforientierten Sprachen. Daraus folgt die Gleichmächtigkeit der *rekursiven* und der *iterativen* Beschreibungen.

Es versteht sich von selbst, dass dies aus Sicht der Informatik von grossem Interesse war, da aufgrund dieser Gleichmächtigkeit unterschiedlichste Programmiersprachen für die verschiedensten Zwecke entwickelt werden konnten. Dies ermöglicht, ein Problem algorithmisch

jeweils in der Sprache zu formulieren, die aus Sicht des Problems am geeignetsten ist.

Diese intensiven Forschungen und die bahnbrechenden Ergebnisse der Informatik an der Schnittstelle von Mathematik und Informatik wären ohne die Anforderungen aus der Informatik so nicht erfolgt. Die gerade entstehende Disziplin Informatik produziert bereits Ergebnisse, die einen entscheidenden Fortschritt in der Wissenschaft Mathematik bedeuten. Von Seiten der Mathematik werden diese Erkenntnisse des Jahres 1936 zu Recht entsprechend gewürdigt: „Fast ist man versucht, das Jahr 1936 als das Geburtsjahr der theoretischen Informatik zu bezeichnen.“<sup>4</sup>

Entscheidende Erkenntnisse über den Begriff Algorithmus und seiner Eigenschaften ergab jedoch nicht nur die Untersuchung von rekursiven und iterativen Formulierungen. Auch die Art und Weise, wie iterative Algorithmen formuliert und programmiert werden sollen, wurde von Seiten der Informatik intensiv erforscht:

Der Übergang von den maschinenorientierten Sprachen zu den problemorientierten Sprachen der 3. Generation ermöglichte die Loslösung von den Strukturen konkreter Hardware, die praktisch in die Programmiersprache übertragen wurde<sup>5</sup>. Aus Sicht der Hardware ist der *Sprungbefehl* eine der wesentlichen technischen und sprachlichen Strukturen, die eine universelle Maschine entsprechend der *von-Neumann-Architektur* ermöglicht haben. Aus programmiersprachlicher Sicht produziert die unkontrollierte Anwendung dieser Möglichkeit bei grossen Programmen mit hoher Wahrscheinlichkeit eine grosse Fehlerquote und kaum überprüfbare Programme. Die Analyse dieser Problematik führte letztendlich zur *strukturierten Programmierung*, die unter anderem den hardware-orientierten Sprung durch Wiederholungsanweisungen ersetzt. *Edsger Wybe Dijkstra* schreibt 1969:

*'In particular: when programs for a sequential computer are expressed as a linear sequence of basic symbols of a programming language, sequencing should be controlled by alternative, conditional and repetitive clauses and procedure calls, rather than by statements transferring control to labelled points.'* [Dij69]

Ein Jahr zuvor fordert er entsprechend

*'[...] I became convinced that the go to statement should be abolished from all higher level programming languages [...]'* [Dij68].

*Ole Johan Dahl*, *Edsger Wybe Dijkstra* und *Charles Antony Richard Hoare* fassen ihre Überlegungen 1972 in dem Buch *Structured Programming* [DDH72] zusammen, in dem Dijkstra Positionen zur strukturierten Programmierung, Hoare zu Datentypen sowie Hoare und Dahl zu hierarchischen Programmkonzepten beziehen. Zu diesen Konzepten zählen unter anderem auch Klassen [DDH72, S.179]. Es überrascht heute, warum diese Idee der *Klassen* nicht schon in der Programmiersprache *Pascal* [Wir71] realisiert wurden.

Diese Überlegungen und Ergebnisse über Eigenschaften von Algorithmen sind direkt aus den Forschungen in der Informatik ableitbar. In der Informatik ist die Korrektheit von Programmen eine zentrale Fragestellung. Aus der Forderung nach Korrektheit von Programmen erfolgen konkrete Anforderungen, mit welchen Programmstrukturen ein Algorithmus zu

<sup>4</sup><http://www.mathematik.de/ger/information/kalenderblatt/turingmaschinen/turingmaschinen.html>, zuletzt überprüft: 26.3.2012

<sup>5</sup>Einen Einblick in die historische Entwicklung gibt *Friedrich Ludwig Bauer* als damals Beteiligter in *Die ALGOL-Verschwörung* [Bau12]

schreiben ist. Die Schwierigkeit, korrekte Programme zu schreiben, wächst mit dem Umfang. Einen kritischen Umfang konnten Algorithmen jedoch erst mit dem Auftreten von Computern erhalten. Insofern hat sich die Wissenschaft Mathematik auch nicht um derartige Fragen bei der Thematik *Algorithmik* beschäftigt.

## Komplexität

Neben der Frage, mit welchen programmiersprachlichen Mitteln ein Algorithmus geschrieben werden soll, ist die Frage nach der Laufzeit eines Algorithmus bzw. eines konkreten Programmes in der Informatik eine zentrale Frage. Natürlich hat auch die Mathematik bei der Aufstellung von Algorithmen die Problematik der Laufzeit zu berücksichtigen<sup>6</sup>. Aus historischer Sicht waren die Kriterien allerdings völlig andere als die heute in der Informatik.

In der Mathematik interessiert häufig nur die theoretische Existenz und Lösbarkeit für ein Problem ohne Berücksichtigung der dafür in der Praxis benötigten Zeit. Für konkrete Berechnungen werden in der Mathematik sehr ungerne Algorithmen verwendet. Die Mathematik bevorzugt Gleichungen und Terme. Zumeist wird nur bei Fehlen einer derartigen Lösung auf einen Algorithmus als Berechnungshilfe zurückgegriffen. Als Beispiel diene das *Heron-Verfahren* für die Berechnung einer Quadratwurzel einer Zahl  $n$ . Dieses Verfahren konvergiert schneller als das *Halbierungsverfahren* für das Intervall  $I = (0, n)$ . Bei der Berechnung ohne Computer ist das nicht unwichtig. Die Anzahl der zu bearbeitenden Schritte möglichst zu verringern, ist bereits bei wenig Rechenaufwand eine sinnvolle Aufgabe und Herangehensweise, da schon wenige Berechnungen je nach Aufgabe die Machbarkeit eines Verfahrens in Frage stellen können. Im Gegensatz zu den meisten algebraischen Berechnungen werden geometrische Konstruktionen durch ein algorithmisches Vorgehen erstellt. Sofern diese Konstruktionen händisch durchgeführt werden sollen, ist jede Reduktion auch um relativ wenige Konstruktionschritte ein deutlicher Gewinn. Die Analyse derartiger geometrischer Konstruktionsalgorithmen waren damit Thema mathematischer Forschung. So beschäftigte sich beispielsweise *Émile Lemoine* im 19. Jahrhundert mit derartigen Problemen und konnte beim *apollonischen Problem* nachweisen, dass mit der von ihm gewählten Herangehensweise eine Reduktion von den damals bekannten notwendigen 479 Schritten auf 199 Schritte eine erhebliche Zeitverbesserung erzielt werden konnte [Coo40, S.59].

Die zeitliche Dimension bei der Lösung mit einem Algorithmus unter Nutzung eines Computers erweist sich als völlig neue Fragestellung, weil trotz der scheinbaren ‚Schnelligkeit‘ von Computern beileibe nicht alle Probleme (sinnvoll) gelöst werden können. Probleme mit einem hohem zeitlichen Aufwand waren in der Zeit vor der Existenz von Computern praktisch unlösbar. Damit wird das *Zeitproblem* einer Lösung paradoxerweise eine informatische Fragestellung.

Informatiker fragen nach der *Komplexitätsklasse* von Algorithmen, um die Sinnhaftigkeit eines konkreten Algorithmus für ein gegebenes Problem beurteilen zu können, da die Durchführung eines jeden Programmes *Zeit* kostet. Dieser Zeitaufwand hängt dabei oft von einer konkreten Grösse  $n$  aus dem Problemraum ab. Die dabei eventuell kritische Grösse eines  $n$  wird in der Zeit vor der Existenz von Computern nicht erreicht.

---

<sup>6</sup>Algorithmen, die als Existenzbeweis z.B. bei Betrachtung von Folgen aus mathematischer Sicht interessant sind, werden hier nicht betrachtet.

Zentrale Fragen dazu wurden allerdings schon um die Jahrhundertwende gestellt: Wie viele andere glaubte auch *David Hilbert* an die Möglichkeit, die Mathematik *axiomatisch* begründen zu können und daran, dass alle formulierbaren Aussagen als falsch oder wahr mit Hilfe eines Algorithmus bewiesen werden können. [Hro08, S.318]. *Kurt Gödel* zeigte 1931 im berühmten *Unvollständigkeitssatz*, dass dieses nicht möglich ist und vor allem: „Es gibt keine Methode (keinen Algorithmus) zum automatischen Beweisen mathematischer Sätze“ [Hro08, S.320]. Das war in jeder Hinsicht eine *revolutionäre* Entdeckung in und für die Mathematik. Damit stellte sich natürlich sofort die Frage, welche Aufgaben(klassen) algorithmisch lösbar waren und welche nicht. Diese Diskussionen führten unter anderem zur Definition von *Turing-Maschinen* und zum *Halteproblem* [Tur36].

Mit der Diskussion um algorithmisch lösbare Probleme war die Frage verbunden, welche von diesen lösbaren Problemen auch *praktisch lösbar*, also in einer *sinnvollen* Zeitspanne lösbar sind. Die dabei wichtigste zeitlichen Grössenordnung ist die nach der *polynomialen* Zeit. Diese Algorithmen können von einer *deterministischen Turingmaschine (DTM)* entschieden werden und werden als Klasse  $\mathfrak{P}$  der *polynomialen Algorithmen* zusammengefasst. Die Algorithmen, die von einer *nichtdeterministischen Turingmaschine (NTM)* entschieden werden können, gehören der Klasse  $\mathfrak{NP}$  der *nicht-polynomialen Algorithmen* an. Diese können in *exponentieller* Zeit gelöst werden. Eines der ‚Grundprobleme‘ der Informatik wird darin gesehen, ob

$$\mathfrak{P} \subset \mathfrak{NP}$$

oder ob eine NTM durch eine DTM dargestellt werden kann und damit sich

$$\mathfrak{P} = \mathfrak{NP}$$

erweist [Weg05, S.43ff], [Hro10, S.231ff]. Aber selbst polynomiale Algorithmen sind oft zeitkritisch, wenn der Zeitaufwand für einen Algorithmus  $A$  stärker als linear wächst, also  $\mathcal{O}(A) > n$  ist. Diese Überlegungen sind neben den grundsätzlichen Überlegungen zur Berechenbarkeit [Hro11] ein weiterer grundlegender Teil der *theoretischen Informatik*.

### 2.3.3 Fazit Algorithmik

Die Wurzeln der Algorithmik entstammen zweifelsohne aus der Mathematik. Nachdenken über Algorithmik und Algorithmen führt jedoch über die bisherigen Überlegungen in der Mathematik hinaus und dient mit dem Aufkommen der Computer als Werkzeuge für das maschinelle Rechnen als theoretische Grundlage für die Möglichkeiten und Unmöglichkeiten in der Informatik. Gleichzeitig wirken die Überlegungen zum und über den Algorithmus unter anderem aufgrund der *Gödelschen Aussage* zentral auf die Mathematik zurück und sind für die heutige Mathematik unverzichtbar.

Ohne die oben dargestellten Überlegungen um und über den Algorithmus wären wichtigste erkenntnistheoretische Überlegungen in der Mathematik nicht möglich und sie gehören für jeden Mathematiker zum grundsätzlichen Rüstzeug.

## 2.4 Mathematik

Im vorigen Kapitel ging es um den Begriff des *Algorithmus*, der von der Mathematik als in erster Linie mathematischer Begriff verstanden wird. Wir haben gesehen, dass die Mathematik Algorithmen benutzt, verwendet und auch sucht, dies allerdings eigentlich nur dann bewerkstelligt, wenn andere, aus Sicht der Mathematik bessere und elegantere Methoden nicht zur Verfügung stehen. Im Folgenden betrachten wir ergänzend weitere Gebiete der Mathematik, vor allem Teildisziplinen, die als im Kern der Mathematik angesehen werden.

Die auch in der Schule (zumindest teilweise) vermittelten Gebiete der Analysis sind durch die Existenz der Informatik nicht verändert worden. Das ist auch naheliegend, ist doch gerade die Analysis deswegen entstanden, weil zur Zeit ihrer Entwicklung keine geeigneten Werkzeuge zur numerischen Berechnung zur Verfügung standen und daher Verfahren, Methoden und Begriffe gesucht wurden, um beispielsweise physikalische Fragestellungen mit möglichst wenig konkreten Berechnungen lösen zu können. Diese Anforderung führte von der diskreten Berechnung weg hin zur Infinitesimalrechnung. Zweifelsohne war dies eine hervorragende Leistung der Mathematik zu der damaligen Zeit. Ähnliches gilt auch für Gebiete wie beispielsweise Funktionentheorie, Funktionalanalysis, Zahlentheorie. Die Algebra — in der Schule ansatzweise in der linearen Algebra angeschnitten — ist allerdings mit einem wichtigen Gebiet ergänzt worden.

### 2.4.1 Numerik

Numerische Methoden gibt es nicht erst seit dem Auftreten der Computer. erinnert sei hier nur an das *Gauß'sche Eliminationsverfahren*. Aufgrund der technischen Entwicklung vor allem im Maschinenbau wurden umfangreiche Berechnungen nötig. In der Festschrift zum 100-jährigen Jubiläum der DMV *Ein Jahrhundert Mathematik 1890 – 1990* schreibt der Numeriker *Lothar Collatz*:

„Die starke Entwicklung der naturwissenschaftlichen und technischen Disziplinen im 19. Jahrhundert führte zu wachsend komplizierteren mathematischen Problemen, die meist nicht mehr in ‚geschlossener Form lösbar‘, sondern nur noch numerisch angreifbar waren. So waren auf dem Gebiet der gewöhnlichen Differentialgleichungen Anfangswertaufgaben über lange Zeitintervalle numerisch zu integrieren., wie z.B. bei Bahnberechnungen in der Astronomie.“ [FMV90, S.272]

Mit dem Auftreten der Computer ist die Notwendigkeit der Anwendung der Differential- und Integralrechnung bei konkreten Anwendungsaufgaben oft nicht mehr gegeben. Die Anzahl der numerischen Berechnungen im Rahmen einer gewünschten Genauigkeit stellt kaum ein Problem mehr dar. Entsprechend entwickelte sich die *numerische Mathematik* zu einer umfangreichen Teildisziplin der Mathematik. „Zu dieser Entwicklung hat wesentlich auch die geradezu stürmische Expansion auf der Sektor der Computerwissenschaften beigetragen.“ [FMV90, S.269] Noch deutlicher formulieren es die Mathematiker *Manfred Feilmeier* und *Hansjoerg Wacker*:

„[...] erlebte die Numerische Mathematik durch die etwa 1950 beginnende Verbreitung digitaler Rechenanlagen einen entscheidenden Aufschwung. [...] Dazu mußten eine Reihe



neuer numerischer Verfahren entwickelt und schon bekannte Verfahren modifiziert werden. [...] *Numerische Mathematik, wie wir sie heute verstehen, bedeutet Mathematik auf digitalen Rechenanlagen.*“ [FW75, S.15]

Entsprechend sind heute selbst in jedem Taschenrechner vielfältige Verfahren und entsprechende Algorithmen realisiert. Die durch die Informatik ermöglichten Werkzeuge, die Computer, haben daher von Anbeginn der Informatik die (numerische) Mathematik wesentlich beeinflusst und erweitert.

## 2.4.2 Statistik

Entsprechend liegt nahe, dass die Disziplinen in der Mathematik, die intensive numerische Berechnungen verlangen, von den Produkten und Ideen der Informatik profitieren. Das gilt auch für die *Statistik*, die sich Ende des 19. Jahrhundert als eigenständige Disziplin etablierte [FMV90, S.781]. Vielfältige Verfahren waren zuvor nur eingeschränkt nutzbar und konnten nur durch umfangreiche Tabellenwerke als umständliches Hilfsmittel benutzt werden. Eine Alternative dazu waren und sind stetige Verteilungen. Diese können jetzt verwendet werden, wenn es die Sachlage verlangt und nicht, weil es mangelnde Werkzeuge förmlich erzwingen. Dadurch konnte die Statistik enorm entwickelt werden.

Dieser innermathematische Fortschritt ist nicht nur allein als Folge der Entwicklung der Werkzeuge in Form von Hardware und Software durch den (in-)direkten Einfluss seitens der Informatik möglich geworden. Die Möglichkeiten der Anwendung der numerischen Mathematik sowie die Notwendigkeit innerhalb der Statistik haben hier zu einer enormen Weiterentwicklung geführt. Der Statistiker *Hermann Witting* (Präsident der DMV 1978/79) schreibt:

„Außerdem ist zu erwähnen, daß die Fortschritte in der *elektronischen Datenverarbeitung* grundlegend neue Möglichkeiten geschaffen haben und zwar nicht nur für die praktisch-statistische und datenanalytische Arbeit, sondern auch für theoretische Untersuchungen, etwa durch den Einsatz neuer rechenintensiver Verfahren. [...] Andererseits haben schnelle Rechner neue Möglichkeiten für Theorie und Anwendungen geschaffen, was bereits jetzt einen deutlichen Einfluß auf die in der theoretischen Statistik bearbeiteten Fragestellungen hat. Die langfristigen Auswirkungen dieses Wandels lassen sich jedoch noch nicht absehen.“ [FMV90, S.810/811]

## 2.4.3 Diskrete Mathematik

Der (diskrete) Mathematikprofessor *Martin Aigner* beschreibt die diskrete Mathematik kurz und knapp mit: „Die Diskrete Mathematik beschäftigt sich vor allem mit endlichen Mengen.“ [Aig06, S.IV] und stellt wenig später fest: „Die Diskrete Mathematik ist heute eine Grundlagenwissenschaft auch der Informatik.“ Deutlicher kann der gegenseitige Erkenntnisprozess und Erkenntnisgewinn von zwei Wissenschaften nicht formuliert werden.

In einigen Jahrzehnten wird die diskrete Mathematik vielleicht auch als *Mathematik des 20. Jahrhunderts* bezeichnet werden, da ihre wesentlichen Erkenntnisse in diesen Jahrzehnten entwickelt worden sind:

**Graphentheorie** Die Graphentheorie ist entstanden aus der Diskussion um das 4-Farbenproblem [FMV90, S.91]. Ein erstes Resultat ist der *Satz von Menger* [Men27] aus Jahre 1927.

Viele algorithmische Probleme der Informatik können mit Hilfe eines Graphen modelliert werden. Mathematische Untersuchungen dieser Graphen ermöglichen gegebenenfalls die Reduzierung des Lösungsaufwandes auf eine Grössenordnung, die bei umfangreichen Graphen noch berechnet werden kann. Als Beispiel sei hier nur der *Dijkstra-Algorithmus* für den kürzesten Weg [Dij59] genannt. Diese häufige Anwendungsmöglichkeit von Graphen in der Modellierung von Problemen in der Informatik hat damit die Graphentheorie von einem kleinen Randgebiet zu einer wichtigen mathematischen Teildisziplin in der (diskreten) Mathematik im 20. Jahrhundert werden lassen. Diese Stellung zwischen Mathematik und Informatik beschreibt *Martin Aigner*:

„Baumstrukturen, Flußdiagramme, Netzwerke gehören heute zum Rüstzeug jeden Informatikers. [...] Seit Beginn der 60er Jahre gibt es eine unübersehbare Anzahl von Optimierungsfragen, die mit graphentheoretischen Methoden befriedigend gelöst wurden — aus den „slums of topology“ (WHITEHEAD) hat sich die Graphentheorie zu einer Grundlagenwissenschaft der angewandten Mathematik entwickelt.“ [FMV90, S.94]

Diese Entwicklung war nur möglich, weil die Programmiersprachen sich von maschinenorientierten Sprachen zu problemorientierten Sprachen entwickelt hatten. Erst in diesen waren (dynamische) Datenstrukturen möglich, in denen diese Graphen abgebildet werden konnten. Der Einfluss der Informatik auf beide Wissenschaftsbereiche *Sprachen* und *Mathematik* wirkte also auf die Informatik zurück und ermöglichte auf diese Weise einen enormen Erkenntnisgewinn.

## Kryptologie

Die Kryptologie ist ein sehr altes Gebiet der Mathematik. Allein der Name des *Caesar-Verfahrens*<sup>7</sup> [Fre10, S.24], [Hro06, S.216] deutet auf diese Vergangenheit hin. Auch weitere Verfahren wie *Vigenère* [Fre10, S.100ff] sind schon einige Jahrhunderte alt. Aber erst die Informatik mit ihren Anwendungen, die nach einer sicheren Datenübertragung fragen, machten umfassende Untersuchungen in der Mathematik notwendig. Als Beispiel seien hier nur die asymmetrischen *Public-Key-Verfahren* [Fre10, S.297] genannt.

Ohne die informatischen Hintergründe beispielsweise über die Datenübertragungen in Netzen bezüglich Sicherheit und Verfügbarkeit wird nicht deutlich, warum es eigentlich dieser Verfahren bedarf [Fre10, S.245ff], [Pas09].

**Codierungstheorie, Informationstheorie und Digitalisierung** Anders als bei der Kryptologie handelt es sich bei der *Codierungstheorie* und der *Informationstheorie* um zwei neuere Gebiete der *diskreten Mathematik*, die im Umfeld der Mathematik entstanden sind. In

<sup>7</sup> In der Literatur wird vielfach allerdings ohne Quellenangabe behauptet, dass sich in den Werken von Caesar eine Beschreibung des Verfahrens befindet. Beim römischen Schriftsteller *Sueton* finden wir eine derartige Beschreibung [Ihm08, Suet.Jul.56.6]:

„[6] . . . extant et ad Ciceronem, item ad familiares domesticis de rebus, in quibus, si qua occultius perferenda erant, per notas scripsit, id est sic structo litterarum ordine, ut nullum uerbum effici posset: quae si qui inuestigare et persequi uelit, quartam elementorum litteram, id est D pro A et perinde reliquas commutet.“

der Informationstheorie, die auf *Claude Shannon* [Sha48] zurückgeht, wird der (technische) *Informationsbegriff* definiert und untersucht.

Dieser *Informationsbegriff* basiert auf den Überlegungen von *Harry Nyquist* [Nyg02], der gezeigt hat, dass Daten bei der Übertragung eine physikalisch-technische Bandbreite benötigen, auch wenn diese Begriffe damals noch nicht bekannt waren. Dieses *Nyquist-Abtasttheorem* begründet das Zeitalter der technischen Informationsübertragung, hier allerdings beschränkt auf Übertragungen analoger Art in erster Linie in Telefonnetzen. Ähnlich wie sich hier Gedanken und Ideen der Physik und der Mathematik berühren, ist die Verschränkung von mathematischen und informatischen Ideen und Strukturen (siehe Abbildung 2.2<sup>8</sup>) in den weiteren Entwicklungen, die mit dem Informationsbegriff von Shannon ihren Anfang nahmen.

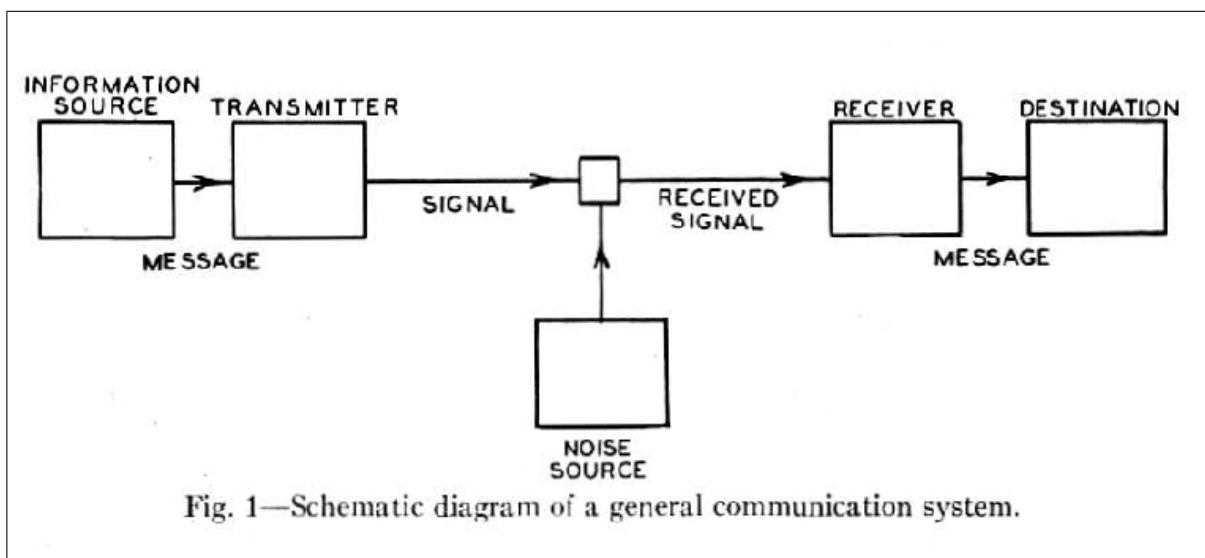


Abbildung 2.2: Grafische Darstellung eines Kommunikationssystems im Originaltext von Shannon 1948[Sha48, S.381]

Ein zentraler Begriff im Text von Shannon ist der der *Entropie*, den Shannon in Anlehnung an den Begriff in der *Thermodynamik* [MG03, S.242ff] aus der Physik entlehnt hat. Die Information(sverteilung) über das technische Signal entspricht dabei in etwa der Wärme(verteilung) (der Energie) in einem abgeschlossenen System. Bei beiden Begriffen geht es um inhaltliche Größen — auf der einen Seite die Information, auf der anderen Seite um die Nutzung von (,höherwertiger’) Energie —, die im Verständnis und in der Nutzung weit über die darunterliegende Kategorie — technisches Signal und Energie — hinausgehen.

Während Nyquist die technische Übertragung von Daten in Form von elektromagnetischen Signalen diskutiert, fragt sich Shannon, welche Bedeutung — Information — diese Signale für den Nutzer haben können. Diese Untersuchungen stellen unter anderem für die Übertragung in Netzen zentrale Grundlagen bezüglich Bandbreite und Störungen [ADH<sup>+</sup>00] bereit.

<sup>8</sup>Quelle des Originals: <http://www.alcatel-lucent.com/bstj/vol27-1948/articles/bstj27-3-379.pdf>, letzter Zugriff: 16.11.2012

Aufbauend auf diesen Überlegungen entstand die *Codierungstheorie*. Sie beschäftigt sich mit der Übersetzung von Daten von einem Zeichensystem in ein anderes. Die ersten Arbeiten erschienen von *Richard Hamming* [Ham50] über fehlerkorrigierende Codes. Sie sind notwendig für eine sichere Datenübertragung in gestörten Kanälen und beschreiben damit theoretische Grundlagen für den Aufbau und die Wirksamkeit von Netzen auf den unteren Schichten beispielsweise im ISO/OSI-7-Schichtenmodell [Zim80] und auch für die Verschlüsselung im Umfeld der Kryptologie beispielsweise beim *Huffmann-Code* [KH05, S.9].

Die praktischen und theoretischen Überlegungen zur Codierung und Information sind Grundelemente der *Digitalisierung*. Die Digitalisierung ist damit ein zentrales Merkmal heutiger (technischer) Welt geworden. Begrifflichkeit und Verständnis davon muss zweifelsohne Teil heutiger Allgemeinbildung sein. Gleichzeitig drückt die Digitalisierung selbst die Verzahnung und Beeinflussung verschiedenster Wissenschaften aus. Aigner schreibt: „Die mathematische Behandlung dieser Probleme in den letzten Jahrzehnten zusammen mit der Erfindung geeigneter Bauteile, beginnend mit dem Transistor, resultierte in einer interdisziplinären Erfolgsbilanz ohnegleichen.“ [FMV90, S.108]

#### 2.4.4 Fazit Mathematik

Informatik und Mathematik haben eine fast symbiotische Beziehung. Auch wenn heute die Informatik viele weitere Wissenschaftsbereiche tangiert, so ist doch das Verhältnis zur Mathematik extrem intensiv und vielschichtig.

Aber die Mathematik als Wissenschaft ist heute ohne Informatik und informatisches Denken nicht denkbar. Wir haben dabei gesehen, dass es sehr verschiedene Rollen sind, die die Informatik für den Zugewinn bei der Erkenntnis der Mathematik spielt. Beim *Algorithmusbegriff* wird seitens der Informatik der Begriff selber thematisiert, Eigenschaften von Algorithmen allgemein werden untersucht und die Zeitproblematik erkannt. Die *Numerik* und die *Statistik* konnten aufgrund der Existenz von Computern einen Aufschwung erleben. Weitere Teilgebiete der *diskreten Mathematik* wie beispielsweise die *Kryptologie* entstehen oder verändern ihren Charakter von einem Nischenbereich zu einem wichtigen Teil der Mathematik.

Die Informatik hat die Mathematik entscheidend verändert. Ein Mathematiker im 21. Jahrhundert muss zumindest einen Teil dieser veränderten, erweiterten oder neuen Gebiete kennen. In der *Schulmathematik* ist allerdings von diesen Veränderungen kaum etwas zu spüren [Min99b]. Der viel zu geringe Stellenwert der Informatik in der Schule scheint offensichtlich keinen grossen Innovationsdruck auf die Mathematik aufzubauen<sup>9</sup>. In den siebziger Jahren hat *Arthur Engel* versucht, mit dem Buch „*Elementarmathematik vom algorithmischen Standpunkt*“ [Eng77] die Schulmathematik algorithmisch zu beeinflussen, hat aufgrund der fehlenden Unterstützung von offizieller Seite allerdings keinen Erfolg gehabt. Ebenso wird das im Jahre 2011 erschienene Buch von *Reinhard Oldenburg* mit dem Titel „*Mathematische Algorithmen im Unterricht: Mathematik aktiv erleben durch Programmieren*“ [Old11] leider

---

<sup>9</sup>Als Nachweis möge dienen: „Ländergemeinsame inhaltliche Anforderungen für die Fachwissenschaften und Fachdidaktiken in der Lehrerbildung“, (Beschluss der Kultusministerkonferenz vom 16.10.2008 i.d.F. vom 16.9.2010, S.30/31), im Internet: [http://www.kmk.org/fileadmin/veroeffentlichungen\\_beschluesse/2008/2008\\_10\\_16-Fachprofile-Lehrerbildung.pdf](http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2008/2008_10_16-Fachprofile-Lehrerbildung.pdf), letzter Zugriff: 20.4.2012)

die Schulmathematik wahrscheinlich nicht beeinflussen. Eine Einführung der Graphentheorie in den schulischen Mathematikunterricht würde sowohl aus innermathematischen als auch aussermathematischen Überlegungen nicht nur aus der Informatik zu begründen sein. Es ist allerdings in der Vergangenheit bei wenigen Versuchen geblieben [Bod76, Huß07, ABZ07]. So obliegt dem Schulfach Informatik die zusätzliche Aufgabe, einerseits notwendige mathematische Voraussetzungen innerhalb des eigenen Faches zu erarbeiten, als auch andererseits damit aktuelle Defizite der Schulmathematik auszugleichen.

## 2.4.5 Ausflug: Der Streit der Fakultäten

Wie bedeutsam die Diskussion um das informatische Denken in der Mathematik ist, zeigt die Auseinandersetzung zwischen führenden Institutionen der Mathematik und Informatik aus den 90er Jahren. Diese Auseinandersetzung um das Schulfach Informatik zwischen dem *Fakultätentag Informatik* und der *GdM*, der Standesorganisation der Mathematiker fand in den Jahren 1993 und 1994 statt und zeigt das auch heute sehr verbreitete Unverständnis bezüglich des informatischen Denkens.

Im Januar 1994 erarbeitete der Arbeitskreis *Mathematikunterricht und Informatik* der *Gesellschaft deutscher Mathematiker (GDM)* auf Veranlassung des GDM-Vorsitzenden eine Stellungnahme [BFH<sup>+</sup>94]:

„Der Behauptung, daß die ‚Informatik die Grundlage vieler Wissensgebiete‘ sei, ist zu widersprechen. Vielmehr besteht die Bedeutung der Informatik für die Gesellschaft darin, daß ein reichhaltiges Prozeßwissen anderem Wissen hinzugefügt wird. In diesem Sinne wird Informatik in vielen Wissensbereichen angewandt und haben Computer auch Methoden und Arbeitsmittel vieler Wissenschaften verändert. Grundlage könnte die Informatik in einigen ingenieurmäßigen Wissensgebieten wie Computer- und Softwaretechnik, Robotik, Fertigungstechnik sein.

...

Der dabei unseres Erachtens zur Allgemeinbildung gehörende Aspekt der *Algorithmierbarkeit*, das Verständnis der prinzipiellen und praktischen Möglichkeiten und Grenzen algorithmischer Beschreibungen und Problemlösungen, wird aber eher durch Mathematik und künstlerische Fächer vermittelbar als durch die Informatik allein, die in diesen ‚Empfehlungen‘ einer Überschätzung der Reichweite ihrer eigenen Problemlöseverfahren zu wenig entgegentritt“ [BFH<sup>+</sup>94]

Diese Erklärung der Mathematiker macht auf der einen Seite deutlich, dass eine Begründung von Informatik als Schulfach jenseits der Bedeutung als ‚Herstellungswissenschaft‘ für Produkte und Prozesswissen dringend notwendig ist, aber auch, dass es nicht selbstverständlich ist, dass die besonderen Anteile informatischen Wissens und Denkens in anderen Wissenschaften als solche auch erkannt und dann anerkannt werden. So wird beispielsweise das wichtige informatische *Problemlösen* zwar anerkannt, aber in der Bedeutung unterschätzt.

## 2.5 Physik

Die *Informationstheorie* und die *Codierungstheorie* leiten uns zur nächsten Wissenschaft: der *Physik*. Die Rolle der Physik und der neu entstehenden Informatik in den letzten Jahrzehnten und Jahrhunderten wird von *Andrew Stuart Tanenbaum* in der Einleitung in seinem grundlegenden Buch über Computer-Netzwerke folgendermassen beschrieben:

„Jedes der letzten drei Jahrhunderte wurde von einer bestimmten Technologie beherrscht. Das achtzehnte Jahrhundert war die Blütezeit der großen mechanischen Systeme, welche die industrielle Revolution mit sich brachte. Das neunzehnte Jahrhundert war das Zeitalter der Dampfmaschine. Im zwanzigsten Jahrhundert spielen das Sammeln, Verarbeiten und Verbreiten von Informationen die wichtigste technologische Rolle. Wir haben unter anderem die Einrichtung weltweiter Telefonnetze, die Erfindung von Radio und Fernsehen, die Geburt und den beispiellosen Aufstieg der Computerindustrie und den Start von Kommunikationssatelliten miterleben können.“ [Tan03]

### 2.5.1 Die Krise der Physik

Die Physik, die Mitte des 19. Jahrhunderts scheinbar (fast) abgeschlossen schien<sup>10</sup>, geriet Ende des 19. Jahrhunderts und Anfang des 20. Jahrhunderts in mehrere Krisen: *Albert Einstein* revolutionierte mit der *Relativitätstheorie* die Physik, indem er die ‚Urfeste‘ *Zeit* im wahrsten Sinn des Wortes *relativierte*, die Entdeckung der *Radioaktivität* brach mit dem *kausalen Prinzip* der Physik, die *Quantenphysik* ausgehend von der Entdeckung von *Max Planck*, dass auf atomarer Ebene die Energie nur in ganzzahligen Vielfachen einer Grundgrösse abgegeben wird, gab der Physik auf mikroskopischer Struktur eine neue Grundlage und letztendlich widersetzten sich die *elektromagnetischen Wellen* der *Wechselwirkungs idee* von (mechanischen) Systemen. Eine zweifellos hochspannende Zeit für einen physikalischen Wissenschaftler. Entsprechend sind die Überlegungen von *Nyquist*, *Shannon* und Anderen nicht allein aus mathematischem Interesse oder Theorien entstanden, sondern aus der sich entwickelnden *Informationsübertragung* beim kabelbasierten Telefonnetz und über elektromagnetische Wellen beim beginnenden Rundfunk. Aus Sicht der Informatik sind von den Entwicklungen in der Physik vor allem die elektromagnetischen Wellen interessant.

### 2.5.2 Elektromagnetische Wellen

Im Jahre 1887 entdeckte *Heinrich Hertz* die elektromagnetischen Wellen [Her87, Her71]. Innerhalb weniger Jahrzehnte, vergleichbar mit den Entwicklungen im Computerzeitalter, haben diese Wellen mit ihren technischen Anwendungen vor allem in der Informationsübertragung<sup>11</sup> die Welt verändert. Bereits 1901 wurden aufgrund eines Vertrages zwischen *Guglielmo Marconi* und der Versicherungsgesellschaft *Lloyd* (technische) Signale in der Schifffahrt über elektromagnetische Wellen von See auf Land gesendet [KW97, S.515].

<sup>10</sup>Max Planck soll geraten worden sein, „[...] auf ein Studium der Physik zu verzichten, weil bestenfalls noch Kleinigkeiten [...] zu klären wären [...].“ [Fis01, S.391]

<sup>11</sup>Eine andere genauso wichtige Anwendung der elektromagnetischen Wellen sind beispielsweise die Nutzung der *Röntgenstrahlen*.

Mit Hilfe der (Amplituden-)Modulation [Mes05, S.146] dieser Wellen konnte dann mit der 1906 von *Robert von Lieben* entwickelten Elektronenröhre [BK97, S.153] auf der Senderseite Sprache und Musik übertragen werden. Vom ersten (militärischen) Sprechverkehr 1913 zwischen Berlin in Nauen bis zum ersten Rundfunksender ab Oktober 1923 [BK97, S.156/157] war es nur noch ein kurzer Weg. In den USA gab es schon 1921 sieben Rundfunkstationen [BK97, S.154]. Diese Rundfunksender stiessen aufgrund der technischen Entwicklung der elektronischen Verstärker mit ihren immer grösseren Reichweiten relativ schnell aufgrund der notwendigen *Bandbreite* an ihre Grenzen. Schliesslich einigten sich die Vertreter verschiedener Staaten auf der *Wellenkonferenz von Genf 1926*<sup>12</sup> auf eine technische Bandbreite von 5 kHz — das bedeutet bei einem amplitudenmodulierten Signal mit zwei Seitenbändern eine Gesamtbreite von 10 kHz — bei den bis dahin verwendeten Sendefrequenzen<sup>13</sup>. Diese Beschränkung wurde erst bei der Einführung des *UKW-Rundfunks* ab 1950 [BK97, S.393] aufgehoben.

### 2.5.3 Telefon

Eine weitere technische Errungenschaft stellte das Telefon dar. Die Teilnehmer waren ab 1881 sternförmig über eine (Orts-)Vermittlungsstelle [KW97, S.506] angeschlossen. Die Verknüpfung der einzelnen Ortsnetze wurde über Fernleitungen realisiert. Diese wurden zunächst nur mit je einem Gespräch zu gleichen Zeit genutzt. Daraus erfolgten verständlicherweise hohe Kosten im Ferngesprächsnetz. Der Wunsch nach einer Mehrfachnutzung führte auch hier zur Modulation von hochfrequenten elektromagnetischen Wellen, dem *Frequenzmultiplexing*. Diese erfolgte zum ersten mal 1918 in den USA mit vier Kanälen auf einer Leitung zwischen Baltimore und Pittsburgh [BK97, S.150]. In Deutschland wurde diese Technik ab 1929 zuerst bei einem Seekabel zwischen Schweden und Deutschland und dann ab 1930 bei Freileitungen verwendet [Ste69, S.92]. Damit tauchte auch hier das Bandbreitenproblem auf. Weltweit ist daher das Frequenzband beim analogen Telefon auf 300 – 3400 Hz festgelegt [ADH<sup>+</sup>00, S.374,375].

Das *Nyquist-Abtasttheorem* ist daher für die physikalische Erkenntnis und darauf aufbauenden Anwendungen zentral. Mit diesem Theorem werden die physikalischen Randbedingungen für die Anwendung bei der Übertragung von technischen Signalen festgestellt. Die Physiker betrachten die Informationsübertragung interessanterweise ausschliesslich als Signalübertragung und erklären die notwendige Bandbreite über die mathematischen Berechnungen über die bei der Modulation entstehenden Frequenzen. Dass diese physikalisch-mathematischen Betrachtungen einen inhaltlichen Grund unabhängig von der konkreten technischen Realisierung haben, wird dabei nicht deutlich. Dies erfolgt durch die Einführung des *Informationsbegriffes* nach Shannon und darauf aufbauend des Begriffes der *Entropie*. Erst diese Überlegungen entheben die Betrachtungen über die Bandbreite in der Physik dem fast zufälligen technischen Beiprodukt einer konkreten technischen Anwendung physikalischer Erkenntnisse. Im Gegensatz zur Mathematik werden diese Überlegungen in der Physik völlig ignoriert.

<sup>12</sup><http://www.wabweb.net/radio/radio/frequenz.htm>, letzter Zugriff: 3.5.2012

<sup>13</sup>Diese Bandbreite wurde auf späteren Wellenkonferenzen auf die noch heute existierende Bandbreite von 4,5 kHz im LW- bis KW-Bereich verringert.

Offensichtlich sieht die Physik im *Informationsbegriff* kein physikalisches Konzept<sup>14</sup> und entsprechend untersucht die Physik diese Phänomene auch nicht. Damit kann die Physik die Prozesse auch nur bedingt erklären und verweist damit indirekt auf die (damals) neue Wissenschaft *Informatik*. Hier unterscheidet sich die Herangehensweise der Physik von der der Mathematik. Die Physik beschränkt sich auf einen Teil der in diesem Gebiet notwendigen Erkenntnisse. Erst im Rahmen der *Elektrotechnik*, *Nachrichtentechnik* und *Kybernetik* werden diese Erkenntnisse registriert und angewendet.

Informationen bzw. Daten entsprechend dem informatischen Verständnis benötigen immer einen (physikalischen) *Träger*. Weil dieser Träger als physikalisches System immer physikalischen Gesetzmässigkeiten unterliegt, sind und bleiben Physik und Informatik in diesem Bereich tatsächlich eng miteinander verwoben. Will ein Physiker die Phänomene der Informationsübertragung mit elektromagnetischen Wellen vollständig erfassen, benötigt er notwendigerweise informatische Erkenntnisse.

## 2.5.4 Ausbildung in der Physik

**Informationsbegriff in der Hochschule** Den geringen Stellenwert der informationstheoretischen Begriffe in der Physik werden auch in der Literatur für die Studierenden der Physik deutlich. In einem der Standardwerke für die Physik, dem ‚*Gerthsen*‘, wird im Kapitel über elektromagnetische Wellen zwar die Modulation auf insgesamt 4 Seiten kurz besprochen, allerdings wird kaum auf die Anwendungen oder auch die Informatik Bezug genommen [Mes05, S.158ff,S.437ff]. Der Begriff *Information* wird nur im Absatz *Ausblick* im Satz „Die Elektrotechnik versorgt den Menschen mit Energie und, was jetzt immer wichtiger wird, mit Information.“ [Mes05, S.440] Die Grösse *Entropie* wird überhaupt nicht angesprochen.

**Informationsbegriff in der Schule** Auch im Schulunterricht der Sekundarstufe II spielen diese Gedanken nur eine untergeordnete Rolle<sup>15</sup>. So wird beispielsweise im *Dorn-Bader* [BD10, S.166-169] unter der Überschrift ‚*Elektromagnetische Wellen übertragen Daten*‘ zwar die Amplitudenmodulation und auch die Frequenzmodulation besprochen und sogar danach unter ‚*Übertragung braucht Frequenzbereiche*‘ die benötigte Bandbreite aus der Frequenzanalyse dargestellt, bei der Zusammenfassung des Kapitels der elektromagnetischen Wellen wird dies jedoch nicht mehr erwähnt. Für den in der Materie nicht ganz vertrauten Schüler oder Lehrer kann zudem die Aussage „Es gibt jedoch raffinierte Verfahren, mit denen die beanspruchte Bandbreite verringert werden kann“ [BD10, S.169] zu erheblichen Fehlvorstellungen führen. Immerhin vier Seiten werden im Physikbuch des Cornelsen-Verlages für diese Thematik verwendet. Auch der Begriff *Bandbreite* wird erläutert [DEa08, S.246-249]. *Information* und *Entropie* sind in beiden Werken nicht einmal als Begriff erwähnt.

Völlig fehlerhaft wird im Physikbuch des Dudenverlages für die gymnasiale Oberstufe sogar behauptet: „Die Informationsübertragung mit Licht ermöglicht den Datentransport größerer Datenmengen, da man z.B. für die Übertragung eines Telefongesprächs nur eine Lichtfre-

<sup>14</sup>Entsprechend sehen die Bildungsstandards Physik [KMK05c] ausschliesslich *Materie*, *Energie*, *Wechselwirkung* und *System* als *Basiskonzepte* der Physik.

<sup>15</sup>Allerdings unterscheidet sich damit die Schulphysik schon positiv von der Schulmathematik.



quenz benötigt [MS<sup>+</sup>11, S.367]. Die Informationsübertragung mit elektromagnetischen Wellen wird auf gerade zwei Seiten behandelt, dabei wird der Begriff *Bandbreite* nicht einmal erwähnt [MS<sup>+</sup>11, S.352/3].

Wie unbedeutend aus Sicht der Physik diese Thematik ist, ist an den beiden letzten Auflagen des *Metzler*-Schulbuches erkennbar. In der 3. Auflage werden dieser Thematik noch vier Seiten gewidmet und dabei auch die Bandbreite zumindest anschaulich erklärt [GK<sup>+</sup>98, S.288-291], in der 4. Auflage schrumpft die Thematik auf ganze zwei Seiten [GK<sup>+</sup>07, S.298,299]. Der Begriff Bandbreite wird nicht mehr erklärt. Allerdings wird die *digitale* Übertragung inklusive *Datenkompression und Datenreduktion*, „die es möglich macht, eine größere Anzahl von Sendern in einem bestimmten Frequenzband unterzubringen“, gleich mit bearbeitet.

Die Behandlung dieses Thema im 2011 erschienenen Buch des Duden-Verlages für die Einführungsphase der gymnasialen Oberstufe mit dem Titel *Selbstverständlich Physik* [Ae11] ist aus physikalischer, aber auch aus informatischer Sicht extrem kritikwürdig. Es wird das physikalisch-informatische Phänomen *Information* nicht besprochen, geschweige denn eingeführt und auch nicht eine in der Physik übliche Definition der physikalisch-informatischen Größe *Information* vorgenommen noch seine Einheit *bit* eingeführt. Als Abschluss des Themas *Energie* wird dann allerdings auf einer Seite mit der Überschrift ‚*Information trägt Energie*‘ [Ae11, S.132] ein Gedankenexperiment aus der Habilitationsschrift von *Leo Szilárd*<sup>16</sup> aus dem Jahre 1929 vorgestellt, mit dem ein *Informations-Energie-Äquivalent*<sup>17</sup> von  $E = k * T * \ln(2)$  ( $k$  ist dabei die Boltzmann-Konstante und  $T$  die absolute Temperatur des Systems) für 1 *bit* nachgewiesen wird. Abgesehen davon, dass diese hoch abstrakten physikalischen Ausführungen von einem Schüler im Anfangsjahr des Physikunterrichts der gymnasialen Oberstufe nur im Ausnahmefall verstanden werden können, stellt sich die Frage, welcher physikalische Sinn diese Ausführungen an dieser Stelle machen. Es drängt sich der Verdacht auf, dass die Darstellung der Existenz eines Informations-Energie-Äquivalents die Beschäftigung mit dem Phänomen *Information* als überflüssig erscheinen lassen soll<sup>18</sup>. Zumindest wird dieser einseitige Einschub nicht der *Information* als physikalisches und informatisches Basiskonzept gerecht.

**Digitalelektronik in der Schule** Computer basieren heute auf *Digitalelektronik* und damit auf hochintegrierten Schaltungen, die aus physikalischer Sicht aus *Festkörpern* zusammengesetzt sind und deren Eigenschaften zu Schaltungszwecken ausgenutzt werden.

Die Entwicklung der Digitalelektronik ist die physikalische Umsetzung aussagenlogischer Aussagen [Ass65, KSS77]. Die Reduzierung der sechzehn möglichen Grundschaltungen mit zwei Eingängen und einem Ausgang auf eine von zwei Schaltungen — *NAND* oder *NOR* — ermöglichten mit den physikalischen Erkenntnissen seit Erfindung des Transistors eine hohe automatisierte Fertigung von Schaltkreisen, die als Basis für die Entwicklung und Bau der

<sup>16</sup>Leo Szilárd stammte wie Edward Teller und John v. Neumann aus Ungarn und war mit der erwähnten Habilitationsschrift einer der Begründer der Informationstheorie. Zudem war er im *Manhattan-Projekt* bei der Entwicklung der amerikanischen Atombombe beteiligt [LS92].

<sup>17</sup>siehe hierzu auch die Erklärung und Animation von *Ernst* aus der FH Rosenheim: <http://www.ernst-team.de/texte/talk2/daemon/kap5.htm>, letzter Zugriff: 30.9.2012

<sup>18</sup>Entsprechend könnte man aufgrund der Existenz des Masse-Energie-Äquivalentes auf die Beschäftigung mit der physikalischen Größe *Masse* verzichten.

Computer dienen.

Im Rahmen der Ausbildung der Physik gehört heute daher nicht nur ausschliesslich die Funktionsweise von Schaltungen wie *Serienschaltung*, *Parallelschaltung* und *Wechselschaltung* als Möglichkeiten, in einem technischen System Ströme ein- und auszuschalten. Ebenso gehören die logischen Grundschaltungen zum Basiswissen. Dementsprechend werden diese auch in den neueren Schulbüchern für den 5. und 6. Jahrgang öfters angeboten. Leider sind dort gelegentlich Ungenauigkeiten bzw. Fehler vorzufinden, da die Autoren diese Thematik offensichtlich nicht verinnerlicht haben. So werden im *Dorn Bader* für den 5. und 6. Jahrgang die *UND*-, *ODER*- und die *ENTWEDER-ODER*-Schaltung vorgestellt. Allerdings hat dort die *ENTWEDER-ODER*-Schaltung einen Eingang und zwei Ausgänge [BD09, S.63].

Bei den Schulbüchern für die Sekundarstufe II zeigt sich ein sehr unterschiedliches und nicht systematisches Bild. Im Buch aus dem Verlag Cornelsen wird der Begriff der *Digitalisierung* [DEa08, S.247] erklärt, die *Digitalelektronik* fehlt jedoch vollständig. Im Buch aus dem Duden-Verlag wird die Digitalisierung sehr knapp beschrieben [MS<sup>+</sup>11, S.257], auch hier fehlt jede Information zur Digitalelektronik. Beim älteren Metzler gibt es ein umfassendes Kapitel zur Digitalelektronik und und darauf aufbauend zur Schaltungstechnik in Computern [GK<sup>+</sup>98, S.470-475]. In der Überarbeitung aus dem Jahre 2011 fehlt dieses völlig.

## 2.5.5 Fazit Physik

Neben vielen weiteren entscheidenden Veränderungen im Weltbild der Physik sowie in ihren Anwendungen in den letzten hundert Jahren sind informatische Kenntnisse zumindest aus dem Gebiet der technischen Informatik unerlässlich, um das eigene Fach vollständig verstehen, nachvollziehen, einschätzen und weiter entwickeln zu können. Im Gegensatz zur Mathematik hat die Physik den Informationsbegriff nicht zu ihren Konzepten und Ideen hinzugefügt. Daher berühren sich Physik und Informatik inhaltlich sehr stark, betrachten sich aber gegenseitig mehr jeweils als Anwendungs- bzw. als Umsetzungstechnik.

Sowohl in der Hochschule als auch in der Schule werden in der Ausbildung von Studenten und Schülern keine systematischen Kenntnisse über den *Informationsbegriff* und die *Digitalelektronik* vermittelt. Im Gegensatz zu den tradierten und auch neueren physikalischen Teildisziplinen hängt es von den Autoren der Lehrwerke ab, ob Themen aus dem Grenzgebiet von Physik und Informatik überhaupt und in welcher Form und Tiefe präsentiert werden. Damit reicht ein physikalischer Unterricht für die notwendigen informatischen Kompetenzen innerhalb der Physik nicht aus.

## 2.6 Philosophie

Auch in der Philosophie ist der Begriff *Information* eine zentrale Kategorie. Während die Physik diese Kategorie *Information* und ein Nachdenken darüber jenseits der Signalübertragung völlig ausblendet, ist dieser Aspekt für die Philosophie völlig unbedeutend.

## 2.6.1 Das Gehirn als informationsverarbeitende Maschine

Die *Philosophie* nimmt diese Diskussion ganz anders auf. Sie führt auf der einen Seite die aus der Informatik kommende Diskussion um die These des *Gehirns als informationsverarbeitende Maschine*, da natürlich diese Diskussion eine ureigene Diskussion der Philosophie ist, die unter anderem auch nach dem Wesen und Sein des Menschen und dabei auch nach der Funktionsweise des Gehirnes und des Geistes — die Philosophie des Geistes — fragt:

“*The basic idea of the computer model of the mind is that the mind is the program and the brain the hardware of a computational system. A slogan one often sees is: ‘the mind is to the brain as the program is to the hardware.’ Let us begin our investigation of this claim by distinguishing three questions:*

1. *Is the brain a digital computer?*
2. *Is the mind a computer program?*
3. *Can the operations of the brain be simulated on a digital computer?‘*

[Sea90],deutsch: [Sea92]

formuliert es der Philosoph *John R. Searle*. Er selber kommt zum Ergebnis:

„Soweit es um seine intrinsischen Operationen geht, betreibt das Gehirn keine Informationsverarbeitung. Es ist ein bestimmtes biologisches Organ, und seine spezifischen neurobiologischen Prozesse verursachen spezifische Formen der Intentionalität.“ [Sea92, S.232]

Auch in der gerade neu entstehenden Informatik beschäftigten sich schon unter anderen *John von Neumann* in einem geplanten, aber aufgrund seines frühen Todes nicht vollendeten Vortrages *Die Rechenmaschine und das Gehirn* [Neu91] für die *Silliman Lectures der University Yale* 1956 mit derartigen philosophischen Fragen. So schreibt Neumann:

„Ich habe [...] das Wesen moderner Rechenmaschinen [...] beschrieben, so daß ich mich jetzt dem anderen Vergleichsobjekt, dem menschlichen Nervensystem, zuwenden kann. Ich werde sowohl die Punkte, in denen die beiden ‚Automaten‘ einander ähnlich sind als auch die Punkte, in denen sie sich unterscheiden, diskutieren.“ [Neu91, S.44]

und erläutert wenig später: „Man macht zuerst die Beobachtung, daß das Nervensystem in erster Linie *digital* arbeitet.“ [Neu91, S.44] Diese eher *neurobiologische* und *neuroinformatische* Frage wird ergänzt durch Aussagen zur Wirkung des *Gehirns als Automaten*:

„Diese Ausführungen zeigen, daß das Nervensystem, betrachtet man es als Automaten, unbedingt sowohl einen arithmetischen als auch einen logischen Teil besitzen muß, und das die Anforderungen an die Arithmetik in ihm von ebenso großer Bedeutung sind wie die Anforderungen an die Logik. Das besagt, daß wir es wieder mit einer Rechenmaschine im eigentlichen Sinn zu tun haben, und daß eine Diskussion in der Terminologie der in der Rechenmaschinentheorie gebräuchlichen Begriffe erlaubt sind.“ [Neu91, S.72]

Diese Diskussion beschäftigt somit gleichermaßen Informatiker und Philosophen. Der Informatiker *Peter Schefe* stellt auf der interdisziplinären *Dagstuhl-Tagung Informatik und Philosophie 1992* entsprechend fest: „Es ist zu beobachten, daß Informatiker von McCarthy bis Winograd, von Weizenbaum bis Hofstadter — um einige prominente zu nennen — Anleihen bei Philosophen machen oder selber Philosophien entwickeln.“ [Sch92, S.1]

Entsprechend fragt die *Philosophie*, aber ähnlich auch die *Psychologie*, und sucht die Antwort auf die Frage, die *Alan Turing* bereits als *Turing-Test* in einem der berühmtesten Artikel in der Geschichte der Informatik *Computing Machinery and Intelligence* 1950 gestellt hat: *“I propose to consider the question, ‘Can machines think?’“* [Tur50]

Der Elektrotechniker, Politologe und Philosoph *Gero Zimmermann* hat sich in seiner Dissertation mit der *Philosophie des Geistes im Spiegel der Informatik und der Komplexitätstheorie* beschäftigt. Er stellt darin fest:

„Das Gehirn kann aber nicht die physikalischen Träger (erregte Neuronen) erkennen, die die Repräsentationen hervorbringen, sondern nur die Repräsentation selbst. Das Gehirn verarbeitet halt Signale, aber seine Funktion lässt sich nicht durch Beobachtung der Signale erschliessen (Ansatz der Hirnforschung), sondern nur durch die Art und Weise, wie die *Signalmuster*, d.h. die Zeichen bzw. Codes, verarbeitet werden (Ansatz der Informatik).“ [Zim11, S.255]

Er betont dabei aber auch:

„Damit wird aber ausdrücklich nicht behauptet, dass das Gehirn ein Computer ist, sondern lediglich dass solche *Computermodelle* geeignet sind, uns gewisse Phänomene zu erklären.“ [Zim11, S.6]

Diese Diskussionen sind einerseits die Fortsetzung der Auseinandersetzungen um das *mechanistische Weltbild* aus dem 18. und 19. Jahrhundert [Gie82], andererseits zwingen sie durch das Nachdenken und Reflektieren über angenommenes intelligentes Verhalten eine Aufklärung des Begriffes *Intelligenz*. Die Erkenntnisse aus der Algorithmik sind sicher dabei eine Hilfe, machen sie doch deutlich, dass zumindest erhebliche Anteile von uns bisher als intelligent angenommenen Verhaltens sehr viel *Routine* enthalten. Philosophen haben bei ihrer ureigenen Suche nach der Funktionsweise des Gehirns und dem Sein des Geistes die Modelle der Informatik zu Hilfe genommen und damit benötigen dafür informatisches Denken

## 2.6.2 Der Begriff *Information* aus philosophischer Sicht

Die Diskussion um den Begriff *Information* wird aus Sicht der Philosophie anders geführt als es in den technischen Fächern geschieht. Wie die Informatik im Rahmen der Informations- und Codierungstheorie den Begriff *Information* als Ergänzung zum digitalen Signal eingeführt hat, fragt sich die Philosophie, was denn und wann beim Menschen aus *Daten Information* werden. Weitere daran anschließende Fragen sind dann die nach den Gemeinsamkeiten und Unterschieden zum Begriff *Wissen*.

Die Philosophie empfindet die technische Information als ein Zeichenmenge oder auch als Daten. Information werden diese Daten erst beim Empfänger, der diesen Daten eine *pragmatische* Bedeutung zumessen kann. Die Philosophin *Sybille Krämer* formuliert: „Menschliches Denken beruht auf Informationsverarbeitung. Informationsverarbeitung geschieht mit Hilfe von Daten, den physischen Trägern von Informationen. [...] Nun gilt als der Ort der menschlichen Datenverarbeitung das Gehirn. Also zielt die Informatik ab auf die Gestaltung von Systemen, die Leistungen des Gehirns, geistige Tätigkeiten also, technisch unterstützen.“ [Krä92, S.69] Der Empfänger kann diese Information dann eventuell mit anderen Informationen verknüpfen, um daraus *Wissen* zu generieren. Der Hamburger Philosoph *Peter Janich* hat mit

diesem Ansatz Probleme: „Information ist längst ein Schlüsselbegriff in der Psychologie, der Physiologie, der Neurobiologie, allgemein, der Erforschung organischer Verhaltens- und Erkenntnisleistungen sowie aller technischer Substitute geworden. Und doch läßt sich, sogar ohne erhebliches Risiko, für die gesamte wissenschaftliche und philosophische Debatte behaupten, daß es keine logisch und definitionstheoretisch geschweige denn erkenntnistheoretisch befriedigende Definition von ‚Information‘ gibt. [Jan92, S.63] Trotzdem konstatiert auch er wenig später: „Informationen zu verarbeiten hatten wir als menschliche Handlung und Leistung beschrieben, die durch Datenverarbeitung technisch substituiert wird.“ [Jan92, S.67]

Die Begrifflichkeiten waren implizit schon vor dem Beginn des ‚Zeitalters der Informatik‘ existent, aber erst mit dem Nachdenken über informatische Probleme und dem Begriff *Information* wurde diese Strukturen offensichtlich. Krämer stellt fest: „So kann eine komplexe geistige Tätigkeit zurückgeführt werden auf das handgreifliche Operieren mit externen Symbolen. Wir können dazu auch sagen: Eine formale Sprache komme zum Einsatz.“ [Krä92, S.73] Dem können wir Chomskys Aussage anfügen: „Daß sich diese Systeme in einer bislang unaufgeklärten Weise wechselseitig durchdringen, ist seit langem bekannt. Ein hinreichendes Verständnis dieses Bereichs kann man schwerlich erreichen, bevor nicht profunde Analysen der semantischen Regeln einerseits und der Systeme von Kenntnissen und Überzeugungen andererseits verfügbar sind. [Cho73, S.201]

Der Physiker und Philosoph *Max Bense* beschreibt die Notwendigkeit der differenzierten Betrachtung des Begriffes Information:

„Da der Begriff ‚sachhaltige Bedeutung‘ der Nachricht jedoch höchst vage und relativ ist, denn ‚Bedeutungen‘ sind ja höchst schwankende Gebilde, kann man den Begriff der Information, will man ihn wissenschaftlich, gar mathematisch präzisieren, nicht an den der ‚inhaltlichen Bedeutung‘ gründen. Man muß den Begriff der Information gewissermaßen bedeutungsfrei und abstrakt einführen, ähnlich wie man ja seit Galilei in der Physik und in der Mathematik vorgeht. Das erreicht man eben dadurch, daß man nur das an einer Sache, hier an der Information, durch den Begriff bezeichnet, was man zahlenmäßig, meßbar fassen kann.“ [Ben62, S.12](zitiert nach [Cub71, S.29])

*Zimmermann* verlangt zwingend eine differenzierte Analyse der ‚Begriffswolke Information‘, um kognitive Prozesse zu verstehen:

„Die strikte Trennung zwischen *Informationslieferant* (Körper mit Rezeptoren und Propriozeptoren) einerseits und *Informationsverarbeiter* (Gehirn) andererseits, zwischen *Informationsverarbeitung* (einem Prozess im Gehirn) und den (im Gehirn in neuronalen Erregungsmuster repräsentierten) *Informationen selbst* sowie zwischen dem *physikalischen Träger* (Signal) und dem in ihm codierten bzw. repräsentierten *propositionalen Gehalt* einer Information ist die entscheidende Voraussetzung zum Verständnis kognitiver Prozesse.“ [Zim11, S.7]

Die drei Begriffe *Signal*, (*technische*) *Information*, (*philosophisch geprägte*) *Information* lassen sich damit vereinfacht so darstellen:

Die (*technische*) *Information nach Shannon* steht zum *Signal (Daten)* so wie die (*philosophisch geprägte*) *Information* zur (*technischen*) *Information*. Bildlich gesprochen betrachten wir denselben Gegenstand durch drei verschiedene *Linsen* [MS06] bzw. *Windows* [Den03] und nehmen ihn jeweils entsprechend anders wahr. Die vollständige Verstehen des Begriffes

*Information* verlangt von den Beteiligten das Hineindenken in alle Dimensionen des Begriffes und damit das der Dimension zu Grunde liegende Denken.

### 2.6.3 Künstliche Intelligenz

Ging es bisher um den Begriff und Bildung der *Information* beim Menschen und beim Automaten, so stellt sich darauf aufbauend die Frage, wie denn aus den *Informationen* Wissen konstituiert wird und noch mehr, wie denn aus diesem Wissen heraus *Schlussfolgerungen* und letztlich *Handlungen* generiert werden.

„Ganz allgemein und philosophisch eher traditionell ausgedrückt, fragt die Philosophie des Geistes nach der ‚Natur‘, nach dem ‚Wesen‘ des menschlichen Geistes.“ [Tet92, S.175] formuliert dies der Philosoph *Holm Tetens* auf der schon erwähnten *Dagstuhl-Tagung 1992*.

Die bisher untersuchte Arbeitsweise kann verstanden werden als das Untersuchen, Nachvollziehen und Verstehen von (automatisierten) Handlungen — also Ausführen von Algorithmen — der Maschinen analog von mechanischen oder geistigen Tätigkeiten des Menschen entsprechend der konkreten Peripherie. Fragen nach dem *Wissen* und dem *Anwenden von Wissen* sind Kernfragen der *künstlichen Intelligenz*. Die *künstliche Intelligenz* ist der Versuch, die Funktionsweise des Denkens und der Herausbildung der Handlungen nachzuvollziehen und zu praktizieren. Die Ergebnisse dieser dem menschlichen nachempfundenen Denken müssen dann wieder in entsprechendes Handeln überführt werden, um die Stimmigkeit dieser Nachbildung zu zeigen.. Damit befinden wir uns auf einer Abstraktionsstufe über dem bisherigen algorithmischen Handeln.

Die Querverbindungen zwischen *Informatik* und *Philosophie* sind auf dem Gebiet der *künstlichen Intelligenz* überdeutlich. Auch wenn in der Öffentlichkeit Computer schon immer als *Elektronengehirne* angesehen und verstanden wurden und werden, ist die Interpretation und Entwicklung der Computer im Rahmen der *künstlichen Intelligenz* eine neue Stufe.

Aus Sicht der Philosophen muss diese Debatte auch aus ihrem Selbstverständnis geführt werden. Beispielhaft mögen das folgende Darlegungen zeigen: Der Philosoph *Tetens* fragt entsprechend:

„Können mentale Zustände Teile der physischen Welt sein und, wenn ja, wie können sie es sein? Das ist die Leitfrage, um die die Diskussionen der Philosophie des Geistes in den letzten Jahrzehnten unaufhörlich kreisen.

Computer spielen in den Antworten vieler Philosophen auf diese Leitfrage eine prominente Rolle. Denn namhafte Vertreter der Informatik, genau der Künstlichen Intelligenz erheben ja immer wieder den Anspruch, es ließen sich Maschinen entwerfen und bauen, die ‚Geist haben‘, und die sich so verhalten, wie sich Menschen deshalb verhalten, weil diese ‚Geist haben‘. [Tet92]

Also erhoffen die Philosophen, mit den Ergebnissen der Forschung auf dem Gebiet der *künstlichen Intelligenz*, die zentrale Frage der *Natur des Geistes* als *physische Existenz* verifizieren oder falsifizieren zu können.

Es versteht sich fast von selbst, dass nicht alle Philosophen die Position teilen, dass der Geist und damit auch intelligentes Verhalten als komplexe Maschine verstanden werden kann und soll. *Ray Kurzweil* schreibt:

„Dennoch läßt sich der Existenzialismus als eine intellektuelle und kulturelle Reaktion auf die starke Tendenz des abendländischen Denkens verstehen, immer mehr auf Rationalität und analytische Betrachtungsweisen zu bauen. Für den Existenzialismus ist menschliche Wirklichkeit nahezu das genaue Gegenteil von dem, was der Logische Positivismus als Realität begreift. Die analytischen und synthetischen Typen von Aussagen, für den Logischen Positivismus die einzig sinnvollen, sind für den Existenzialismus bedeutungslos oder trivial. Im geistigen Sein und im Gefühlsleben — sinnlose Begriffe für den Logischen Positivisten — verbirgt sich der wahre Sinn.“ [Kur93, S.35]

Der amerikanische Philosoph und Kritiker der künstlichen Intelligenz *Hubert Dreyfus* formuliert:

„Ich fragte mich — wie ich meinen Studenten mitteilte — mehr und mehr, wie Computer überhaupt Intelligenz besitzen können – schließlich hatten sie keinen Körper, keine Kindheit und waren nicht vertraut mit unserer Kultur.“ [DD87, S.23/24]

Es ist gar nicht so einfach, den Begriff *Intelligenz* begriffsmässig überhaupt zu fassen. Wie kann man denn dann aber *Künstliche Intelligenz* beschreiben und betreiben, wenn *Intelligenz* als Begriff nicht klar definiert ist. *Marvin Minski*, einer der prominentesten Vertreter und Begründer dieser Teildisziplin, meint dazu: „Deshalb ist wohl die Bedeutung des Wortes ‚Intelligenz‘ so schwer zu fassen: Es beschreibt nicht etwas fest Umrissenes, sondern lediglich den gegenwärtigen Stand unserer Unwissenheit darüber, wie der Geist funktionieren könnte.“ [Min93]

Also verwundert es nicht, wenn die Computer in ihrer Anfangszeit als *Elektronengehirn* oder auch *Supergehirn* bezeichnet wurden, bis die Wissenschaft gezeigt hat, dass diese Geräte doch zumeist nichts anderes tun, als relativ ‚dumme‘ *Algorithmen* auszuführen. Kein Mensch heute würde die Fähigkeit, irgendwelche Buchungsdaten korrekt zu erfassen und zu berechnen als intelligentes Verhalten deuten. Diese Verschiebung des Begriffes *Intelligenz* in Bezug auf den Computer erleben wir allerdings auch bei dem Begriff *Intelligenz beim Menschen*. Eine der ursprünglichen Definitionen — Benutzung von Werkzeugen — ist mehr als fragwürdig geworden, seitdem *Klaus Köhler* 1921 erstmals die Verwendung von Werkzeugen in der Tierwelt publiziert hat [Köh21],[Köh59].

Eine adäquate Definition von Intelligenz versucht daher nicht, ein konkretes Verhalten oder konkrete Fähigkeiten als zwingenden Teil von Intelligenz festzulegen, sondern diese durch eine rekursive Definition zu umschreiben: „Ein intelligenter Prozeß besteht aus einer Verbindung intelligenter und unintelligenter Prozesse, die untereinander kommunizieren und sich gegenseitig beeinflussen.“ [Kur93, S.145]

Ausgehend von dieser Definition der Intelligenz stellt die *Künstliche Intelligenz* ein Gebiet dar, in dem versucht wird, verschiedene Facetten intelligenten Verhaltens zu erkennen und mit Maschinen — also Computern — nachzubilden. Jede dieser Teilbereiche, ob *Expertensysteme*, *Mustererkennung*, *Sprachanalyse*, *Robotik*, *Problemlösung*, [...], verlangt dabei eigene Lösungsansätze und Strategien.

Mit jeder gefundenen (Teil-)Lösung erarbeiten Informatiker wie Philosophen einen kleinen Stein in dem übergrossen Mosaik auf der Suche nach der Klärung des Seins und der Eigenschaften des (menschlichen) Geistes und der Intelligenz. Beide — Informatiker und Philosophen — benötigen entsprechend das Denken der jeweilig anderen Disziplin.

## 2.6.4 Ethik

Damit ist die *Intelligenz* auch ein *Thema der Informatik*. Folglich ist auch die *Ethik* ein solches Thema, ebenfalls auch ein klassisches Thema der Philosophie. Das bedeutet zwangsläufig, dass sich ein weiteres Mal die *Philosophen* in die Debatte informatischer Themen einmischen und auch verstehen müssen.

Aus Sicht der Informatik hat schon sehr früh *Joseph Weizenbaum* aus ethischen Gründen die Informatik und vor allem die künstliche Intelligenz kritisiert [Wei78]. Der oben schon erwähnte *Dreyfus* ergänzt diese vom Standpunkt des Philosophen. *Ekkehard Martens* als Philosoph formuliert folgerichtig: „Die ambivalenten Folgen der gegenwärtigen drei Schlüsseltechnologien der Atom-, Gen- und Computertechnik fordern die Philosophie gegenwärtig in erster Linie zur Lösung oder Beratung in einzelnen Problemfeldern in ihrer praktischen Kompetenz heraus.“ [Mar92]

Folgerichtig hat sich in den verschiedensten Organisationen im Umfeld der Informatik eine Diskussion entwickelt, die das Handeln der in diesem Feld arbeitenden Menschen einer ethischen Verpflichtung „als Teil und Ausdruck eines sich gestaltenden berufsständischen Ethos“ [Cap92b, S.137] unterwerfen will. Entsprechend dieser ethischen Verantwortung hat sich die *Gesellschaft für Informatik(GI)* im Jahre 1994 *ethische Leitlinien* gegeben, die im Jahre 2004 überarbeitet wurden<sup>19</sup>.

*Ethisches Handeln* ist ohne eine Form der *Freiheit* nicht denkbar. Nur wenn ein Mensch zumindest teilweise frei ist, ist ein ethisches Handeln möglich, da ich als Mensch Entscheidungen treffen kann und muss. Wenn der Mensch ein ausschliesslich physikalisch determiniertes Wesen ist, scheint diese Entscheidungsvielfalt gegen Null zu gehen. Zimmermann „[...] zeigt auf, dass der Begriff der Freiheit aus Sicht der Chaos- und Komplexitätstheorie aus einer neuen Perspektive verstanden werden muss und sich die Verantwortung für Entscheidungen und Handlungen nicht daraus ableitet, dass es auch alternative Möglichkeiten gegeben hätte, sondern aus den Situationen und Motiven, die der Entscheidung und Aktion zugrunde lagen. Und dies gilt für Mensch und Maschine gleichermaßen.“ [Zim11, S.8] Damit werden Informatik und Philosophie endgültig verknüpft.

## 2.6.5 Fazit Philosophie

Mit dem Aufkommen der ersten vereinzelt, aus heutiger Sicht völlig leistungsschwachen Computer wurden diese in Worten mit dem menschlichen Gehirn verglichen. Spätestens als nicht nur mehr numerische Probleme mit dem ‚Rechner‘ bearbeitet wurden, stellte sich die Frage auf der einen Seite der Ethik, auf der anderen Seite nach den Möglichkeiten dieser neuen Technik im Verhältnis zu den Menschen. Spätestens mit dem Auftreten des Begriffes der *Künstlichen Intelligenz* wird dieser Anspruch auch in Worten formuliert. Dies fordert *Philosophie* und *Informatik* gleichermaßen heraus. Für beide bedeutet das allerdings auch ein Hineindenken in die Gedanken- und Arbeitswelt der jeweils ‚anderen‘ Seite. Diese Diskussion ist noch lange nicht zu Ende. Je mehr Fähigkeiten Computer erhalten, desto wichtiger und intensiver werden diese Diskussionen werden. Ohne Übertreibung lässt sich feststellen: Phi-

---

<sup>19</sup><http://www.gi.de/?id=120>, zuletzt überprüft: 24.5.2012



losophie betreiben ohne Informatik-Kenntnisse wird in der Zukunft kaum noch sinnvoll sein können.

Der Technik-Philosoph *Rafael Cappuro* formuliert es so: „Sowohl als Verstandes- als auch als Vernunftwesen ist der Mensch ursprünglich ein Handelnder. Philosophie als Wissenschaft des vom Handeln (griechisch ‘praxis’) her aufgefaßten Menschen ist ursprünglich ‚praktische Philosophie’. [...]“

Es geht daraus also klar hervor, daß die Informatik, sofern sie in alle Bereiche menschlicher Praxis eingreift, eine Herausforderung für die praktische Philosophie ist.’ [Cap92a, S.344/345]

## 2.7 Biologie

Das Thema *Intelligenz* ist nicht nur ein philosophisches, sondern auch ein biologisches Thema. Natürlich interessiert die Biologen im Rahmen der Verhaltensbiologie, warum sich ein Lebewesen in bestimmten Situationen wie verhält und über welche intelligenten Fähigkeiten ein konkretes Tier verfügt. Zudem interessiert die Biologen zusätzlich, welche biologischen und eventuell auch chemischen Prozesse ein Lebewesen zu einem denkenden Lebewesen führen.

### 2.7.1 Neuronale Netzwerke

Damit befinden wir uns in einer dritten Schicht der Betrachtung von Intelligenz und intellektuellem Handeln. Diese oberste Schicht ist verantwortlich für die Durchführung und Steuerung einer konkreten Handlung, die in der zweiten Schicht nach bestimmten Kriterien, sprich Überlegungen, begründet und ausgewählt wird. Die unterste Schicht sind die technische Konstruktionen, die solche Denkprozesse ermöglichen und produzieren. Bei den Lebewesen handelt es sich dabei um das Gehirn, das nach bestimmten Prinzipien aufgebaut ist und diese Leistung vollbringt.

Die oberste Schicht ist aus informatorischer Sicht der Prozess, der aufgrund eines Algorithmus und dem darauf aufbauenden Programm durchgeführt wird. Die zweite Schicht ist, sofern vorhanden, beispielsweise ein Expertensystem, das aus einer Wissensbasis und einem Regelsystem besteht und dann selber die Handlungen durchführen lässt oder einem anderen System mit einer Schnittstelle die Parameter für einen konkreten Prozess mitteilt. Die unterste Schicht ist in diesem Fall eine konkrete Computerhardware, auf der diese oberen Schichten eingesetzt werden.

Die Alternative zu dieser Ausführung ist ein *neuronales Netz*, das ohne die Konstruktion derartiger beispielhafter Expertensysteme zu den konkreten Ergebnissen kommt, wie es vergleichsweise bei einem menschlichen Gehirn organisiert und praktiziert wird.

„Ein künstliches Neuronales Netzwerk als DV-technisches Modell des Gehirns besteht folglich nicht aus einer komplexen Zentraleinheit(CPU) und einem Arbeitsspeicher wie ein konventioneller Computer, sondern wie das Gehirn [...] aus sehr vielen einfachen Rechenelementen, den Neuronen (Nervenzellen). Diese Neuronen sind hochgradig miteinander verknüpft und können dadurch gleichzeitig untereinander Informationen austauschen.“ [SHG90, S.14]

beschreibt *Eberhard Schöneburg* ein *neuronales Netzwerk*. Ein künstliches neuronales Netzwerk ist damit die versuchte Nachbildung — also eine Simulation — des biologischen Gehirns. Dabei werden „die für das Lernen verantwortlichen synaptischen Veränderungen zwischen Neuronen DV-technisch durch reelle Zahlen(Gewichte) und Lernregeln(Algorithmen zur Veränderung der Gewichte während der Lernphase) simuliert“ [SHG90, S.14]. Im Gegensatz zum klassischen Computer realisieren und regeln die Algorithmen nicht mehr das Umsetzen von vorgedachten und geplanten Handlungen, sondern sind jetzt verantwortlich für das Trainieren und Einstellen des künstlichen neuronalen Netzwerkes bei einem bestimmten Problem.

Neuronale Netzwerke sind dabei keine Erfindung der letzten Jahre. Schon 1943 zeigten der Neurophysiologe *Warren MC Culloch* und der kognitive Psychologe *Walter Pitts*, dass neuronale Netzwerke logische Schaltungen simulieren können:

*'It is easily shown: first, that every net[...] can compute only such numbers as can a Turing machine; second, that each of the latter numbers can be computed by such a net; and that nets with circles can be computed by such a net; and that nets with circles can compute [...] some of the numbers the machine can, but no others; and not all of them.'* [MP90, S.113]

Die darauf einsetzende stürmische Entwicklung führte dann 1958 zur Entwicklung eines Modells *Perceptron* des Psychologen *Frank Rosenblatt*. „Das Perceptron konnte sich teilweise selbst organisieren, war lernfähig, konnte einfache Muster klassifizieren, war fehlertolerant und konnte in einem gewissen Sinne Erfahrungen verallgemeinern, indem es auf bislang unbekannte und untrainierte Eingaben korrekt antwortete, wenn die Eingaben nur eine ‚gewisse‘ Ähnlichkeit zu den dem Netzwerk vorher präsentierten Eingaben hatten“ [SHG90, S.70]

Die intensiven Arbeiten an derartigen Netzen gerieten ins Stocken, nachdem 1969 *Marvin Minsky* und *Seymour Papert* in ihrem Buch *Perceptrons* [MP69] Grenzen dieser Technik beschrieben hatten [SHG90, S.70],[Kur93, S.140]. Erst ab den 80er Jahren mit den Erkenntnissen des Molekularbiologen *John Joseph Hopfield* [Hop82] wurden die Arbeiten wieder aufgenommen. Die zuvor berechtigte Kritik an den neuronalen Netzen konnte er einschränken, da diese sich auf sehr einfache Netzwerke mit nur einer Schicht bezogen.

In beiden Fachdisziplinen Informatik und Biologie entstanden die Spezialisierungen *Neuroinformatik* und *Neurobiologie*. Die Beziehungen sind so eng, dass die Grundlagen sogar schon in einem gemeinsamen Lehrbuch mit dem Titel ‚*Grundlagen zur Neuroinformatik und Neurobiologie*‘ [CS97] zusammengefasst werden .

## 2.7.2 Genetik

Im Jahre 1953 wurde von den Biochemikern *James D. Watson* und *Francis H. C.Crick* die Bedeutsamkeit der *DNA* dargelegt [WC53]. „Innerhalb eines Jahrzehnts erkannten die Wissenschaftler, daß die vier Basen das Alphabet des genetischen Codes bilden“[Kev93, S.26]. Dies war nicht allein das Ergebnis von Biologen: „[...] innerhalb von 10 Jahren hatten es Mathematiker, Physiker, Chemiker, Biologen und Nachrichtentechniker in enger Zusammenarbeit

geschafft, diese Bedeutung aufzuklären und durch das Experiment zu bestätigen.“<sup>20</sup> [Bog67, S.35] Je drei Basen bilden ein sogenanntes *Triplet* das bei der Synthese eines Proteins für den Einbau einer konkreten Aminosäure steht [Bog67, S.36]. Insgesamt gibt es ca. 3 Milliarden Basenpaare, die etwa 100 000 bis 300 000 Gene bestimmen [Gil93, S.95].

Der menschliche Code ist also ‚nichts‘ anderes als ein Bitfolge, indem man jede Base mit je zwei Bit codiert. Damit lassen sich vielfältige Untersuchungen unter anderem mit den Ergebnissen der Codierungstheorie am Genmaterial durchführen, wie es *Hans Joachim Bogen* 1967 auch formuliert hat. Auch *John von Neumann* stellt in seinem Buch *Die Rechenmaschine und das Gehirn* bei der Betrachtung über *digitale und analoge Vorgänge im Nervensystem* bereits 1965 fest: „In diesem Zusammenhang spielt nun das genetische Phänomen eine ganz besondere Rolle. Die Gene sind selbst eindeutig Teile eines digitalen Systems von Bausteinen, ihre Wirkung, die jedoch analoger Natur sind, besteht darin, die Erzeugung spezifischer Chemikalien, d.h. bestimmter Enzyme, zu veranlassen, die für das in Frage kommende Gen charakteristisch sind.“ [Neu91, S.67]

Im Jahr 2003 wurde schliesslich bekannt gegeben, dass das menschliche Genom entschlüsselt ist<sup>21</sup> Damit hat sich in der zweiten Hälfte des zwanzigsten Jahrhunderts das Bild von der Biologie völlig verändert, eine vergleichbare Entwicklung wie in der Physik in der ersten Hälfte des zwanzigsten Jahrhunderts.

Der Wissenschaftsjournalist *Horace Freeland Judson* schreibt dazu: „Die Genetik ist das zentrale Problem der Biologie“ [Jud93, S.50]. Und wenig später: „Bis zur Entdeckung der DNA-Struktur konnten die Biologen nur tun, was bereits Mendel getan hatte, nämlich das Vorhandensein und die Wechselwirkungen von Genen anhand ihrer sichtbaren Anzeichen herzuleiten.“

Die Zusammenarbeit von Biologen mit Informatikern und anderen Wissenschaftlern ist auf dem Gebiet der Genetik auch weiterhin zwingend notwendig. Der Mediziner und Biologe *Leroy Hood* stellt fest:

„Interdisziplinäre Zusammenarbeit wird im nächsten Jahrhundert grundlegend für die Zukunft der Biologie sein, wenn sie sich der Analyse von komplexen Systemen, wie Moleküle, Zellen oder Zellenverbände, widmet“ [Hoo93, S.169]. Und: „Die kombinatorischen Probleme der Biologie, wie beispielsweise der Sequenzvergleich, werden die Entwicklung neuer Algorithmen und neuer Hardware wie spezialisierte Koprozessoren und in zunehmenden Maße den Einsatz von Parallelrechnern erfordern. [...]

In Zukunft werden Biologen in hohem Maße darauf angewiesen sein, Computermodelle von komplexen System und Netzwerken herzustellen, um neue Hypothesen an biologischen Systemen oder lebenden Organismen zu überprüfen.“[Hoo93, S.182/833]

Also ist nicht nur Interdisziplinarität verlangt, sondern auch die Erweiterung des Wissens und Könnens der Biologen selber in Richtung informatisches Wissen. Es wundert daher nicht,

---

<sup>20</sup>Statt *Nachrichtentechniker* würde der Autor wahrscheinlich heute *Informatiker* schreiben, aber diese Bezeichnung war 1967 noch nicht gebräuchlich

<sup>21</sup>Die abgeschlossene Genom-Entschlüsselung im Jahre 2003 wird auf der ‚offiziellen‘ Website des *Human Genome Project* [http://www.ornl.gov/sci/techresources/Human\\_Genome/home.shtml](http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml), letzter Zugriff: 29.5.2012, angegeben.

dass sich in den letzten Jahren die *Bioinformatik* als Bindeglied zwischen Informatik und Biologie etabliert hat<sup>22</sup>.

### 2.7.3 Fazit Biologie

Der Biochemiker und Nobelpreisträger für Chemie 1939 *Adolf Butenandt* schreibt in einem Geleitwort zum Buch *Knaurs Buch der modernen Biologie*:

„[...] in dem ersten Kollegheft, das ich als junger Marburger Student im Sommersemester 1921 geführt habe, ist als noch gültige Lehrmeinung zu lesen, man könne die Naturwissenschaften in zwei nach Problemstellung und Methodik grundsätzlich unterschiedene Bereiche einteilen: Physik und Chemie seien ‚exakte‘, Biologie und Geowissenschaften dagegen ‚deskriptive‘ (beschreibende) Wissenschaften. [...] heute stellt man die Frage: ‚Wieweit lassen sich Grundphänomene des Lebens durch systematische Anwendung chemischer und physikalischer Denkweisen lösen?‘“ [Bog67]

In der Zwischenzeit ist aus dieser deskriptiven Wissenschaft Biologie nicht nur eine exakte Wissenschaft geworden, sondern auch eine Wissenschaft, die neben den exakten Methoden der Chemie und Physik auch Techniken, Modelle und Strukturen der Informatik anwendet, nutzt und reflektiert.

## 2.8 Zusammenfassung und Fazit

Ob *Informatik* als Schulfach berechtigt ist, kann unter anderem aus fachwissenschaftlicher Sicht abgeleitet werden. In diesem Kapitel haben wir die Beantwortung dieser Frage von einer Warte aus betrachtet: Wir haben nicht hinterfragt, ob Informatik eventuell mit seinen Produkten für den Einzelnen hilfreich und notwendig sein kann, auch nicht, ob die Ideen der Informatik für die Welterkenntnis dienen können, sondern ob Informatik für den Erkenntnisgewinn in anderen anerkannten Disziplinen notwendig oder zumindest wichtig ist.

### 2.8.1 Bedeutung der Informatik für andere Disziplinen

Wir haben uns die *Sprachwissenschaften*, die *Mathematik*, die *Physik*, die *Philosophie* und letztendlich die *Biologie* angesehen. In allen Fächern konnten wir feststellen: Ideen und Gedanken aus der Informatik sind zum Verständnis des entsprechenden Faches nicht nur am Rande bedeutsam, sondern diese Gedanken greifen weit in das Ideengefüge des entsprechenden Faches ein. Dabei wurde deutlich, dass zumindest in den Fächern *Mathematik*, *Physik* und *Biologie* regelrechte Paradigmenwechsel eingetreten sind, aber auch die *Sprachwissenschaften* und *Philosophie* erhebliche Erweiterungen erfahren haben.

Umso überraschter muss man sein, wenn diese zentrale Bedeutung der Informatik für den Erkenntnisgewinn vieler Fächer nicht im öffentlichen Bewusstsein vorhanden ist. Auch in diesen Fächern wird die Informatik viel zu sehr als technische Wissenschaft gesehen, die ähnlich beispielsweise dem Maschinenbau wichtige Produkte auf hohem Niveau produziert.

<sup>22</sup>Bioinformatik und ähnliche Studiengänge kann man inzwischen in der Bundesrepublik an ca. 20 Hochschulen studieren: <http://studieren.de/bioinformatik.0.html>, letzter Zugriff: 29.5.2012

## INFORMATIK: Automatische Informationsverarbeitung

Von K. Steinbuch · Standard Elektrik AG, Informatikwerk

DK 481.14-523.8

### 1. Allgemeines

Die elektrische Nachrichtentechnik hatte bis vor wenigen Jahren eine einzige Aufgabe: die Übertragung von Signalen über räumliche Entfernungen hinweg. Zum Beispiel übertragen Telegrafiesysteme die Fernschreibsignale über Tausende von Kilometern; Telefonsysteme ermöglichen Sprechverbindung mit Amerika oder Australien, und Funkverbindungen tragen das Fernsehbild über Kontinente hinweg in unsere Wohnung. In all diesen Beispielen ist die Überwindung der räumlichen Entfernung die wesentliche Leistung.

Vor etwa zwanzig Jahren entdeckten Ingenieure in USA und Deutschland unabhängig voneinander, daß die Verfahren der Nachrichtentechnik auch für andere Aufgaben nützlich sind, Aufgaben, bei denen die Überwindung der räumlichen Entfernung ganz unwesentlich ist. Sie fanden, daß man mit elektrischen Schaltungen Zahlenrechnungen durchführen kann, und zwar mit einer Schnelligkeit, wie sie bis dahin einfach unvorstellbar war. Damit begann die automatische Informationsverarbeitung. Wir nennen sie „INFORMATIK“.



Abb. 3 Aufbau eines Rechenplatzes

darstellen, entweder auf mehreren Drähten gleichzeitig, oder auf einem einzigen Draht nacheinander angeordnet sein können (Parallel- bzw. Seriendarstellung).

Abbildung 2.3: Definition des Begriffes *INFORMATIK* in den SEG-Nachrichten 1957

## 2.8.2 Geschichte des Begriffs Informatik

Eine Erklärung könnte auch in der etwas unglücklichen Geschichte der Bezeichnung der eigenen Disziplin liegen. Der Begriff *Informatik* wurde erstmals von *Karl Steinbuch* in den „SEG-Nachrichten Heft 4/1957“ (Abbildung 2.3) für die *Automatische Informationsverarbeitung* [Ste57] verwendet und hat sich dann mit der immer grösseren Bedeutung der dazu gehörigen Maschinen — den Computern — Ende der 60er Jahre als Fachbezeichnung durchgesetzt. Steinbuch schreibt im gleichnamigen Artikel:

„Sie [Ingenieure in den USA und Deutschland] fanden, daß man mit elektrischen Schaltungen Zahlenrechnungen durchführen kann, und zwar mit einer Schnelligkeit, wie sie bis dahin einfach unvorstellbar war. Damit begann die automatische Informationsverarbeitung. Wir nennen sie ‚INFORMATIK‘.“ [Ste57, S.171]

Für viele war damit offensichtlich der Begriff mit dem *technischen Gerät Computer* verknüpft<sup>23</sup> Der allgemeinere über die Technik hinausgehende Begriff in den 50er bis 70er Jahren war der der *Kybernetik*. *Norbert Wiener* begründet dieses Begriff in seinem berühmten Buch *Cybernetics or Control and Communications in the Animal and the Machine* [Wie48] (Original-Titelblatt siehe Abbildung 2.4<sup>24</sup>) und definiert 1948: „Wir haben beschlossen, das

<sup>23</sup>Ganz deutlich sieht man diese Verbindung bei der englischen Bezeichnung *computer science*.

<sup>24</sup>Quelle: <http://books.google.de/books?id=NnM-uISyywAC&printsec=frontcover&hl=de>

gesamte Gebiet der Regelungstechnik und der Informationstheorie, ob bei Maschine oder Lebewesen, mit dem Namen Kybernetik zu belegen.“ [Wie68, S.32] Der Biologie- und Mathematiklehrer und spätere Professor für Erziehungswissenschaften *Felix von Cube* stellt einige Jahre später richtigerweise fest: „Daß die Kybernetik auf diese Weise zur Brücke zwischen den Wissenschaften wird, heißt insbesondere, daß bestimmte, ursprünglich aus der Technik stammende, mathematische Methoden auch in Geisteswissenschaften und Sozialwissenschaften Eingang gefunden haben.“ [Cub71, S.40] Cube formuliert die Definition von Kybernetik folgendermassen um:

**Definition Kybernetik:**

„Faßt man alle Gesichtspunkte zusammen, so könnte man den Begriff der Kybernetik im engeren Sinne so erklären:

Unter Kybernetik im engeren Sinne versteht man die Wissenschaft und Technik von den informationsverarbeitenden Maschinen; unter Kybernetik im weiteren Sinne versteht man die mathematische und konstruktive Behandlung allgemeiner struktureller Beziehungen, Funktionen und System, die verschiedenen Wirklichkeitsbereichen gemeinsam sind.“ [Cub71, S.41]

Noch kürzer und knapper formuliert etwa 10 Jahre später *Steinbuch*:

**Definition Kybernetik:**

„Kybernetik ist die Wissenschaft von den informationellen Strukturen — gleichgültig, ob diese nun in der Biologie, Technik, Soziologie, Ökonomie usw. auftreten.“ [Ste78, S.39]

Logischerweise werden die entsprechend dieser Definition entwickelten Maschinen auch *kybernetische Maschinen* genannt. „Bei den kybernetischen Maschinen, programmgesteuerten Rechenanlagen, Computern, oder wie man diese neuen Maschinen noch bezeichnen mag, handelt es sich nicht — wie in der traditionellen Technik — um Energieerzeugung und Arbeitsleistung (im physikalischen Sinne), sondern um die Aufnahme, Speicherung und Verarbeitung von Information.“ [Cub71, S.17]

Bevor die *Informatik* als eigenständige Disziplin definiert wurde, war sie eigentlich Teil der *Kybernetik*. In der Kybernetik wird die *Regelung und Nachrichtenübertragung in Lebewesen und Maschine* betrachtet, wie *Wiener* in seinem Werk *Kybernetik* 1948 schreibt. Und dementsprechend werden diese Strukturen und Ideen auf allen Gebieten der Wissenschaft untersucht. Da die Anzahl der Computer zu der Zeit verschwindend gering war — *Steinbuch* beschreibt die Anzahl der Computer mit unter 200 im Jahr 1955 in den USA und in der BRD im Jahr 1959 [Ste69] — ist das leicht nachzuvollziehen. Da mag es vielleicht manchen Nicht-Techniker gefreut haben, das mit dem Aufkommen der technischen(?)

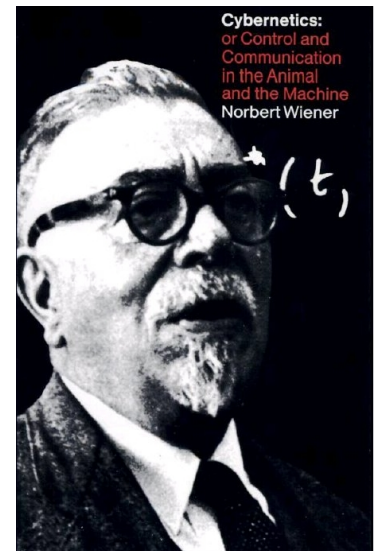


Abbildung 2.4: Cybernetics

---

#v=onepage&q&f=false,  
letzter Zugriff: 16.11.2012

*Teildisziplin Informatik* diese kybernetischen Gedanken und Ansprüche auf den technischen Bereich der Maschinen reduziert wurde.

### 2.8.3 Informatik und informatisches Denken

Der zentrale Gedanke „Information ist Information, weder Materie noch Energie. Kein Materialismus, der dieses nicht berücksichtigt, kann heute überleben.“ [Wie68, S.166] von Wiener aus dem Jahre 1948 hat allerdings zur Folgerung, dass in allen Wissenschaften das Nachdenken über informationelle Prozesse mit informatischen Methoden und Ideen durchgeführt werden muss. Und das ist nichts anderes als *Jeannette M. Wing* mit *Computational Thinking* gemeint hat. Der Informatikdidaktiker *Norbert Breier* hat mit der Definition eines *informativ-orientierten Ansatzes* [Bre05] für ein Wiederaufgreifen dieser Gedanken in der Informatikdidaktik argumentiert. Auch *Peter Hubwieser* schliesst sich diesem Gedanken an [Hub07, S.78].

In ‚leichter‘ Abwandlung der Definition von Steinbuch können wir damit als Verdeutlichung der Gedanken von Wing formulieren

*Informatisches Denken* ist das Denken in informationellen Strukturen — gleichgültig, ob diese nun in der Biologie, Technik, Soziologie, Ökonomie usw. auftreten.

Wir haben in diesem Kapitel gesehen, dass die Informatik erkenntnistheoretisch in vielen anderen Disziplinen mindestens hilfreich, wenn nicht sogar notwendig ist. Daraus ergibt sich, dass es zumindest sinnvoll ist, Informatik auch in der Schule verstärkt als Schulfach anzubieten, da ansonsten im späteren (Aus-)Bildungs- und Arbeitsleben diese Kenntnisse angeeignet werden müssten, um im eigenen Fach die notwendigen Qualifikationen zu erhalten.

### 2.8.4 Curriculare Konsequenzen

Schule hat eine allgemeinbildende Aufgabe. Inhalte können für sich einen Allgemeinbildungswert besitzen. So ist es sicher sinnvoll, allen Schülern Grundkenntnisse in der Atomphysik zu vermitteln. Dieser Inhalt ist dann ohne Berücksichtigung anderer Kriterien als Gegenstand gerechtfertigt.

Andere Gegenstände lassen sich rechtfertigen, weil sie für das Verständnis an anderer Stelle im selben oder in einem anderen Fach notwendig sind. So versteht sich der enge Zusammenhang der Schulmathematik mit der Schulphysik. Deutlich wird dies bei der Begründung der Veränderung der Schulmathematik anfangs des 20. Jahrhunderts durch *Felix Klein*. Als eine Begründung für seine Ideen formuliert Klein 1904 auf den *Verhandlungen zu Breslau der Gesellschaft deutscher Naturforscher und Ärzte*: „Wir werden in der Lage sein, die mathematische Behandlung physikalischer Aufgaben, die in den physikalischen Stunden die freie Entfaltung des physikalischen Gedankens so häufig hemmt, in die mathematischen Stunden hereinzunehmen, ebenso beispielsweise die für die Schule unerläßlichen mathematischen Entwicklungen aus den Gebieten der Geographie und Astronomie.“ [Gut08, S.50] An diesem Beispiel wird deutlich, dass Fächer und Inhalte wie hier die Mathematik als geistiges Rüstzeug für andere Inhalte und Fächer wie hier die Physik legitimiert werden. Sicherlich gilt das nicht nur

die Beherrschung des Kalküls der Infinitesimalrechnung als Werkzeug für die Physik, sondern generell für das mathematische Denken für die Physik.

In der Schule bilden wir die Schüler in vielfältigem Denken aus. Wir unterrichten das mathematische, physikalische, informatische Denken nicht als Selbstzweck, sondern als geistiges Werkzeug in vielen verschiedenen Disziplinen. Dementsprechend hat auch die curriculare Auswahl der Inhalte diesem Zwecke zu entsprechen. Das gilt für alle Fächer und ist damit generell eine Absage an ein didaktisches Abbild der universitären Fächer in der Schule. Wenn das informatische Denken heute ein notwendiges geistiges Rüstzeug darstellt, so muss das im Curriculum deutlich werden.

So müssen beispielsweise Mathematik und Informatik darüber nachdenken, wer welche Inhalte der Graphentheorie vermittelt. Ebenso könnte die Relationenalgebra als mathematische Grundlage von (relationalen) Datenbanken, erstmals von *Edgar Frank Codd* [Cod70, Cod72] beschrieben, in der Mathematik oder Informatik behandelt werden. In der linearen Algebra könnten theoretische Betrachtungen von endlichen Automaten durch Übergangsmatrizen angestellt werden. Auf der anderen Seite könnte die Computermathematik ein Thema in der Informatik sein. An dieser Stelle soll kein Vorschlag gemacht werden oder eine Entscheidung gefällt werden, welches Fach in der Schule welche Inhalte vermitteln soll. Schon 1998 fordert *Rüdeger Baumann*: „Neuerdings fällt der Mathematik (zusammen mit Logik und Linguistik) die Rolle als Grundlagendisziplin auch der Informationswissenschaften, insbesondere der Informatik, zu. In dieser Funktion hat sie die elementare *Algorithmik*, Beispiele aus *Graphentheorie* und *Kombinatorik*, *Optimierung* über endlichen Mengen, Anfänge der *Zahlentheorie* zu vermitteln (was gern unter dem Stichwort ‚Diskrete Mathematik‘ zusammengefasst wird)“. [Bau98, S.44/45, (Hervorhebungen im Original)] Die Mathematikdidaktikerin *Astrid Beckmann* sieht dies eher realisiert als fächerübergreifenden Unterricht und entzieht sich damit einer curricularen Festlegung im Schulfach Mathematik: „Mit dem Einbeziehen von informatischen Aspekten in den Mathematikunterricht wird eine Form des fächerübergreifenden Unterrichts zwischen Mathematik- und Informatikunterricht angesprochen. . . . Eine grobe Einteilung der Informatik führt auf fünf Aspekte . . . . Neben den typisch informatischen Aspekten gibt es Methoden und Konzepte, die beide Disziplinen betreffen und damit einen besonderen Ansatz für eine Kooperation versprechen.“ [Bec05, S.9/10]

In der Informatik müssen kryptologische Ideen so vermittelt werden, dass neben Mathematikern beispielsweise auch Biologen davon profitieren. Philosophie und Informatik müssen gegenseitig Verständnis für die Frage Mensch und Maschine entwickeln. Die Idee der *Digitalisierung* wird in der Physik und Informatik benötigt. Dies sind erste völlig unvollständige Diskussionsansätze für ein Curriculum, das aus Sicht der Informatik dem *informatischen Denken* als geistiges Werkzeug verpflichtet ist.



# 3 Bildungswissenschaftliche Legitimation

## Überblick

Damit eine Fachdisziplin in der Schule als Unterrichtsfach gerechtfertigt ist, sind mehrere Rechtfertigungsebenen zu berücksichtigen. Dies ergibt sich allein aufgrund des Mangels an Zeit in der Schule. Auf der einen Seite muss das Fach seine fachliche Bedeutung für sich und andere Fächer unter Beweis gestellt haben. Dies ist im Kapitel 2 beschrieben worden.

Zusätzlich muss aber gezeigt werden, dass sich dieses Fach für eine Allgemeinbildung eignet oder sogar notwendig ist. Um dies festzustellen ist Voraussetzung, dass das Begriffspaar Bildung — Allgemeinbildung analysiert und genau beschrieben wird. Für diese Allgemeinbildung sind dann Kriterien heraus zu arbeiten, sodass für ein Fach der allgemeinbildende Charakter festgestellt werden kann. In den letzten Jahrzehnten sind hierzu vor allem in der Zeit der Schulreform vor ca. 50 Jahren wichtige wissenschaftliche, aber auch politische Debatten geführt worden. In den 90er Jahren hat dann Heymann diese Debatte erneut aufgegriffen, um die Frage nach Bildung und Allgemeinbildung zu versachlichen. Heymann hat entsprechend Kriterien für die Allgemeinbildung von Fächern und Lerngegenständen aufgestellt. Diese eignen sich gut, die Allgemeinbildung von Informatik als Schulfach zu untermauern. Die Debatte der letzten 10 Jahre, die nach den sogenannten PISA-Untersuchungen die Bildungsdiskussion bestimmt und sich eigentlich nur auf wenige Fächer beschränkt, darf dabei nicht die notwendige Diskussion über sämtliche Schulfächer und deren Kompetenzen zuschütten.

Da Bildung und Schule in erster Linie in der gesellschaftlichen Wirklichkeit und nicht in der akademischen Wissenschaft definiert und gestaltet wird, existieren neben diesen sachlichen Argumenten für und gegen ein Fach auch historische und normative Zwänge. Daher ist es empfehlenswert, auch die Schule in ihrer historischen Entwicklung zu betrachten und die Auswirkungen dieser Geschichte auf die heutige und eine mögliche zukünftige Fachstruktur zu berücksichtigen.

Ausgehend von den Überlegungen im vorigen Kapitel könnte man erwarten, dass Informatik inzwischen seit Jahren ein bedeutsames Fach in der Schule ist. Das ist jedoch nicht der Fall. Wir wollen feststellen, warum das so ist und welche zusätzlichen bildungswissenschaftlichen Anforderungen zu erfüllen ist.

Informatik ist als Schulfach von Beginn der reformierten gymnasialen Oberstufe 1972 durch einen Beschluss der *ständigen Konferenz der Kultusminister* möglich [INF84]. Allerdings nehmen nur relativ wenige Schülerinnen und Schüler am Informatik-Unterricht teil. So wurden beispielsweise im Schuljahr 2010/2011 in der gymnasialen Oberstufe an den Gymnasien und Gesamtschulen knapp 90 000 Schülerinnen und Schüler an ca. 750 Schulen in Physik unterrichtet, aber nur ca. 40 000 Schülerinnen und Schüler an knapp 600 Schulen in Informatik<sup>1</sup>.

Das heutige Fächerangebot in der Schule ist teilweise über viele Jahrzehnte historisch gewachsen, wofür es vielfach ideologische Gründe gegeben hat. Mit ‚ideologisch‘ ist gemeint, dass beispielsweise wie bei der *klassischen Bildung* im 19. Jahrhundert ein konkretes Menschenbild die Inhalte der Schulbildung wesentlich bestimmt haben. Die Schwierigkeiten gegen ein solches gesetztes Menschenbild neue Inhalte in und für die Schule durch zu setzen,

<sup>1</sup>[http://www.schulministerium.nrw.de/BP/Schulsystem/Statistik/2010\\_11/StatUebers373.pdf](http://www.schulministerium.nrw.de/BP/Schulsystem/Statistik/2010_11/StatUebers373.pdf), S. 82, letzter Zugriff: 17.10.2012

zeigte die Auseinandersetzung um die naturwissenschaftlichen Fächer um die Jahrhundertwende [Hir87, Gut08]. Mit ‚ideologisch‘ ist auch gemeint, dass Personen oder Gruppen so fest an die Sinnhaftigkeit bestimmter Inhalte und Fächer glauben, dass gegen diese Positionen kaum eine Argumentation bestehen kann. Und nicht zuletzt ist ‚ideologisch‘ gemeint, dass Bildung immer wieder eine Auseinandersetzung in der konkreten Politik in Wahlkämpfen, weil viele glauben, hier mitreden zu können und sollen. Dieser Prozess ergibt eine *normative* Setzung von Bildungsinhalten. Veränderungen können dagegen nur sehr schwer durchgeführt werden. Um hier die Chancen zu erhöhen, ist es hilfreich, die eigenen Positionen belegen zu können. In diesem Rahmen sind wissenschaftliche Ergebnisse von nicht geringem Wert.

Die konkrete Fachstruktur heute ist u.a. ein Ergebnis von bildungspolitischen und bildungswissenschaftlichen Auseinandersetzungen vor allem ab den 60er-Jahren. Ein Fach, das in der Schule stärker als bisher vertreten sein möchte, muss sich in einer Auseinandersetzung mit den etablierten Fächern qualifizieren. Daher werden in diesem Kapitel die wesentlichen bildungspolitischen Auseinandersetzungen, die zum heutigen Schulsystem geführt haben, analysiert. Wir haben dabei ergänzend zu untersuchen, nach welchen Kriterien aus bildungswissenschaftlicher Sicht ein Fach als Schulfach generell begründet wird und welche Konsequenzen dies für das Schulfach Informatik hat.

Vor allem in den 60er Jahren begann eine heftige Diskussion über Inhalte und Strukturen von Schule und Hochschule, deren politische Ergebnisse die Voraussetzungen für die heutige Struktur des Bildungssystem geschaffen haben. Wir betrachten daher zunächst einige wichtige Positionen der damaligen Diskussion. Anschliessend widmen wir uns den fast parallel laufenden theoretischen Auseinandersetzungen um den Bildungsbegriff, der in dieser politisch so brisanten Zeit eine Wandlung erfuhr, sowie der daraus gefolgerten didaktischen Positionen.

Erst Mitte der 90er Jahre lebte die Debatte um die Bildung wieder auf und führte schliesslich ab dem Jahre 2000 zu den bis heute andauernden Auseinandersetzungen um die Leistungsfähigkeit des deutschen Bildungssystems. Mitten in dieser Debatte befindet sich die Diskussion um die Informatik als allgemeinbildendes Fach in der Schule.

## **3.1 Das Bildungssystem der Bundesrepublik Deutschland**

Durch den Wiederaufbau des Bildungssystems in der Bundesrepublik Deutschland entsprechend der Strukturen des Schulsystems in der Weimarer Republik war die Fächerstruktur im gymnasialen Schulsystem extrem verengt worden. Schülerinnen und Schüler hatten maximal die Wahl zwischen verschiedenen ‚Zweigen‘ des Gymnasiums und einige wenige Wahlmöglichkeiten zwischen einzelnen Fächer. Weitere Angebote konnten höchstens in Form von Arbeitsgemeinschaften wahrgenommen werden.

Mit dem Beginn der politischen Veränderungen in den 60er Jahren wurden auch diese Strukturen und die Inhalte des Schulsystem in der Bundesrepublik hinterfragt. Damit bestand auch die Möglichkeit, über die bisherigen und eventuell neuen Fachangebote zu diskutieren und deren Rolle in der Schule neu zu definieren.

Die Ursachen der Diskussion um die Schule können auf wenige Probleme zusammengefasst

werden:

- Das konservativ-restaurierte Bildungssystem geriet mit den Ansprüchen der modernen Gesellschaft immer mehr in Widerspruch.
- Das Bildungssystem ‚produzierte‘ eindeutig zu wenig qualifizierte Absolventen vor allem mit höheren Qualifikationen.
- Die Situation wurde durch den Mauerbau und damit abebbenden Zustrom von gut ausgebildeten Bürgern aus der DDR verschärft.

### 3.1.1 Die deutsche Bildungskatastrophe

In der bundesrepublikanischen Gesellschaft wurden diese Probleme vor allem mit einer Artikelserie *Die deutsche Bildungskatastrophe* des *Religionsphilosophen Georg Picht* in der Zeitung *Christ und Welt*<sup>2</sup> öffentlich und lösten eine Diskussion aus, die das in den Jahren nach dem 2. Weltkrieg restaurierte bisherige Bildungswesen bis in die Grundfesten erschütterte.

Picht war allerdings nicht der Erste, der auf dieses Problem hingewiesen hat. Einige Jahre zuvor im Jahr 1956 hat der Journalist *Wilhelm Bittorf* in einem Buch mit dem Titel *Automation, Die zweite industrielle Revolution*<sup>3</sup> das beginnende Computerzeitalter und seine wahrscheinlichen Folgen beschrieben. Er schreibt:

„Gebraucht wird ein großes und vielschichtiges Programm für die Umschulung der Arbeitskräfte, für die Forschung, für die Heranbildung von Ingenieuren und Wissenschaftlern, für die Erwachsenenbildung, für die Hebung des Niveaus der Volksbildung insgesamt. Die Automation ist nicht so sehr kapital- oder maschinenintensiv als *wissensintensiv*, und das Wissen avanciert unzweifelhaft zum Betriebsmittel Nr. 1. Nie haben Männer der Industrie so händeringend um Unterstützung durch kapitalkräftige Bankiers gefleht wie heute um Gunst und Dienste der jungen Techniker und Ingenieure auf den Hochschulen.

Soll die Entwicklung nicht durch einen Mangel an ausgebildeter Intelligenz gehemmt oder gar unterbunden werden, dann muß man ganz unten anfangen, die geistigen Reserven der Gesellschaft zu mobilisieren: durch eine (*Rechtschreibfehler im Original*) neuntes und zehntes Volksschuljahr, durch Ausdehnung der Mittelschulbildung auf die Mehrheit aller Jugendlichen, durch die endgültige Beseitigung des Bildungsprivilegs an den Oberschulen und Universitäten.“ [Bit56, S.184]

Auch das Entstehen neuer Berufe im Umfeld der neuen Technologien sah er voraus: „Diese Kommunikation zwischen Mensch und Maschine wird heute nur erst von einer kleinen Zahl hochbefähigter Experten beherrscht. Wird es möglich sein, genügend Menschen die hohe Kunst der Programmierung zu lehren? Davon hängt die technische Seite der Automation ab.“ [Bit56, S.84/85] Picht formuliert ähnlich ein neues Verständnis von einem modernen Bildungssystem: „In der modernen ‚Leistungsgesellschaft‘ heißt soziale Gerechtigkeit nichts anderes als gerechte Verteilung der Bildungschancen; denn von den Bildungschancen hängen der soziale Aufstieg und die Verteilung des Einkommens ab.“ [Pic65, S.21] Das Bildungssystem

<sup>2</sup><http://www.spiegel.de/spiegel/print/d-46174255.html>, letzter Zugriff 11.6.2012

<sup>3</sup><http://www.spiegel.de/spiegel/print/d-41120955.html>, letzter Zugriff: 14.7.2012

hat nicht mehr die soziale Schichtstruktur zu stabilisieren, im Gegenteil, es muss den sozialen Aufstieg ermöglichen. Dies ist nicht nur ein politischer Anspruch, sondern wirtschaftliche Notwendigkeit: „Bildungsnotstand heißt wirtschaftlicher Notstand. Der bisherige wirtschaftliche Aufschwung wird ein rasches Ende nehmen, wenn uns die qualifizierten Nachwuchskräfte fehlen, ohne die im technischen Zeitalter kein Produktionssystem etwas leisten kann.“ [Pic65, S.9/10] In einer vergleichenden grafischen Darstellung stellt Picht den erwarteten Zuwachs der Abiturienten in der BRD von 1959 bis 1970 von 51400 auf 53300 dar, während beispielsweise in derselben Zeit Frankreich eine Steigerung der Abiturienten von 59100 auf 105000 oder Italien von 55600 auf 116000 erwartet. Die mittlere Reife erreichten beispielsweise 1960 in Deutschland je nach Bundesland 5 bis 24 Prozent aller Schülerinnen und Schüler, während beispielsweise in Norwegen 35,7% und in den Niederlanden 32,8% diesen Bildungsabschluss erreichten. [Pic65, S.18/23]

Mit dieser Absolventenzahl des Schulsystems sind die Probleme der deutschen Wirtschaft Mitte der sechziger Jahr vorprogrammiert: „Daß wir genötigt sein könnten, die vakanten Stellen von Ingenieuren, Naturwissenschaftlern oder Ärzten durch ausländische Kräfte zu besetzen, ist mehr als eine Sensationsmeldung; diese Möglichkeit wird heute von Sachkennern wie Professor Edding wissenschaftlich in aller Nüchternheit diskutiert.“ [Pic65, S.51/52] Diese Situation sieht er als so dramatisch an, dass er neben den strukturellen Reformen des bisherigen Bildungssystem eine enorme Erweiterung der bisherigen Institutionen anmahnt:

„Es bleibt deshalb kein anderer Weg, als das zu tun, was der Philologenverband seit jeher forderte, nämlich die höhere Schule ‚unangetastet‘ beiseite zu lassen, und die Probleme unseres Bildungswesens unter Umgehung der höheren Schule zu lösen. So behalten sowohl die Philologen als auch die Eltern, die eine solche Schule wünschen, jenes Gymnasium, das ihnen gefällt.

Aufgeben muß man allerdings das Dogma, die höhere Schule habe ein Monopol auf die Erteilung von Reifezeugnissen. [...] Besondere Sorgfalt muß darauf gewandt werden, daß sich der Anteil der Arbeiterkinder an den Abiturienten und späteren Studenten erhöht.“ [Pic65, S.54-55]

Damit unterscheidet sich Picht von den Ansätzen des späteren Bildungsgesamtplans, der zuerst eine Struktur- und dann eine inhaltliche Reform fordert und erwartet. Wie dramatisch Picht die Situation sieht, kann man im letzten Absatz lesen: „Jedes Volk hat das Bildungswesen, das es verdient. Noch ist es möglich, zu verhindern, daß die Bildungskatastrophe in ihrer vollen Gewalt über uns hereinbricht. Deutschland kann als Kulturstaat noch erhalten werden.“ [Pic65, S.68]

### 3.1.2 Bildung ist Bürgerrecht

Auch der Soziologe und liberale Politiker *Ralf Dahrendorf* fordert wie Picht einen enormen Ausbau des Bildungssystems. Allerdings empfindet er in seinem Plädoyer *Bildung ist Bürgerrecht* [Dah65] die Positionen von Picht als eine reaktive Bildungspolitik. Es ist ihm viel zu kurz gegriffen, ausschliesslich aufgrund des technischen Fortschritts und der demographischen Entwicklung mit steigenden Schülerzahlen und zu erwartender Pensionierungswelle im Schulbereich einen Anstieg von Abiturienten zu fordern. Er fordert eine *aktive Bildungspolitik*. Darunter versteht er, dass es neben den bekannten Bürgerrechten auch ein *Bürgerrecht auf*

*Bildung* gibt, das unabhängig von wirtschaftlichen und sozialen Zwängen ausgeübt werden darf und soll.

„Die Verfassungsartikel aus denen eine aktive Bildungspolitik sich entwickeln läßt, müßten lauten:

- (1) Jeder Mensch hat ein Recht auf eine intensive Grundausbildung, die ihn befähigt, von seinen staatsbürgerlichen Rechten und Pflichten wirksamen Gebrauch zu machen.
- (2) Jeder Mensch hat ein Recht auf eine seiner Leistungsfähigkeit entsprechende weiterführende Ausbildung.
- (3) Es ist die Pflicht der staatlichen Instanzen, dafür Sorge zu tragen, daß diese Rechte ausgeübt werden können.“ [Dah65, S.23]

Das mag aus heutiger Sicht fast unverständlich klingen, aber vor etwa 50 Jahren war es vor allem in ländlichen Gebieten noch üblich, beispielsweise Mädchen den Übergang zu einer höheren Schule mit dem Argument ‚die heiratet doch sowieso‘ zu erschweren. Natürlich hatte ein Sohn des Bäckers oder Landwirts den elterlichen Betrieb zu übernehmen. Wozu sollte dieser dann eine höhere Schule besuchen. Der Lebensweg war für die meisten Jugendlichen durch derartige soziale und religiöse Schranken vorgegeben oder zumindest eingegrenzt. [Dah65, 67/68/71]

Die Forderung und tatsächliche Umsetzung eines derartigen *Recht auf Bildung* war daher nicht nur eine bildungspolitische Forderung, sondern schlichtweg ein Fanal für eine offene, pluralistische Gesellschaft, in der ein jeder bei voller Chancengleichheit sein Leben frei planen und gestalten können soll. Im Rahmen dieses Rechts wird der Ausbau und Umstrukturierung des bisherigen Systems gefordert, da in der Vergangenheit vielen jungen Menschen gerade diese Möglichkeit verwehrt blieb.

Entsprechend widersetzt sich Dahrendorf auch dem Vorschlag Pichts, notfalls das Gymnasium in Ruhe zu lassen und mit einem Ausbau eines Schulsystem neben dem Gymnasium zu beginnen. [Dah65, S.36]

Dahrendorf übersieht jedoch, dass die Bildung zu allererst die Chancengleichheit dadurch realisiert, dass durch sie Chancen auf mögliche Berufe und damit den Lebensunterhalt geschaffen werden. Und mit dieser vorrangigsten Aufgabe von Bildung stellt sich die Frage der Inhalte. Ohne diese soziale Bindung kann das Recht auf Bildung zu einer Beibehaltung bisheriger Strukturen verkümmern. Zwar erkennt auch er die Wandlungsnotwendigkeit von Bildung, indem er erwartet: „[...] so wie heute die elementare *literacy*, das Lesen- und Schreibenkönnen, als Ausstattung des Staatsbürgers nicht mehr ausreicht, mag morgen schon die Kenntnis einer Fremdsprache, gewisser Elemente der Mathematik und der Grundtatsachen des sozialen, wirtschaftlichen und politischen Lebens — sämtlich noch längst nicht selbstverständlich — ein unzulänglicher Inhalt dieses Bürgerrechts sein.“ [Dah65, S.23] Auch erwartet er eine veränderte Qualifikationsstruktur der Beschäftigten: „[...] so steht heute zuweilen dort ein Fachschulingenieur, wo früher ein Facharbeiter und dort ein Facharbeiter, wo früher ein Ungelernter stand. [...] endlich wird dann auch Deutschland eine *new working class*, eine moderne Arbeiterschicht bekommen.“ [Dah65, S.62/80] Allerdings schreckt ihn die Vision eines *akademischen Proletariats* nicht, denn es „kann zum Beispiel die außerberufliche Welt dem Abiturienten, der unter dem von ihm erwarteten Statusniveau beschäftigt ist, kompensierende Befriedigung liefern.“ [Dah65, S.63] Mit dieser Position entspricht er dem liberalen

Gedankengebäude und entzieht letztendlich dem Bildungswesen die Pflicht, auch auf die gesellschaftlichen Notwendigkeiten der Arbeitsplatzentwicklung aus Sicht von Gesellschaft und zukünftigem Arbeitnehmer zu achten.

Die Begründungen für einen Ausbau des Bildungssystems waren also teilweise unterschiedlich, wenn auch nicht widersprechend. Trotzdem hatten und haben diese unterschiedlichen Ansätze Konsequenzen. Je nach schon bei Picht und Dahrendorf feststellbaren Standpunkten können inhaltliche oder strukturelle Fragen im Vordergrund der Aufmerksamkeit stehen. Das macht überdeutlich, dass erst mit einem genaueren Kriterienkatalog der Bildung und Allgemeinbildung überprüft werden kann, ob die Gesellschaft tatsächlich dem Recht auf Bildung in sozialer Verantwortung gerecht wird.

### 3.1.3 Der Bildungsgesamtplan

Der *Deutsche Bildungsrat* entwickelte in den nächsten Jahren eine völlige neue Konzeption der Bildungsstruktur und -Inhalte und veröffentlichte Anfang 1970 einen *Strukturplan für das Bildungswesen* [Deu70], der Bildung in der Bundesrepublik Deutschland tatsächlich anders als in den ersten Jahrzehnten der BRD definierte. So geht dieser Plan von der ‚Chancengleichheit‘ und einer ‚Individualisierung des Lernens‘ aus im Rahmen einer ‚Selbstentfaltung der Person‘, die zur einer ‚Mannigfaltigkeit im curricularen Angebot‘ führen wird. Der Bildungsrat sah daher die Notwendigkeit erheblicher Reformen in ‚Struktur und Curriculum‘. Er schreibt: „Ein besonders enger Zusammenhang besteht zwischen Strukturveränderungen und Curriculum-Veränderungen. [...] Daher genügt es nicht, einzelne Mängel der bestehenden Schulorganisation zu beheben. Vordergründig drängen sich die strukturellen und organisatorischen Veränderungen auf; auf die Dauer aber werden die curricularen Veränderungen bedeutender sein.“ [Deu70, S.27]

Es wurde Mitte des Jahres 1970 eine *Bund-Länder-Kommission für Bildungsplanung* eingesetzt, die schliesslich im Juni 1973 einen *Bildungsgesamtplan* den Regierungen vorgelegt hat. Dieser formuliert als Zielvorgabe:

„Ziel qualitativer und aller quantitativen Ausbaupläne ist die Entwicklung eines Bildungswesens, das unter Berücksichtigung der gesamtgesellschaftlichen Entwicklung den Anspruch des einzelnen auf Förderung und Entfaltung seiner Begabungen, Neigungen und Fähigkeiten erfüllt und in dadurch befähigt, sein persönliches, berufliches und soziales Leben selbstverantwortlich zu gestalten. Damit werden Chancengleichheit und Leistungsfähigkeit zu einander ergänzenden und bedingenden Prinzipien des künftigen Bildungswesens; die Reform der Curricula sowie der Differenzierung und Individualisierung des Bildungsangebots werden die Mittel sein, die Prinzipien zu verwirklichen.“ [Bun73, S.8]

Den Autoren ist bewusst, dass in der Zukunft kein Bildungssystem mehr so starr und verändert bleiben kann, wie das in der Vergangenheit war. So schreiben sie: „Abhängigkeit von Ausbildungssystem und Beschäftigungssystem macht eine stetige Korrektur der Prognosen erforderlich“ [Bun73, S.7]. Für den Sekundarbereich I wurde unter anderem gefordert:

„Sicherung einer allgemeinen wissenschaftsorientierten Grundbildung für alle.  
Vermeidung vorzeitiger Festlegung auf bestimmte Bildungsgänge.

Berücksichtigung der Neigung und der Befähigung des einzelnen durch eine zunehmende Wahl- und Leistungsdifferenzierung unter Beibehaltung eines verpflichtenden Kernbereichs gemeinsamer Inhalt.

Die genannten Ziele und Prinzipien bedingen die Organisationsform der integrierten Gesamtschule<sup>4</sup>.“ [Bun73, S.24]

### 3.1.4 Auswirkungen

Auch wenn in den darauffolgenden Jahrzehnten die Einmütigkeit bezüglich der Veränderungen der Bildungslandschaft zwischen den politischen Parteien in heftige Auseinandersetzungen und Gegensätze beispielsweise zur Schulform der *Gesamtschule* übergegangen ist, so hat dieser Gesamtplan in seiner Umsetzung die bildungspolitische Landschaft in der Bundesrepublik Deutschland wesentlich verändert. Dies wird allein schon die Zahl der Abiturienten deutlich, die im Jahr 2011 die Anzahl 360 000 (inklusive Fachhochschulreife ca. 500 000) erreicht hat.<sup>5</sup> Diese Steigerung um das Sechsfache in den letzten 50 Jahren ist eine positive Entwicklung. Die zweifelsohne erwirkten Veränderungen in Struktur und vor allem in der Zahl höherer Bildungsabschlüsse wurden im curricularen Bereich nicht so ergänzt wie es der Bildungsrat 1970 erwartet hatte. Hier hat die von Picht schon erahnte Diskussion um das Gymnasium und die Gesamtschule die inhaltlichen Fragen weitgehend verdrängt. *Hans Werner Heymann* schreibt schon fast resignierend: „Historische Erfahrungen sprechen dafür, daß eine über ‚Fächer‘ repräsentierte Ordnung des von der Schule zu vermittelnden Wissens zu den nur schwer veränderbaren Rahmenbedingungen schulischen Lernens gehört.“ [Hey96, S.29] Und damit sind sicher generell die Fächer in der Schule, ihre Rolle in der Schullaufbahn des einzelnen Schülers sowie auch teilweise die Inhalte in den Fächern gemeint. Der Ausbau des Schulsystems war dagegen aufgrund der Anforderungen der veränderten Bedingungen durch den technischen Fortschritt erfolgt und dieses hätte zumindest erfordert, dass nach dem strukturellen Umbau des Schulsystems durch das jetzt mehr mögliche Auswählen von Schulfächern nicht gerade die von den Schülerinnen und Schülern weitgehend abgewählt werden können, die für die Weiterentwicklung des technischen Fortschritts notwendig sind.

In einer vorläufigen Bilanz im Jahre 1986 stellt der Technikdidaktiker *Günther Schnuer* fest: „Heute drängen sich Abiturienten in das Medizinstudium, die auf dem Gymnasium keine Stunde Physik-Unterricht hatten, den Chemie-Unterricht nach einer Minimaldauer mit einem Grundkursus und der Note ‚sehr gut‘ abwählten [...].“ und ergänzt: „Das Abitur von heute hat seine Bedeutung für den Nachweis einer allgemeinen höheren Bildung weitgehend verloren. Es beinhaltet, was die naturwissenschaftlichen Fächer angeht, weniger an Substanz als das, was Abiturienten vor 20, 50 oder gar 70 Jahren an naturwissenschaftlichem Grundlagenwissen für ein Universitätsstudium mitbrachten. Aber auch die Fremdsprachenkenntnisse und sogar die der deutschen Muttersprache waren von diesem Verfallsprozeß nicht ausgeschlossen.“

---

<sup>4</sup>In einem ‚Besonderen Votum‘ der Länder BaWü, Bayern, Rheinland-Pfalz, Saarland und Schleswig-Holstein wird diese Forderung eingeschränkt, indem die aktuellen Schulversuche vor einer endgültigen Entscheidung abgewartet werden sollen.

<sup>5</sup>[http://www.focus.de/wissen/mensch/campus/bildung-kuerzeres-gymnasium-zehn-prozent-mehr-abiturienten\\_aid\\_719481.html](http://www.focus.de/wissen/mensch/campus/bildung-kuerzeres-gymnasium-zehn-prozent-mehr-abiturienten_aid_719481.html), letzter Zugriff: 13.6.2012

sen.“ [Sch86, S.63] Noch kritischer formuliert der Erziehungswissenschaftler *Hermann Giesecke*: „In den 60er und 70er Jahren war man euphorisch der Meinung, eine höchstmögliche Bildung für möglichst alle zahle sich wirtschaftlich aus, schaffe Innovationen und Arbeitsplätze. [...] Seitdem wurde die Abiturientenquote zu einer magischen Zahl. Aber schon damals zeigte sich, dass Massenarbeitslosigkeit nicht nur ungelernete Tätigkeiten, sondern auch akademische Berufe treffen kann, und die regierenden Sozialdemokraten hatten große Mühe, den Hochschulabsolventen zu erklären, dass aus dem Recht zum Studium nicht auch das auf einen angemessenen Arbeitsplatz abzuleiten sei. Tatsächlich gibt es keinen zuverlässigen Zusammenhang zwischen Nachfrage am Arbeitsmarkt und Qualifikationsangebot, [...]“ [Gie05] Es verwundert daher nicht, dass der von Picht schon 1964 erwartete Fachkräftemangel doch mit Verzögerung eingetreten ist.

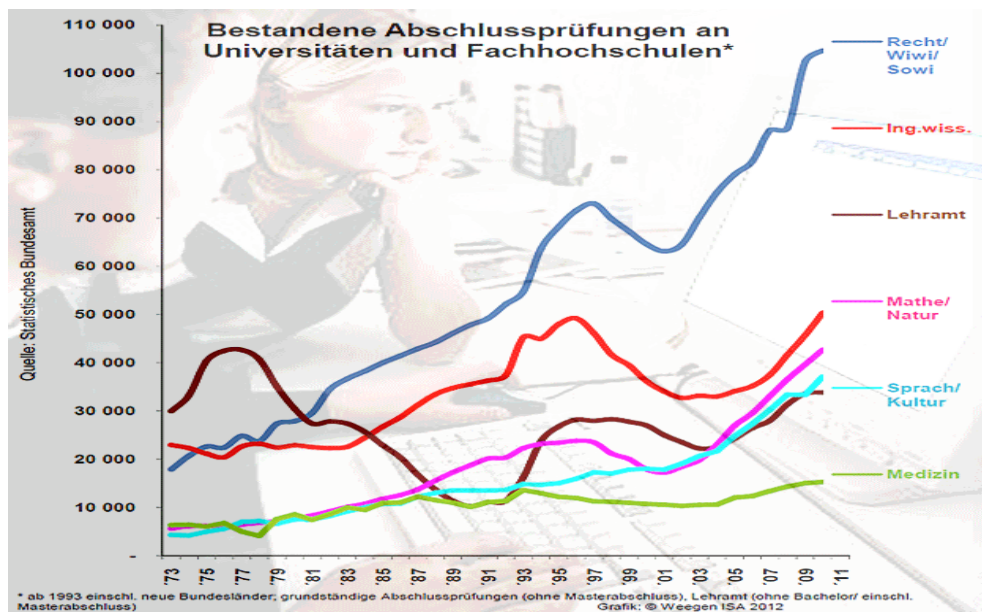


Abbildung 3.1: Bestandene Abschlussprüfungen an (Fach-)Hochschulen 1973 - 2011

Quelle: <http://www.uni-due.de/isa/absolventen.png>, letzter Zugriff: 13.6.2012

Die Abbildung [3.1] zeigt, dass die in den letzten Jahren erhöhte Zahl an Abiturienten nicht in die Studiengänge übergegangen sind, die für die Weiterentwicklung der Gesellschaft so dringend notwendig waren und sind. Die Bildungsreform hat in den letzten Jahren also nicht das Ziel erreicht, das sie angestrebt hat. Umso mehr stellt sich die Frage nach den Fächern und den Inhalten in diesen Fächern in der Schule, die offensichtlich den Erwartungen bisher nicht gerecht geworden sind.

### 3.2 Definition des Bildungsbegriffes

Es ist notwendig, festzulegen, was generell unter *Bildung* verstanden wird und welche daraus abgeleiteten *Bildungsinhalte* in Institutionen wie beispielsweise der Schule unterrichtet



werden sollen. Besonders in Zeiten politischer Veränderungen wird sehr intensiv um den Bildungsbegriff gerungen. Entsprechend setzte nach dem zweiten Weltkrieg eine heftige Diskussion um das Schulwesen ein. Einen grossen Einfluss in dieser wissenschaftlichen Diskussion hatte *Wolfgang Klafki*. Er untersucht 1958 in einem Vortrag *Bildung und Erziehung im Spannungsfeld von Vergangenheit, Gegenwart und Zukunft* [Kla58a] den Bildungsbegriff. Ausgangspunkt seiner Überlegungen ist, „daß der Mensch das Wesen sei, das Vergangenheit, Gegenwart und Zukunft habe“. Im Gegensatz zu etablierten Positionen besteht der Ausgangspunkt der Bildungsüberlegungen nicht nur aus der Vergangenheit.

### 3.2.1 Bildung aus der Vergangenheit, der Gegenwart oder für die Zukunft?

Nach Klafki muss Bildung aus den drei Zeitkategorien begründet werden. Die *Vergangenheit* wird repräsentiert durch den *pädagogischen Traditionalismus*, der einen erheblichen Einfluss auf die Bildung bis heute hatte und hat:

„Das abendländische Bildungswesen hat seinen Ausgang im christlichen Mittelalter von der Klerikerbildung genommen. [...] In diesen beiden Grundpfeilern [Lateinschulen und außertheologische Universitätsfakultäten] findet der pädagogische Traditionalismus eine fast reine Verwirklichung. [...] in der zweiten Hälfte des 19. Jahrhunderts setzte sich das traditionalistische Erziehungsdenken weitgehend erneut durch. Die Aufgabe der Bildungsarbeit — mindestens in den Gymnasien — schien es jetzt zu sein, die jungen Menschen durch die immer größer werdende Menge der wissenschaftlichen Einzelergebnisse hindurch, und zwar mittels der Auswahl des Wichtigsten, möglichst nah an den bereits erreichten Erkenntnisstand heranzuführen.“ [Kla58a, S.11,12].

Klafki verweist auf den Pädagogen und Philosophen *Friedrich Paulsen*, der die Meinung vertritt, „daß Erziehung und Unterricht der allgemeinen Wissenschafts- und Kulturentwicklung immer um etwa ein Menschenalter nachfolgen“. [Kla58a, S.12] Er stellt dann fest, „daß diese traditionalistische Erziehungsauffassung bis heute hin einen recht erheblichen Teil der Lehrerschaft, besonders der höheren Schulen, bestimmt.“ [Kla58a, S.12] Es ist nicht unwahrscheinlich, dass diese Position auch nach über 50 Jahren einer Überprüfung standhalten würde.

Die *Gegenwart* wird repräsentiert durch den *pädagogischen Aktualismus*. Die Grundannahme dieses Ansatzes lässt sich folgendermassen beschreiben: „Um in die Zukunft hineinzuwachsen, braucht das sich entwickelnde Wesen nur seinen inneren Antrieben zu gehorchen, die ohne Vorgriff und Verfrühung immer zur rechten Zeit den Übergang in die nächste Phase in Gang setzen.“ Diesen pädagogischen Aktualismus sieht Klafki in den so genannten *Trends*: „Der Erziehung aber macht man zur Auflage, sich solchen ‚Trends‘ ‚anzupassen‘“ [Kla58a, S.16] und kritisiert: „In Wahrheit macht man also pure Fakten der Gegenwart zum Maße und zur Norm der Zukunft.“ [Kla58a, S.17]

Die *Zukunft* wird repräsentiert durch den pädagogischen Utopismus, der beispielsweise in der inzwischen jüngeren Geschichte durch die *Pädagogik des dialektischen Materialismus* repräsentiert wird. Dieser geht aus von der zukünftigen „Lebenssituation der klassenlosen Gesellschaft“, deren Ziel die Befreiung des Einzelnen von unter anderem ökonomischen Zwängen erreicht haben will. „Um dieser zukünftigen Freiheit willen glaubt man sich dann als Po-

litiker und als Erzieher befugt, dem Kinde die Freiheit der Gegenwart und das heißt zugleich: die Erziehung zur wirklich innerlich freien Entscheidung zwischen verschiedenen Möglichkeiten nehmen zu dürfen.“ [Kla58a, S.19]

Alle drei isoliert betrachteten Herangehensweisen stellen nur Teilaspekte dar und können alleine keine Bildung im Interesse des Kindes erzeugen. Klafki folgert, „daß der gesuchte Ansatzpunkt nur in der Gegenwart der zu Erziehenden gefunden werden kann, einer Gegenwart allerdings, die nun nicht mehr statisch oder deterministisch gesehen werden darf, sondern die als zutiefst geistig-dynamisch betrachtet werden muß.“ [Kla58a, S.20,21]

### 3.2.2 Kategoriale Bildung

Aus dieser Beschreibung und seiner Kritik an bisher vorhandenen Theorien, die entweder die Inhalte über die Methoden oder umgekehrt die Methoden über die Inhalte gestellt haben, fordert Klafki eine Bildung, die beide Aspekte gleichermaßen berücksichtigt: die *kategoriale Bildung*. „Eine solche Bildungsauffassung aber [...] klingt, wenn wir recht sehen, in den fruchtbaren Ansätzen und Vorschlägen der neuen Didaktik des Exemplarischen, Typischen, Repräsentativen, Elementaren an.“ [Kla59] Er beruft sich dabei auf *Martin Wagenschein*, der für den naturwissenschaftlichen Unterricht entsprechende Konzepte fordert [Wag68, S.7ff]. Für einen solchen Unterricht gilt: „Wesentlicher ist die Einsicht, daß die Inhalte der Bildung gar nicht richtig verstanden werden können, ohne daß der Schüler den ‚Weg‘, der zu ihnen führt, mindestens in vereinfachter Form selbst geht. Inhalt und Methode sind unlöslich korrelativ aneinander gebunden.“ [Kla59, S.41] Konsequenz zu Ende gedacht bedeutet dies ein Überarbeiten und Kürzung aller damaligen Lehrpläne und Curricula:

„Alles, was nicht repräsentativ für grundlegende Sachverhalte und Probleme ist, sondern nur Einzelwissen oder Einzelkönnen, das nicht kategorial erschließend zu wirken vermag; alles was nur ‚auf Vorrat‘, auf eine Zukunft hin Bedeutung hat, die der junge Mensch nicht bereits in seiner Gegenwart als *seine* Zukunft zu erfahren vermag; alles, was nur tradierte Vergangenheit ist, ohne daß es dem Schüler im Bildungsvorgang selbst als *seine* Vergangenheit transparent würde; alles, was für den Schüler nicht auf *seine* Wirklichkeit bezogen ist, [...] alles das sollte in unserem Bildungswesen keinen Ort — jedenfalls keinen zentralen Ort — mehr haben.“ [Kla59, S.44/45]

Klafki fordert damit eine aufgrund von Kriterien überlegte Auswahl der Unterrichtsinhalte.

### 3.2.3 Die lerntheoretische Didaktik

Die geschilderten Prinzipien haben in verschiedener Ausprägung die bildungspolitische Diskussion vor allem in den 60-er und 70-er Jahren bestimmt. Dabei sind unterschiedlichste alternative Ansätze formuliert worden. Einige orientieren sich mehr am Bildungsbegriff, andere versuchen mehr, Bildung unter lernpsychologischen Gesichtspunkten zu betrachten. Die mit den Namen *Paul Heimann*, *Gunter Otto* und *Wolfgang Schulz* verknüpfte *lerntheoretische Didaktik* verlangt, dass der Lehrer in seiner Planung und Handlung sich an die ‚Vorgaben der Lernpsychologie‘ [Asc83, S.92] halten muss. So kann Lernen als *Wissen aneignen, sich etwas einprägen, Fertigkeiten erwerben, zu einem bestimmten Verhalten kommen* verstanden werden [Asc83, S.92/93]. Dazu gibt es im sogenannten *Berliner Didaktikmodell* verschiedene

Strukturmomente, unter denen sich auch inhaltliche ‚Entscheidungsfelder‘ befinden [Asc83, S.93ff], [Bla00, S.101].

### 3.2.4 Die curriculare Didaktik

Während das lerntheoretische Modell im Verhältnis zur bildungstheoretischen Modell nach Klafki den konkreten Lernvorgang mehr in den Vordergrund stellt und damit die *formale Bildung* [Kla59, S.27] im Gegensatz zur *materiellen Bildung* mehr betont, stellt die *Curriculare Didaktik* im Gegensatz dazu die Inhalte noch deutlicher als Klafki in den Vordergrund. In seinem Buch *Bildungsreform als Revision des Curriculums* [Rob67] fordert *Saul Benjamin Robinsohn* 1967, die Inhalte unabhängig ihrer geschichtlichen Bedeutung für den Schüler zu begründen. Er stellt in seinen Überlegungen zur Bildung den Begriff *Curriculum* in den Mittelpunkt. „Der Rückgriff auf diesen indessen aus der deutschen Pädagogik entschwundenen Begriff hat gute Gründe, kannte doch die Pädagogik des Barock noch die enge Verbindung der Bemühungen um Auswahl und Planung der Lerninhalte, [...].“ [Rob67, S.1] Robinsohn kritisiert die bisherigen Lehrpläne als unreflektiert und nicht gerechtfertigt:

„Glaubt man denn, gegenwärtige Lehrpläne reflektieren einen de facto bestehenden Konsens von Wertvorstellungen oder auch einen in ‚Freiheit‘ sich manifestierenden Pluralismus und nicht vielmehr die Ideologie einer vergangenen Epoche? Die Frage ist gar nicht, ob Schule Fortschritt initiieren kann, sondern, viel bescheidener, ob sie als fortschritthemende Institution fungieren muß; ferner, ob lediglich der in Parlamenten und Behörden artikulierte Wille Entwicklungen im Bildungswesen initiieren darf und es ausschließlich an ihm ist, die Rationalisierung bildungspolitischer Entscheidungen zu fördern oder zu unterlassen.“ [Rob67, S.8]

Im Gegensatz zu *Georg Picht* und *Ralf Dahrendorf*, die die konkreten Inhalte im Bildungskanon nur indirekt ansprechen, fordert Robinsohn ein generelles ‚Redesign‘ schulischer Bildung:

„Dabei geht es keineswegs um eine bloße ‚Verjüngungskur‘ für die Lehrpläne im Lichte fachwissenschaftlicher Entwicklungen — sie wird, wenn gleich mit unzulänglichen Mitteln, in einzelnen Fächern laufend versucht —, sondern um die Überprüfung der pädagogischen Relevanz des gesamten Gefüges in einem Prozeß, in dem gesellschaftliche Kräfte und wissenschaftliche Erkenntnisse mittelbar und unmittelbar bestimmend werden können.“ [Rob67, S.10]

Nach seiner Ansicht wird dies mit grosser Wahrscheinlichkeit die Fächerstruktur verändern: „Hat aber das neunzehnte Jahrhundert die Naturwissenschaften im Bildungsplan rezipiert, so hat das zwanzigste ähnliches für die Sozialwissenschaften noch nicht geleistet<sup>6</sup>., [Rob67, S.18] Das sieht nicht nur Robinsohn so. Auch der Pädagoge *Werner Loch* schreibt 1970 im Vorwort der deutschen Ausgabe von *Jerome Seymour Bruners* Buch *Der Prozeß der Erziehung*:

„In West und Ost, in den USA wie in der Sowjetunion, in England wie in Japan, in Frankreich wie in Schweden besteht das allgemeine Problem darin, daß der Bildungskanon aus einer Reihe von typischen, in der Struktur industrieller Gesellschaften nicht mehr einfach als Kumulation der kulturellen Überlieferung zu verstehen und anzuwenden ist, sondern fortwährend veränderten Bedürfnissen angepaßt werden muß.“ [Bru70, S.7]

<sup>6</sup>Informatik als Fachdisziplin gab es 1967 noch nicht.

Und er fügt hinzu: „Wenn der Abstand zwischen dem wissenschaftlichen Fortschritt und dem ‚natürlichen‘ Rückstand der Schulen zu groß wird, ist dieser ‚cultural lag‘ nur über eine Reorganisation der Lehrpläne aufzuholen.“ [Bru70, S.8]

Als wesentliches Kriterium für legitimierte Bildungsinhalte gibt Robinsohn an:

„Wir gehen also von den Annahmen aus, daß in der Erziehung Ausstattung zur Bewältigung von Lebenssituationen geleistet wird; daß diese Ausstattung geschieht, indem gewisse Qualifikationen und eine gewisse ‚Disponalität‘ durch die Aneignung von Kenntnissen, Einsichten, Haltungen und Fertigkeiten erworben werden; und daß eben die Curricula und — im engeren Sinne — ausgewählte Bildungsinhalte zur Vermittlung derartiger Qualifikationen bestimmt sind.“ [Rob67, S.45]

Der Bezug auf *Qualifikationen* ist später vielfach kritisiert worden. Dieser Begriff orientiert sich zu sehr an kurzfristigen und schnell verwertbaren Wissen vor allem für die Arbeitswelt und selbst die Verallgemeinerung auf *Schlüsselqualifikation* behebe nicht diese Problematik. [Hey96, S.56-59] Mit der genaueren Beschreibung durch *Aneignung von Kenntnissen, Einsichten, Haltungen und Fertigkeiten* wird deutlich, dass hier Qualifikationen viel umfassender gemeint sind.

Dementsprechend verlangt er auch für Bildungsinhalte die Erfüllung folgender Kriterien:

- (1) „die Bedeutung eines Gegenstandes im Gefüge der Wissenschaft [. . .];
- (2) die Bedeutung eines Gegenstandes für [. . .];
- (3) die Funktion eines Gegenstandes in spezifischen Verwendungssituationen des privaten und öffentlichen Lebens.“ [Rob67, S.47]

Durch die Betonung des aus dem Amerikanischen *rückimportieren*<sup>7</sup> Wortes *Curriculum* soll versucht werden, die immer wieder unreflektierte Übernahme aus dem alten tradierten Bildungskanon zu verhindern. Man bedenke, „daß es Robinsohn vorrangig um die Legitimation von Unterrichtgegenständen und die Auswahl von Unterrichtszielen und -inhalten ging“ [Asc83, S.103]. Zum Zeitpunkt der Veröffentlichung Ende der 60-Jahre und sicher auch später war diese Radikalität sehr provozierend. Dies ist verständlich, wenn man sich vor Augen hält, mit welcher Selbstverständlichkeit keine zehn Jahre zuvor der aus Vertretern des Bundes und der Länder zusammengesetzte *Deutsche Ausschuss für das Erziehungs- und Bildungswesen* in einem *Rahmenplan zur Umgestaltung und Vereinheitlichung des allgemeinen öffentlichen Schulwesens* ohne jegliche Begründung an überkommenen Traditionen festhält: „Die höhere Schule dient heute [. . .] einem doppelten Auftrag: einerseits muß sie den ständig wachsenden Bedarf der modernen Zivilisation an qualifizierten Nachwuchskräften decken; zum andern steht sie nach wie vor im Dienst ihres alten Bildungszieles: der Überlieferung der klassischen Gehalte der europäischen Kultur.“ [Deu59, S.193] Dafür werden zwei verschiedene Schulen verlangt, ein Gymnasium und eine *Studienschule*, „die in besonderem Maße zu den geschichtlichen Quellen unserer Kultur führen soll“ [Deu59, S.192]. Und diese besondere Schule können Schüler besuchen, die „mit hoher Wahrscheinlichkeit die Hochschulreife erreichen werden“. Diese Planungen sind zwar so nicht umgesetzt worden, die Diskussion macht aber deutlich; wie schwer damals die Forderung nach Begründung von Bildungsinhalten durchsetzbar war. Selbst in den Gewerkschaften wurden diese Inhalte für die Oberstufe

<sup>7</sup>Der Rückgriff auf diesen amerikanischen Begriff wird bewusst intensiv begründet. [Rob67, S.1]

der gymnasialen Oberstufe unkritisch gesehen. Im *Bremer Plan zur Neugestaltung des Deutschen Schulwesens* aus dem Jahre 1960 wird neben ersten vorsichtigen Forderungen nach einer (eher kooperativen) Gesamtschule für die Inhalte der Oberstufe gefordert: „Das Bildungsziel für beide Typen des Gymnasiums ist das gleiche: Die Überlieferung der klassischen Gehalte der abendländischen Kultur und die Einführung in die heutige Welt und die Zeitgeschichte. Beim altsprachlichen Zug liegt im Zusammenhang mit der Pflege der alten Sprachen der Schwerpunkt bei der Überlieferung.“<sup>8</sup> [Arb60, S.200]

### 3.2.5 Die lernzielorientierte Didaktik

Als Kritik an der normativen Setzung von Bildungsinhalten wurde eine sehr genaue Begründung von jeglichen Lernzielen gefordert. Als eine spezielle Form entwickelte sich aus der curricularen Didaktik die *lernzielorientierte Didaktik*. In ihr wurden *Taxomonien* für Lernziele aller Stufen formuliert, mit denen der konkrete Unterricht geplant und untersucht werden konnte [Asc83, S.107]. Der als Hilfe gedachte lernzielorientierte Unterricht unterlag und unterliegt aber immer der Gefahr, den Lehrer einem relativ hohen Rechtfertigungszwang zu unterwerfen. Zusätzlich bleibt als Gefahr: „Der lernzielorientierte Unterricht vernachlässigt alle Lernzielbereiche, die sich nicht oder nur unter Schwierigkeiten operationalisieren lassen.“ [Asc83, S.113] Und *Blankertz* fügt fast vernichtend hinzu: „Lernzielorientierung ist bisher fast ausschließlich unter der Voraussetzung behavioristischer Lernpsychologie durchdacht und entwickelt worden.“ [Bla00, S.159] Insofern haben Taxomonien für den konkreten Unterricht nur bedingte Bedeutung und Unterstützung erlangt. Sie ermöglichen zweifelsfrei eine Analyse von Unterricht beispielsweise zu Forschungszwecken. Die bedeutendste Taxonomie auf internationaler Ebene geht auf *Benjamin Bloom* [Blo56] zurück. Diese wurden Ende des Jahrhundert von einer Gruppe um *Lorin W. Anderson* und *David R. Krathwohl* [AKA<sup>+</sup>00] weiterentwickelt. Die in den Taxomonien beschriebenen Ebenen tauchen in abgewandelter Form in den *Kompetenz(stufen)* auf.

### 3.2.6 Didaktische Analyse

Entsprechend den bildungstheoretischen Ansprüchen ändern sich die Aufgaben des Lehrers erheblich und verlangt mehr Nachdenken über Stoff und Vermittlung als in der Vergangenheit. Er ist nicht mehr entsprechend der materiellen Bildungstheorien im Extremfall alleiniger Wissensvermittler, aber auch nicht mehr nur der Trainer in methodischen Techniken nach der formalen Bildungstheorie. Es gilt nun, „daß die methodische Besinnung nur der letzte, wenn auch unabdingbare und in gewisser Weise krönende Schritt einer rechten Vorbereitung sei. Der methodischen Vorbereitung wird immer wieder die Beschäftigung mit der im Unterricht zu vermittelnden bzw. zu erarbeitenden *Sache* als erster Vorbereitungsschritt gegenübergestellt.“ [Kla58b, S.127] Dies erfordert natürlich den Lehrer, der sich in seiner Aus- und Fortbildung so qualifiziert hat, dass die für den Unterricht ausgewählten Gegenstände und Methoden einer kritischen Prüfung Stand halten. Damit dies erreicht werden kann, bedarf es

---

<sup>8</sup>Trotzdem wurde dieser Plan von kirchlichen Kräften heftigst kritisiert: <http://www.spiegel.de/spiegel/print/d-43067068.html>, letzter Zugriff: 9.6.2012

Kriterien, nach denen Lehrplangestalter und Lehrer, aber auch Schüler und Öffentlichkeit die geforderten Inhalte und die zugeordneten methodischen Absichten beurteilen können

### 3.3 Der (Allgemein-)Bildungsbegriff nach Heymann

Nach der Reform der gymnasialen Oberstufe 1972 und den Einführung der Gesamtschule als alternative Schulform wurde etwa zwanzig Jahre weitgehend über die Struktur von Schule gestritten. Inhaltliche Fragen gerieten in den Hintergrund. Weiterentwicklungen der verschiedenen Bildungstheorien fanden kaum noch statt. Volker Huwendiek formuliert dies positiv: „Seit den 1980er-Jahren sind die Abschottungen und Alleinvertretungsansprüche der verschiedenen Entwürfe zurückgegangen. . . . Für die meisten Didaktiker . . . ist es selbstverständlich, dass die je spezifischen Ansätze nicht universale Geltung beanspruchen können, sondern zum Teil einander ergänzen und entsprechend kombiniert werden können.“ [BH06, S.64].

Die gesellschaftlichen Strukturen änderten sich aufgrund der technischen Weiterentwicklung, aber auch aufgrund der Folgen der Diskussionen in Folge der 60er Jahre. Über das, was *Bildung* und *Allgemeinbildung* in einer immer komplexer werdenden Welt wirklich ausmacht und wie sie eventuell teilweise neu begründet werden muss, wurde lange aufgrund der Strukturdiskussion wenig nachgedacht. Neue Inhalte und Fächer benötigen diese Reflektion, ansonsten erfährt das Bildungssystem eine zumindest inhaltliche Stagnation.

Erst durch die von Hans Werner Heymann 1996 in seiner Habilitationsschrift über den Mathematikunterricht formulierten Aussagen wurden weite Kreise der Mathematiklehrer und die Öffentlichkeit verunsichert. Seine Argumentation bezüglich Bildung und Allgemeinbildung wurden in der Öffentlichkeit ausschliesslich auf die Akzeptanz von Mathematikunterricht bis etwa Jahrgangsstufe 8 in der Schule reduziert. Doch damit wird man seinen Analysen nicht gerecht. Sie sind für die Schule — und damit natürlich auch für das Fach Informatik — insgesamt interessant und wichtig.

Heymann analysiert Gemeinsamkeiten und Unterschiede der Begriffe *Bildung* und *Allgemeinbildung* und versucht daraus die Aufgabe von *Schule als allgemeinbildende Institution* zu ermitteln und folgert schlussendlich als mathematischer Fachdidaktiker daraus ein Konzept für den Mathematik-Unterricht.

#### 3.3.1 Der Begriff *Bildung*

Zum Begriff *Bildung* formuliert Heymann zusammenfassend:

„Auf die Frage, was den Menschen zum Menschen macht, kann es in einer pluralistischen, demokratisch verfaßten Gesellschaft keine verbindliche, für alle Mitglieder dieser Gesellschaft gültige *inhaltliche* Antwort mehr geben. Bildung als Weg zur individuellen Lebensgestaltung kann sich an unterschiedlichen Bildungs- und Persönlichkeitsidealen orientieren, von unterschiedlichen Weltanschauungen, Philosophien, Religionen und den darin zum Ausdruck kommenden Menschenbildern ausgehen. Jedes Mitglied der Gesellschaft hat das Recht auf freie Entscheidung, auf welche persönlichen Ziele hin es sich verwirklichen will, an welche Traditionen es in seiner Lebensführung anknüpft und mit welchen es bricht.“ [Hey96, S.45/46]

Diese Definition von Bildung verbietet es geradezu, bestimmte Inhalte für einen einzelnen Menschen vorzuschreiben. Es bedeutet, dass das, was mich zum *gebildeten Menschen* macht, bestimme ich selbst. Und damit ändert sich die Funktion von Schule: Sie dient dem Einzelnen in der Entwicklung seiner Bildung und darf damit nicht mehr den Schülern eine fremdbestimmte Bildung angedeihen lassen.

### 3.3.2 Der Begriff *Allgemeinbildung*

Welche Aufgabe hat dann die Schule als *allgemeinbildende Institution* und woraus besteht diese *Allgemeinbildung* im Unterschied zur *Bildung*? Heymann formuliert:

„Allgemeinbildung ist so zu konzipieren, daß sie individuelle Bildung in großer Vielfalt möglich macht. Allgemeinbildung muß Raum lassen für eine Fülle unterschiedlicher, eventuell auch konkurrierender individueller Bildungsideale. Schulische *Allgemeinbildung* wird so zur *Bedingung der Möglichkeit von Bildung*: Allgemeinbildung ist für den Einzelnen Voraussetzung vernünftiger Selbstverwirklichung; sie eröffnet ihm Zugänge zu allem Besonderen, auf das er sich einlassen, für das er sich einsetzen sollte, um ganz Mensch zu sein.“ [Hey96, S.46]

Nur in der Einschränkung, dass dem Einzelnen Einsichten in bestimmte Teilbereiche möglicher Bildung gegeben werden, damit eine individuelle Auswahl nicht zufällig, sondern aus bekannten, möglichst vielen Alternativen erfolgt, ist ein Verpflichtung als Teil der Allgemeinbildung zulässig. Andersherum hat die Schule auch diese Vielfalt zu präsentieren und nicht selber durch eine Einschränkung ihrerseits die individuelle Bildungsentwicklung zu behindern. Dieser Anspruch an die allgemeinbildende Schule rechtfertigt sich auch aus der Erkenntnis, dass diese Breite an Möglichkeiten durch das persönliche Umfeld eines jungen Menschen bis auf Ausnahmen nicht gegeben ist. Teil dieser Allgemeinbildung muss dann auch sein, den Einzelnen zu befähigen, im Rahmen der Gesellschaft sein Einkommen zu erwirtschaften, um seinen Lebensunterhalt und sein Bildungsinteresse finanzieren zu können.

### 3.3.3 Aufgaben der allgemeinbildenden Schule

Ausgehend von dieser Beschreibung sieht Heymann folgende *Aufgaben der allgemeinbildenden Schule* [Hey96, S.47]: Lebensvorbereitung, Stiftung kultureller Kohärenz, Weltorientierung, Anleitung zum kritischen Vernunftgebrauch, Entfaltung von Verantwortungsbereitschaft, Einübung in Verständigung und Kooperation, und Stärkung des Schüler-Ichs.

Diese sieben Kriterien lassen sich weiter zusammenfassen in:

- Lebensvorbereitung
- Demokratischer Bürger
  - Stiftung kultureller Kohärenz;
  - Weltorientierung;
  - Anleitung zum kritischen Vernunftgebrauch;
  - Entfaltung von Verantwortungsbereitschaft;
  - Einübung in Verständigung und Kooperation;
- Individueller Bürger

- Stärkung des Schüler-Ichs.

Jedes Bildungsgegenstand und damit auch jede Fachdisziplin hat sich an diesen Kriterien zu überprüfen, ob er bzw. sie in der Schule eine Berechtigung hat oder nicht.

**Lebensvorbereitung** Die Lebensvorbereitung ist zweifelsohne eine der zentralen, wenn nicht sogar die wichtigste Forderung.

„Aus der Sicht der Schülerinnen und Schüler sowie ihrer Eltern, als den unmittelbar betroffenen ‚Abnehmern‘ von schulischem Unterricht, entspricht der Forderung nach Lebensvorbereitung der Anspruch auf die persönliche Verwertbarkeit des zu Lernenden, auf seine lebenspraktische Nützlichkeit. Aus der Sicht der Gesellschaft wiederum sichert die Schule dadurch, daß sie sich der Lebensvorbereitung annimmt, den Fortbestand der Gesellschaft: Strukturelles Korrelat der Lebensvorbereitungsphase ist die Reproduktionsfähigkeit der Schule.“ [Hey96, S.51]

Da die Entwicklung des Arbeitslebens von etwa 40 Jahren einer einzelnen Person zur Schulzeit kaum vorhergesagt werden kann, ist dies sicher keine einfache Aufgabe für die Schule. Die letzten 50 Jahre seit Pichs Analysen machen aber deutlich, dass diese Kernaufgabe der Diskussion um Struktur und Inhalt der Schule zielgerichteter als bisher geführt werden muss. Aufgabe von Schule gegenüber dem Einzelnen muss aber sein, ihm Möglichkeiten zu zeigen und zu erschliessen, an die der Einzelne zuvor vielleicht nicht gedacht hat.

**Demokratischer Bürger** Unumstritten in einer demokratischen Gesellschaft ist, dass im Rahmen der Allgemeinbildung eine *Weltorientierung* vermittelt werden muss. Was das bedeutet, mag ein Beispiel beleuchten: Jeder Schülerin und jeder Schüler hat das Recht, Physik nicht als besonders erstrebenswerten Inhalt ihrer bzw. seiner individuellen Bildung zu empfinden und anzustreben. Um in der Gesellschaft sachgerecht mitreden und mitentscheiden zu können, gehören Kenntnisse aus der Physik dazu, um beispielsweise die Katastrophe im Kernkraftwerk Fukushima in Japan beurteilen zu können. Dazu bedarf es Wissen über die Radioaktivität als Teil der Natur und seiner technischen Anwendung und Gefahren. Gerade, wenn dies nicht Teil einer selbst gewählten individuellen Bildung sein soll, muss dies Teil der schulischen Bildung sein.

Aufbauend auf dieser Weltorientierung dient die *Anleitung zum kritischen Vernunftgebrauch*. In einer demokratischen Gesellschaft hat die Schule die Aufgabe, einen Beitrag zur Entwicklung eines reflektierten Handelns jeder Person beizutragen. Daher ist die Erziehung zu einem mündigen, aber auch emanzipierten Bürger notwendig, der kraft seines Verstandes seine Interessen und Positionen innerhalb der Gesellschaft als Teil der Gesellschaft kund tun und gegebenenfalls umsetzen kann. Im Rahmen dieses reflektierten Handelns kann *Verantwortung* übernommen werden. Als Teil der Gesellschaft kann dieses nur in *Verständigung und Kooperation* mit anderen Mitmenschen erfolgen. Eine besondere und immer wichtigere Rolle spielt dabei die Verständigung zwischen Experten und Laien. Heymann verdeutlicht, dass das „Funktionieren unserer Gesellschaft [...] in zunehmenden Maße von der richtigen Handhabung hochspezialisierten Wissens und Könnens“ abhängt, „über das jeweils nur wenige Experten verfügen“. Daher sind „wir alle in den meisten Angelegenheiten, die uns berühren, Laien“ und es wäre gut, „wenn nicht so oft die Autorität des Experten auch da hervorgekehrt würde,



wo der Experte nur als Laie urteilt“. Für die Allgemeinbildung in der Schule ist aber wichtig: „Der Dialog zwischen Experten und Laien setzt, wenn er gelingen soll, *allgemeingebildete Experten* voraus, die über den Zaun ihres Spezialgebiets hinüberzuschauen vermögen, und *allgemeingebildete Laien*, die zumindest soviel von den Fragen verstehen, für deren Lösung sie Experten brauchen, daß sie ihnen nicht blind vertrauen müssen.“ [Hey96, S.113/114]

**Individueller Bürger** Die *Stärkung des Schülerichs* „zielt auf die Entwicklung von Selbstbewußtsein, Selbstvertrauen, personale Identität, auf die Fähigkeit, eigene Ziele, Wünsche und Vorstellungen klar zu erkennen und handelnd zu verwirklichen, mit den eigenen Stärken und auch Schwächen realistisch umzugehen.“ [Hey96, S.117] Ein solches starkes Schülerich entsteht nicht automatisch. Schule und die in ihr arbeitenden Menschen haben die Schüler als Persönlichkeiten ernst zu nehmen. Es müssen die Bedingungen geschaffen werden, die solche Persönlichkeit ermöglichen. Dazu müssen Bildungsinhalte präsentiert und erlebbar gemacht werden, bei und in denen der einzelne Schüler sich wiederfindet. Die Beschäftigung mit beispielsweise *klassischer Musik* ermöglicht es dem Schüler, entscheiden zu können, ob diese klassische Musik Teil seiner *personalen Bildung* werden soll. Dies ist gerade eine grosse Herausforderung und Pflicht der Schule gegenüber den sogenannten ‚bildungsfernen‘ Schichten. Schüler, die sich in dieser Hinsicht in der Schule *erleben* dürfen, können leichter auch nach der Schulzeit ihre Persönlichkeit entfalten.

**Allgemeinbildung in der Schule** Heymann fasst zusammen: Es „läßt sich Allgemeinbildung dem Einzelnen nicht aufzwingen. Allgemeinbildung ist eher als Angebot zu verstehen, von dem die Einzelperson Gebrauch machen kann oder nicht. [. . . ]

Die sieben Aufgaben der allgemeinbildenden Schule, wie sie hier entwickelt wurden, konkretisieren Allgemeinbildung als einen Qualitätsanspruch, an dem schulischer Fachunterricht und Vorgaben für Fachunterricht gemessen werden können.“ [Hey96, S.129/130] Diese Forderung bedeutet, dass Schule überprüfen muss, welche Inhalte und damit auch Fächer überhaupt angeboten werden. Und jedes Fach, und damit auch die Informatik, muss in einem permanenten Prozess die durch das Fach angestrebten Kompetenzen hinterfragen und gegebenenfalls anpassen.

## 3.4 Allgemeinbildung und Informatik

Mit den *sieben Aufgaben der allgemeinbildenden Schule* hat Heymann einen Kriterienkatalog formuliert, mit dem wesentlich deutlicher als zuvor bestimmt werden kann, ob ein Bildungsgegenstand in der allgemeinbildenden Schule gerechtfertigt ist oder nicht. Diesen Kriterien hat sich selbstverständlich auch das Schulfach Informatik zu unterwerfen.

### 3.4.1 Lebensvorbereitung

Der wesentlichste Teil der *Lebensvorbereitung* kann sehr gekürzt mit *Verwertbarkeit* umschrieben werden. Die Verwertbarkeit ist dann auf jeden Fall eingelöst, wenn die erlernten Kompetenzen in der Berufswelt genutzt werden können bzw. müssen.

Unabhängig davon, ob ein eigenständiges Schulfach Informatik gefordert wird oder nicht, wird die Verwertbarkeit informatischer Kenntnisse von kaum jemandem in Frage gestellt. Selbst die Gegner eines eigenständiges Faches fordern zumindest eine Grundbildung, die zu Anfang ‚Grundbildung Informatik‘ [BHW85] genannt wurde, später jedoch teilweise in ‚Informations- und kommunikationstechnologische Grundbildung‘ umbenannt wurde. In dem dazugehörigen Rahmenkonzept steht als Begründung: „Man muß davon ausgehen, daß nahezu alle bedeutsamen Bereiche menschlichen Lebens in der nahen und weiteren Zukunft zunehmend von den Auswirkungen der Neuen Informations- und Kommunikationstechnologien beeinflusst, verändert und geprägt werden.“ [NRW85, S.10] Im Folgenden werden diese ‚bedeutsamen‘ Bereiche etwas differenzierter dargestellt.

### 3.4.2 Berufswelt

In allen Bereichen unserer heutigen Berufswelt existieren Berufe, die sich direkt mit der Informatik beschäftigen. Nicht nur die Schülerin oder der Schüler, die bzw. der nach dem Abitur ihren bzw. seinen beruflichen Werdegang mit einem Studium beginnen will, sondern auch Schüler ohne Abitur oder ohne Fachhochschulreife können einen Beruf ergreifen, der direkt mit der Informatik zu tun hat.

**Studium** Ein Studium kann an einer Universität oder einer Fachhochschule absolviert werden. Damit stehen dem oder der Studierenden eine entweder mehr theoretisch-abstrakte oder praktische universitäre Auswahl zur Verfügung. Die Berufsaussichten sind in den nächsten Jahren mehr als zufriedenstellend. Die *Agentur für Arbeit* bestätigt dies und formuliert noch vorsichtig: „Im Bereich der hochqualifizierten IT-Spezialisten mehren sich sogar Anzeichen für zunehmende Schwierigkeiten von Unternehmen bei der Besetzung offener Stellen.“ [Bun12, S.84]

**Ausbildung** Es gibt derzeit mehrere Ausbildungsberufe mit den unterschiedlichsten Schwerpunkten wie Anwendungsprogrammierung oder Systemintegration. Es handelt sich hierbei um Berufe, die auch ohne Abitur angestrebt werden können. Die Aussichten für die Absolventen sind derzeit und in den nächsten Jahren sehr gut. Die BITKOM veröffentlichte auf Basis von Zahlen der Arbeitsagentur im September 2011 Zahlen zur aktuellen Situation auf dem Ausbildungsmarkt: „Die Zahl der neuen Auszubildenden in den IT-Berufen blieb 2010 mit 13.700 stabil. [...] Die Branche bietet damit mehr Ausbildungsstellen an als Bewerber registriert sind. ‚Wer sich heute für eine Ausbildung im IT-Sektor entscheidet, hat hervorragende Jobaussichten‘, sagte Kempf<sup>9</sup>.“

**‚Hilfswissenschaft‘** Kompetenzen in der Informatik und deren Anwendungen werden in vielfältigster Weise im Berufsleben benutzt. Es geht auf der einen Seite um die konkrete Benutzung von IT-Systemen als Werkzeug, aber auch um informatische Strukturen wie beispielsweise Netze, die bei der Anwendung solcher Systeme hilfreich sind. Darüber hinaus werden

---

<sup>9</sup>[http://www.bitkom.org/de/markt\\_statistik/64054\\_69467.aspx](http://www.bitkom.org/de/markt_statistik/64054_69467.aspx), letzter Aufruf: 18.6.2012

auch spezifische Kompetenzen benötigt, die über die Benutzung solcher Werkzeuge hinausgehen. Beispielsweise können im Rahmen von Automatisierungen Scriptsprachen verwendet werden. Die Beschränkung der Bedeutung des informatisches Wissens und der informatischen Kompetenzen hat zur Forderung nach der oben erwähnten *informations- und kommunikationstechnologischen Grundbildung (ITG)* geführt. Unabhängig vom schulischen Ort der Vermittlung solcher Kompetenzen gehören sie zweifelsohne zu den notwendigen schulischen (Schlüssel-)Qualifikationen und damit zur Lebensvorbereitung.

**Erkenntnistheorie** Im Kapitel 2 haben wir gesehen, wie weit informatisches Denken in anderen Disziplinen Einzug gehalten hat. Damit sind diese auch Teil einer möglichen Verwertung und damit Lebensvorbereitung für den Einzelnen. Diese Erkenntnisse gehen weit über den Bereich der ITG hinaus und sind damit unter anderem Basis und Grundlagen für ein Fach Informatik in der Schule.

### 3.4.3 Privat- und Freizeitwelt

Auch wenn viele Produkte, die einen informatischen ‚Kern‘ enthalten — digitale Artefakte [SD08] — auch ohne wesentliche informatische Kenntnisse nach kurzer Einweisung in Teilen verwendet werden können, sind die oben beschriebenen Kenntnisse, Fertigkeiten und Kompetenzen notwendig, diese Produkte umfassend zu nutzen. Beispielsweise ist es für die Bearbeitung einer Grafik oder Bildes nicht unwichtig, ob das ausgewählte Werkzeug zur Bearbeitung von Pixel- oder Vektorgrafiken konzipiert worden ist. Die selbstbestimmte Nutzung derartiger Produkte verlangt daher entsprechendes informatisches Denkens. Ein weiteres Beispiel stellt der gesamte Bereich der digitalen aufgezeichneten Musik dar. Aber auch der mediale und kommunikative Bereich ist nur mit informatischen Kompetenzen verantwortlich zu gestalten und bewusst zu nutzen. Beispielsweise sind Sicherheitsmassnahmen bei der Kommunikation im Internet wie das sichere Protokoll `https` nur mit derartigen Kenntnissen nachvollziehbar. „In Form vielfältiger digitaler Artefakte nimmt Informationstechnologie eine starke Bedeutung im Leben und im Lebensweg des Einzelnen ein. Das Erwerben und Messen informatischer Kompetenzen muss diese Vielfalt berücksichtigen. Im Sinne von IT-Schlüsselkompetenzen gilt es, transferierbare Kompetenzen zu vermitteln, die lebenslanges Lernen ermöglichen.“ [SD08]

### 3.4.4 Demokratischer Bürger

Die Notwendigkeit, informatische Kompetenz in der Schule zu vermitteln, um in der modernen Gesellschaft qualifiziert am demokratischen Dialog teilnehmen zu können, begleitet die didaktische Diskussion um das Schulfach Informatik von Beginn an. Die *Zielsetzungen und Inhalte des Informatikunterrichts* der GI aus dem Jahre 1976 geben bereits an: „Um die Zielsetzungen zu erreichen, soll der Schüler erlernen: [...] 4. Die Fähigkeit, die Auswirkungen der Datenverarbeitung auf die Gesellschaft zu erkennen.“ [Fac76] *Eva Pilz* und *Bärbel Tiesler* formulieren schon 1980, „[...] daß die wesentliche Begründung für Informatik in der Schule in Begriffen wie Information, Algorithmus, Rolle der Informatik in der Gesellschaft und Rolle der Informatik als konstruktive Wissenschaft liegt.“ [PT80, S.11] In seiner Replik

auf das Buch *Schwanitz: Bildung, Alles was man wissen muß* [Sch02] schreibt der Mathematiker, Physiker, Biologe und Professor für Wissenschaftsgeschichte *Ernst Peter Fischer* generell bezüglich der Naturwissenschaften:

„Wie viel Wissenschaft braucht also ein gebildeter Mensch. [...] Erstens: Er braucht soviel wissenschaftliches Wissen, um verstehen zu können, wie die Wissenschaft den Ort und das Bild des Menschen bestimmt. [...] Zweitens: Er sollte begreifen können, dass Wissenschaft in ihm steckt und zu ihm gehört. Nur aus dieser Verbindung können Teilnahme und Dialogbereitschaft entstehen, die nötig sind, damit alle die Verantwortung übernehmen können, die Wissenschaft heute benötigt. Humanistisch gebildete Menschen werden in dieser Antwort das griechische Modell erkennen, an dem sich der Begriff der Bildung orientiert. Im antiken Verständnis konnte sich nur der gebildete Bürger an der Demokratie beteiligen. [...] Mit dem Festhalten an Traditionen und der Bewahrung von vorhandenem Wissen ist es nicht getan.“ [Fis01, S.46/47]

Diese Aussagen generell über die Wissenschaft lassen sich direkt auf die Informatik übertragen.

Der breiten Öffentlichkeit wurde bei der *Volkszählung 1983* [Tae83] spätestens bewusst, welche Bedeutung Informatik und deren Produkte und Verfahren in der Gesellschaft besitzen. Die Auseinandersetzung mit dieser Volkszählung führten schlussendlich zur Festlegung des *Grundrechts auf informationelle Selbstbestimmung* durch das Bundesverfassungsgericht am 15. Dezember 1983. Kernaussage dieses Beschlusses ist: „Das Grundrecht gewährleistet insoweit die Befugnis des Einzelnen, grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten zu bestimmen.“ [TT89, S.10]. Dieses Recht kann nur wahrgenommen werden, wenn über die Möglichkeiten und Gefahren der Datenverarbeitung in der Gesellschaft ein entsprechendes Bewusstsein vorhanden ist. Der Informatikunterricht ist ein Ort in der Schule, an dem dies entwickelt werden kann und muss.

### 3.4.5 Stiftung kultureller Kohärenz

Die auf den ersten Blick kritischste Forderung von Heymann ist die *Stiftung kultureller Kohärenz*. Leicht lässt sich damit einfordern, dass in einer Gesellschaft die ‚Jungen‘, sprich die Schülerinnen und Schüler in die Vorstellungen eingeführt und eingefügt werden sollen, die von den ‚Älteren‘ vorgeprägt werden. Heymann meint aber dagegen:

„Die allgemeinbildende Schule hat mit dafür zu sorgen, daß — auch in einer Zeit der beschleunigten Entwicklung, schnell veraltenden Wissens und sich ändernder Wertvorstellungen —, eine *Verständigung zwischen den Generationen* möglich bleibt. [...]

Die allgemeinbildende Schule hat Voraussetzungen dafür zu schaffen, daß die Heranwachsenden eine *reflektierte kulturelle Identität* gewinnen können:

- ...
- daß sie Verbindendes innerhalb der eigenen Kultur erkennen können, jenseits ihrer Aufspaltung in disparate Teil- und Subkulturen;
- ....

Dies bedeutet, dass in einer Gesellschaft Alt und Jung, aber auch die Angehörige der verschiedenen Generationen untereinander einen Dialog führen können. Dazu gehört ein gemeinsamer

Begriffs- und Wissenskanon, ohne den eine Verständigung nicht möglich ist. Hierfür ist bei der jungen Generation die Schule in erster Linie verantwortlich. Da die Informatik eine relativ neue Disziplin ist, bedeutet dies, dass zumindest die jetzigen und zukünftigen Schülerinnen und Schüler qualifiziert werden müssen, miteinander unabhängig von ihrer zukünftigen Position in der Gesellschaft über informatische Probleme kommunizieren zu können.

### 3.4.6 Weltorientierung

Genau wie die Elektrizität im Verlauf der ersten Hälfte des 20. Jahrhunderts die Welt im Beruf, dann aber auch im Privat- und Freizeitleben verändert hat, hat die Informatik vor allem durch den heute allgegenwärtigen Computer das tägliche Leben in Beruf und Privatleben stark verändert. Aber es ist nicht nur die konkrete Anwendung, sondern die damit veränderten Denkstrukturen im Alltäglichen bis hin zum wissenschaftlichen Arbeiten, die es notwendig machen, Grundlagen dieses Denkens in der Schule zu vermitteln. *Wilhelm Steinmüller* spricht in diesem Zusammenhang von der *Transdisziplinarität* [Ste93, S.86] im Gegensatz zur Interdisziplinarität der Informatik. In vielen Bereichen der Technik stellt die Anwendung der Informatik nicht nur eine neue Quantität dar, sondern auch neue Qualität. Als Beispiele seien genannt: Anwendungen in der Kriegsführung, Überwachung und Kontrolle durch Erfassung und Analyse grosser Datenmengen, die sogenannte *digitale Spaltung*<sup>10</sup>. Derartige Problembereiche lassen sich nur sinnvoll erschliessen und einschätzen, wenn entsprechende informatische Grundlagen vorhanden sind.

### 3.4.7 Anleitung zum kritischen Vernunftgebrauch Entfaltung von Verantwortungsbereitschaft Einübung in Verständigung und Kooperation

Im vorhergehenden Absatz zur Weltorientierung ist gezeigt worden, wie bedeutsam Informatik und informatische Anwendungen und Produkte in der heutigen Welt geworden sind. Das zwingt natürlich, dazu, in der Schule auch genau zu diesen Techniken bei der *Anleitung zum kritischen Verbrauch* Stellung zu beziehen. *Helmut Witten* schreibt dazu: „Wenn man die Rede von den ‚Chancen und Risiken‘ nicht nur als leere Worthülse nimmt, müssen Lernende durch den Unterricht befähigt werden, Einsatzmöglichkeiten der Informationstechnik kritisch zu bewerten und vernünftige Kriterien für die Gestaltung von IT-Systemen zu entwickeln. Hierfür benötigen sie die Fähigkeit zum kritischen Vernunftgebrauch, verbunden mit der geistigen Haltung, den Dingen auf den Grund gehen zu wollen.“ [Wit03] Ein Teilaspekt des *informatischen Denkens* ermöglicht diesen kritischen ‚Vernunftgebrauch‘.

Entsprechend verhält es sich bei den weiteren Forderungen nach der Verantwortungsbereitschaft und der Verständigung und Kooperation. Anzumerken bleibt, dass gerade die Koopera-

---

<sup>10</sup>Ein Teil dieser digitalen Spaltung besteht darin, dass aufgrund mangelnder informatischer Kompetenz viele Mitbürger trotz vorhandener technischer Geräte viele Möglichkeiten nicht nutzen können: <http://www.welt.de/wirtschaft/webwelt/article6832954/Die-meisten-Deutschen-verstehen-das-Internet-nicht.html>, letzter Aufruf: 25.6.2012

tion und Verständigung heute sehr intensiv mit Anwendungen und Produkten der Informatik unterstützt werden kann und wird. Diese nutzen, reflektieren und beurteilen zu können, ist Teil einer Verantwortungsbereitschaft.

### 3.4.8 Individueller Bürger

Ein individueller Bürger mit einem gestärkten *Ich* ist in der Lage, seinen Lebensunterhalt selbst zu verdienen, sich in die Gesellschaft mit seinen beruflichen und persönlichen Kompetenzen einbringen zu können und für die Gesellschaft sinnvolle Arbeit zu leisten. Dazu weiss er mit seiner Freizeit für sich und seine Angehörigen bewusst umzugehen und kann sich auch dort entsprechend seiner Vorlieben, Bedürfnisse und Interessen entfalten.

**Stärkung des Schüler-Ichs** Die Schülerin bzw. der Schüler hat in der Schule in den verschiedensten Bereichen Anregungen zu erhalten und kann entscheiden, welche dieser Anregungen ihren bzw. seinen Vorstellungen entspricht. Aufgrund dieser Basis ist ein selbstbestimmtes Handeln in Richtung dieser ausgewählten Tätigkeit möglich. Beispiele für diese personale Bildung sind üblicherweise Musik, Kunst, Literatur. Aber auch die Informatik enthält einige interessante Beschäftigungsmöglichkeiten jenseits der Berufstätigkeit. Programmieren basiert auf der einen Seite auf einer wissenschaftlichen Theorie und verlangt zudem noch hohe handwerkliche intellektuelle Fertigkeiten und Fähigkeiten und kann im Rahmen von Freizeitbeschäftigung in verschiedenster Form zur Entfaltung der Persönlichkeit führen. Es ist allerdings eine Beschäftigung, zu der eine Schülerin bzw. Schüler nur in den seltensten Fällen aus eigenem Antrieb oder über Vorbilder aus der persönlichen Umgebung kommt. Hier hat die Schule möglichst für alle Schüler ein Angebot zu schaffen. Heymann schreibt dazu allerdings bezogen auf allgemeinbildende Qualifikationen: „[...] sie sollten in der Regel nicht beiläufig und ohne systematischen Unterricht von jedem Heranwachsenden erworben werden, etwa durch Vermittlung von Familienangehörigen oder Gleichaltrigen“. [Hey96, S.62] Was hier für allgemeinbildende Qualifikationen in der Schule gefordert wird, gilt entsprechend auch für die von der Schule zu fördernde *personale Bildung*.

### 3.4.9 Fazit Allgemeinbildung und Informatik

Die Aufgabe von Schule bezüglich der Allgemeinbildung befindet sich in einer dauerhaften Diskussion. In der Bundesrepublik wurde das Bildungssystem in und nach den 60er Jahren in erster Linie quantitativ, teilweise auch qualitativ umgestaltet. Ausgangspunkt dafür waren sowohl die wachsenden technischen Anforderungen an die Beschäftigten der Zukunft als auch ein aus den Grund- und Menschenrechten abgeleitetes Bürgerrecht auf Bildung. Von beiden Ansätzen aus ergab und ergibt sich die Verpflichtung, zu begründen, warum welche Bildungsinhalte gerechtfertigt sind. Unterschiedlichste Ansätze — teils mehr inhaltsorientiert, teils mehr lernorientiert — sind seitdem in der bildungspolitischen Debatte diskutiert worden. Heymann formulierte einen Kriterienkatalog, mit dem es möglich ist, zu überprüfen, ob ein Gegenstand oder ein Fach den Ansprüchen für eine Allgemeinbildung erfüllt. Wir haben diese

Kriterien auf die Informatik angewendet und festgestellt, dass das Schulfach Informatik diesen Forderungen gerecht wird.

### 3.5 Pisa und die Informatik

Im Jahre 2001 wurde die Öffentlichkeit durch empirische Untersuchungen seitens ‚*Programme for International Student Assessment (PISA)*‘ aufgeschreckt. PISA wurde 1997 seitens der OECD gestartet und ab dem Jahr 2000 wurden erstmalig Daten erhoben. Damit lagen zum ersten Mal vergleichbare empirische Daten aus vielen Ländern vor, die jenseits einer wie auch immer politisch geprägten Diskussion zur Analyse herangezogen werden konnten. In diesen Vergleichen schnitten die Schüler aus Deutschland unterdurchschnittlich ab. Die Ergebnisse sind zwar in den folgenden Untersuchungen bis 2009 besser geworden, allerdings verbleiben die Schüler aus Deutschland weitgehend im Mittelfeld der OECD-Staaten. Der Abstand zu den ‚Siegern‘ ist noch erheblich [KAH<sup>+</sup> 10a, KAH<sup>+</sup> 10b].

Als Begründung dieser Untersuchungen wurde angegeben:

„Hintergrund hierfür waren wirtschaftliche Interessen der aufstrebenden Industrienationen, mithilfe von Bildungsplanung die langfristige Verfügbarkeit qualifizierter Arbeitskräfte zu sichern. Neben wirtschaftlichen Interessen wurden schon früh auch gesellschaftspolitische Konsequenzen diskutiert, die insbesondere auf Bildungsgerechtigkeit abzielten; [...]“ [KAH<sup>+</sup> 10a, S.12]

Der Ansatzpunkt von PISA lässt sich daher problemlos mit der Bildungsreform in der BRD der 60er und 70er Jahre vergleichen. Auch dort ging es um die Verfügbarkeit von qualifizierten Arbeitskräften und auf der anderen Seite um das Bürgerrecht auf Bildung. Auch wenn es sich bei PISA um eine empirische Untersuchung handelt, bedarf es einer Begründung dessen, was denn in dieser Untersuchung als Basis dient.

„Als grundlegende Kompetenzbereiche betrachtet PISA die Lesekompetenz, die mathematische Kompetenz und die naturwissenschaftliche Kompetenz.“ [KAH<sup>+</sup> 10a, S.13] „Schon diese Kompetenzbeschreibungen machen deutlich, dass PISA eine breite, pragmatische Konzeption von allgemeiner Grundbildung aufgreift, wie sie in Deutschland seit Ende der 1960er Jahre vertreten wird (Deutscher Bildungsrat, 1974; Klieme & Hartig, 2007). Bildung ist ein Prozess, der jungen Menschen verschiedene Weltsichten erschließt und ihre Persönlichkeit entfaltet. Sich die eigene Lebenswelt mithilfe von Sprache, aber auch mit Methoden und Modellen aus den Naturwissenschaften und der Mathematik zu erschließen, ist Grundlage von Bildung. Die Kompetenzmessung in PISA erfasst nicht Bildung an sich, wohl aber diese unverzichtbaren Grundkomponenten.“ [KAH<sup>+</sup> 10a, S.13]

Zweifelsohne zählen diese drei Kompetenzbereiche zur Grundbildung hinzu. Es stellt sich zu Recht aber die Frage, warum andere Kompetenzen, beispielsweise einer Fremdsprache, Gesellschafts- und Sozialwissenschaften und eben auch Informatik, ausgeklammert werden. Mit der Beschränkung auf diese Kompetenzen wird auf der einen Seite von Anfang der Bildungsbegriff verengt sowie in der Öffentlichkeit ein unvollständiges Bild von Bildung aufgedrängt. Eine Auseinandersetzung um einen ‚vollständigen‘ Bildungsbegriff wird nicht geführt. Es verwundert nicht, wenn seit PISA mit (Allgemein-)bildung dann auch ausschliesslich diese drei Kompetenzen gemeint werden.

Die Kompetenzen werden genauer ausgeführt:

„Naturwissenschaftliche Kompetenz wird als die Fähigkeit verstanden, die charakteristischen Eigenschaften sowie die Bedeutung der Naturwissenschaften in unserer heutigen Welt zu verstehen, naturwissenschaftliches Wissen anzuwenden, um Fragestellungen zu erkennen, naturwissenschaftliche Phänomene zu beschreiben und aus Belegen Schlussfolgerungen zu ziehen, sowie die Bereitschaft, sich reflektierend mit naturwissenschaftlichen Ideen und Themen auseinanderzusetzen.“ [KAH<sup>+</sup> 10a, S.13]

Ersetzt man in diesem Satz das Wort *naturwissenschaftlich* durch *informatisch*, so erhält man eine Formulierung, mit der informatische Kompetenzen beschrieben werden können.

Der Ansatz von PISA, Bildungsprozesse quantitativ und vergleichbar messbar zu machen, um damit in den OECD-Staaten verlässliches Material zur Verfügung zu stellen, greift zu kurz, wenn als Grundlage ein unvollständiges Allgemeinbildungsmodell vorliegt und damit die scheinbare Objektivität in Subjektivität umschlägt<sup>11</sup>.

### 3.5.1 Bildungsstandards

Ergebnis der Diskussion sind auf der einen Seite formale Veränderungen: Verkürzung der Schulzeit bis zum Abitur und zentrale Prüfungen beim Abitur und zusätzlich neu (zumindest in NRW) zum Abschluss der Jahrgangsstufe 10. Andererseits wurden auf nationaler Ebene Bildungsstandards in den Fächern Deutsch, Mathematik, 1. Fremdsprache und den naturwissenschaftlichen Fächern erstellt. [KMK04a, KMK04b, KMK05a][. . . ] [KMK05c] Es sind dies die Fächer aus dem OECD-Kanon zuzüglich der Fremdsprache als sogenannte *Kernfächer*. Kritisch bleibt auch hier anzumerken, dass die Auswahl der Fächer nicht ausgehend von einem Bildungsbegriff begründet wird, sondern gesetzt wird.

„Bildungsstandards greifen allgemeine Bildungsziele auf. Sie legen fest, welche Kompetenzen die Kinder oder Jugendlichen bis zu einer bestimmten Jahrgangsstufe mindestens erworben haben sollen. Die Kompetenzen werden so konkret beschrieben, dass sie in Aufgabenstellungen umgesetzt und prinzipiell mit Hilfe von Testverfahren erfasst werden können.“ [Kli07, S.9] Diese Formulierung erinnert sehr stark an den lernzielorientierten Unterricht und die Operationalisierung, aber gerade von dieser Lernzielorientierung grenzt sich die nun angestrebte Kompetenzorientierung ab. Die praktizierte Definition von Lernzielen wurde schon von *Hans Aebli*, Schüler von Piaget, kritisiert als Definition von „*Prüfungsmöglichkeiten* für Lernziele“ [Aeb76, S.287]. Aebli fordert stattdessen für Lernziele: „Wir gehen davon aus, daß der Unterricht Instrumente der Lebensbewältigung zur Verfügung stellt: Vorstellungen, Begriffe, Operationen, Methoden, Strategien des Denkens und des praktischen Verhaltens.“ [Aeb76, S.293] Diese Definition ist nahe an der ursprünglichen Forderung von *Robinsohn*, der fordert, „[. . . ] daß in der Erziehung Ausstattung zur Bewältigung von Lebenssituationen geleistet wird; daß diese Ausstattung geschieht, indem gewisse Qualifikationen und eine gewisse ‚Disponalität‘ durch die Aneignung von Kenntnissen, Einsichten, Haltungen und Fertigkeiten erworben werden; und daß eben die Curricula und — im engeren Sinne — ausgewählte Bildungsinhalte zur Vermittlung derartiger Qualifikationen bestimmt sind.,“ [Rob67, S.45] Mit

<sup>11</sup>Zur Kritik an Pisa siehe auch *Jahnke/Meyerhofer: Pisa & Co: Kritik eines Programms* [JM07]



der Aufstellung der Bildungsstandards wird die curriculare Diskussion mit ähnlichen Begriffen wie in den 60er-Jahre wieder aufgenommen allerdings mit der Einschränkung, dass die Fachstruktur in der Schule nicht hinterfragt wird.

### 3.5.2 Der Kompetenzbegriff

Neu an der Definition ist, dass sie sich nicht an Begriffen wie *Instrumente der Lebensbewältigung* bzw. *Qualifikation* orientiert, sondern sich auf den Begriff *Kompetenz* bezieht und damit eine schärfere Akzentuierung vermuten lässt. Dieser Kompetenzbegriff fundiert in der Definition von *Franz Weinert*: Unter Kompetenzen versteht man

„die bei Individuen verfügbaren oder durch sie erlernbaren kognitiven Fähigkeiten und Fertigkeiten, um bestimmte Probleme zu lösen, sowie die damit verbundenen motivationalen, volitionalen und sozialen Bereitschaften und Fähigkeiten, um die Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können“. [Wei01, S.27/28].

*Gregor Wittke* hat in seiner Dissertation eine Reihe unterschiedlicher Kompetenzmodelle ausgehend von Chomskys Definition aus der Kommunikationswissenschaft vorgestellt. [Wit07] Insofern ist der Bezug auf den Begriff *Kompetenz* genauso mehrdeutig wie der Begriff (*Schlüssel-)**Qualifikation* nach Robinsohn oder *Instrumente der Lebensbewältigung* nach Aebli. Es bleibt zudem fraglich, ob es sinnvoll ist, bei einem Kompetenzbegriff auf eine Definition zu verweisen, die sich an der Kompetenz eines *Experten* orientiert. In der Schule ist es sicher nicht möglich, Schüler zu Experten welcher Ausrichtung auch immer auszubilden. Positiv ist bei allen Definitionen, dass schulisches Lernen sich am Nutzen im zukünftigen Leben rechtfertigen muss.

Die Betonung des Begriffes *Kompetenz* statt des Begriffes *Lernziel* rechtfertigt sich an der früheren Praxis vieler Schulen und Lehrer, im Vordergrund ihres Handelns das ‚Erledigen‘ konkreter Richtziele in ihnen vorgegebenen Lehrplänen zu sehen und weniger darauf zu achten, dass die Schüler für die Lebenswirklichkeit vorbereitet werden.

„In der Bildungspolitik und der Bildungsverwaltung haben TIMSS und PISA eine grundsätzliche Wende eingeleitet. Wurde unser Bildungssystem bislang ausschließlich durch den ‚Input‘ gesteuert, d.h. durch Haushaltspläne, Lehrpläne und Rahmenrichtlinien, Ausbildungsbestimmungen für Lehrpersonen, Prüfungsrichtlinien usw., so ist nun immer häufiger davon die Rede, die Bildungspolitik und die Schulentwicklung sollten sich am ‚Output‘ orientieren, d.h. an den Leistungen der Schule, vor allem an den Lernergebnissen der Schülerinnen und Schüler.“ [Kli07, S.11/12]

Die OECD formuliert das deutlich vorsichtiger:

„Die Ansätze der OECD-Staaten konzentrierten sich lange Zeit jedoch hauptsächlich auf die Beschreibung quantitativer Inputmerkmale, das heißt von Bildungsausgaben und Ressourcen. Die entsprechenden Ergebnisse aus den nationalen Bildungsstatistiken wurden und werden in dem seit 1992 jährlich erscheinenden Bericht *Bildung auf einen Blick* (OECD, 1992, 2010a) veröffentlicht. [...] Ende der 1990er Jahre war die OECD bestrebt, ihr Indikatorenprogramm durch Messungen von erworbenen Kompetenzen zu erweitern, um auch die Outcome-Dimension des Bildungssystems abzudecken. Aufbauend auf Erfahrungen der IEA wurde PISA etabliert.“ [KAH<sup>+</sup> 10a, S.12/13]

Unter *Input* versteht die OECD administrative Grössen wie Bildungsausgaben und Ressourcen und keine pädagogischen Grössen wie beispielsweise Lernziele. Erst die *Outcome*-Dimension hat eine pädagogische Bedeutung. Aus diesem Missverständnis heraus wird vielfach in Deutschland verstanden, dass es im Rahmen einer Kompetenzorientierung keine Lernziele geben kann. Das führt dazu, dass viele Lehrer das Wort *Lernziel* durch das Wort *Kompetenz* ersetzen. Damit ändert sich der Unterricht natürlich nicht im Geringsten. Es versteht sich von selbst, dass Kompetenzen als mittel- und langfristige Ziele nur erreicht werden können, wenn die Lehrkraft mit konkreten Zielen in jeder konkreten Unterrichtsstunde — eingebettet in einen längerfristigen Unterrichtsprozess — die Kompetenzen über einen längeren Zeitraum reifen lassen kann.

### 3.5.3 Die Bildungsstandards Informatik

In den sogenannten Kernfächern war mit dem Schritt zur Kompetenzorientierung endlich ein Weg beschritten worden, der seit den 60er Jahren von Bildungsreformern gefordert wurde. Diese hatten jedoch nicht nur eine didaktische Reflektion einiger Fächer angemahnt, sondern auch verlangt, dass alle Bildungsinhalte, damit auch alle Fächer mit ihren Inhalten, einer didaktischen Überprüfung und Rechtfertigung unterworfen werden. Indem nur ein Teil dieser Fächer als Kernfächer in Form von Bildungsstandards beschrieben werden, ist dieser wesentliche Teil einer Bildungsreform unterlaufen worden und eine notwendige Anpassung der gesamten schulischen Bildung an die aktuelle gesellschaftliche Situation und Struktur verhindert worden.

Die Festlegung, dass Bildungsstandards nur in wenigen Fächern, nicht aber Informatik, erstellt werden sollten, fand nicht die Zustimmung vieler Informatiklehrer. In ihr wurde eine Abwertung des Schulfaches Informatik gesehen. *Steffen Friedrich* [Fri03] und *Hermann Puhlmann* [Puh04] haben in Beiträgen aufbauend auf den PISA-Ideen Bezüge zum Schulfach Informatik hergestellt. Diese Diskussionen wurden auf den jährlich in Königstein (Sächsische Schweiz) stattfindenden informellen Treffen — *Fachdidaktische Gespräche zur Informatik*<sup>12</sup> — bestehend aus Lehrern und Hochschullehrern aus den deutschsprachigen Ländern aufgenommen und führten dann nach mehrjährigen Beratungen zu den *Bildungsstandards Informatik*, die von der GI im Januar 2008 als *Empfehlung der GI* zum Beschluss erhoben wurde<sup>13</sup>. Diese Bildungsstandards unterscheiden sich damit von den Bildungsstandards in den anderen Fächern, da sie von einer Fachgesellschaft und nicht von einer Kultusbürokratie beschlossen wurden und stellen damit eine Besonderheit in der Bildungslandschaft der Bundesrepublik Deutschland dar.

Die angestrebten Kompetenzen werden in diesen *Bildungsstandards Informatik* folgendermassen beschrieben:

„Die Vision ist, dass informatisch gebildete Menschen alle informatischen Probleme, die ihnen in ihrem Leben begegnen werden, mit Selbstvertrauen anpacken und selbstständig allein oder im Team bewältigen können. Und die Lehrenden helfen den Lernenden dabei, ihre Kompetenz zum Lösen solcher Probleme einzusetzen, zu vertiefen und auszubauen.“ [Arb08, 2]

<sup>12</sup><http://koenigstein.inf.tu-dresden.de>, letzter Zugriff: 27.6.2012

<sup>13</sup><http://www.gi.de/service/publikationen/empfehlungen.html>, letzter Zugriff: 27.6.2012

Eine tiefere Begründung erfolgt in:

„Die Informatik ist Grundlage der Informations- und Kommunikationstechniken, die als Schlüsseltechnologien unserer Epoche gelten. ‚Neben Schreiben, Lesen und Rechnen wird die Beherrschung grundlegender Methoden und Werkzeuge der Informatik zur vierten Kulturtechnik‘ (GI, 2006, S. 26). Wenn Deutschland wieder erstklassig werden soll, so formulierte es die Gesellschaft für Informatik in einem Memorandum (GI, 2004), dann dürfe niemand mehr ohne grundlegendes Verständnis moderner digitaler Hilfsmittel bleiben (vgl. auch GI/BITKOM, 2007).“ [Arb08, S.2]

Die Bildungsstandards sind im Gegensatz zu den Bildungsstandards der anderen Fächer nicht als *Regelstandard*, sondern als Mindeststandard, den alle Schüler erreichen sollen, formuliert. Diese ‚schärfere‘ Form der Standards wird auch begründet mit:

„Eine sich darauf gründende informatische Bildung gehört zur Allgemeinbildung, denn das Unterschreiten dieser Mindeststandards lässt erhebliche Schwierigkeiten beim Übergang ins Berufsleben und bei ihrer künftigen Position im gesellschaftlichen Leben erwarten.“ [Arb08, S.2]

Diese Formulierung erinnert an Aussagen, die schon in den Anfangsjahren des Schulfaches Informatik formuliert worden sind. In den Richtlinien des Faches Informatik im Bundesland NRW aus dem Jahre 1981 finden wir beispielsweise: „Die sich durch die Datenverarbeitung verändernde moderne Gesellschaft benötigt gerade in diesem Bereich Bürger, die auf der Basis fundierter Kenntnisse über Möglichkeiten und Grenzen des Einsatzes informationsverarbeitender System (vom Mikroprozessor bis zu Rechnernetzen) zu selbständigem Urteil und zu eigenverantwortlichem Handeln befähigt sind. So können die negativen Auswirkungen in Grenzen gehalten und die positiven Leistungen gefördert werden.“ [Der81a, S.27]

Damals ist eine solche Aussage fast ausnahmslos akzeptiert worden, inzwischen ist aber eine ganze Generation Schüler weitgehend ohne entsprechende Ausbildung durch Schule und teilweise Universität gegangen und diese Schüler stehen im Beruf. Scheinbar haben ihnen die Defizite bezüglich informatischer Bildung wenig geschadet. Allerdings haben viele sich schon daran gewöhnt und ‚umschiffen‘ individuell oder institutionell die eigentlich geforderten Kompetenzen. Beispielhaft wird dies deutlich an der ersten Frage im FAQ der Broschüre *Computeralgebrasysteme (CAS) im Mathematikunterricht des Gymnasiums, Jahrgangsstufe 10*, die sich an Mathematiklehrer an Gymnasien in Bayern richtet: „Müssen Lehrkräfte über vertiefte Computerkenntnisse verfügen oder gar eine Programmiersprache erlernen?“ [SL<sup>+</sup>11, S.10] Hätten diese Lehrer des affinen Fach Mathematik eine informatische Bildung erfahren, würde sich eine solche Frage erübrigen.

Dieses regelrechte ‚Verdrängungsszenario‘ macht damit aber gerade die Notwendigkeit einer solchen informatischen Bildung notwendig. Dies wird entsprechend in der Expertise zur Entwicklung nationaler Bildungsstandards gesagt, nur Konsequenzen jenseits einer ‚Medienbildung‘ werden dabei nicht gezogen: „Aber als generelle Prämisse für die Teilhabe an gesellschaftlicher Kommunikation reicht die Dimension der einfachen Kulturtechniken nicht mehr aus. Die Heranwachsenden müssen vielmehr fähig werden für den Gebrauch der Computer, für den Umgang mit Medien, [...]“. [Kli07, S.67]

Insofern war die Formulierung von Bildungsstandards Informatik ein richtiger und wichtiger Schritt. Diese können damit fortan als Grundlage bei der Entwicklung von Curricula in der Sekundarstufe I dienen.

### 3.6 Curriculare Konsequenzen

Aus den Überlegungen zur Bildungswissenschaft folgen curriculare Konsequenzen. Beispielsweise fordert die *Gesellschaft für Fachdidaktik*:

„Die GFD geht davon aus, dass das Ziel von Mindeststandards die Befähigung zur aktiven Beteiligung am beruflichen und öffentlichen Leben sowie zur Gestaltung des privaten Lebens sein muss. Zur persönlichen Entfaltung und Enkulturation der Jugendlichen ist ein breites Fächerspektrum unabdingbar. Eine Konzentration allein auf die Kulturtechniken, wie dies im Nationalen Ausbildungspakt geschieht, reicht dafür nicht aus. Die GFD formuliert daher im Rahmen der Fachlichkeit von Bildungsprozessen Mindeststandards aus einer Perspektive der Verantwortung der Schule als Ganzes und der Fächer gegenüber dem Einzelnen und der Gesellschaft. Schulunterricht muss explizit solche Kompetenzen, die alle Schülerinnen und Schüler als Mindestmaß für die eigene individuelle Entfaltung und die gesellschaftliche Partizipation sowie die Grundlage für lebenslanges Lernen erwerben müssen, über Schularten hinweg stärker in den Blick nehmen und sichern.“ [Ges09]

Es ist offensichtlich, dass aus der Bildungstheorie keine Fachdidaktik entsprechend der Fachsystematik gefolgert werden kann. Gehen wir von den Kriterien von Heymann aus, so folgt aus der *Lebensvorbereitung*, die in erster Linie die Voraussetzungen für die Aufnahme einer Berufstätigkeit schaffen soll, einerseits curricular die im Kapitel *Curriculare Konsequenzen* 2.8.4 beschriebenen Positionen, andererseits aber auch das Verständnis für den Aufbau und die Wirkungsweise von Infrastrukturen wie wir sie beim Internet vorfinden. Unabhängig von fachinternen Bedeutungen sind Kenntnisse von *Datenbanken* und *HTML* notwendig. Bisher wird völlig vernachlässigt, dass im Rahmen der *Stiftung kultureller Kohärenz* Mitglieder verschiedener Alters- und Gesellschaftsgruppen mit einander — auch mit Informatikern — über informatische Themen reden können. Dazu ist ein Mindestwissen über informatische Zusammenhänge in der Gesellschaft vonnöten.

Aus den Bildungszielen, die den *demokratischen Bürger* betreffen, kann gefolgert werden, dass Verständnis für die Sicherheit im Internet bezüglich Datenschutz und Datensicherheit vermittelt werden muss. Dies ist ohne konkretes technisches Wissen nicht möglich. Das Ziel des *individuellen Bürgers* kann vergleichbar dem praktischen Musizieren in der Anwendung auf der Basis des Verständnisses der grundlegenden Strukturen im Multimedia-Bereich wie Sound und Video und auch dem Programmieren als intellektuell fordernde Tätigkeit erfolgen. Die Struktur von Ton und Bild sowie deren analoge und digitale Repräsentanz gehört genauso in dieses Feld wie die Informationsübertragung in verschiedenen Infrastrukturen wie Computernetze, Rundfunk und Fernsehen. Hier sind als Fächer vor allem die Informatik und die Physik gefordert.

Auch die hier angeschnittenen Themengebiete sind auf unterschiedlichste Weise mit anderen Fächern verknüpft und müssen entsprechend unterrichtlich sinnvoll verteilt werden. Ein Teil der Gebiete überdeckt sich mit den curricularen Anforderungen aus der Fachwissenschaft,

ein Teil lässt sich mit fachwissenschaftlichen Forderungen kaum begründen. Es ist zu vermuten, dass aufgrund der Fülle der Themen eine sehr kleine Auswahl getroffen muss.

### 3.7 Zusammenfassung

Das wissenschaftliche Fach *Informatik* ist faktisch genauso alt wie die *Bildungsreform* der 60er Jahre. Die von *Picht* angemahnte *Bildungskatastrophe* hat darin eine nicht unwesentliche Ursache, besteht doch in der Anwendung der Erkenntnisse der Informatik bzw. ihrer Herkunftswissenschaften eine der Triebfedern der rasanten technologischen Entwicklung der letzten 50 Jahre. Die Folgen aus den Untersuchungen von *Picht*, *Dahrendorf* und auch *Robinsohn* waren ein immenser Ausbau des Bildungssystem, von dem man eigentlich erwartete, dass es sich auch inhaltlich erneuerte, wie *Loch* es vorhersagte: „Dem Abbau überholter Stoffe entspricht der Einbau neuer“ [Bru70, S.8] So wurde zwar *Informatik* als Schulfach 1972 zugelassen, mehr als eine sogenannte ‚Orchidee‘ im Fächerkanon stellt sie bis heute allerdings nicht da.

Eine fachliche inhaltliche Veränderung von Schule ist in den letzten 50 Jahren kaum umgesetzt worden. Die wesentlichen Änderungen waren organisatorischer Art. Auch die Diskussionen um didaktische Positionen der verschiedensten Richtungen haben die bisherige Fächerstruktur nicht verändert und damit die eigentlichen Auseinandersetzungen um die zentralen *Bildungsinhalte* nicht ausreichend geführt. Die von *Robinsohn* geforderte *Curriculumrevision* ist damit im Sande verlaufen. Eine neue Diskussion um den Bildungsbegriff ergab sich in den 90er Jahren im Anschluss an einen von *Heimann* formulierten *Bildungs- und Allgemeinbildungsbegriff*. Der *PISA-Schock* führte zwar zu einer ‚neuen Sicht‘ der Ziele von Unterricht, indem *Kompetenzen* und nicht einzelne *Lernziele* Perspektive von Schule sein sollen. Bei dieser Diskussion wurde die Fächerstruktur in der Schule als Grundfeste gesetzt und nicht weiter hinterfragt. Das ist verständlich, werden bei derartigen Untersuchungen empirischer Art vorhandene Bildungsergebnisse untersucht und nicht mögliche sinnvolle Alternativen angedacht, die erst in Jahren empirische Ergebnisse zeitigen können.

Bezieht man in diese Auseinandersetzungen um Bildung in der Schule die Informatik mit ein, so haben wir feststellen können, dass Informatik als Schulfach in vielfältigster Weise die Ansprüche erfüllt, die ein Bildungsgegenstand heute haben muss.



# 4 Lernpsychologie

## Überblick

Kann Informatik in der Schule unterrichtet werden? Und wenn ja, welche Inhalte auf welchem Niveau sind für Kinder und Jugendliche in den verschiedenen Altersstufen zu verstehen. Um diese Frage zu beantworten, muss geklärt werden, wie der Mensch lernt. Im 20. Jahrhundert sind auf diesem Gebiet sehr viele Erkenntnisse gewonnen worden. Im Grossen und Ganzen sind drei zeitliche und inhaltliche Abschnitte zu betrachten: Behaviorismus, Kognitivismus und Konstruktivismus. In jeder dieser Perioden wurde Aspekte des Lernens untersucht, die sich teilweise widersprechen, sich aber auch teilweise ergänzen. Wir möchten, das Schülerinnen und Schüler im Informatikunterricht vielfältige Kompetenzen für ihr zukünftiges Leben erwerben können. Dazu müssen wir wissen, wie ausgehend von ihrem anfänglichen Entwicklungsstand neue Informationen aufgenommen und verknüpft werden können. Ein Übertragen der Hochschulformatik auf die Schule würde sehr viele Schüler nicht erreichen. Informatiklehrer als Teil der ‚technischen Intelligenz‘ müssen sich bewusst werden, wie denn gerade die Schülerinnen und Schüler angesprochen werden können, die aufgrund ihrer Umgebung bisher nur wenig mit diesen Themen in Berührung gekommen sind.

Kinder lernen anders als Erwachsene. Dies gilt gerade für abstrakte Ideen und Gegenstände, die in der Informatik sehr vielfältig vorhanden sind. Eine Didaktik der Informatik muss sich dieser Psychologie des Kindes bewusst sein, um die Lerngegenstände in einem Curriculum so zu positionieren, dass im Verlaufe einer informatischen Lernbiographie die gewünschten Ziele erreicht werden. Piaget und Bruner haben in ihren Beiträgen grundlegende Positionen entwickelt, die berücksichtigt werden sollen. An dieser Stelle sei nur das Spiralprinzip erwähnt, das in der bisherigen informatischen fachdidaktischen Arbeit viel zu wenig berücksichtigt worden ist.

Das Schulfach Informatik hat die Chance, als relativ neues Fach die Fehler nicht zu wiederholen, die beispielsweise die naturwissenschaftlichen Fächer, die vor etwa 100 Jahren in die Schule integriert wurden, erlebt haben.

Sowohl aus fachwissenschaftlicher als auch aus bildungswissenschaftlicher Sicht rechtfertigt sich Informatik als Schulfach. Dann stellt sich die Frage nach dem *Wie und Was*. Mit *Was* ist die Auswahl der Gegenstände gemeint, die aus der Fachwissenschaft Informatik in der Schule schülergerecht aufbereitet unterrichtet werden sollen. Mit *Wie* ist gemeint, dass untersucht werden muss, auf welche Art und Weise, zu welcher Zeit und in welcher ‚Verpackung‘ diese Gegenstände im Unterricht bearbeitet und gelernt werden sollen. Denn es versteht sich von selbst, dass eine *Abbilddidaktik* der wissenschaftlichen Fachdisziplin Informatik nicht das ist, was den Anforderungen von Schule gerecht wird.

Um diese Fragen beantworten zu können, müssen wir uns vor Aufgaben führen, wie das Kind bzw. der Jugendliche in seinem Lernprozess unterstützt werden kann. Dies leitet uns zur Betrachtung der *Lernpsychologie*. Beim Lernen spielt die Psyche des Menschen eine nicht unwesentliche Rolle. Dabei müssen zwei unterschiedliche Bereiche betrachtet werden: Auf der einen Seite geht es generell um die Beziehung *Psyche und Lernen*, auf der anderen Seite um die Entwicklung des Kindes mit seiner besonderen *kindlichen Psyche*.

Gut gemeinte und durchdachte pädagogische Ansätze ‚verpuffen‘ daher regelrecht, wenn nicht die Erkenntnisse der Psychologie bezüglich des Lernens berücksichtigt werden. Pädagogik und Psychologie sind hierbei symbiotisch verknüpft.

## 4.1 Lerntheorien

### 4.1.1 Begriff des Lernens

Bevor man unterschiedliche Theorien über das Lernen aufstellen will, muss der Begriff *Lernen* definiert werden. Unter *Lernen* versteht man die *Veränderung des Verhaltens*. Dabei geht es beim Begriff nicht darum, ob das Lernen gezielt, bewusst oder zufällig passiert. Zu unterscheiden ist vom Lernen eine *Verhaltensänderung aufgrund von Reifung* [RA69, S.26], [Aeb69, S.174ff]. Schule ist zweifelsfrei ein Ort, an dem das Lernen mit Unterstützung der Lehrer gezielt praktiziert wird. Um das Lernen auf der einen Seite *effizient* durchzuführen und auf der anderen Seite für den Lernenden *zum optimalen Erfolg aus seinen Begabungen und Lernvoraussetzungen* führt, ist es erforderlich, die psychologischen Abläufe beim Lernen zu betrachten und entsprechende Schlussfolgerungen für den Unterricht daraus zu ziehen. Im Vordergrund der Betrachtung steht dabei immer der Lernende.

### 4.1.2 Behaviorismus

Zum *Behaviorismus* wollen wir auch die verwandten Ansätze der *Konditionierung* und *Instruktion* zählen [ZG04, S.167-209]. Bei allen Formen dieses Lernens geht es darum, dass der Lernende durch im weitesten Sinn Training, Schulung, Übung, Dressur und Drill sein Verhalten verändert. Das unterrichtliche Ziel dabei ist, am Ende dieses Prozesses eine gewünschte Verhaltenveränderung, eine *Lernleistung* wie beispielsweise das Beherrschen einiger Vokabeln in einer Fremdsprache zu erreichen. Ob dabei eine reine Konditionierung aufgrund eines *Reiz-Reaktions-Schema* nach *Iwan Petrowitsch Pawlow* oder eine *operante Konditionierung* nach *Burrhus Frederic Skinner* vorliegt, spielt dabei aus Sicht des Lernens vom Prinzip eine untergeordnete Rolle. Allein schon aus Gründen der Effizienz sind natürlich Lernergebnisse mit positiver Verstärkung im Sinne von Skinner vorzuziehen. Wie wenig dies allerdings Grundlage pädagogischen Handelns zumindest war, ist dem Verfasser als Schüler durch stundenlanges ‚Päckchen-Rechnen‘, durch Auswendiglernen ungebräuchlicher Wörter für ein geplantes Diktat oder ‚Vokabel-Pauken‘ noch in Erinnerung. Wie hätten es sich die Schüler zu der Zeit gewünscht, wenn zumindest Skinners Ergebnisse pädagogische Realität gewesen wären, stattdessen wurden sie mit ‚Weisheiten‘ wie ‚Es ist noch kein Meister vom Himmel gefallen‘ abgespeist.

Natürlich haben sich durch die verschiedensten Formen der Konditionierung Lernerfolge eingestellt. Ob die Lehrinhalte inhaltlich verstanden oder nur angewendet werden konnten, war zumindest für die frühere Volksschule fast nebensächlich, denn ihr Lehrauftrag bestand darin: „Die Volksschule hat ihre Schüler **auszurichten** und **auszurüsten**; ihr Erziehungsauftrag umfaßt **Haltung, Weltbild** und **Leistungsfähigkeit** der Schüler.“ [Hub44, S.60, (Fettdruck im Original)] Beispielsweise heisst es in den Erläuterungen zu den am 10.10.1945 erlassenen ‚Übergangsrichtlinien für die Bayrischen Volksschulen‘: „Die Volksschule muß wieder eine Stätte schlichter, geordneter Arbeit werden, in der religiöser Geist, echte Autorität, sinnvolles Leistungsstreben, Zusammenarbeit mit Elternhaus und Kirche, der Geist des Verstehens und der Freude die Jugend zu edlem Menschentum formt und die notwendigen Grundlagen für die Lebenstüchtigkeit erworben werden.“ [Kla71, S.174] Ein solches Menschenbild ist mit



Leitbild im Behaviorismus verträglich.

### 4.1.3 Kognitivismus

Der Hauptvertreter des Behaviorismus *Skinner* veröffentlichte 1957 das Buch '*Verbal Behavior*' [Ski57], in dem er eine Theorie aufstellte, dass auch das Lernen einer Sprache behavioristisch zu erklären sei. Mit diesem Ansatz stiess er auf die Kritik von *Chomsky*, der mit seiner *generativen Grammatik* Strukturen innerhalb verschiedenster Sprachen analysiert hat und den Ansatz vertrat, dass Sprachvermögen nicht durch Verhaltensanpassung, sondern durch Nutzung kognitiver Strukturen entwickelt wird. In der Antwort auf Skinner '*A Review of B. F. Skinner's Verbal Behavior*' schreibt *Chomsky* vernichtend: „*The questions to which Skinner has addressed his speculations are hopelessly premature.*“ [Cho67] Als Hauptargument in der Kritik wird ausgesagt: „Die reine Akkumulation von Reiz-Reaktions-Ketten kann niemals — und schon gar nicht in so kurzer Zeit — zur Ausbildung der komplexen Regelsysteme führen, über die die sprachliche Kompetenz des Kindes etwa am Ende des 6. Lebensjahres verfügt. [...] oder, um es aus der Sicht von *Chomsky* zu formulieren: Die Lerntheorie (auch die Theorie des Erfolgslernen) kann nicht erklären, warum das Kind mit einer beschränkten Menge grammatisch-syntaktischer Regeln unendlich viele Sätze bilden und verstehen kann, auch solche, die es vorher noch nicht gehört hat.“ [Vol97, S.215]

Ausgehend von dieser Auseinandersetzung erfolgte eine Umorientierung in der Analyse des Sprechens, Denkens und Lernens, die man im Nachhinein als *kognitive Wende* oder *kognitive Revolution* [Mil03] bezeichnet hat. Sie basiert auf den Erkenntnissen von *Jean Piaget*, *Jerome Seymour Bruner* und Anderen. Auch die Entwicklung der digitalen Computer mit ihrer Interpretation als *Elektronengehirn* und die darauf aufbauenden Untersuchungen von *John von Neumann*, *Marvin Minsky et al.* wie in Kapitel 2 beschrieben zur künstlichen Intelligenz haben das Bild des Denkens mit verändert. Die Bedeutung dieser Wende wird von *Jakob Muth* beschrieben mit: „Die Auffassung des Lernens, die der beschriebenen Situation zugrunde liegt, markiert eine ‚Wende im didaktischen Denken‘, eine ‚Revision des Unterrichts‘. Sie relativiert nämlich ein vom Beginn der Neuzeit an absolut geltendes Methodenprinzip. Es handelt sich um das Prinzip der Kleinschrittigkeit und der Kontinuität<sup>1</sup>“ [Mut67, S.89] Kritiker empfinden die Bezeichnung *kognitive Wende* für die Auseinandersetzung um Behaviorismus und Kognitivismus allerdings nicht für gerechtfertigt [DFN03].

Mit *Michael Charlton*, *Christoph Käßler* und *Helmut Wetzl* können wir folgende Aussagen als Beschreibung des Kognitivismus auffassen:

„Kognitiv erweiterte Lerntheorien sind sehr viel stärker als behavioristische Lerntheorien an den Fragen des Wissenserwerbs interessiert. Gleichzeitig wurde der Gedanke aufgegeben, dass durch Lernen ganz spezifische Außenreize mit spezifischen Reaktionen zusammengeslossen werden. Wenn man etwas gelernt hat, lassen sich mit Hilfe dieses Wissens unendlich viele, ganz verschiedene Aufgaben lösen. [...] Das Lernen erweist sich dabei nur noch zum teil als fremdgesteuerter Prozess. [...] Überwiegend steuern die Lernenden den Wissenserwerb jedoch selbst.“ [CKW03, S.77]

<sup>1</sup>Mit den beiden Begrifflichkeiten ‚Wende im didaktischen Denken‘ und ‚Revision des Unterrichts‘ bezieht Muth sich auf *Resag* und *Velthaus*.

Wenn sicher mit dem Aufkommen des Kognitivismus alle Aussagen des Behaviorismus nicht als widerlegt zu gelten haben, so ist der entscheidende Lernansatz im Kognitivismus nicht mehr das Training, die Übung, der Drill, sondern wie es *Aebli* schreibt:

„So ist die Zielvorstellung klar: die Schule vermittelt ein Wissen, das direkt oder indirekt der Lösung der Alltagsprobleme dient und dem jungen Menschen ein Bild der Wirklichkeit vermittelt; sie entwickelt Interessen und Werthaltungen, die ihm helfen, sein Verhalten zu ordnen und auszurichten. Handlungsschemata, Operationen und Begriffe und das Weltwissen, zu dem sie sich verknüpfen, haben genau diese Funktion, einenteils im kognitiven Bereich, anderenteils im Bereich der Interessen und Motive, sobald ihre Strukturen mit Wert belegt und intrinsisch interessant werden.“ [Aeb98, S.354]

„Didaktisches Nachdenken beginnt mit der Doppelüberlegung: Welches ist der nächste Schritt im Denken und Handeln des Kindes, und was hat die wissenschaftliche Erkenntnis bzw. die beste verfügbare Analyse der Lebenspraxis zu seiner Verwirklichung anzubieten.“ [Aeb98, S.388]

Entsprechend sind auch heutige politische und gesetzliche Ansätze und Vorschriften für schulische Aufgaben völlig anders formuliert als in der Restaurationsphase der Bundesrepublik:

So lesen wir beispielsweise im Schulgesetz in NRW im §4, Absatz 4:

„(4) Die Schule vermittelt die zur Erfüllung ihres Bildungs- und Erziehungsauftrags erforderlichen Kenntnisse, Fähigkeiten, Fertigkeiten und Werthaltungen und berücksichtigt dabei die individuellen Voraussetzungen der Schülerinnen und Schüler. Sie fördert die Entfaltung der Person, die Selbstständigkeit ihrer Entscheidungen und Handlungen und das Verantwortungsbewusstsein für das Gemeinwohl, die Natur und die Umwelt. Schülerinnen und Schüler werden befähigt, verantwortlich am sozialen, gesellschaftlichen, wirtschaftlichen, beruflichen, kulturellen und politischen Leben teilzunehmen und ihr eigenes Leben zu gestalten. Schülerinnen und Schüler werden in der Regel gemeinsam unterrichtet und erzogen (Koedukation).“ [Min12]

Die *kognitive Wende* erfolgte natürlich nicht im luftleeren Raum. Sie war der pädagogische und didaktische Widerhall auf die politischen Verhältnisse der 50er und 60er Jahre, so wie es in einer Darstellung des *Goethe-Institutes* (Dossier 1968) zum 40-jährigen ‚Jubiläum‘ von 1968 über *Hartmut von Hentig* formuliert wird: „1925 geboren erlebte der junge von Hentig das Dritte Reich. Seine wichtigste Lehre daraus: Kinder so zu erziehen, dass sie starke und freie Persönlichkeiten werden. Damit entsprach er den Idealen der 1968er und ging gleichzeitig weit darüber hinaus. Er kritisierte nicht nur das System, er machte sich und andere stark für Aufklärung und Selbstbestimmung.“<sup>2</sup> Mit dieser kritischen Einstellung zur Schule geprägt durch die sogenannten ‚68er‘ versuchten viele Junglehrer ein neues Denken, Handeln und Unterrichten in der Schule einzuführen und zu praktizieren.

---

<sup>2</sup><http://www.goethe.de/ges/pok/dos/dos/wdp/bil/de3160390.htm>, letzter Zugriff: 15.7.2012

#### 4.1.4 Konstruktivismus

Mit dem kognitiven Ansatz haben viele Lehrer ihren Unterricht durchgeführt und gehofft, dass ihre Schülerinnen und Schüler nun auch wirklich verstehen, was sie als Lehrer mit ihnen besprochen und bearbeitet haben. Sie wurden vielfach enttäuscht, weil noch viel zu viele Schüler den Stoff nicht so verstanden hatten wie sie es als moderne Lehrer nun eigentlich erwartet hatten. Abgesehen von eigenen Fehlern, Unzulänglichkeiten konnte und musste dies doch einen systematischen Grund haben.

Der systematische Fehler ist einsichtig: Der vom Lehrer angestrebte und vermittelte Verstandesprozess ist zumeist dem eigenen Verstandesprozess angelehnt. Damit erreicht man die Schülerinnen und Schüler, die in etwa eine vergleichbare Entwicklung wie der Lehrer mit in die Schule bringen. Für die anderen Schüler muss diese wahrhaft fremde kognitive Struktur oft unverstanden bleiben. Daraus folgt für die Lehrperson, dass dem Schüler Gelegenheit gegeben werden muss, den Lernprozess selbst konstruieren zu können.

Beispielhaft kann dies mit einem Gegenstand aus der Physik illustriert werden. Atomare Vorgänge sind für Schülerinnen und Schüler meist sehr abstrakt. Die Bedeutung von nur noch selten in der Schule durchführbaren Versuchen für physikalische Modellvorstellungen ist daher oft nicht einsichtig. Ein möglicher Weg ist der Versuch, die atomaren Vorgänge in eine makroskopische Welt zu übersetzen. Beispielhaft können die *Rutherford'schen Streuversuche* [GK<sup>+</sup>98, S.406] als Wurfversuche mit verschiedenen grossen Bällen durch Netze erklärt werden. Dies hat aber zur Voraussetzung, dass die Schülerinnen und Schüler Erfahrungen mit Bällen und dem Verhalten von fliegenden Bällen haben. Diese Erfahrungen haben sie normalerweise im Sport in der Schule und Verein und auch im Spiel auf Felder und Wiesen gemacht. Aufgrund des veränderten Freizeitverhaltens gibt es aber immer mehr junge Menschen, die über diese eigentlich spielerische Grunderfahrung nicht mehr verfügen. Damit scheitert der Erklärungsversuch der Rutherford'schen Streuversuche, da diese Schüler aufgrund des fehlendes Gefühles für Ballwürfe die physikalischen Erkenntnisse sich nur erschwert konstruieren können.

Der Psychologe *Aebli* formuliert folgendermassen: „Wir sind der Meinung, daß alle neuen Inhalte des geistigen Lebens durch Konstruktion aus einfacheren Elementen hervorgehen. ‚Von außen‘ nehmen wir nichts auf, weder durch Wahrnehmung noch durch Mitteilung. Das ist die Grundthese des Konstruktivismus.“ [Aeb98, S.389]

Auch wenn im pädagogischen Bereich Positionen erst in den letzten Jahren [Aeb98], [Rei06] immer mehr mit dem Begriff des Konstruktivismus begründet werden, sind derartige Gedanken nicht neu. Erfolgreiche Pädagogen haben im Prinzip immer entsprechend dieser Prinzipien gehandelt und gedacht. Unter anderen hat beispielsweise *Jürgen Henningsen* in einem kleinen Band *Atome, Algen, Automaten: Futurologie in der Schule* bereits 1968 formuliert: „Lernen ist nicht primär Hinzuaddieren, sondern Umstrukturierung: neue Information wird, wenn Erfahrung ermöglicht werden soll, eingefädelt und verknüpft mit dem jeweiligen immer schon mitgebrachten erworbenen Zusammenhang des Wissens, der sprachlich erschlossenen Erfahrung. [...] Es ist nicht sinnlos, eine ‚enzyklopädische‘ Systematik des Wissens anzustreben; diese enzyklopädische Systematik ist aber nicht die der Wissenschaft oder des Bücher insgesamt, sondern die des erworbenen Wissenszusammenhangs, sie befindet sich, salopp gesprochen, im Gehirn des Adressaten.“ [Hen68, S.23/24]

Umfassend stellt *Nicole Schobert* die Idee des Konstruktivismus als Schnittmenge unterschiedlicher Definitionen dar:

„Der aus Kybernetik, Biologie, Psychologie und Philosophie entstammende Hauptargumentationsstrang besagt, dass keine objektive Wirklichkeit bestehen kann, da sie nicht wahrnehmbar ist. Jeder Mensch verarbeitet seine Sinneswahrnehmungen auf der Grundlage seiner persönlicher Erfahrungen und seines Weltverständnisses. Daraus resultiert der Schluss, dass jeder Mensch seine eigene, einzigartige Wirklichkeit konstruiert und diese mit keiner Wahrnehmung eines anderen Individuums übereinstimmt. Die Erfahrung eines jeden Menschen bildet die Welt, in der er lebt“ [Sch11]

Dies Vorstellung sind nicht unbedeutend für den Aufbau eines Curriculum für das Fach Informatik. Die von der Schülerin oder dem Schüler in seinem Umfeld erfahrenen Aspekte der Informatik prägen sein Bild von der Informatik. In ihrem bzw. seinem Bewusstsein ist eine bestimmte Struktur konstruiert worden. Diese Strukturen sind im Gegensatz zu manchen ‚klassischen‘ Fächern sehr unterschiedlich, da es in unserer Gesellschaft bisher kein allgemeingültiges Bild der Informatik ausgehend von Familie, Kindergarten und Schule gibt. Wird auf diese unterschiedlichen Strukturen keine Rücksicht genommen, werden im informatischen Bildungsprozess die Schülerinnen und Schüler ‚verloren‘, die sich nicht in dem im Informatikunterricht vorgelebten Bild der Informatik wiederfinden können und damit sich anschliessen und dann weiterentwickeln können.

Die entsprechenden Folgen hat *Maria Knobelsdorf* in ihrer Dissertation untersucht und kommt zur folgenden Erkenntnis:

„Die untersuchte biographische Computernutzung der StudentInnen verdeutlicht, dass der IU dann eine prägende Wirkung auf das Interesse und die Lernbereitschaft der Personen hatte, wenn er Anknüpfungspunkte an ihre Vorgeschichte angeboten und dabei vor allem ihre Vorkenntnisse und Erwartungen an den Unterricht aufgegriffen hatte.“ [Kno11, S.227]

„Beschränkt sich der mit der biographischen Computernutzung einhergehende Lernprozess auf die Wahrnehmung des Computers als Arbeitsgerät und der ausschließlichen Nutzung von Anwendungsprogrammen, dann können die Lernenden die Unterrichtsthemen nicht mit ihren eigenen bisherigen Erlebnisse sinnhaft verknüpfen und sind dementsprechend wenig motiviert am IU aktiv teilzunehmen.“ [Kno11, S.219].

„Erwartungen an den IU rühren jedoch auch daher, dass in ihrem Weltbild in Bezug auf Informatik keine anderen Themen verankert sind. Die im IU vermittelten Kenntnisse und Kompetenzen werden damit nicht nur abgelehnt, weil sie den Erwartungen der Lernenden nicht entsprechen, sondern auch weil sie von den Lernenden nicht in eigene Bedeutungsstrukturen übertragen werden können. Hier wäre es entscheidend, fachdidaktische Brücken zwischen dem Weltbild der Lernenden und den tatsächlichen Unterrichtsinhalten zu entwickeln, die beispielsweise über ein einführendes Beispiel hinaus gehen.“ [Kno11, S.228].

Ähnliche Ergebnisse lassen sich beispielsweise bei Chemiestudenten nachweisen. Auch dort stellt sich die Frage, ob und wie der Chemieunterricht die Schüler erreicht und motiviert, sich auch nach der Schule mit der Chemie zu beschäftigen. *Jasmin Nadine Mnich* folgert in ihrer Masterarbeit aus Interviews mit Chemie- und Chemielehrerstudenten:

„Bei genauerer Betrachtung der Interviews fällt jedoch auf, dass alle Eltern, ausgenommen die der Fachwechsler und die der Nichtchemikerin, ein gewisses naturwissenschaftliches Interesse zeigen. [...]

So scheint das vom Elternhaus erweckte Interesse der Schülerinnen und Schüler sich kaum von den schulischen Erfahrungen beeinflussen zu lassen. Bedenklich ist jedoch, dass eben auch sehr interessierte Jugendliche den Unterricht in der Sekundarstufe I negativ wahrnehmen. So haben die Schülerinnen und Schüler, die von ihren Eltern kaum Interesse an naturwissenschaftlichen und speziell chemischen Inhalten erfahren haben, kaum eine Chance dieses Interesse überhaupt zu entdecken. [...]

Das familiäre und soziale Umfeld der Studierenden kann als ein entscheidender Faktor für den Werdegang festgehalten werden. [...] Biographische Parallelen in den Interviews und Lebensläufen der Befragten ergaben einen ersten Hinweis auf komplexe Zusammenhänge zwischen in Interessenbildung und Lebensbedingungen.“ [Mni07, S.81,84]

Diese Beobachtungen sind von den Unterrichtenden und auch von den Forschenden in der Curriculumentwicklung zu berücksichtigen, damit alle Schüler (in der Informatik) sich in ihrem Bildungsprozess wiederfinden können. Das erfordert eine umso höhere Aufmerksamkeit, weil Informatiker und Informatiklehrer nach diesen Erkenntnissen im Wesentlichen eine ähnliche Biografie bezüglich des Faches haben.

#### 4.1.5 Zusammenfassung Lerntheorien

Mit der Entwicklung der Psychologie sind im Verlaufe des letzten Jahrhunderts wesentliche Erkenntnisse bezüglich des Lernen gewonnen worden. Wir wissen heute, dass der Lernende sich seine ‚Wissenswelt‘ entsprechend seiner Biografie und Erfahrungen konstruieren muss. Da jeder Mensch seine eigene Biografie mitbringt, ist das Lernen ein sehr individueller Prozess. Das bedeutet nicht, dass Schule als Einzelunterricht stattfinden soll und muss, sondern, dass Lehrer darauf zu achten haben, über welche Strukturen die Schülerinnen und Schüler konkret verfügen, an die wir im Unterrichtsprozess anknüpfen können.

Oft werden die lerntheoretischen Ansätze als Widerspruch angesehen. Das sind sie nicht. Natürlich soll kognitiv vorgegangen werden. Der Lernende soll die Unterrichtsinhalte verstandesmäßig nachvollziehen können. Der Konstruktivismus hat den Kognitivismus nicht ‚erledigt‘, sondern er hat ihn konkretisiert, ihn vom allgemeinen Prinzip auf den Lernprozess beim einzelnen Schüler übertragen. Ob es daher gerechtfertigt ist, den vielen didaktischen Modellen noch eine *konstruktivistische Didaktik*, wie es Reich [Rei06] versucht, hinzuzufügen, mag allerdings dahingestellt sein.

Auch der Behaviorismus hat — korrekt und vorsichtig verstanden — seine Berechtigung und Aufgabe. Hat der Lernende einen Sachverhalt verstanden, ihn in sein Abbild der Welt hineinkonstruiert, so ist es oft vonnöten, bestimmte Routinetätigkeiten zu automatisieren, auf die in späteren Zeiten zurückgegriffen werden soll. Als Beispiel sei hier auf die Bruchrechnung in der Mathematik verwiesen. Da, wo Übungen notwendig sind, sollen sie auch weiterhin durchgeführt werden. Auch im Informatikunterricht sind wir gelegentlich gehalten, bestimmte Erkenntnisse zu üben und zu trainieren. Beispielsweise müssen die Erkenntnisse bei den Kontrollstrukturen durch Übungen gefestigt und auch variiert werden.

## 4.2 Psychologie des Kindes

Die im vorigen Kapitel angeschnittenen Probleme gelten für jeden Lernprozess einer beliebigen Person unabhängig vom Alter. In der Schule haben wir es überwiegend mit Kindern und Jugendlichen zu tun. Bei ihnen muss ergänzend berücksichtigt werden, dass aufgrund der psychologischen Entwicklung des Kindes für den Lernprozess unterschiedliche Bedingungen gelten. In dieser Arbeit liegt der Fokus auf den Unterricht in den Anfangsjahren der Sekundarstufe I. Kinder und Jugendliche in diesem Alter befinden sich in einer wichtigen Phase eines Schulkindes und noch nicht eines Erwachsenen. In dieser Phase gibt es für das Lernen Bedingungen, denen man sich als Unterrichtender bewusst sein muss.

Die frühe österreichische pädagogische Psychologin *Hildegard Hetzer* schreibt schon 1948 sehr treffend: „Das Kleinkind ist kein kleines Schulkind, das Schulkind keine Taschenausgabe des Erwachsenen. Aus dem Kleinkind wird auch kein Schulkind, wenn zu seiner Ausstattung etwas hinzukommt und man ihm dafür etwas anderes wegnimmt, wenn man das Kleinkind also mit Hut, Mantel oder sonst etwas bekleidet, und ihm sein Spielschürzchen auszieht.“ [Het48, S.10] Bei der Planung von Unterricht darf demnach nicht vergessen werden, dass in der Sekundarstufe I keine kleinen Erwachsenen von uns unterrichtet werden. Wir haben die (nicht immer rationale) kindliche und jugendliche Psyche zu berücksichtigen.

### 4.2.1 Die Psychologie der Intelligenz nach Piaget

Die entscheidenden und sehr umfassenden Grundlagen zur Psychologie der kindlichen Intelligenz hat *Jean Piaget* und sein Team vor allem in der ersten Hälfte des 20. Jahrhunderts erarbeitet. Folgende Strukturen sind nach Piaget dabei von Belang: Während der Entwicklung des Kindes findet eine permanente *Anpassung an die Umwelt (Adaption)* statt. Dabei werden *Schemata* als Grundbausteine verwendet und entwickelt. Diese Schemata können *Verhaltensschemata* oder *kognitive Schemata* sein. Neue Erfahrungen werden in die bisherigen Strukturen entweder durch *Assimilation* oder *Akkommodation* eingearbeitet. Assimilation ist die Einpassung neuer Erfahrung anhand oder durch Abwandlung bisheriger Schemata, Akkommodation ist der Aufbau neuer Schemata, wenn die bisherigen Schemata zur Einpassung nicht ausreichen. [Pia74, S.337ff], [ZG04, S.122-125]

Dabei verläuft die *Entwicklung der Intelligenz* des Kindes nach diesen Vorstellungen in vier Phasen:

- 1.) Sensomotorische Phase (0 bis 2 Jahre – Säuglingsalter) [Pia00, S.113ff]
- 2.) Präoperationale Phase (2 bis 7 Jahre – Kindergarten- und Vorschulalter) [Pia00, S.140ff]
- 3.) Phase der konkreten Operationen (7 bis 12 Jahre – (erweitertes) Grundschulalter) [Pia00, S.157ff]
- 4.) Phase der formalen Operationen (ab ca. 12 bis 15 Jahre – Jugendalter) [Pia00, S.167ff]

Kinder und Jugendliche befinden sich zu Beginn der Sekundarstufe I in der *Phase der konkreten Operationen*, in den mittleren Jahren wechseln sie dann in die *Phase der formalen Operationen*. Mit diesen konkreten Operationen sind *logisch-arithmetische und räumlich-zeitliche Operationen* gemeint. „Es ist aber wichtig zu betonen, daß diese verschiedenen, logisch-arithmetischen und räumlich-zeitlichen Gruppierungen noch weit davon entfernt sind, eine formale Logik zu bilden, die auf alle Begriffe und Schlüsse anwendbar wäre.“ [Pia00, S.164]

„Die Operationen, um die es sich hier handelt, sind also noch ‚konkret‘ und nicht ‚formal‘: immer mit der Handlung verbunden, geben sie dieser eine logische Struktur, in der die sie begleitenden sprachlichen Ausdrücke miteinbezogen sind, was aber noch nicht die Fähigkeit einschließt, einen von der Handlung unabhängigen Schluß zu entwickeln.“ [Pia00, S.165] Haben die Jugendlichen die *Phase der formalen Operationen* erreicht, sind sie in der Lage „hypothetisch-deduktiv zu denken“ [Pia00, S.167]. „Das formale Denken besteht im Gegensatz dazu [zum konkret-operationalen, der Verf.] in Reflexionen über diese Operationen (im eigentlichen Sinn des Worts), also im Operieren mit Operationen oder ihren Ergebnissen, [. . .]“ [Pia00, S.168].

Die von Piaget und seinen Mitarbeitern angegebenen Altersstufen sind allerdings mit grosser Vorsicht zu betrachten. *Manfred Bormann* und *Hermann Klinger* schreiben 1978: „Amerikanische Studien zeigen, daß bei den meisten Schülern des Sekundarbereichs in den USA formal-operationale kognitive Strukturen nur unzulänglich ausgebildet werden und daß darüberhinaus viele Schüler, die in ‚Piaget-Tests‘ die Fähigkeit zu formalen Denken zeigen, im naturwissenschaftlichen Unterricht auf ein konkret-operationales Niveau zurückfallen.“ [BK78, S.55] In den von ihnen durchgeführten Experimenten stellte sich heraus, dass bis auf Ausnahmen kein Schüler der Sekundarstufe I sowohl im Gymnasium als auch an der Gesamtschule dieses Niveau bezüglich physikalischen Denkens erreicht hatte und sich auch nur eine Minderheit der Schüler im Übergang von der konkret-operationalen Phase zur formal-operationalen Phase befand [BK78, S.61], [Mer08, S.85]. Bormann und Klinger stellen daraufhin fest: „Es ist vielmehr so, daß viele Schüler der gesamten Sekundarstufe ihre intellektuelle Anbindung an die konkret-operationale Denkweise nicht überwinden und formal-operationales Denken nicht entwickeln oder nur unzureichend anwenden können.“ [BK78, S.62]

Der Übergang zwischen den Phasen ist also individuell sehr unterschiedlich. Entsprechend muss curriculare Entwicklung, aber auch das konkrete Lehrerhandeln in den (ersten) Jahren der Sekundarstufe I immer bedacht sein, an diese beiden Phasen anzuknüpfen. Die heute allseits anerkannte Stufenentwicklung der Intelligenz führte zur Definition der *operativen Didaktik* durch den Piaget-Schüler *Hans Aebli* [Aeb76, Aeb98]. Dabei geht Aebli davon aus, dass Unterricht über *Handlungsschema, Operation und Begriff* als *genetischem Gesetz* folgen muss [Aeb98, S.179].

## 4.2.2 Die Erziehung nach Bruner

Aufbauend, aber auch teilweise unabhängig von diesen Entwicklungen hat *Jerome Seymour Bruner* wesentlich in den 60er Jahren seine Positionen zur Erziehung und zum Unterricht aufgebaut. Bruner wird allerdings heute zu sehr auf das *Spiralprinzip* reduziert. Dieses Prinzip reiht sich ein in umfassendere Vorstellungen von Unterricht und Erziehung.

Bruner geht vom Konstruktivismus aus und folgert daher: „Die Struktur eines Themas begreifen heißt, es so zu verstehen, daß viele andere Dinge dazu in eine sinnvolle Beziehung gesetzt werden können.“ [Bru70, S.22]. Diese Grundstruktur zu begreifen macht Sinn, denn „Vielleicht das Grundlegende, was man nach einem Jahrhundert intensiver Forschung über das menschliche Gedächtnis sagen kann, ist, daß Einzelheiten schnell wieder vergessen werden, wenn sie nicht in eine strukturierte Form gebracht worden sind.“ [Bru70, S.36] Daher gilt entsprechend: „Kenntnisse, die man erworben hat, ohne daß eine Struktur sie genügend verbind-

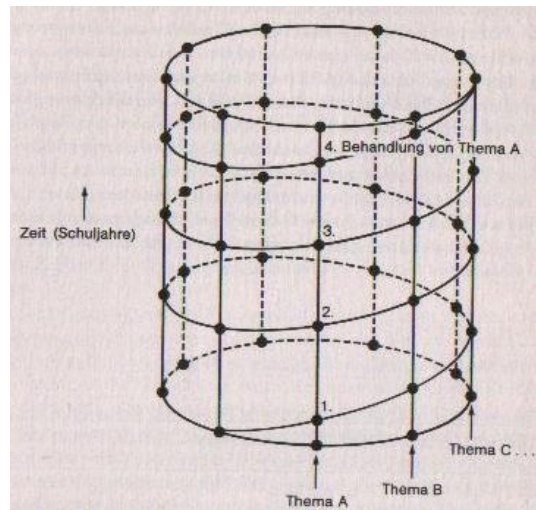


Abbildung 4.1: Das Spiralprinzip von Bruner in der Darstellung von Aebli [Aeb76, S.311]

det, sind Wissen, daß man wahrscheinlich bald wieder vergißt.“ [Bru70, S.43] Es kommt also darauf an, diese grundlegenden Begriffe herauszuarbeiten und diese dem Schüler Stück für Stück erschliessen zu helfen: „Um das Kind grundlegende Begriffe zu lehren ist es vor allem wichtig, daß man ihm hilft, vom Denken im Bereich des Konkreten allmählich zu Denkweisen überzugehen, die dem Bereich des Begrifflichen mehr angemessen sind.“ [Bru70, S.49] Daraus ergibt sich dann folgerichtig die „[...] Hypothese: Jedes Kind kann auf jeder Entwicklungsstufe jeder Lehrgegenstand in einer intellektuell ehrlichen Form erfolgreich gelehrt werden.“ [Bru70, S.44] Dieses Prinzip wird als *Curriculum-Spirale* [Bru70, S.61] bezeichnet und kann entsprechend der Abbildung 4.1 von Aebli dargestellt werden.

Bruner zieht aus diesen Erkenntnissen Folgerungen für die mathematisch-naturwissenschaftlichen Fächer: „Betrachtet man das Verständnis von Zahl, Maß und Wahrscheinlichkeit als unumgänglich für die Beschäftigung mit exakter Wissenschaft, dann sollte die Unterweisung in diesen Gegenständen so geistig-aufgeschlossen und so früh wie möglich beginnen, und zwar in einer Weise, die den Denkformen des Kindes entspricht.“ [Bru70, S.63] Auf die Informatik übertragen bedeutet dies: Auch die zentralen Ideen der Informatik können und sollten den Kindern und Jugendlichen schon so früh wie möglich unterrichtet werden.

### 4.2.3 Die Lernstufen nach van Hiele

Piaget hat deutlich gemacht, dass wir beim Unterrichten die geistige Entwicklung der kindlichen Psyche und Intelligenz zu berücksichtigen haben, Bruner ergänzte, dass wir jeden Lerngegenstand trotzdem oder gerade deswegen sinnvollerweise dem lernenden Kind und Jugendlichen in einem Spiralcurriculum wiederholt entsprechend seinem augenblicklichen Entwicklungsstand offerieren können und sollten.

Zusätzlich hat *Pierre M. van Hiele* noch *Lernstufen* (oder auch Denkstufen bzw. Denkebenen) bei der Erschliessung von Kompetenzen beobachtet [Hie86, S.39ff], [Hie76, HHG78].



Auch van Hiele geht vom Konstruktivismus aus und sieht sich in einer Linie mit Piaget und Bruner, dass sich der Mensch ‚seine‘ Wirklichkeit konstruiert. Handelt es sich nun um einen neuen Gegenstandsbereich, muss dieser allerdings (Denk-)Stufe für (Denk-)Stufe erarbeitet werden.

„Folgende Punkte können dazu beitragen, die Denkebenen genauer zu definieren:

- a) Auf jeder Ebene erscheint das als nebensächlich, was auf der vorhergehenden Ebene wesentlich war. [...]
- b) Jede Ebene hat ihre eigenen sprachlichen Symbole und ihr eigenes Beziehungsnetz, das diese Zeichen verbindet. [...]
- c) Zwei Menschen, die auf verschiedenen Ebenen denken, können sich nicht verstehen. [...]
- d) Der Reifeprozess, der zu einer höheren Ebene führt, läuft nach einem besonderen Schema ab. Man kann in ihm mehrere Phasen unterscheiden. [...]

“ [HHG78, S.129/130]

Hiele und Hiele-Geldorf betonen: „Man konnte sehen, daß die Denkebenen dem Denken selbst inhärent sind: sie sind also nicht nur für diejenigen, die sich mit Didaktik befassen, von Interesse.“ [HHG78, S.132]

Wenn eine Stufe bereits in einem frühen Stadium der Entwicklung verstanden werden kann, heisst das nicht, dass ein Mensch in einem höheren Alter und höherer Entwicklungsstufe diese nun einfach überspringen kann oder in kürzester Zeit absolvieren kann. Jede Stufe mit ihrer eigenen Repräsentation in einer Sprache muss absolviert werden. Hiele verdeutlicht diesen Zusammenhang zwischen den Stufen nach Piaget und den Denkebenen folgendermassen: „In mehreren Untersuchungen Piagets haben die Stadien eine gewisse Übereinstimmung mit meinen Denkstufen. Wenn das aber der Fall ist, so kann der Übergang von einem Stadium zum andern durch einen geeigneten Unterricht beschleunigt werden. Was aber viel wichtiger ist — unterliesse man den Unterricht, so wäre das Kind ausschliesslich nicht geleiteten Lernprozessen überlassen, bei denen es nur zufälliger Fetzen von Kenntnissen teilhaftig würde.“ [Hie76, 4]

Leider ist diese recht einsichtige Stufentheorie, die van Hiele im geometrischen Mathematikunterricht untersucht hat, relativ unbekannt. Dies mag damit zusammenhängen, dass durch den grossen Einfluss von *Hans Freudenthal* in der Mathematikdidaktik diese nicht so öffentlich wurden [AK05], obwohl sie Freudenthal in seinem Buch *Mathematik als pädagogische Aufgabe* beschreibt [Fre73, S.116-124]. Gerade die Folgerungen bezüglich der Sprachebenen zwischen Gesprächspartnern wie beispielsweise Lehrer und Schüler machen deutlich, wie wichtig die Beobachtungen von van Hiele sind. Aufgrund dieses Missstandes ist bisher auch nicht untersucht worden, wie konkret die verschiedenen Denkstufen bei den Begriffen und Prozessen in der Informatik sind. Es ist daher nicht ausgeschlossen, dass aufgrund des genauen Aussehens dieser Denkstufen und der jeweils in ihnen enthaltenen Phasen und der damit verbundenen (Unterrichts-)Sprachen manche Schülerin und Schüler in Informatik unnötigerweise den Anschluss verliert und daher (bei zumeist wahlfreiem Unterricht in Informatik) diesen abwählt. Hier liegt also ein grosses Forschungsfeld vor der Informatikdidaktik.

#### 4.2.4 Zusammenfassung Psychologie des Kindes

Die verschiedenen Erkenntnisse von *Piaget*, *Bruner* und *van Hiele* widersprechen sich nicht, sondern ergänzen sich und müssen, wenn wir Informatik in der Schule so unterrichten wollen, dass ein möglichst grosser Effekt eintritt, berücksichtigt werden.

Wir müssen darauf achten, dass in den verschiedenen Altersstufen entsprechend der Reife des Kindes die unterrichtlichen Ziele festgesetzt werden. Es ist naheliegend, dass gerade in den Anfangsjahren spielerische Themen beispielsweise aus dem Grafikbereich eine grössere Rolle spielen. Diese entsprechen der Lebenswirklichkeit des Kindes. Erst in späteren Jahren können mehr die Themen behandelt werden, die aus fach- und bildungswissenschaftlicher Sicht besprochen werden sollen. Da die verschiedensten Gegenstände spiralig im Curriculum wieder auftreten sollen und müssen, laufen wir dabei nicht Gefahr, wesentliche Inhalte streichen zu müssen. Sie müssen nur an der richtigen Stelle unterrichtet werden. Wir müssen allerdings bei jeder Thematik darauf achten, dass wir alle Denkstufen einer konkreten Materie durchschreiten.

Es ist davon auszugehen, dass diese Ansprüche der bisherige Informatikunterricht bis auf Ausnahmen nicht erfüllen konnte. Das ist aber weder den Lehrplangestaltern oder den Lehrern vorzuwerfen. Dies ist ein Folge des fehlenden durchgängiges Informatikunterrichtes von der Sekundarstufe I bis zur Sekundarstufe II in der allgemeinbildenden und berufsbildenden Schule.

# 5 Das Schulfach Informatik

## Überblick

In den vorigen Kapitel wurde gezeigt, dass sich ein Schulfach Informatik aus fach- und bildungswissenschaftlicher Sicht rechtfertigen lässt und ebenso lernpsychologische Anforderungen erfüllen kann. Wie lassen sich jetzt konkrete Inhalte für ein solches Schulfach heraus filtern, damit die angestrebten Kompetenzen erreicht werden? Ein Versuch einer inhaltlichen Basis stellen die Bildungsstandards Informatik dar. Vergleichbare Überlegungen gibt es auch in anderen Ländern, die es Wert sind, betrachtet zu werden.

Das Schulfach Informatik erweist sich dabei als in einer Weise völlig unterschiedlich von allen anderen Fächern. Der Schüler nimmt gegenüber dem Lehrer und dem Computer als Arbeitsgerät völlig konträre Positionen ein, die ihn damit immer wieder zwischen einer Schüler- und Lehrerrolle wechseln lässt. Dies führt zu einem besonderen didaktischen Dreieck der Informatik. Die Kommunikationsbasis ist dabei nicht unwesentlich eine Programmiersprache, an die entsprechende Anforderungen zu stellen sind.

Informatik ist heute in allen gesellschaftlichen Bereichen vertreten. Unterrichtliche Gegenstände können und sollen daher aus verschiedensten Sichten betrachtet werden. Diese Great Principles, Leitlinien und Fundamental Ideen dienen zur Orientierung und Auswahl von Unterrichtsthemen und -Gegenständen. Bei der Fülle der möglichen Themen bietet sich ein exemplarisches Lernen an, das häufig in lebenswirklichen Kontexten stattfinden sollte.

Die Aussagen zur Psychologie waren völlig fachunspezifisch. Das bedeutet, dass wir bei der Umsetzung eines Schulfaches Informatik diese Erkenntnisse berücksichtigen müssen. Die Aussagen vor allem von Bruner und von Hiele machen deutlich, dass ein Lernprozess in einem derartigen Fach wie Informatik vergleichbar beispielsweise in Mathematik oder den Fremdsprachen über einen langen Zeitraum organisiert werden soll bzw. muss. Daher ist ein spirales Curriculum zumindest ausgehend von der Sekundarstufe I bis zum Ende der Sekundarstufe II zu entwickeln und durchzusetzen, wenn tatsächlich ernsthaft ein Schulfach Informatik als Pflichtfach gewünscht werden soll. Es versteht sich dabei von selbst, dass in einem solchen Prozess mit überlegt werden muss, welche bisherigen Bildungsinhalte in anderen Fächern heute obsolet sind, da die Anzahl der zur Verfügung stehenden Stunden für den Unterricht seitens der Schülerinnen und Schüler nicht zunehmen soll.

## 5.1 Didaktisches Dreieck der Informatik

Informatisches Wissen und Denken ist notwendig für die zukünftige Generation. Aber wie soll das in der Schule verwirklicht werden. Zwei Möglichkeiten bieten sich an:

- Informatisches Denken wird in einem eigenständigen Schulfach Informatik in der Sekundarstufe I und/oder Sekundarstufe II unterrichtet.
- Informatisches Denken wird ganz oder teilweise in anderen Fächern mitvermittelt.

### 5.1.1 Umsetzung der informatischen Bildung in den 80er Jahren

Während in den 80er Jahren entsprechend der Struktur der Sekundarstufe II das Schulfach Informatik eingerichtet wurde, ist die schulische Umsetzung informatischer Bildung in der Sekundarstufe I immer wieder umstritten. Zu Beginn waren es vorrangig schulische personelle und apparative Strukturen und Bedingungen, die vor Ort die konkreten Entscheidungen geprägt haben. Es entwickelte sich ab ca. Anfang der 80er Jahre eine prinzipielle Diskussion, wie diese grundlegende informatische Bildung in der Schule realisiert werden soll.

Die überwiegende Mehrheit der Verantwortlichen bevorzugte ein *integratives Modell*, das entsprechend in verschiedenen Varianten in den Bundesländern eingeführt wurde. „In Nordrhein-Westfalen gibt es Überlegungen, diesen Block nicht einem Leitfach zuzuordnen (z.B. der Mathematik), sondern zur gleichen Zeit ein entsprechendes Stundenkontingent von den 3 Fächern Mathematik, Deutsch, Technik/Physik zu erhalten.“ [BHW85, S.114]

Doch sehr schnell zeigte sich, dass dieses Konzept keinen Erfolg bezüglich der gewünschten informatischen Bildung erzielte. Der Autor formulierte bereits 1996:

„Die Konzepte aus den 80-Jahren einer I.u.K.-Grundbildung sind gescheitert. Dies war auch vorherzusehen, da die Wirkungen und Auswirkungen einer so umfangreichen Technologie so jungen Schülern mit so wenig Unterrichtsstunden höchstens im Ansatz vermittelbar sind. Von Anfang an fragwürdig war es, diese Ausbildung im wesentlichen von Nichtfachleuten ohne tatsächliche fachliche und fachdidaktische Ausbildung durchführen zu lassen. Gerade in einem inhaltlich und technisch sich so rasch wandelnden Fachgebiet ist die fachliche Kompetenz des Unterrichtenden unabdingbar, um nicht morgen schon von vorgestern zu sein. So mußte die (extrem platte) angestrebte Ganzheitlichkeit zum institutionalisierten Dilettantismus verkommen.“ [Pas96]

Sehr ausführlich hat sich *Engbring* mit diesem Ansatz auseinandergesetzt und kommt abschliessend zu einem ähnlichen Ergebnis:

„Die ITG ist mit ihrem fachübergreifenden Anspruch auch daran gescheitert, dass die Inhalte aus diesen Bereichen nicht genügend didaktisch aufbereitet waren und sind. Viele politisch interessierte Informatik-Lehrende fühlen sich zwar kompetent, Inhalte aus dem Kontext der Informatik zu behandeln, sind es aber nicht wirklich. Sie politisieren lediglich. Umgekehrt fühlen sich auch viele Lehrer aus dem Bereich Sozialkunde kompetent, wenn sie Erfahrungen in der Nutzung von Computern haben, die Rolle von Computern für die Gesellschaft zu analysieren.

Beide Gruppen von Lehrern wildern letztlich in fremden Gefilden. Auch in den Schulen muss man wohl der Tatsache ins Auge sehen, dass es nur wenige Universalgelehrte geben kann und dass Fachlehrer vor allem Fachkompetenzen und keine fachübergreifenden Kompetenzen haben. Deswegen ist es auch eine Frage der Pragmatik, allgemeine Bildung an Fächern auszurichten.

Daraus resultiert die Forderung nach einem Fach Informatik, das sich aber nicht auf die Unterrichtung von Programmierung und ein wenig Hardwarekunde beschränkt sondern zumindest eine Informatik im Kontext miteinbezieht.“ [Eng04, S.160]

Es versteht sich von selbst, dass ein Ansatz, der allein schon aus Kompetenzgründen seitens der Lehrer in der Sekundarstufe I scheitern musste, erst recht nicht für die Sekundarstufe II

geeignet sein kann. Allerdings wirkt diese Begründung etwas defensiv, würde sie doch Informatik als eigenes Schulfach nicht aus sich heraus, sondern nur aus vielleicht nur unglücklichen Rahmenbedingungen zulassen.

### 5.1.2 Problematik integrativer Ansätze

Gerade, wenn die Informatik auch als *Querschnittswissenschaft* verstanden werden kann, weil informatisches Denken wie bereits gezeigt in vielen Wissenschaften vorausgesetzt wird, wirkt es doch faszinierend, zukünftige Lehrer so zu qualifizieren, dass sie das notwendige informatische Wissen und Denken in ihren Fächern quasi kontextorientiert mit vermittelt bekommen. So könnte beispielsweise der Mathematiker argumentieren, das Thema Datenbanken mit der Erarbeitung der Relationenalgebra theoretisch klären zu wollen, ebenso Listen und vor allem Bäume als spezielle Graphen in der Graphentheorie besser als Informatiker besprechen zu können oder in der linearen Algebra als Anwendung der Matrizenrechnung Übergangsmatrizen von endlichen Automaten zu verwenden. Der Biologie erarbeitet Grundlagen der Codierungstheorie und der Kryptologie im Rahmen der Behandlung der Vererbungslehre mit Unterstützung der Sozialkundelehrer zum Thema Sicherheit in Netzen. Im Philosophieunterricht wird die Frage nach der Philosophie unter Berücksichtigung der Komplexitätstheorie unterrichtet. Der Physiker übernimmt gerne den Part der technischen Informatik einschliesslich mit Betrachtungen von informatischen Lösungen bezüglich solcher Themen wie Leitungsvermittlung versus Paketvermittlung sowie Netztopologien. Abgesehen davon, dass bei einem solchen Unterricht hohe Anforderungen an das informatische Wissen des jeweiligen Fachlehrers gestellt werden, wird sicherlich sofort und zu Recht eingewendet, dass in diesem Fall die Summe der informatischen Teilgebiete nicht zu einem gesamten informatischen Denken führt genauso wenig, wie die Behandlung mathematischer Teilgebiete in der Physik einen ordnungsgemässen Mathematikunterricht ersetzt.

In einem anderen Zusammenhang klärt der Pädagoge *Hermann Giesecke* auf:

„Aber die Fächer sollen ja in fächerübergreifende ‚Lerndimensionen‘ aufgelöst werden; diese jedoch kennen weder klare Sachverhalte, noch Begriffe, Regeln und Prinzipien, weil sie entweder gar keine oder gleich mehrere Bezugswissenschaften haben, deren Beziehung zueinander das Gutachten nicht einmal anspricht, geschweige denn klärt. Will man auf diese Bezugswissenschaften ganz verzichten, fragt sich erst recht, woher die Begriffe und Regeln eigentlich kommen sollen. Es geht in der Schule ja nicht nur um Fächer, sondern auch um die unterschiedlichen Methoden der Erkenntnisgewinnung, die zugleich Grenzen des Wissens beschreiben und die etwa für Mathematik und Geschichte keineswegs identisch sind. Das Fächerübergreifende hat kein entsprechendes Set an Methoden.“ [Gie98, S.152]

und an anderer Stelle:

„Die Schulfächer müssen insgesamt die wichtigsten Aspekte der Realität abdecken – die Natur, die Kultur, die Politik. Sie unterscheiden sich nicht nur durch ein bestimmtes Wissen, sondern auch durch spezifische Methoden, mit denen es gewonnen wird. Das Lebendige verhält sich anders als die tote Materie, Biologie ist etwas anderes als Physik. Der bildende Unterricht bedarf also eines Kanons von Fächern, Stoffen und Methoden – was zusätzliche Wahlgebiete nicht ausschließt.“ [Gie02, S.24/25]

Wenn diese Aussagen richtig für alle Fächer sind, so gilt das auch für die Informatik. Auch wenn die Erkenntnisgewinnung in vielen Fächern inzwischen informatisches Denken voraussetzt, so kann informatisches Denken nur erlernt und erworben werden, wenn Informatik in einem Schulfach Informatik gelehrt wird. Daher ist ein Fach Informatik zwingend erforderlich, wenn denn tatsächlich informatisches Denken und Wissen als unverzichtbar angesehen werden.

Es stellt sich daraufhin natürlich die Frage, wie denn nun dieses informatische Denken und Wissen im Informatik-Unterricht anders bzw. besser erworben werden kann als in den Wissenschaften, in denen es entweder als Denkbasis mit verwendet wird oder in Informatiksystemen angewendet wird. Es muss wie bei der Mathematik deutlich werden, dass dies in einem eigenständigen Unterricht sinnvoller ist als ausschliesslich in Anwendungszusammenhängen. In der Physik werden zwangsläufig mathematisches Denken und mathematische Kenntnisse benötigt, aber kein Physiklehrer würde es als seine Aufgabe ansehen, Herleitungen und Beweise zu mathematischen Aussagen in seinem Physik-Unterricht zu thematisieren.

### 5.1.3 Das didaktische Dreieck der Informatik

Mit der Frage, wie sich denn tatsächlicher Informatikunterricht denn von einem Unterricht in anderen Fächern unterscheidet, beschäftigte sich auch eine zehnköpfige Arbeitsgruppe auf der *ItiCSE 2009* in Paris. Sie betonen in ihrem Arbeitsgruppenbericht zunächst die Bedeutung eines eigenen Faches Informatik, das sich — wie sie es nennen — dem *computationalist thinking* verpflichtet ist:

*“We take the position that computing is a discipline unto itself — neither math nor science nor engineering nor anything else, though it overlaps with many of these — and that it is distinguished by a mindset that we call computationalist thinking.“ [ISC<sup>+</sup>09, S.195]*

Der Begriff *computationalist thinking* hat eine etwas andere Bedeutung als *computational thinking*: *“Here, we use the term computationalist merely to mean someone who does computing, and nothing more nor less.“ [ISC<sup>+</sup>09, S.195]* Dieser Begriff bezieht sich also auf das Denken der Personen, die konkret Tätigkeiten in der Informatik ausüben, während *computational thinking* die Bedeutung und Notwendigkeit informatischen Denkens auch in anderen Disziplinen und im täglichen Leben kennzeichnet. Genauso wie ein Physiker nicht dieselbe Ausprägung des mathematischen Denkens wie ein Mathematiker benötigt, gilt Entsprechendes für das informatische Denken. Als Kernaufgabe dieses Faches sieht diese Arbeitsgruppe:

*“As a discipline, computing brings together models, languages, and machines to represent and generate processes. The heart of computing is not the particular artifacts around which our curricula often revolve. Instead, this key idea — that models, languages, and machines are equivalent — is the fundamental core of computing.“ [ISC<sup>+</sup>09, S.195]*

Das informatische Denken ist durch die Entwicklung der automatischen Informationsverarbeitung erst entwickelt und angewandt worden, auch wenn informatisches Denken schon teilweise vorher notwendig und sinnvoll gewesen ist bzw. wäre und mit diesem Denken heute in vielen Disziplinen notwendigerweise gearbeitet werden muss. In einem konkreten Fach Informatik steht aber unzweideutig die automatische Informationsverarbeitung im Vordergrund. In ihr und mit ihr kann dieses Denken entscheidend herausgebildet und entwickelt werden. Dies

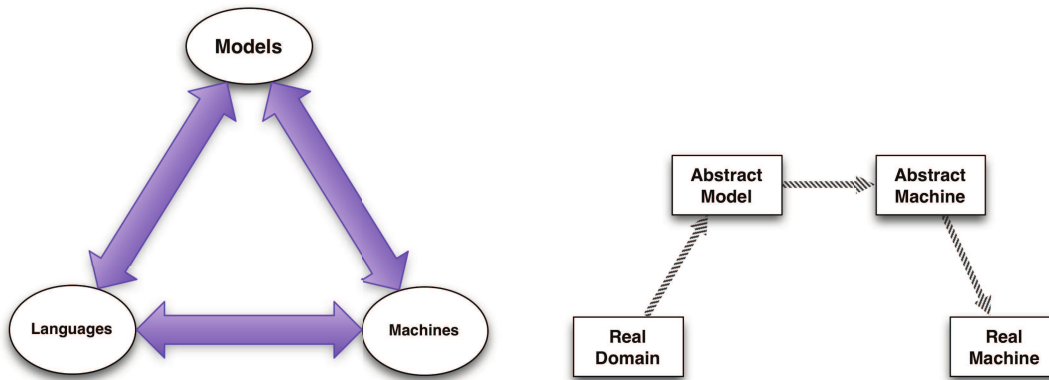


Abbildung 5.1: links: Das didaktische Dreieck der Informatik [ISC<sup>+</sup>09, S.198]  
rechts: Modelle und Maschinen in der Informatik [ISC<sup>+</sup>09, S.199]

verhält sich ähnlich wie die Teildisziplin *Elektrik* in der Physik. Elektrische Phänomene gab es in der Natur schon immer, aber erst durch die umfassende technische Anwendung physikalischer Erkenntnisse erhielt das Gebiet die Bedeutung in der Physik und auch im Schulfach Physik und wird entsprechend heute intensiv mit und über diese technischen Anwendungen unterrichtet.

Auch wenn Informatik als Schulfach sich im Wesentlichen aus den Anforderungen der automatischen Informationsverarbeitung begründet, ist die Bedeutung und Notwendigkeit breiter angelegt: *Computing has become an inter- and intradisciplinary field of intertwined concepts pervading not just most technical fields, but society at large*“ [ISC<sup>+</sup>09, S.196] Dieses wird später noch ausführlicher begründet:

*“Like mathematics, we build models; unlike mathematics’, our models are active and effect-making: they cause things to happen. Like science, we study a system that exists in nature; however, like engineering, our systems are artificial technology and subject to complex trade-offs in implementation. Computing also bears resemblance to the arts — the creation of artifacts — to humanities — the study of texts — and to the social sciences — the study of humans and societies.“ [ISC<sup>+</sup>09, S.196]*

Das Fach Informatik hat nach Ansicht der Arbeitsgruppe folgende Aufgabe:

*“We can now define computing as: any purposeful activity that marries the representation of some dynamic domain with the representation of some dynamic machine that provides theoretical, empirical or practical understanding of that domain or that machine. Often but not always, computationalists then further actualize those representations by executing them on a physical computing artifact.“ [ISC<sup>+</sup>09, S.198]*

Grafisch dargestellt wird das durch das *didaktische Dreieck der Informatik* entsprechend der Abbildung 5.1 links.

Als Maschine ist dabei nicht immer an eine reale Maschine gedacht, es kann auch eine virtuelle Maschine sein, die auf einer realen Maschine abläuft oder auch ‚nur‘ eine virtuelle Maschine wie beispielsweise der Turingmaschine. Entscheidend ist, dass für diese Maschine

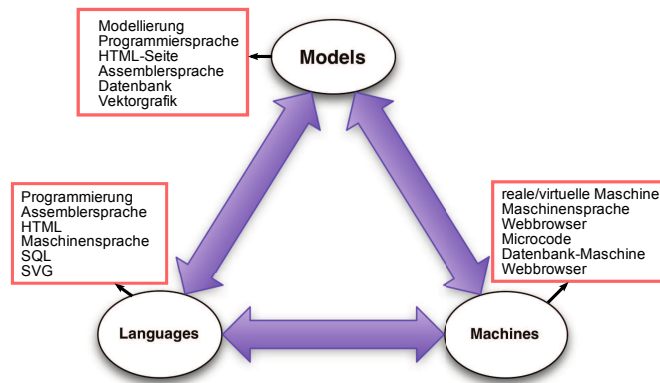


Abbildung 5.2: Das didaktische Dreieck der Informatik in der Praxis [ISC<sup>+</sup>09, nach:S.199]

eine Sprache im Sinne der Informatik definiert und korrekt ist. In vielen konkreten Situationen sieht es speziell entsprechend der Abbildung 5.1 rechts aus. Bedeutsam ist das Verhältnis von Sprache zu Maschine. Es geht in der konkreten Betrachtung und Umsetzung eines konkreten Modelles der Wirklichkeit immer um eine Darstellung in einer Sprache, die von einer Maschine ausgeführt werden kann oder könnte. Dabei besteht Sprache aus Zeichen, die von einer zeichen(symbol)-verarbeitenden Maschine zur Ausführung gebracht werden. Natürlich können gegebenenfalls auch grafische Hilfsmittel verschiedenster Art genutzt werden, die Übertragung in die konkrete zeichenbasierte Darstellung muss aber erkennbar und nachvollziehbar sein. Es ist natürlich denkbar, in einzelnen Unterrichtsszenarien auf einige Schritte in diesem Prozess begründet zu verzichten. Es muss dann aber allen Beteiligten klar sein, dass es ein bewussten Auslassen ist.

Im Verlaufe einer viel zu selten vollständigen Schülerlaufbahn in Informatik könnte die konkrete Realisierung des informatischen Dreiecks der Informatik beispielhaft wie in Abbildung 5.2 aussehen. Diese didaktische Grundposition führt automatisch zu einer leicht realisierbaren methodischen Vielfalt. Es müssen neben den Phasen der Erarbeitung der Modelle und Sprachen auch die Ausführbarkeit in der entsprechenden konkreten Maschine nachgeprüft und auch geübt werden. Damit unterscheidet sich der Unterricht in Informatik essentiell von anderen Fächern. Die Arbeit am Computer ist nicht eine methodische Variante unterrichtlichen Geschehens, sondern notwendiger didaktisch begründeter Anteil der Vermittlung informatischen Wissens.

Noch mehr als die didaktische begründete Erweiterung methodischen Handelns unterscheidet sich der Informatikunterricht auf der Sprachebene von jedem anderen Unterricht in der Schule. Dieses wird deutlich, wenn man das didaktische Dreieck der Informatik mit dem ‚generellen‘ didaktischen Dreieck vergleicht. Dieses didaktische Dreieck ist das erste Modell einer unterrichtsbezogenen Didaktik. [Hub44, S.8] [Asc83, S.65] Es blendet — wie jedes Modell — einige Aspekte unterrichtlichen Geschehens aus, aber auf die kommt es hier nicht an. Wie man in der linken Abbildung 5.4 erkennt, geht es im Unterricht darum, dass die Schule den Stoff vermittelt. Dies kann natürlich auf die verschiedenste Weisen und Methoden direkt (siehe



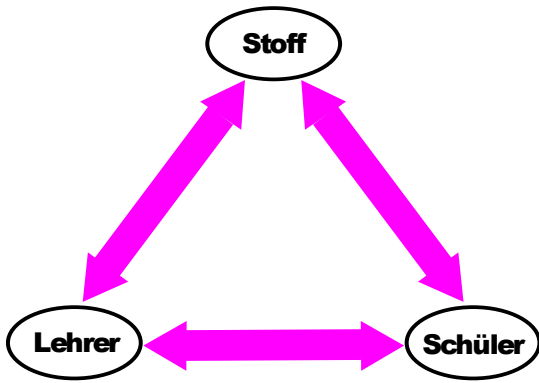


Abbildung 5.3: Das didaktische Dreieck (links)



Abbildung 5.4: Einfachstes Modell für das Lernen (rechts)

das einfachste Modell für das Lernen entsprechend der rechten Abbildung 5.4 [Hub44]) oder vermittels des Einflusses des Lehrers erreicht werden. Das wichtigste Werkzeug, über das der Lehrer verfügt, ist seine Sprache. Das Modell wird über die Sprache des Lehrers an den Schüler kommuniziert. Der Lehrer hofft, dass der Schüler seine Sprache versteht. „Sprache verstehen wir in einem Vorgang der *Nachkonstruktion* dessen, was wir hören, wobei sich diese Nachkonstruktion auf der syntaktischen Ebene und auf der Ebene der extrinsischen Bedeutungen (Vorstellungen) vollzieht: wir bauen die Satzkonstruktion nach, die wir hören, und wir bauen in unserem Geiste die entsprechenden Vorstellungen auf.“ [Aeb76, S.63] Und wenn die entsprechenden Vorstellungen aufgebaut sind, besteht berechnete Hoffnung, dass der Schüler auch die semantische und pragmatische Bedeutung des vom Lehrer vermittelnden Stoffes verarbeiten kann. Schulische Praxis zeigt, dass auf diesem Weg so viele Missverständnisse und Fehlvorstellungen möglich sind, die bekannterweise das schulische Lernen so schwer macht.

Und hier befindet sich der eklatante Unterschied zum didaktischen Dreieck der Informatik: Im informatischen Dreieck ‚schlüpft‘ die Maschine in die Rolle des Schülers. Der Schüler — der eigentlich Lernende — übernimmt die Rolle des Lehrers, indem er das, was er im Unterricht gelernt und verstanden hat, in der für den ‚Maschinenschüler‘ verständlichen Sprache formuliert und der Maschine dies mitteilt. Der Schüler hat somit eine Zwitterfunktion: Er lernt und kommuniziert gleichzeitig als Lehrer und Schüler. Er bringt sich damit in letzter Konsequenz den Stoff weitgehend selbst bei. Der reale Lehrer wird damit letztlich zum Mentor des Unterrichtsgeschehens. Der Maschinenschüler gibt auch sofort eine Rückmeldung: Entweder hat er den realen Schüler nicht verstanden, oder er hat ihn verstanden, aber der Schüler hat ihm etwas Unsinniges erzählt oder im günstigsten Fall reagiert der Maschinenschüler so, wie es der reale Schüler erwartet hat. Sicherlich ist manches Ergebnis nicht immer auf das bewusste und reflektierte Überlegen und Verhalten des realen Schülers zurück zu führen, manches mag auch dem *Trial and Error* - Verhalten des Schülers geschuldet sein, aber durch die Rückkopplung zwischen Maschine und Schüler ist der Lernprozess gegenüber dem schwer kontrollierbaren Kommunikationsproblem mit allen seinen möglichen Missverständnissen deutlich verbessert.

Maschinen verlangen bekannterweise eine hohe Exaktheit in der Formulierung in der ihnen zugeordneten Sprache. Dieser Exaktheit kann der Schüler nicht ausweichen, wie es manchmal

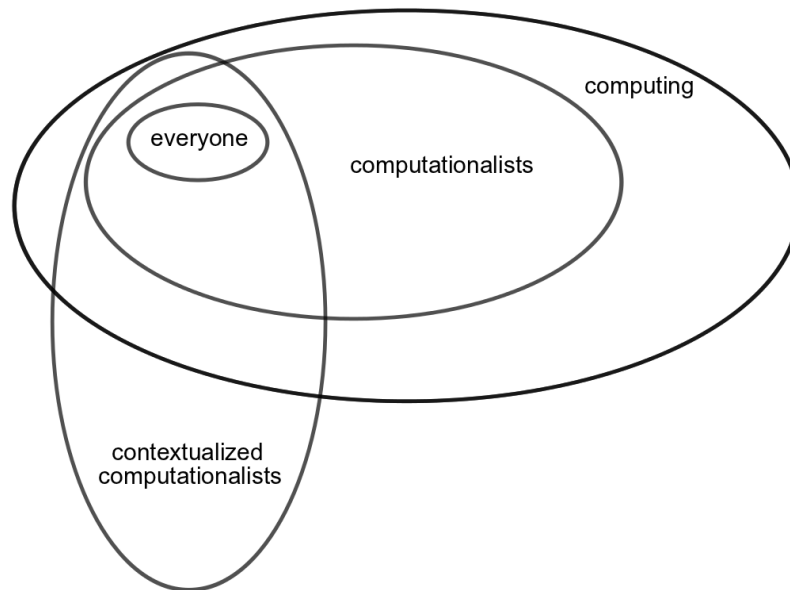


Abbildung 5.5: Notwendiges Informatikwissen für verschiedene gesellschaftliche Gruppen [ISC<sup>+</sup>09, S.200]

im sonstigen Unterricht möglich ist. Dies ist auf der einen Seite sicher eine Erschwernis im Unterrichtsprozess, andererseits ist dies auf Dauer ein Gewinn für den Schüler auch für den Unterricht in anderen Fächern. Auch auf diesen positiven Nebeneffekt sollte die schulische Bildung nicht verzichten.

Was von einem Menschen jetzt und in der Zukunft an informatischen Wissens und Denkens verlangt wird, kann zu Schulzeiten nicht vorhergesagt werden. Die Schule hat allerdings soweit die Grundlagen zu legen, dass der Schüler in seiner weiteren Ausbildung nicht behindert wird. Das bedeutet, das zumindest das in in der Abbildung zum *Notwendiges Informatikwissen 5.5* dargestellten Bereich *Everyone* jedem Schüler vermittelt wird. Die *Bildungsstandards Informatik* [Arb08] sind in Deutschland ein erster Versuch, dieses *Everyone*-Feld auszuleuchten. Diese Standards müssen sinnvollerweise für die Sekundarstufe II fortentwickelt werden. Dass der interessierte Schüler ein zusätzliches Angebot erhält, ist wünschenswert und eigentlich selbstverständlich.

#### 5.1.4 Fazit didaktisches Dreieck der Informatik

Seit etwas über 30 Jahren wird in Deutschland über die Vermittlung einer informatischen Bildung in der Sekundarstufe I diskutiert. Dabei konnte bis auf Ausnahmen nicht erreicht werden, dass Informatik in einem eigenständigen Fach unterrichtet wird. Es konnte sich die Position durchsetzen, dass Informatik ganz oder teilweise in anderen Fächern mit vermittelt wird. Dabei wurden in den Ländern unterschiedlichste Modelle durchgeführt.

Es zeigte sich in der Praxis, dass ein solcher integrativer Ansatz mit grossen Schwierigkeiten behaftet ist. Im wesentlichen mangelte es an Fachwissen der unterrichtenden Kollegen.

Insofern schmolz der informatische Anteil immer mehr zusammen.

Dabei beinhaltet Informatik als Schulfach eine Besonderheit. Im Gegensatz zu allen anderen Fächern kann und muss der Schüler sowohl die Rolle des Schülers wie des Lehrers einnehmen. Auf der einen Seite findet ein ‚üblicher‘ Vermittlungsprozess zwischen Lehrer und Schüler statt, in dem als Kommunikationsform die Sprache zwischen Lehrer und Schüler dominiert. Auf der anderen Seite verhält sich der Schüler wie ein Lehrer gegenüber der Maschine, auf der er praktische Experimente durchführt. Der Schüler muss mit der Maschine in einer Sprache kommunizieren. Diese Besonderheit kann in einem *didaktischen Dreieck der Informatik* dargestellt werden und verleiht der Informatik als Schulfach einen erstrebenswerten Ausnahmecharakter.

## 5.2 Visuelle und textuelle Programmiersprachen

In der Informatik geht es um Modelle der Wirklichkeit und deren Umsetzung in Darstellungen, die auf und mit einer Maschine interpretiert werden. Die Kommunikation erfolgt mit *Sprachen*. Damit sind *Sprachen* in der Informatik und im Informatikunterricht eine zentrale Kategorie im Informatikunterricht. Wie sind diese nun zu unterrichten bzw. zu erlernen. Sehen wir uns dazu in Kurzform an, was *Aebli* aus psychologischer Sicht zur Sprache sagt. Zunächst müssen wir uns daran erinnern, dass es sich bei der Sprache nicht ausschliesslich, aber in zentraler Hinsicht um eine Kommunikationsform zwischen Erzähler und Zuhörer (in der verbalen Form) handelt: „Vor dem geistigen Auge des Erzählers stehen gewisse *Vorstellungen* [Hervorhebung im Original] [...] Vor dem Geiste des Schülers erstehen Vorstellungen, Erlebnisse werden in ihm wach, die weitgehend mit denjenigen des erzählenden Lehrers übereinstimmen.“ [Aeb76, S.22/23] Dies ist zumindest das Ziel des Erzählers. Er muss versuchen, seine Erlebnisse so zu formulieren, dass trotz individueller Vorstellungen der Zuhörer diese weitgehend mit der des Erzählers übereinstimmt.

### 5.2.1 Sprachenlernen

*Hans Aebli* hat sich in seinem Werk *Grundformen des Lehrens* intensiv mit dem Thema Sprache und das Sprachenlernen auseinandergesetzt. Für einen Lehrer ist das Bewusstsein über Sprache und seine Funktion von essentieller Bedeutung. In einigen Auflagen seines Buches hat er die Aussagen dazu in einem eigenen Kapitel *Sprachdidaktik* niedergelegt.

So stellt er klar, dass die „Sozialisation und ihr Teilprozeß des Spracherwerbs [...] durch den Einbezug des Kindes in Situationen des gemeinsamen Handelns, Erlebens und Wahrnehmens“ geschieht [Aeb76, S.52]. Entsprechend sind die Kinder an diesem Prozess durch *mit-handeln, mit-erleben, mit-sehen* beteiligt und erlernen und praktizieren darin *sprachliche Kommunikation*. Weitergehende Untersuchungen und Erklärungen zu diesem interaktiven Prozess hat u.a. *Michael Tomasello* durchgeführt [Tom05].

„Handeln erfordert aber *Informationsverarbeitung* und damit die *Beachtung bestimmter Teile oder Aspekte der Situation*. Das Gesagte gilt natürlich auch für das *Wahrnehmen*, stellt es doch einen notwendigen Bestandteil des Handelns dar.“ [Aeb76, S.53] Dieses Handeln drückt

den wesentlichen Kern des Wahrnehmens und Erlebens aus und die *Worte der Sprache* repräsentieren diesen Handlungskern. „Sprache wird damit in einem zweiten, tieferen Sinn zu einem *stellvertretenden Handlungssystem*.“ [Aeb76, S.58] Das Wort steht für die Sache, „das Verstehen von Sprache erfolgt also durch Nachkonstruktion.“ [Aeb76, S.59] Sprache wird erst gelernt, wenn das Kind durch sein Lernen Mittel und Zweckbeziehungen kennen gelernt hat. Diese werden dann durch die Sprache als Darstellung des Handlungssystems ausgedrückt. „Die Leistung wäre dann ‚bloß‘ diejenige eines Übersetzens aus dem Bereich der Handlung in denjenigen der Sprache“. [Aeb76, S.62] Handlung und Sprache lernen gehören damit zusammen. „Im Falle des Fremdsprachenunterrichtes sieht man das besonders deutlich. Es geht nicht nur darum, einen Ausdruck wie ‚un kilo de pommes‘ in sein Vokabular aufzunehmen. Vielmehr muß dem Lerner dieser Ausdruck in den Sinn kommen, wenn er in Paris vor einem Fruchtestand steht und er ein Kilogramm Äpfel möchte.“ [Aeb76, S.66] Sprache lernen verlangt zumindest in effizienter Form den Kontext, in dem die Sprache angewendet wird. Dieser Kontext soll dementsprechend möglichst im Unterricht nachgebildet werden. [Aeb76, S.52–71]

## 5.2.2 Übertrag auf den Informatikunterricht

Ähnliche Überlegungen gelten für das Erlernen von *Sprache* im Informatikunterricht: Wir haben *Handlungen* von den Schülern *erleben* zu lassen, die möglichst in einem Kontext aus ihren Lebenssituationen enthalten sind, die sie in *Sprache* nachbilden sollen. Auf der einen Seite ist das Sprachenlernen im Informatikunterricht vergleichbar mit dem Erlernen einer ersten Muttersprache, da die Handlungssituationen, Begriffe etc. in dieser Form ihnen bisher nicht bekannt sind, auf der anderen Seite handelt es sich wie eine Form des Fremdsprachenlernens, einer Art Übersetzung von den Schülern bereits auch sprachlich Bekanntem in eine neue Fremdsprache. Dieser Prozess dürfte umso einfacher sein, je häufiger sich die Schüler in einer Situation eines Fremdsprachenlernens und nicht in der Situation eines Lerners einer Muttersprache sind. Auf jeden Fall gilt: Dem Sprachenlernen muss *Zeit* gegeben werden. Das Erlernen neuer Handlungsformen — beispielsweise einer neuen Anweisungsart — muss behutsam angegangen werden, da im Rahmen des Erlernens einer Programmiersprache beide Ebenen, die Handlungsebene und Sprachebene erarbeitet werden müssen.

Diese Aussagen über Sprache verdeutlichen, dass wir in der Schule nicht eine Unterweisung in das (Handwerk) *Programmieren* durchführen, sondern eine Einführung in die Abbildung der Realität in Modelle geben, die in Form von *Programmiersprachen* kommuniziert werden. Damit wird intellektuelles Gedankengut der Informatik vermittelt. Die Unterscheidung lässt auch dem nicht Informatik-Kundigen eher verstehen, warum in dem Schulfach Informatik programmiert werden muss, auch wenn der spätere Nutzer wahrscheinlich nie etwas selbst programmieren wird. Das Begriffspaar *Programmieren* – *Programmiersprache* im Informatikunterricht entspricht dem Begriffspaar *Schreiben* – *Sprache* beispielsweise im Deutschunterricht. Die Informatik betreibt damit *Sprachförderung* im Sinne von *Avi Cohen* and *Bruria Haberman*:

“We suggest that youngsters acquire five basic languages as tools for communication: a mother tongue, an elective international foreign language, a language of science, a language of art and body, and a language of technology, each of which is used to express

*themes and ideas or feelings associated with specific domains and contexts. We believe that long-term study of these languages, along with intelligent practice and utilizing communication skills, is highly useful for citizens to function successfully on personal, national, and global levels.*“ [CH10]

**Andere Sprachformen** Sprache in der oben beschriebenen Form besteht aus Zeichen. Aus diesem Grunde ist es möglich, bei aller Unterschiedlichkeit Vergleiche zwischen dem Erlernen natürlicher und künstlicher Sprachen anzustellen. Natürlich müssen Sprachen nicht aus den üblichen Zeichen bestehen. Es kann sich um andere Symbole, also andere Zeichen handeln. Die *Taubstummgesten* sind beispielsweise eine derartige Sprache. Demnach ist Denken und Sprechen nicht von einer verbalen Sprache abhängig. [Zim11, S.266]

Daher ist es berechtigt, darüber nach zu denken, ob für das Erlernen einer Programmiersprache dem Lernenden unnötige Schwierigkeiten vorübergehend ausgeschaltet werden können. Dazu gibt es mehrere Möglichkeiten, die man unter dem Stichwort *Visuelle Programmiersprachen* zusammenfassen kann:

- Eine Darstellung des Programmtextes, in der die syntaktische Struktur des Textes sofort ersichtlich wird. Dies kann durch andersartige Ausgabe der Schlüsselworte und bzw. oder automatische Einrückungen von Kontrollstrukturen erfolgen. Ergänzend oder alternativ können durch Verbindungslinien die Zusammenhänge von Programmblöcken veranschaulicht werden. Im Grunde genommen hat sich auf der Ebene der Programmiersprache noch keine Veränderung ergeben. Diese Darstellungen haben aus syntaktischer Sicht nichts mit der Programmiersprache zu tun, sondern sind ausschliesslich als bessere Lesbarkeit des Quellcodes gedacht. Sie sind de facto eine Abwandlung und Erweiterung der Editoren mit Syntax-Highlighting. Zwangsweise von der Sprachdefinition ist dies beispielsweise in der Sprache *Python* angelegt. Bei dieser Sprache gehen Darstellung und syntaktische Struktur ineinander über. Eine vergleichbare Darstellung hat die objektorientierte Sprache *Kids Programming Language*<sup>1</sup>, die die Blockstruktur durch farbige Rahmen kennzeichnet und dafür kein zusätzliches Schlüsselwort verwendet.
- Eine Abwandlung stellt die Möglichkeit dar, den Quelltext nicht mehr als Text aus Zeichen oder Strings für die Schlüsselworte zu generieren, sondern grafisch auf einer Oberfläche durch eine Art Baukasten aus Blockbausteinen beispielsweise für die Kontrollstrukturen aufzubauen. Der Programmierer sieht dann nur noch die grafisch gestalteten Blöcke, in denen in freien Feldern die möglichen Eingaben eingetragen werden können, der diese Blöcke-Struktur repräsentierenden Programmcode sieht er gar nicht mehr. Aber jenseits dieser bunten Blöcke existiert tatsächlich noch ein ganz üblicher Quelltext, der durch diese Darstellung aus Sicht der Befürworter leichter lesbar ist.
- Zusätzlich oder alternativ kann nicht nur die Sprachebene, sondern die Modellebene geändert bzw. eingeschränkt werden. Diese Idee wurde in vielen Varianten realisiert. Die Grundüberlegung ist, dass die Modellierungsphase sehr vereinfacht wird und über einsehbare und überschaubare Modelle die Konzentration auf die Programmierungstechniken gelegt werden kann. Dies widerspricht allerdings der Forderung nach dem Lernen

<sup>1</sup><http://www.kidsp1.de>, letzter Zugriff: 1.7.2012

in sozialen und Handlungssituationen des Lebens der Schüler. Beispiele für derartige Lernumgebungen sind:

- Karel, the Robot [Pat81]
- Niki, der Roboter (eingedeutschte Version von Karel, the Robot)<sup>2</sup>
- GuidoRobot<sup>3</sup>
- Kara [RNH04]
- ...

Diese Umgebungen erleichtern die Einführung in grundlegende Strukturen der Programmierung. Das Problem dieses Vorgehens besteht darin, dass zu irgendeinem Zeitpunkt die Modellwelt verlassen werden muss und völlig andere reale Welten modelliert werden müssen. Bei der Umsetzung dieser Modelle in die verwendete Programmiersprache können sich dann unerwartete Transferschwierigkeiten der bereits gelernten Strukturen ergeben, sodass der Lerneffekt aus und in diesen Programmierwelten fragwürdig erscheint. Empirische Untersuchungen zu dieser Problematik sind dem Autor allerdings nicht bekannt.

Alle diese Umgebungen haben aber gemein, dass sie in ‚übliche‘ Programmiersprachen einführen wollen. Gelingt der Übergang von der Modellumgebung in allgemeine Probleme, haben diese Umgebungen ihren Sinn erfüllt.

- Auf einer ähnlichen Idee einer Modellwelt basiert die Programmiersprache *Alice* [DCP09]. Allerdings wird versucht, die reale Lebenswirklichkeit der Schüler mit Alice besser zu erreichen. Dies soll dadurch gelingen, dass die Kinder Geschichten erzählen sollen — *story telling*. Kinder erzählen gerne Geschichten und konstruieren auch gerne Bilder dazu. Im Gegensatz zu den oben dargestellten Programmierumgebungen wird eine eigene Programmiersprache verwendet. Der Übergang zu einer ‚normalen‘ Programmiersprache beinhaltet nun zwei Problembereiche: Wechsel zu einer anderer Sprache und Wechsel der zu bearbeitenden Welten bzw. Modelle [DCP09].
- Ähnliches gilt für die Sprachen *Scratch*<sup>4</sup>, *Etoys* und *Squeak*<sup>5</sup> [For08]. Auch mit ihnen werden in einer jeweils eigenen Programmiersprache in einer Umgebung mit einem Editor, in dem die Programmanweisungen durch Blöcke dargestellt werden, Szenarien aus Objekten, beispielsweise Grafiken, aber auch Sound, hergestellt. Auch bei diesen Sprachen müssen, wenn übliche Programmiersprachen weiter verwendet werden sollen, Übergänge auf verschiedenen Ebenen durchgeführt werden.

Diese Programmiersprachen(-umgebungen) sind entwickelt worden, um Ideen und Gedanken der algorithmischen Problemlösung von den Schwierigkeiten der syntaktischen Korrektheit zu befreien. Zusätzlich wird durch Präsentation einer konkreten Welt und damit weitgehenden Vorgabe der Modellierung die Algorithmisierung in den Vordergrund gestellt. Ist dies das Ziel des Unterrichtenden, können diese Werkzeuge geeignet sein. Ist die Auseinandersetzung um die Programmierung nur ein Teil des gesamten angestrebten unterrichtlichen Vorgehens, reichen diese Werkzeuge nicht aus. Es stellt sich dann allerdings die Frage, ob diese

---

<sup>2</sup><http://www.hupfeld-software.de/pmwiki/pmwiki.php?n=Main.Niki>, letzter Aufruf: 18.10.2012

<sup>3</sup><http://gvr.sourceforge.net>, letzter Aufruf: 2.7.2012

<sup>4</sup>Scratch im Netz:<http://scratch.mit.edu>, letzter Aufruf: 18.10.2012

<sup>5</sup>Squeak und Etoys im Netz:<http://www.squeakland.org>, letzter Aufruf: 6.7.2012

Werkzeuge dann überhaupt noch geeignet sind, ob ihre Vorteile von den oben beschriebenen Effekten nicht überlagert werden.

**Die Programmiersprache LOGO** Eine völlig andere Art von Programmiersprache für Kinder stellt *LOGO* [Sen83] dar. Diese Sprache wurde von *Seymour Papert* konzipiert, nicht, um Programmieren bzw. Informatik zu unterrichten, sondern um Mathematik anschaulicher zu gestalten [Pap85]. Daher verwundert es auch nicht, dass der wesentliche Kern in einer Umgebung besteht, geometrische Objekte zu zeichnen. Dies geschieht mit einer sogenannten Schildkröte (in deutschen Varianten auch als Igel realisiert), der *turtle graphic*. Diese Idee wurde dann auch in anderen Programmiersprachen wie beispielsweise *COMAL* [Chr85], einer Ausbildungssprache aus Dänemark ‚zwischen Basic und Pascal‘ realisiert. Die entsprechenden Sprachen Logo und Comal beispielsweise stellen im Wesentlichen vollwertige Programmiersprachen dar, sodass hier nur ein Wechsel aus der (mathematischen) Modellwelt in die reale Welt nötig ist.

Die pädagogische Umgebung der Schildkröte bzw. des Igels sind in diesen Sprachen nur eine Möglichkeit, bestimmte Elemente des Programmierens durch damit realisierte Animationen zu vermitteln. Hier dienen die Animationen direkt der Darstellung der im Programmtext niedergelegten Anweisungen.

### 5.2.3 Visuelle Programmiersprachen ideal für Anfänger?

Sofern wir davon ausgehen, dass Programmiersprachen nur, wenn auch ein nicht unwesentlicher Teil von Informatik sind, und wir die Kenntnisse an verschiedensten Stellen im Unterricht verwenden wollen, müssen wir Programmiersprachen in einer nicht-visuellen Sprache textbasiert unterrichten. Der Start mit einer visuellen Sprache mag aus Motivationsgründen aus Lehrer- wie Schülersicht interessant und zielführend sein, an einer bestimmten Stelle muss der Übergang zu einer anderen Sprache erfolgen. Dies gilt vor allem dann, wenn eine visuelle Programmiersprache sich auf eine Modellwelt bezieht und nicht zur Lösung weiterer Problem(klassen) geeignet ist. Eine didaktische Begründung für eine visuelle Sprache ist, dass aufgrund angenommener Schwierigkeiten beispielsweise syntaktischer Art durch die Verwendung einer nicht-visuellen Sprache bis zum Übergang zu dieser Sprache so viele Schüler den Bezug zur Informatik verloren haben, dass der Umweg über die visuelle Sprachumgebung geboten und hilfreich scheint.

*Colleen M. Lewis* hat dagegen in einer Untersuchung Schüler betrachtet, die entweder mit *Scratch* oder mit *LOGO* als Anfangssprache unterrichtet wurden. Unter anderem stellte sie folgende Fragen.

- *Writing computer programs is easy.*
- *I am good at writing computer programs.*
- *I plan to continue programming after the class is over.*
- *I want to take another computer programming course.*

Die Ergebnisse der Fragen 1, 2 und 4 sind vergleichbar. Besonders überraschten allerdings die Antworten zur Frage 2: *I am good at writing computer programs*. 17% der ‚LOGO‘-Schüler gegenüber 10% der ‚Scratch‘-Schüler stimmten dieser Aussage vollständig zu. [Lew10] Diese

Untersuchung stellt die Annahme der Befürworter der visuellen Programmiersprachen, mehr Schüler beim schwierigen Thema Programmieren zu erreichen in Frage. Daraus folgt, dass es vorteilhaft sein kann, das Programmieren mit einer nicht-visuellen Sprache zu unterrichten.

## 5.2.4 Visualisieren von Algorithmen

Die visuellen Programmiersprachen verstehen *Visualisierung* entweder als *Visualisierung des Programmcodes* bzw. *Visualisierung durch eine visuelle Modellwelt*. Eine weitergehende Möglichkeit der Visualisierung besteht darin, die Wirkungsweise eines Algorithmus grafisch zu veranschaulichen, ohne zuvor als Lernender überhaupt einen Programmtext zu verfassen. Auch wenn Veranschaulichungen gerade abstrakter Inhalte dem Lernenden fast immer eine grosse Hilfe sein können und genutzt werden sollen, ist der Erfolg dieses Ansatzes kein Automatismus. *Christopher D. Hundhausen*, und *Sarah A. Douglas* untersuchten einige bekannte Experimente und stellten fest:

*“Despite the enthusiasm and high expectations of AV [gemeint ist: algorithm visualization] software developers, a review of ten experiments [...] fails to substantiate the software’s pedagogical benefits. Indeed, only six of those experiments [...] showed that some aspect of AV technology or its pedagogical application significantly impacted learning outcomes.” [HD00]*

Genauso verdeutlichen *Michael Byrne e.a.* bei der Analyse eines der oben untersuchten Experimente:

*“The results from the two experiments suggest that the benefits of animations are not obvious and that in order to determine whether animations can truly aid understanding, teachers and researchers should consider a careful task analysis ahead of time to determine the specific pieces of knowledge that an animation can help a learner acquire and/or practice.” [BCS96]*

Das deutet darauf hin, dass erst die intensive Beschäftigung mit dem in einem Programmtext niedergelegten Algorithmus das tiefere Verständnis seiner Wirkungsweise offenbart. Hat der Lernende aufgrund der Visualisierung den Ablauf eines Algorithmus verstanden, wird er in der Lage sein, diesen (relativ) exakt zu verbalisieren. Dann kann dieser zumindest zusätzlich in eine Programmiersprache überführt werden und getestet und damit verifiziert werden.

## 5.2.5 Fazit visuelle und textuelle Programmiersprachen

Als Zusammenfassung können wir mit *Aebli* zitieren:

Alle kognitiven Funktionen sind auf Zeichensysteme und Sprachen angewiesen. Die meisten höheren Denkprozesse vermögen die notwendige Beweglichkeit nur dadurch zu erreichen, daß die Gegebenheiten, welche sie verarbeiten, durch Zeichen vertreten sind. So besteht ein wesentliches Ziel aller Bildungsprozesse darin, den Schüler die Sprachen erwerben zu lassen, welchen den verschiedenen Bereichen des kognitiven Verhaltens zugeordnet sind (Wortsprachen, technische Sprachen, Schriften der Mathematik und der Naturwissenschaften, Musiknoten, phonetische Schriften usw.). [...] Daher ist es kein Zufall, daß in allen Bildungsprozessen die Entwicklung und die Handhabung von Sprachen, Schriften und Zeichensystemen so große Aufmerksamkeit findet.“ [RA69, S.157]



Die Informatik ist als Wissenschaft unter anderem die *Wissenschaft der Zeichenverarbeitung*. Diese ist damit geradezu prädestiniert für die von Aebli angesprochene *Aufmerksamkeit für Zeichensysteme*. Die Besonderheit der Informatik, mit Hilfe einer Maschine in Sprache dargestellte Informationen verarbeiten zu können, muss als ein nicht unwesentlicher Grundpfeiler im Schulfach Informatik gezeigt und erlebt werden. Hierzu reichen visuellen Darstellungen nicht aus.

## 5.3 Leitlinien der didaktischen Orientierung

In den ersten Kapiteln haben wir gezeigt, dass aus fachwissenschaftlicher und aus bildungswissenschaftlicher Sicht das Schulfach Informatik heute in die Schule gehört. Nun gibt es das Fach in der Sekundarstufe II inzwischen in Form von Grundkursen und in einigen Bundesländern auch als Leistungskurs. Auch in der Sekundarstufe I gibt es sehr unterschiedliche Strukturen, in der Informatik als eigenständiges Fach oder in Zusammenhang mit anderen im geringen Umfang unterrichtet wird. Zusätzlich kommen noch Angebote im Form von Arbeitsgemeinschaften [Wee07, Sta10].

Betrachtet man die Lehrpläne in den verschiedenen Bundesländern oder in einem Bundesland in ihrer historischen Entwicklung, so ist auffallend, dass sowohl inhaltlich als auch didaktisch sehr grosse Unterschiede bestehen. Sie sind deutlich unterschiedlicher als die Lehrpläne in anderen vergleichbaren Fächern wie beispielsweise Mathematik oder Physik [Der81a, Der81b, Der81c, Min99a, Min99b, Min99c]. Zweifelsohne haben tradierte Fächer — vor allem Mathematik — zumindest inhaltlich ein sehr starkes Beharrungsvermögen, das sicher nicht zu unterstützen ist, allerdings setzt sich das Fach Informatik einer möglichen Kritik als allgemeinbildendes Fach aus, was denn an einem Fach allgemeinbildend ist, wenn es zu verschiedenen Zeiten und an verschiedenen Orten inhaltlich, didaktisch und methodisch völlig unterschiedlich gestaltet wird. Es sind daher Kriterien vonnöten, die bei der Fülle an möglichen Themen im Informatikunterricht helfen können, Entscheidungen didaktischer Art zu treffen.

### 5.3.1 Exemplarisches Lernen

Nach den Aussagen von *Bruner* ist es sinnvoll, ein Curriculum *spiralig* zu konzipieren. Das bedeutet folglich, dass dieselben Inhalte und Verfahren wiederholt im Unterrichtsverlauf der Jahre auftreten müssen. Damit kann also nicht Jahr für Jahr eine Schicht auf die nächste Schicht fachlicher oder sonstiger Begriffe etc. aufgebaut werden. Wir haben auszuwählen und zwar so, dass diese ausgewählten Inhalte *exemplarisch* für das Fach stehen. Bruner schreibt dazu: „Er [gemeint ist der Transfer, der Verf.] besteht im Wesentlichen darin, daß man am anfangs nicht eine Fertigkeit (skill) erlernt, sondern einen allgemeinen Begriff (general idea). [...] Je fundamentaler der Begriff ist, den er gelernt hat, umso weiter ist — fast schon der Definition nach — der Bereich seiner Anwendbarkeit auf neue Probleme.“ [Bru70, S.30/1]

In der Physik hat sich vor allem *Martin Wagenschein* für das exemplarische Lernen eingesetzt. Er schreibt dazu weitgehend fachunabhängig:

„So gibt es für jedes ‚Fach‘ ein seiner Eigenart gemäÙes und seine Eigenart bestimmendes Kategoriensystem, und jedes Fach hat seine ihm eigenen Funktionsziele.

Was ist also von einem Thema zu verlangen, damit es zu einer exemplarischen Behandlung geeignet sei? Das Exemplarische Lehren will auf das Ganze des Faches hinaus, und zwar nicht durch ein Auf-Schichten von Stoffen längs vorgezeichneter Ordnungsstrebepfeiler (vom Einfachen zum Komplizierten hin), bis zu einem (den besonderen Anforderungen zwar angepaÙten, aber doch relativ) kompletten Bestand, sondern durch jenes soeben beschriebene ‚ein-für allemal‘ .

Die dabei zuletzt genannte Phase, die im eigentlichen Sinne menschenbildende, von der Physik sich abhebende (also die das *Ganze* der ‚geistigen Welt‘ berührende Phase des Ausstrahlens [...] macht nun deutlich, daÙ der *Aspekt*-Charakter der Physik (und so einer jeden wissenschaftlichen Disziplin) mit dem exemplarischen Verfahren unlösbar verbunden ist: Indem es den Aspekt tranzendiert, enthüllt das exemplarische Verfahren den Aspektcharakter und umgekehrt: *Gerade und nur weil Physik nur EIN Aspekt is [sic!] kann sie exemplarisch und dadurch menschenbildend gelehrt werden.*“ [Wag71, S.230/231]

Der Mathematiker *Alexander Israel Wittenberg* forderte aufbauend auf den Vorschlägen Wagenscheins auch für die mathematische Bildung ein exemplarisches Vorgehen [Wit63]. Die klassische gymnasiale Bildung war damit stark in Frage gestellt. Diese Beschreibungen eines *Exemplarischen Lehrens und Lernen* sind noch sehr unbestimmt. Sie machen aber deutlich, dass wir in der Schule es nicht mit einer Ausbildung von zukünftigen Fachleuten zu tun haben, sondern mit einer Ausbildung für alle Schüler, die mit den in den Schulfächer erlernten unter anderem ein Weltverständnis erwerben wollen. Die konkreten Themen und Inhalte sind mit diesem Ansatz noch nicht geklärt, sie ergeben sich aus dem von Bruner angesprochenen *fundamentalen Ideen bzw. general ideas*.

### 5.3.2 Fundamentale Ideen

Noch bevor psychologische Kenntnisse im Bereich des Lernens bekannt waren, hat sich der Mathematiker und Philosoph *Alfred North Whitehead* am Anfang des letzten Jahrhunderts mit der Didaktik in der Mathematik befasst [Voh00, S.6]. Schon er forderte ein Unterrichten *fundamentaler Ideen* und hat 1911 zu diesen aus seiner Sicht fundamentalen Ideen auch ein heute noch empfehlenswertes kleines Buch geschrieben, in dem er zu Beginn gleich feststellt: „Der Grund für diese Unfähigkeit unserer Wissenschaft ihrem Rufe nachzuleben ist der, daÙ ihre fundamentalen Begriffe dem Schüler nicht losgelöst von dem technischen Beiwerk erklärt werden, das erfunden ist, um ihre genaue Darstellung in besonderen Fällen zu erleichtern.“ [Whi58, S.5] Aber erst mit der Durchsetzung des Kognitivismus und des Konstruktivismus als grundlegende Dimension des Lernens konnten sich diese Vorstellungen in der didaktischen Diskussion verbreitern.

Wagenschein erkannte, dass die Forderung nach einem exemplarischen Lernen für einen Lehrer sehr abstrakt verbleibt und entwickelte daraufhin einen Kanon für die Physik. Dieser kann als Kanon fundamentaler Ideen in der Physik verstanden werden, auch wenn er den Begriff an dieser Stelle nicht verwendet und damit auch keine Definition dafür angibt. Wagenschein fordert folgenden Kanon: „Das Greifbare – Drei Arten von Stoffen, das Hörbare, drei

Zustandsarten der Stoffe, der Wärmefuhr der Materie, zusammenhaltende Kräfte, Wärmestrahlung, Fernwirkung, Leuchten und Licht, die magnetische Kraft, die elektrische Kraft, die magnetische Kraft der bewegten Elektrizität, die elektrische Kraft des bewegten Magnetismus, zerfallende Grundstoffe.“ [Wag71, Kapitelaufzählung] Zu Recht mag kritisiert werden, dass aus diesen Kapitelüberschriften nicht das Exemplarische und das Fundamentale für die Physik deutlich wird, allerdings geben (teilweise) die Überschriften an, dass die Sichtweise über die Physik durch diesen Kanon eine andere als eine rein fachliche ist.

Lange Erfahrungen wie die Physik oder sogar die Mathematik kann die Didaktik der Informatik nicht vorweisen. Entsprechend kann von den ersten Jahren einer Unterweisung in informatischen Themen nicht erwartet werden, dass eine entsprechende didaktische Durchdringung wie oben dargestellt erfolgt ist. Es muss dazu noch berücksichtigt werden, dass im Gegensatz zu den anderen Fächern kaum ausgebildete Informatiklehrer zur Verfügung standen. Allerdings gab es schon recht früh intensive Bemühungen, ein sich stimmiges didaktisches Konzept für die Informatik zu formulieren<sup>6</sup> [DFKT78], [Der77, S.355ff], [Del80, PT80, Bre82, Bau90]

Mit etwas zeitlicher Verzögerung fand diese Diskussion dann in der sich entwickelnden Fachdidaktik Informatik ihren Widerhall. *Andreas Schwill* formuliert als Bedingungen für eine *fundamentale Idee*:

**„Definition:**

Eine fundamentale Idee (bezgl. einer Wissenschaft) ist ein Denk-, Handlungs-, Beschreibungs- oder Erklärungsschema, das

- (1) in verschiedenen Bereichen (der Wissenschaft) vielfältig anwendbar oder erkennbar ist (Horizontalkriterium),
- (2) auf jedem intellektuellen Niveau aufgezeigt und vermittelt werden kann (Vertikalkriterium),
- (3) in der historischen Entwicklung (der Wissenschaft) deutlich wahrnehmbar ist und längerfristig relevant bleibt (Zeitkriterium),
- (4) einen Bezug zu Sprache und Denken des Alltags und der Lebenswelt besitzt (Sinnkriterium).“ [Sch93]

Als Beispiel gibt Schwill (als Nicht-Physiker) fundamentale Ideen der klassischen Physik an:

- Kausalität [...]
- Determinismus [...]
- Invarianz [...]
- Extremalprinzip [...]

[Sch93]

Diese Ideen unterscheiden sich deutlich von den exemplarischen Ideen im Kanon von Wagenschein. Sie sind deutlich mehr von konkreten fachlichen Inhalten entfernt. Um den Kanon von Wagenschein auf diese Struktur abzubilden, müssten die Kanonpunkte der elektrischen und magnetischen Felder zu einer gemeinsamen fundamentalen Idee zusammengefasst werden:

---

<sup>6</sup>Als Beispiel diene ein Bericht aus der *Computerzeitung* vom 27.5.1977 mit einem Bericht aus dem *Arbeitskreis Schulsprache (ASS)* des *FEOLL* in Paderborn: <http://www.computerwoche.de/heftarchiv/1977/22/1199004/>, letzter Zugriff: 31.7.2012

das *Feldprinzip* oder die *Idee des Feldes*. Dies macht deutlich, dass es in den verschiedenen Fachdisziplinen einer Diskussion bedarf, welches denn nun die fundamentalen Ideen dieser Disziplin sind. Es ist dabei nicht zwangsläufig, dass alle Fachleute derselben Meinung sind. Die *fundamentalen Ideen* nach Schwill unterscheiden sich beispielsweise von den *Great Ideas in Computer Science* [Bie97] von Alan W. Biermann. Diese *Great Ideas* verstehen sich mehr als die Repräsentation der *subareas of computing* [CGM<sup>+</sup>89] von Peter J. Denning.

In der Informatik sieht Schwill folgende fundamentale Ideen, wobei er diese als *Masterideen* kennzeichnet, denen viele weitere Ideen untergeordnet sind:

- Algorithmisierung [...]
- Sprache [...]
- strukturierte Zerlegung [...]

[Sch93]

Ohne hier die Diskussion um einzelne fundamentale Ideen führen zu wollen, fällt auf, dass es sich ausschliesslich um Ideen aus der Softwareentwicklung handelt. Dies ist eine Einengung der Informatik, die wie schon ausgeführt, fraglich ist. Bezieht man die technische Informatik ein, so sind sicher in und mit der *von Neumann-Architektur* und in der theoretischen Grundsteinlegung durch die *Turingmaschine* fundamentale Ideen der Informatik realisiert [Ste09, S.31ff]. Es versteht sich von selbst, dass diese fundamentalen Ideen immer wieder im Unterricht auftauchen und sich im Laufe der Zeit über die verschiedensten Themen als Ideen in den Köpfen der Schülerinnen und Schüler auch so konstruiert werden.

Aber es stellt sich die Frage, ob diese sehr abstrakten, übergeordneten fundamentalen Ideen als Kriterien für Inhalte und Kompetenzen im Informatikunterricht tatsächlich hilfreich sind. Diese Ideen tauchen als methodischer Hintergrund zwangsläufig bei allen Themen auf. Damit ist noch lange nicht gesagt, welche Themen exemplarisch im Unterricht behandelt werden sollen. Die *general ideas* nach Wagenschein, die in einem Kanon münden, sind da schon deutlich unterrichtsnäher.

Unabhängig vom Nutzen bei der Definition eines konkreten Curriculums für die Informatik zeigen diese fundamentalen Ideen sehr deutlich, dass es eines eigenständigen Faches bedarf, denn diese Ideen sind Ideen in der Informatik und können sich auch dort nur entsprechend entfalten. Im Mathematikunterricht beispielsweise würden diese Ideen so nicht repräsentiert werden können.

### 5.3.3 Basiskonzepte

Ein den fundamentalen Ideen vergleichbares Konzept ist das der *Basiskonzepte*. Sie werden in den letzten Jahren aufgrund der durch die *PISA*-Ergebnisse in den Naturwissenschaften verstärkt diskutiert und immer mehr in Richtlinien und Lehrplänen verankert. Entsprechend waren diese Basiskonzepte auch Grundlage bei der Entwicklung des Ansatzes *Chemie im Kontext* [IPDF08, S.12]. Die Bildungsstandards in den naturwissenschaftlichen Fächern basieren auf diesen Basiskonzepten [KMK05a, KMK05b, KMK05c]

Als Beispiel diene wieder die Physik. Die der Physik zugeordneten Basiskonzepte sind: Materie, Wechselwirkung, Systeme, Energie. Diese Aufzählung unterscheidet sich sehr von

den fundamentalen Ideen der Physik nach Schwill und dem exemplarischen Kanon von Wagenschein. Dies könnte eventuell darin begründet sein, dass sich *Energie* zwar als *Konzept*, aber nicht als *Idee* formulieren lässt. Auch für diese Basiskonzepte lässt sich feststellen, dass sie das Fach beschreiben, aber nicht umgekehrt die Berechtigung eines Gegenstandes bzw. Themas im Curriculum zumindest alleinstehend begründen können.

### 5.3.4 Fazit Leitlinien der didaktischen Orientierung

In einer Fachdisziplin existieren bestimmte dem Fach innewohnende *fundamentale Ideen* und *Basiskonzepte*. Diese begründen teilweise ein Schulfach. Es ist u.a. das Ziel des Unterrichts in einem Fach, dass die fachimmanenten fundamentalen Ziele und Basiskonzepte am Ende der Schulzeit im fachlichen Bewusstsein des Schülers angelegt werden. Diese Ziele und Basiskonzepte können allerdings nur über fachliche Inhalte und Begriffe, Methoden etc. vermittelt werden. Beispielsweise kann der Lehrer die physikalische Idee des Feldes nur mit den konkreten Feldern (*magnetisches*, *elektrisches* und *Gravitationsfeld*) vermitteln, die zugleich als Inhalt mit ihren gemeinsamen und unterschiedlichen Eigenschaften Ziel des Unterrichtes sind. Ebenso kann die Idee des Algorithmus nur an konkreten Algorithmen verdeutlicht werden. Dabei sollte das Ziel sein, solche Algorithmen auszuwählen, die im Sinne der Exemplarität für sich selbst schon ein Ziel des Informatikunterrichtes sind. Diese beiden Beispiele aus der Physik und Informatik machen deutlich, dass diese Ansprüche — Vermittlung von fundamentalen Zielen und Basiskonzepten an und mit begründeten fachlichen Inhalten — ausgebildete Fachlehrern des entsprechenden Faches verlangen und damit auch einen fachbezogenen Unterricht begründen.

Es ist unumstritten, dass unterrichtlichen Themen nicht mehr die gesamte Breite des Faches darstellen können. Daher ist ein exemplarischer Kanon im Sinne von Wagenschein so auszuwählen, dass diese Ideen und Konzepte besonders gut zur Geltung kommen. Hierzu bedarf es weiterer Kriterien der Auswahl.

## 5.4 Auswahl von Inhalten

Gesucht sind Auswahlkriterien für ein begründetes Curriculum, das auf der einen Seite gewährleistet, dass die fundamentalen Ziele und Basiskonzepte und auf der anderen Seite fachliche Kompetenzen in ihrer Inhalts- und Prozessdimension bei einem exemplarischen Lernen erreicht werden.

### 5.4.1 Great Principles

Dieser Aufgabe stellt sich *Peter James Denning*. Er stellt zunächst fest: *“Indeed, you cannot understand a principle without knowing where it came from, why it is important, why it is recurrent, why it is universal, and why it is unavoidable.”* [Den03] Und er folgert richtigerweise: *“I portray principles and practices as two equal dimensions of computing.”* Als problematisch für die Lehre erweist sich, dass sich die Basistechnologien in den letzten 50

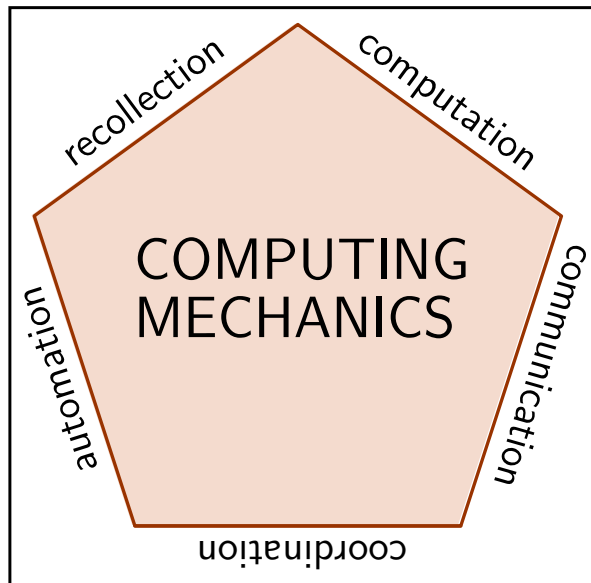


Abbildung 5.6: Die fünf Fenster nach Denning [Den03]

Jahren mehr als verdreifacht haben. Diese Technologien lassen sich strukturell zusammenfassen und unter verschiedenen Aspekten, den 'Windows' betrachten: *Computation, Communication, Coordination, Automation, Recollection*. Dabei zeigt sich, dass bestimmte fachliche Inhalte nicht nur unter einem Aspekt, sondern unter verschiedenen Aspekten betrachtet werden können und müssen. "For example, the Internet protocol stack is an element of both communication and coordination; naming and caching are both elements of communication and recollection." [Den03] Aus dieser Erkenntnis heraus schlägt Denning vor, die fachlichen Ideen und Konzepte nicht linear darzustellen, sondern in einem Pentagramm mit fünf Fenstern wie in Abbildung 5.6 darzustellen.

Diese Basistechnologien haben in der Informatik nicht allein für sich eine Bedeutung. Diese Technologien werden nach bestimmten Prinzipien konstruiert sowie für bestimmte Einsatzzwecke konzipiert. Damit haben wir oberhalb der Basistechnologieebene noch weitere Metaebenen: a) *Computing Mechanics*, b) *Design Principles*, c) *Core Technologies*, d) *Application Domains*

Die Grundlage für die didaktischen curricularen Entscheidungen kann bildlich wie in Abbildung 5.7 dargestellt werden. Für die exemplarische Auswahl entsprechend der Vorstellungen nach Wagenschein bedeutet dies, dass nach Themen gesucht wird, die sich gut eignen, aus verschiedenen Sichten, den 'Windows' nach Denning betrachtet zu werden und wenn möglich, auch ihre Bedeutung für die verschiedenen Metaebenen besitzen.

## 5.4.2 Didaktische Linsen

Einen ähnlichen Ansatz verfolgen die *didaktischen Linsen* von Johannes Magenheimer und Carsten Schulte [MS06, S.319-339]. Magenheimer und Schulte gehen von *Soziotechnischen*

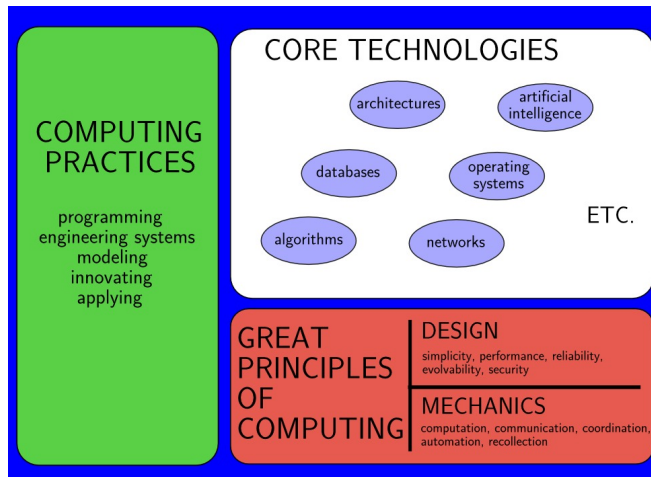


Abbildung 5.7: *Principles based computing* nach Denning [Den03]

*Informatiksystemen (SIS)* aus, da jedes Informatiksystem von sozialen Bedingungen in der Gesellschaft ausgeht und entsprechend derartiger Annahmen von den Konstrukteuren dieses Systems gestaltet wird. Gleichzeitig wirkt es ihn diese Gesellschaft zurück und verändert diese sozialen Bedingungen [MS06, S.320]. Dementsprechend müssen Schüler und Studenten in ihrer Informatik-Ausbildung einen gesamten Blick auf diesen Prozess erhalten [MS06, S.321]. Die Konstruktion eines solchen Systems sollte daher in einer didaktisch geprägten *Dekonstruktion* erfolgen [MS06, S.324]. Soziotechnische Informatiksysteme gibt es in verschiedenen gesellschaftlichen Bereichen wie *Produktion, Distribution, Bildung, Gesundheitswesen, Unterhaltung, Forschung, Militär* und *elektronische Demokratie* [MS06, S.326-328]. Diese können aus verschiedenen *didaktischen Linsen*<sup>7</sup> betrachtet werden: *Automation, Interaktion, Informationsverarbeitung, Netze, Normen* und vor allem auch *soziale Fragen* [MS06, S.328-332]. Ausgehend von ausgewählten soziotechnischen Informatiksystemen sind entsprechend Abbildung 5.8 Informatikkurse zu konzipieren, in denen durch Dekonstruktion unter Verwendung geeigneter didaktischer Linsen sowie ausgewählter Medien und Methoden informatische Kompetenzen erreicht werden [MS06, S.332-338]. Ergänzend gehen *Carsten Schulte* und *Christina Dörge* davon aus, dass *digitale Artefakte* zwei besondere Eigenschaften haben: *Struktur* und *Funktion*. Um neben der vom Benutzer gewünschten Funktion auch die Struktur zu erkennen, bedarf es unterschiedlicher Ansätze je nach digitalem Artefakt. Diese werden durch die unterschiedlichen *didaktischen Linsen* betrachtet [SD08].

Es sind offensichtlich Überschneidungen, aber auch Unterschiede zu den *Windows* von Denning erkennbar. Ein wesentlicher Unterschied ist, dass Magenheim, Schulte und Dörge von einem *anwendungsorientierten Ansatz* ausgehen, während Denning bezüglich des Ansatzes völlig frei ist.

<sup>7</sup>englisch auch: *educational lenses*

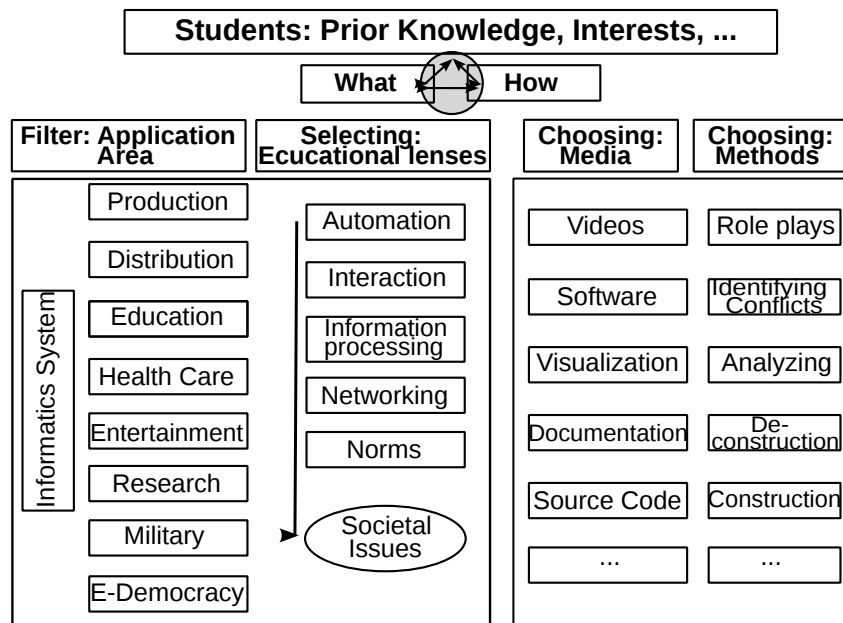


Abbildung 5.8: Konstruktion eines Informatikkurses anhand eines Informatiksystemes nach Magenheimer und Schulte [MS06, S.332]

### 5.4.3 Fazit Auswahl von Inhalten

Aus beiden Ansätzen kann gefolgert werden, dass Unterrichtsthemen gesucht werden, die aus verschiedenen Gesichtspunkten betrachtet werden können. Damit wird auch ansatzweise verhindert, dass aus Sicht des Schülers im Unterricht scheinbare unabhängige Themen behandelt werden, wenn immer wieder an den Unterricht aus früheren Stunden ‚erinnert‘ und darauf Bezug genommen wird.

## 5.5 Informatik und Kontext

Aus konstruktivistischer Sicht folgt, dass der Anschluss neuen Wissens umso besser gelingt, je mehr dieses neue Wissen Anschluss beim Vorwissen des Schülers findet. Inhalte, die mit dem bisherigen Erfahrungen eines Lerners nichts zu tun haben, werden nur schwerlich erfasst und in das Wissen integriert. Hieraus folgt, dass Schule gut daran tut, an der Lebenswirklichkeit der Schüler anzusetzen. Eine Folgerung daraus ist, die Ausbildung an den Hochschulen nicht den Schulen überzustülpen — Abbilddidaktik —, sondern sich an der Lebenswirklichkeit und der Psychologie des Kindes zu orientieren.

Didaktische Ansätze hierzu hat es in den naturwissenschaftlichen Fächern verschiedene gegeben. Sehr deutlich wird das im *situativen Ansatz* deutlich. „Beim situativen Ansatz werden Situationen, die für das jetzige oder zukünftige Leben des Lernenden relevant sind, erfaßt und, soweit die Naturwissenschaften einen Beitrag dazu leisten können, die notwendigen Qualifikationen zu deren Bewältigung bereitgestellt.“ [Häu76, S.45] Aber auch weitere ähnliche Ansätze wie *problem-, umwelt- und anwendungsorientierte* Ansätze können als Spezialfälle



des situativen Ansatzes aufgefaßt werden.“ [Häu76, S.45] Auch in der Mathematik gibt es immer wieder derartige didaktische Forderungen. So stellt *Walther Lietzmann* 1949 fest: „Eine ganz besonders wichtige Fähigkeit ist es, in den Vorgängen des praktischen Lebens oder wissenschaftlichen Untersuchens ein mathematisches Problem zu erkennen und durchzuführen.“ [Lie49, S.76]

**Anlass für den kontextorientierten Unterricht** Das Interesse für naturwissenschaftlichen Unterricht, aber auch der Informatik hat nachgelassen. Dies hat sicher vielfältige Gründe: In einer notenfixierten Welt werden die Fächer seltener gewählt, die als schwer gelten, einige Schulen schränken die Wahlfreiheit aus den verschiedensten Gründen wie beispielhaft Lehrermangel ein und die Rolle dieser Fächer für die Schullaufbahn ist für diese nicht förderlich. Dies erklärt aber nur teilweise den geringen Stellenwert dieser Fächer für die Schule.

Der Physikdidaktiker *Gottfried Merzyn* hat die Beliebtheit von Schulfächern und das Interesse für sie untersucht [Mer08] und dabei festgestellt: „Sehr auffällig sind große Unterschiede zwischen den vier Fächern. Biologie ist ein überdurchschnittlich beliebtes Fach. Ganz anders Physik und Chemie. Sie liegen in allen Studien weit hinten in der Beliebtheitskala, und sie liegen — eine weitere Auffälligkeit — stets dicht beieinander. Eine markante Eigenheit der Mathematik ist, dass sie stark polarisiert: Sie ist gleichzeitig bei einem ganzen Teil der Schüler sehr beliebt und bei einem erheblichen Teil sehr unbeliebt. Ähnliches gilt für Physik und Chemie. Biologie dagegen wird viel seltener abgelehnt<sup>8</sup>.“ Dies wird durch eine Darstellung unterschiedlichster Untersuchungen in beiden Sekundarstufen aus Deutschland, Österreich und der Schweiz sowie auch dem Ausland aus den letzten 30 Jahren belegt [Mer08, S.15-28]. Merzyn stellt dann weitere Untersuchungen vor, aus denen deutlich wird, dass Schüler mehr Interesse für eine Vernetzung der naturwissenschaftlichen Fächer mit Anwendungen der realen Welt haben als für die ‚reine‘ Wissenschaft selbst. Dies nimmt mit zunehmenden Alter sogar zu [Mer08, S.29-41]. Es gilt nicht nur für die Sekundarstufe II, sondern auch schon für die Sekundarstufe I: „Die mangelnde Akzeptanz der Fächer in der Breite der Schülerschaft ist bereits in der Sekundarstufe I angelegt. Das gilt in besonderer Weise für Mädchen. Die reine Wissenschaft — mathematisch-formale Betrachtungen, Naturgesetze, Laborexperimente — spricht nur sehr wenige Schülerinnen und Schüler an. Vermisst werden Anwendungsfelder und Bezüge zu interessanten Phänomenen aus der Erfahrungswelt der Schülerinnen und Schüler.“ [Sch09a] schreibt der Physikdidaktiker *Horst Schecker* in einer Zusammenfassung für eine physikalische Didaktik-Zeitschrift. Merzyn merkt allerdings bezüglich *Alltagssituationen* besorgt an: „Derartige Situationen sind nicht so häufig, wie es sich ein Fachlehrer wünscht. Und fatal ist, dass die technische Entwicklung unter diesem Blickwinkel ungünstig fortschreitet.“ [Mer08, S.47] Eine Tatsache, die so auch für die Informatik bei der zunehmenden Komplexität gilt.

Sehr überraschend sind diese Tatsachen nicht, schreibt doch schon *Wagenschein* 1962: „Ein Grund für den geringen Erfolg des Physikunterrichts an den allgemeinbildenden Schulen — besonders an den Gymnasien, die ja eine Schlüsselstellung haben — scheint mir der folgende

---

<sup>8</sup>Aus einer Kurzbeschreibung des Autors der Untersuchung: [http://www.schulportal-thueringen.de/c/document\\_library/get\\_file?folderId=573998&name=DLFE-83210.pdf](http://www.schulportal-thueringen.de/c/document_library/get_file?folderId=573998&name=DLFE-83210.pdf), letzter Aufruf: 25.7.2012

zu sein: Fachliche Ausbildung und vom Fach ausgehende Menschenbildung werden nicht genügend unterschieden und nur als ein Mehr oder Weniger desselben, nämlich einer gewissen Art der Ausbildung, praktiziert!“ [Wag71, S.139] Sofern der Informatikunterricht auch ein ‚Abklatsch‘ der universitären Fachausbildung darstellt, kann Ähnliches von ihm gesagt werden.

**Chemie im Kontext** Ab dem Jahr 1997 engagierte sich die Universität Oldenburg für neue Ideen im Rahmen einer *Chemie im Kontext*<sup>9</sup>, das sich an das englische Chemiecurriculum „Salters Advanced Chemistry Course<sup>10</sup>“ anlehnte. Aus dieser Initiative entwickelte sich ein vom Bundesministerium für Bildung und Forschung unterstütztes Forschungsprojekt in den drei Naturwissenschaften Chemie, Biologie und Physik, das seinen Schwerpunkt in den Jahren 2002 bis 2005 hatte. Im Teilprojekt *Chemie im Kontext*<sup>11</sup> waren die Universitäten Dortmund, Wuppertal, Oldenburg und Kiel involviert.

Ausgangspunkt der naturwissenschaftlichen Fächer ist: „Naturwissenschaftliche Grundbildung (Scientific Literacy) ist die Fähigkeit, naturwissenschaftliches Wissen anzuwenden, naturwissenschaftliche Fragen zu erkennen und aus Belegen Schlussfolgerungen zu ziehen, um Entscheidungen zu verstehen und zu treffen [...] (OECD 2000)“ [IPDF08, S.18]

Das etwas später (2003) formierte Projekt *Piko* der Physiker [DMSW07], [Dui10] ist 2007 abgeschlossen. In dem zum Projekt gehörigen *PIKO-Brief 5* wird festgestellt: „Ein fachlicher Inhalt kann immer nur in einem für Schülerinnen und Schüler relevanten Kontext gelernt werden. Deshalb sollte der Kontext so gewählt werden, dass er für die Lernenden ‚sinnstiftend‘ (Muckenfuß, 1995) ist. Solche Kontexte können Themen aus dem Alltag der Schülerinnen und Schüler sein, also Alltags- und Naturphänomene oder technische Anwendungen, aber auch Aspekte der Bedeutung der Physik für Technik und Gesellschaft.“<sup>12</sup>.

Die Biologen formulieren im Projekt *bik*<sup>13</sup> „Ein zentrales Anliegen von Biologie im Kontext ist es, biologische Konzepte, Arbeitsweisen und Strategiewissen bei ihrer Vermittlung im Biologieunterricht so in ausgewählte Kontexte einzubetten, dass die Schülerinnen und Schüler damit gezielt in ihrer Kompetenzentwicklung unterstützt werden. Den Lernenden werden Anknüpfungspunkte an ihr Vorwissen und ihre Alltagserfahrungen geboten sowie wissenschaftliche Anwendungsfelder vermittelt.“

Die Intensität in diesen Projekten war in der Chemie am grössten. Daher verwundert es nicht, dass inzwischen im Cornelsen-Verlag auch mehrere Chemie-Lehrbücher erschienen sind, die sich diesem Gedanken verpflichtet fühlen. Entsprechende Angebote für Physik und Biologie fehlen.

Die Chemiker legten für einen Unterricht ein reglementiertes *Stufenkonzept* vor:

- Begegnungsphase
- Neugier- und Planungsphase
- Erarbeitungsphase

<sup>9</sup><http://www.chemiedidaktik.uni-oldenburg.de/17220.html>, letzter Zugriff 26.7.2012

<sup>10</sup><http://www.york.ac.uk/org/seg/salters/chemistry/>, letzter Zugriff: 26.7.2012

<sup>11</sup><http://www.chik.de>, letzter Zugriff: 20.10.2012

<sup>12</sup><http://www.ipn.uni-kiel.de/projekte/piko/pikobriefe032010.pdf>, letzter Zugriff: 26.7.2012. Die eigentlichen Projektseiten sind im Internet nicht mehr verfügbar

<sup>13</sup><http://bik.ipn.uni-kiel.de/typo3/index.php?id=30>, letzter Zugriff: 26.7.2012

- Vernetzungs- und Vertiefungsphase [IPDF08, S.27]

Die Umsetzung war und ist nicht immer unproblematisch. „Im Anfangsunterricht ist darauf zu achten, dass durch den Kontextbezug die fachliche Struktur nicht zu stark verdeckt wird. [...] Es ist von Bedeutung, dass trotz Kontextorientierung der ‚fachliche Faden‘ nicht verloren geht und bewusst auf ein Basiskonzept hin gearbeitet wird.“ [IPDF08, S.35] Der Reiz eines Kontextes aus Sicht der Schüler ist nicht unbegrenzt. So stellten *David Di Fuccia, Wolfgang Schwarz und Bernd Ralle* fest: „Aus dieser Reihe von Unterrichtseinheiten konnte für die Gegebenheiten im bayerischen Schulset abgeleitet werden, dass eine Länge von sieben bis zehn Unterrichtsstunden am besten geeignet ist, um die Motivation der Schülerinnen und Schüler zu erhalten und gleichzeitig den Kontexten in ihrer übergreifenden Anlage noch gerecht zu werden.“ [FSR05, S.4] Diese Stundenanzahl ist doch recht klein, wenn ausgehend von einem Kontext, der zuerst erschlossen werden muss, ein fachlicher Zusammenhang extrahiert werden soll und anschliessend noch vernetzt und vertieft werden soll. Es ist dabei zudem auf keinen Fall zu vergessen, dass im konkreten Unterricht immer wieder Wiederholungs- und Übungsphasen eingebunden werden müssen. Aus dieser Sicht erweist es sich als problematisch, ein ganzes Curriculum aus derartigen geschlossenen Kontexten zu gestalten.

Einzelne Lehrpläne wie beispielsweise der Kernlehrplan Naturwissenschaften für die Gesamtschule in NRW [Min11] sind inzwischen mit Ideen der Kontexte konzipiert worden. Ob sich dieser Ansatz auch nach dem Ende der Projekte langfristig durchsetzen wird, wird die Zukunft zeigen.

### 5.5.1 Informatik im Kontext

Die kontextorientierten Projekte, unterstützt durch das BMBF, waren ausschliesslich in den ‚klassischen‘ naturwissenschaftlichen Fächern angesiedelt. Auf der einen Seite gibt es in der Informatik zweifelsohne die Gefahr gleichartig wie in den Naturwissenschaften, dieses Fach in Form einer *Abbilddidaktik* zu unterrichten, andererseits gibt es bisher kein etabliertes Fach Informatik in der Sekundarstufe I, auf die eine solche Kritik vergleichbar zu der in den naturwissenschaftlichen Fächern zutreffen könnte. Eine solche fachliche Konsolidierung der Inhalte wurde allerdings gerade durch die *Bildungsstandards* angestrebt. Es ist denkbar, dass sich die Initiative *Informatik im Kontext (INIK)*<sup>14</sup> bildete, um einem solchen Fehlverstehen der Bildungsstandards von Anfang an entgegenzutreten.

Die auf dieser Plattform vorgestellten Einheiten sind aber nicht alle nach der Struktur eines Unterrichtes im Kontext wie im Fach Chemie vorgestellt entwickelt worden. Es stellt sich damit die Frage, wie sich der Unterricht nach *Informatik im Kontext* von anderen Konzepten abgrenzt. Auch der zeitliche Umfang der verschiedenen Reihen ist sehr unterschiedlich.

Natürlich macht es aus konstruktivistischer Sicht Sinn, Informatik aus Kontexten der Lebenswirklichkeit zu unterrichten. Zudem liegt ein Bezug auf einen Kontext für die Informatik nahe, da sie sich in ihrem fachsystematischen Handeln beispielsweise in der Software-Entwicklung zwangsläufig in einem Kontext bewegt<sup>15</sup>. Dieses wird unter anderem im *anwendungsorientierten Ansatz* und der sich daran anschliessenden Debatte um eine *Grundbildung Informatik*

<sup>14</sup>[www.informatik-im-kontext.de](http://www.informatik-im-kontext.de), letzter Zugriff: 26.7.20121

<sup>15</sup>siehe dazu auch: [Eng04]

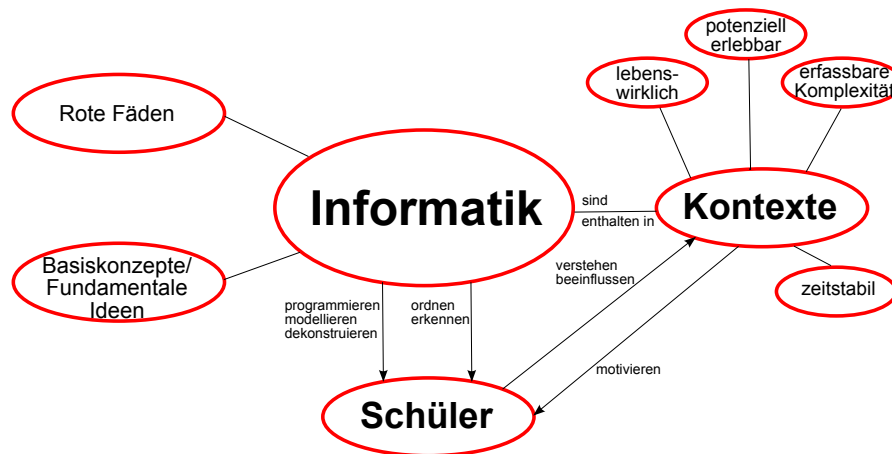


Abbildung 5.9: Kontexte und Informatik [EP10b, S.113]

in den 80-Jahren deutlich.

Aufgrund eines Mangels eines klaren didaktischen Profils sowie auch einem Mangel an qualifizierten Lehrern sowie dem Druck der aus der Grundbildung Informatik entwickelten *informations- und kommunikationstechnologischen Grundbildung* hat sich in der Sekundarstufe I häufig eine mehr oder weniger intensive Applikationsschulung als Informatik etabliert. Soll eine Initiative *Informatik im Kontext* einer tatsächlichen Informatik-Ausbildung gerecht werden, muss sie den Unterrichtenden Material an die Hand geben, das sie weg von dieser Applikationsschulung führt und sichtbar und nachweisbar informatische Kompetenzen, wie es in Abbildung 5.9 dargestellt ist, in den Mittelpunkt stellt.

Die Gefahr für Projekte dieser Initiative besteht darin, dass Kontexte aus dem Informatikfeld bearbeitet werden, bei denen tiefgehende Informatik-Kompetenzen nicht mehr thematisiert werden. Ein derartig aufgebauter Informatikunterricht ist dem allgemeinbildenden Informatikunterricht abträglich. Ein solches Beispiel — hier allerdings aus dem Bereich der Sekundarstufe II — ist beispielsweise in einer Bachelorarbeit zu sehen:

Diese Arbeit *Informatik im Kontext — Web 2.0 — Anwendungen* führt den Anspruch bereits im Titel. Die Einsortierung der in dieser Einheit bearbeiteten Zellen in die Bildungsmatrix<sup>16</sup> entsprechend Abbildung 5.10 zeigt, dass die eigentliche Zielrichtung des Projektes *Informatik im Kontext* nicht erreicht wird. Die in dieser Arbeit beschriebene Unterrichtseinheit in der Form des Stufenprojektes, wie es in der Chemie vorgeschlagen wird, hätte so genauso im Fach Sozialkunde durchgeführt werden können und ist damit eindeutig kein Informatikunterricht. Auch eine Masterarbeit im Themenbereich ‚Informatik und Kontext‘ zum *Ecommerce-Verhalten Jugendlicher* [Led11] deutet in der Umsetzung auf einen weitgehend sozialkundlichen Unterricht hin. Auch das aktualisierte *Planspiel ‚Datenschutz 2.0‘* [DO11] — nach eigenen Angaben *eine Unterrichtsreihe des Projektes ‚Informatik im Kontext‘* — ist ohne Erweiterungen kein Informatik-Unterricht. Wenn diese Gefahr bereits in der Sekundarstufe II gegeben ist, muss befürchtet werden, dass Konzepte für die Sekundarstufe I noch mehr dieser

<sup>16</sup>siehe dazu Kapitel 5.7.1

	Modellieren und Implementieren	Begründen und Bewerten	Strukturieren und Vernetzen	Kommunizieren und Kooperieren	Darstellen und Interpretieren
Information und Daten					
Algorithmen					
Sprachen und Automaten					
Informatik- systeme					
Informatik, Mensch und Gesellschaft		XX	X	XX	X

Abbildung 5.10: Bildungsmatrix für den Kontext *Web 2.0* [Rau08]

Versuchung unterliegen werden.

Der Vergleich der Geschichte und der Darstellungen der verschiedenen *Unterricht im Kontext* - Initiativen zeigt, dass die naturwissenschaftlichen Fächer und Informatik unterschiedliche Schwerpunkte und Zielrichtungen setzen müssen. Geht es in den naturwissenschaftlichen Fächern darum, den bisher fachsystematisch dominierten Unterricht zu verändern, so muss in Informatik darauf hingewirkt werden, dass zumindest in der Sekundarstufe I die informatischen Kompetenzen deutlich herausgearbeitet werden. Die Abbildung 5.11 verdeutlicht die Gemeinsamkeiten und die Unterschiede in diesen didaktischen Konzepten.

## 5.5.2 Grundbildung Informatik

Offensichtlich ist die Gefahr gross, dass ein derartig verstandenes Konzept *Informatik im Kontext* schnell weg von informatischen Inhalten führen kann. Damit könnte sich eine Entwicklung wiederholen, wie wir sie vor ca. 30 Jahren im Umfeld der *Grundbildung Informatik* erlebt haben.

Die *Grundbildung Informatik* entsprach einer Forderung der Informatiker nach einer Informatik für die Sekundarstufe I [INF84, S.93-131], [BHW85, RZ85, AKHK<sup>+</sup>90]. Da sich das Fach nicht unwesentlich aus der gesellschaftlichen Bedeutung der Anwendungen dieser Disziplin begründete, verstand es sich fast von selbst, dass ein solcher Unterricht nicht nur Themen aus beispielsweise der Algorithmik oder Programmierung behandeln sollte. Die Situation war der Anfang des 19. Jahrhunderts nicht unähnlich, als der *Verein zur Förderung des Mathematisch Naturwissenschaftlichen Unterrichts (MNU)* auf seiner Jahrestagung 1904 in Breslau die naturwissenschaftlichen Fächer in den Schulen forderte. In seinem einleitenden Referat sagte das Mitglied der 'Unterrichtskommission' *K. Fricke*: „Nicht Fachbildung ist es, die wir bei dem Unterricht an den höheren Schulen im Auge haben, auch nicht einseitige mathematische oder naturwissenschaftliche Schulung, sondern wir wollen den mathematischen und naturwissenschaftlichen Unterricht auffassen im Rahmen einer allgemein menschlichen Bildung, wie sie für das Verständnis der gegenwärtigen Kultur und für die lebendige Mitwirkung der heranwachsenden Jugend an der Weiterbildung unseres Kulturzustandes erspriesslich erscheint.“ [Gut08, S.22], [Hir87, S.119] Wie wir wissen, sind die naturwissenschaftlichen

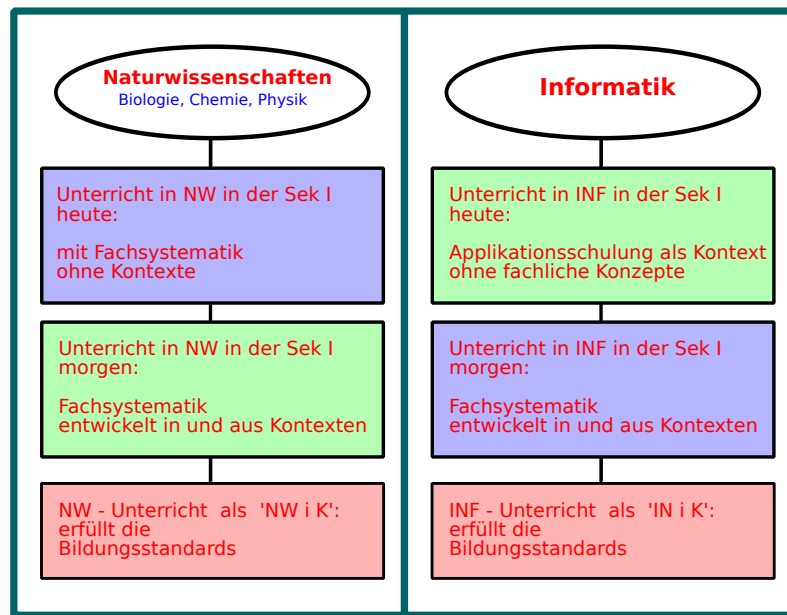


Abbildung 5.11: NW und Informatik im Kontext [EP10a]

Fächer wenig später in den Schulen endgültig eingeführt worden, allerdings hat sich ihre Didaktik doch lange noch an der Fachwissenschaft orientiert.

Es bestand in den 80-Jahren des zwanzigsten Jahrhunderts die Chance, ein Fach Informatik in der Schule zu etablieren, das im Gegenstand und unterrichtlichen Inhalten von Anfang an nicht nur als Abbild des entsprechenden Faches konstruiert werden sollte. Im Gegensatz zu der Etablierung der naturwissenschaftlichen Fächer Anfang des zwanzigsten Jahrhundert im Gefolge der Meraner Beschlüsse [Gut08] mit ähnlichen sachlichen Gründen waren kognitive und konstruktivistische Überlegungen in der Diskussion um die Einführung eines solchen Lerngegenstandes prägend.

Als didaktischer Ausgangspunkt für die *Grundbildung Informatik* sollte daher die *Anwendungsorientierung* und nicht die in der Sekundarstufe II prägende *Algorithmenorientierung* sein. Dieser Ansatz wird durch *Schulz-Zander* beschrieben:

“Der anwendungsorientierte Ansatz unterscheidet sich vom algorithmenorientierten durch die Betonung der Problemanalyse und Modellbildung einerseits und der Modellanwendung und Beurteilung ihrer Auswirkungen.

Die Wahl der Anwendungen bzw. Probleme sollte dadurch bestimmt sein, daß sie ’

- auf dem Erfahrungshintergrund der Schüler aufbauen,
- das individuelle Betroffensein beinhalten,
- zum Handeln herausfordern, indem sie Konfliktsituationen, durch Interessengegenstände hervorgerufen, beschreiben und
- beispielhaft für die gesellschaftliche Bedeutung der Informatik sind’ [SZ78, S.47]

Der anwendungsorientierte Ansatz zielt also auf eine gesellschaftsbezogene und individuelle Handlungsfähigkeit des durch den Einsatz der Informationstechnologien *betroffenen* Schülers ab.“ [SZ86, S.20]

*Handlungsorientierung* ist hier als *Prozesskompetenz* zu verstehen, die der Schüler am Ende seiner Schullaufbahn für sein Leben gewinnen soll, um seine eigenen Interessen vertreten zu können und nicht, wie es oft missverstanden wird, ausschliesslich als *Unterrichtsmethodik*. Eine derartiger Anspruch erfordert eine hohe Kompetenz des Unterrichtenden sowohl auf fachlichem als auch auf sozialpolitischem Gebiet. Unabhängig davon ist dieses Ziel allein aus fachlichen Gründen kaum erreichbar. „Dies liegt auch an der Komplexität und Voraussetzungshaftigkeit der Algorithmisierung und Programmierung.“ [For92, S.229] Insofern musste sich die Anwendungsorientierung bestenfalls auf die Analyse, Auswahl und Benutzung von Programmsystemen für konkrete Problemklassen reduzieren oder im schlechtesten Fall zur reinen *Benutzerorientierung* [BHW85, S.22,S.25], [Hum05, S.48] als Applikationsschulung mutieren.

Diese Entwicklung kann aus Sicht der Informatik nicht unterstützt werden. Aber die Informatiker waren nicht die Einzigen, die an der curricularen Konzeption einer derartigen Grundbildung gearbeitet haben. Dies wird beispielsweise an dem Kreis deutlich, die zu einer *Curriculum-Konferenz* in Kiel geladen wurden. Diese bestand aus Schülern, Lehrern und Personen aus der Öffentlichkeit. Die Fächer der Lehrer reichten von Deutsch über Mathematik, Geschichte bis Philosophie, nur einige Teilnehmer hatten eine Aus- oder Vorbildung in Informatik. Diese Vorgehensweise ist sicher ein Novum in der Geschichte von Schule, zumindest in einem Fach des mathematisch-naturwissenschaftlich-technischen Feldes und sehr bedenklich. Ihre eigene Berechtigung haben Sozialwissenschaftler nicht angezweifelt:

“Im Falle der Informationstechnologie läßt sich ein gewaltiger Unterschied im Ausmaß der Informationen und Erfahrungen feststellen. Die Kenntnisse im technischen Bereich sind recht gut, wogegen die Auswirkungen im sozialen und pädagogischen Bereich unzureichend sind. Aufgrund dieses Ungleichgewichts wird befürchtet, daß die informationstechnische Grundbildung zu Lasten der sozialen Problematik geht, da die Akzente sehr stark zugunsten der technischen Grundlegung und ihrer methodischen Vermittlung gesetzt werden.

Für die Überwindung der Einseitigkeit der informationstechnischen Grundbildung schlägt H. Simmon die Einbettung von Sozialwissenschaften im naturwissenschaftlichen Unterricht vor. Er betrachtet den vermuteten Gegensatz zwischen wissenschaftlichen und technischen Wissen zur humanistischen Ausbildung als ein Vorurteil, weil nach einer traditionellen Haltung 'die wissenschaftlichen Fächer hauptsächlich unterrichtet werden, um bestimmte Fertigkeiten hervorzubringen und zu entwickeln. Die humanistischen Fächer werden dagegen unterrichtet, um das Verstehen zu lernen.'“ [Bik90, S.41]

Eine solche Einstellung macht es natürlich leicht, an curricularen Entscheidungen teilnehmen und beschliessen zu wollen, auch wenn man von der Thematik des entsprechenden Faches nichts oder nur wenig versteht.

*Knobelsdorf* hat vier Modell-Kategorien vorgestellt, in die ein Nutzer von informatischen Systemen eingeteilt werden kann. Es sind dies: *Ausprobieren, Anwenden, Verändern, Erzeugen* [Kno11, S.137] Jemanden, der sich in den unteren Stufen aufhält, sind die oberen Stufen

nicht bewusst [Kno11, S.137]. Er oder sie kann daher kein Bedürfnis für das Erlangen oder Kennenlernen dieser Stufen entwickeln. Allein aus diesem Grunde konnte der curriculare Prozess mit den beteiligten Personen über eine Grundbildung Informatik nicht gelingen, da die Wahrnehmungen der Teilnehmer zu unterschiedlich waren. Diese Form der informatischen Unterweisung gilt daher als gescheitert<sup>17</sup> [Bru09], [Bre03], [Pas96].

**Medienkunde** Zur Zeit des Aufkommens der 'Grundbildung Informatik', später dann 'Informations- und kommunikationstechnologischen Grundbildung', drang die Eigenschaft der (persönlichen) Computer als *Werkzeuge* in das Bewusstsein der relativ informatikfernen Pädagogen und Bildungspolitiker. Etwa zwanzig Jahre später wird Vielen bewusst, dass der Computer auch *Medium* sein kann. Eine Aktualisierung der Konzepte findet daher in Form von *Medienkunde* statt. Auf bundesstaatlicher Ebene ist entsprechend ein Konzept von einer 'Expertenkommission' erstellt worden mit dem Titel: „Kompetenzen in einer digital geprägten Kultur, Medienbildung für die Persönlichkeitsentwicklung, für die gesellschaftliche Teilhabe und für die Entwicklung von Ausbildungs- und Erwerbsfähigkeit!“ [Bun10] Viele der Argumentationen für und wider die *ITG* könnten hier wiederholt werden. Eine umfassende Darstellung und Würdigung der Medienkunde kann und soll hier nicht geleistet werden. Ein Ersatz für einen Informatikunterricht stellt sie auf keinen Fall dar.

### 5.5.3 Kontextorientierter Unterricht

Die Erfahrungen mit der Grundbildung Informatik und die mögliche Fortsetzung über nur wenige Zellen der Bildungsmatrix füllen Kontextbeispiele mit der Idee nach *Informatik im Kontext* verdeutlichen, dass ein Ansatz, wie Unterricht organisiert werden soll, nicht gewährleisten kann, dass die angestrebten Kompetenzen tatsächlich erreicht werden. Übersehen wird dabei, dass es auch ähnliche Möglichkeiten bei anderen Organisationsformen gibt.

**Projektunterricht** Aus Sicht der Informatik sind Projekte eine sehr sinnvolle Unterrichtsform, da bei aller Unterschiedlichkeit von Projekten in der Schule und in der realen Welt grose Softwareprojekt nur auf diese Weise realisiert werden können und Schüler bei der Umsetzung beispielsweise einer Idee bis zu einem Softwareprodukt sehr viel lernen können [Leh85], [Fre83], [Fis83]. Dennoch lässt sich nicht der gesamte Unterricht in Informatik in Projektform durchführen und es muss begründet sein, welche Ziele und Kompetenzen mit dieser Unterrichtsform erreicht werden sollen.

**Projektorientierter Unterricht** Gelegentlich möchte die Lehrkraft einige Elemente des Projektunterrichtes beibehalten, aber nicht ein vollständiges Projekt durchführen. Dies erfüllt dann unter Umständen die Kriterien eines projektorientierten Unterrichts. *Ralph Hepp* und *Wolfgang Münzinger* stellen für einen derartigen *projektorientierten Unterricht* folgende Merkmale auf: „

- Es werden Themen bearbeitet, welche die Schüler wirklich interessieren bzw. mit denen sie sich identifizieren können (Thema).

<sup>17</sup><http://www.heise.de/tp/artikel/16/16082/1.html>, letzter Zugriff: 27.7.2012



- Das Prinzip der Schülerorientierung wird wesentlich konsequenter verfolgt als im herkömmlichen Unterricht. Die Schüler arbeiten zunehmend selbstständig, eigenverantwortlich und produktiv (Schüleraktivität).
- Es wird in Gruppen gearbeitet, so dass Kooperation und Kommunikation zwingend erforderlich sind (Gruppenarbeit).
- Die konkrete Realisierung erstreckt sich über einen längeren Zeitraum und ist immer produktorientiert (organisatorischer Rahmen).
- Der Lehrer versucht, durch eine partnerschaftliche und moderierende Führung die Arbeit zu koordinieren und durch Würdigung der Zwischenergebnisse und des Resultates die Schülerleistungen zu fördern (Präsentation).“ [HM01]

Gegenüber einem Projekt liegen lehrerseits Einschränkungen bezüglich der Themenwahl, Materialbeschaffung, Gruppenstruktur, Wege zur Erarbeitung notwendigen Vorwissens, Zeitorganisation, Umfang etc. vor. Im Gegensatz zu einem Projekt, das zumindest öfters fachübergreifend ist, findet projektorientierter Unterricht ausschliesslich im Unterricht eines Faches statt. *Münzinger* betont bezüglich eines solchen Unterrichtes in der Mathematik, dass der projektorientierte Unterricht damit die übliche ‚Praxisferne‘ verhindert, indem er „(a) auf die Interessen und Erfahrungswelt des Schülers eingeht, (b) mit seinen Anwendungsaufgaben gesellschaftliche Praxis realistisch und exemplarisch behandelt [...]“. [Mün76, S.77]

Gelingt es dem Lehrer, eine derartige Unterrichtsform nicht nur einmal durchzuführen, sondern zu einem regelmässigen Bestandteil seines Unterrichtes zu etablieren, kann man den Gesamtunterricht projektorientiert bezeichnen. Der Projektunterricht und der projektorientierte Unterricht sind damit die ‚obere Grenze‘ für Kontexte.

**Problemorientierter Unterricht** Die ‚untere Grenze‘ stellt der problemorientierte Unterricht dar. Nach *Aebli* ist er beschrieben mit:

„Es gibt einen *Geist des Problemlösens*, der wichtiger als alle Techniken ist. Ihn sollte der Lehrer immer wieder zu wecken suchen, indem er mit den Schülern zusammen Probleme löst. Der Geist ist derjenige der Rationalität, des klaren Sehens von etwas, des widerspruchsfreien, gesetzmässigen Schließens. [...]

Die Fähigkeit, Probleme zu lösen und schöpferisch zu forschen und zu gestalten, reift nicht einfach heran. Es genügt nicht, die Schüler sich selbst zu überlassen, sie selbständig arbeiten zu lassen, damit kreative Leistungen entstehen. [...]

Dies bedeutet, daß die erste Schule des Problemlösens *das Gespräch mit dem Klassenlehrer* ist, nicht die Gruppenarbeit und nicht die individuelle Arbeit. Nichts vermag dieses gemeinsame Nachdenken von Lehrer und Klasse zu ersetzen, keine Form der selbständigen Arbeit und kein Trainingsprogramm für Kreativität.“ [Aeb76, S.272]

*Martin Wagenschein* sagt ergänzend mit Bezug auf den Philosophen und Psychiater *Karl Jaspers*: „Nichts setzt unsere Gedankenmassen so sehr in Bewegung wie das Gespräch. ‚Wechselgespräch ist eine Grundform, in der die Wahrheit an den Tag kommt‘ (Jaspers). Das Gespräch macht auch den von sich aus Trägern oder Gleichgültigen zu einem Beteiligten und Angesteckten; er wird ‚hineingezogen‘.“ [Wag71, S.125]

Diese Aussagen gelten trotz oder gerade wegen der von *Michael Winterhoff* in [Win08] beschriebenen Veränderungen in der Kindheit, durch die es unter anderem immer schwieri-

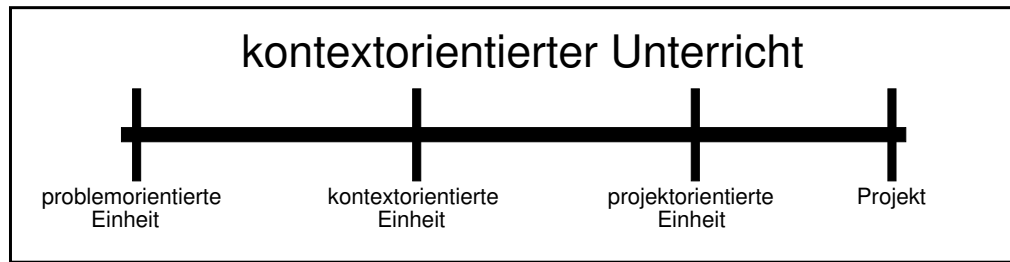


Abbildung 5.12: kontextorientierter Unterricht

ger wird, aufgrund der gestiegenen Anzahl von Störungen und Undiszipliniertheiten in einer Klasse ein gemeinsames Gespräch zu führen und die Versuchung besteht, durch ein Erarbeiten von Arbeitsblättern in Einzel- oder Gruppenarbeit dieser Problematik zu entgehen.

**Kontextorientierter Unterricht** Die von der Fachdidaktik Chemie vorgelegten Kriterien für einen *Unterricht im Kontext* sind sehr eng. Gelegentlich ist die Länge einer solchen Einheit kürzer oder länger, eine Vertiefungsphase lässt sich nur bedingt anschliessen oder es gibt praktisch keine Neugier- und Begegnungsphase. Vergleichbar mit dem projektorientierten Unterricht definieren wir eine solche Unterrichtseinheit als *kontextorientiert*.

**Definition 1** Eine Unterrichtsreihe wird als **kontextorientierte Unterrichtsreihe** bezeichnet, wenn sie folgende Kriterien erfüllt:

- Eine kontextorientierte Unterrichtseinheit versucht, zumindest drei der vier Phasen (Begegnungsphase, Neugier- und Planungsphase, Erarbeitungsphase, Vernetzungs- und Vertiefungsphase) eines Unterrichts im Kontext zu realisieren.
- Die Länge kann dabei die Länge eines üblichen Kontextes von ca. 10 Stunden unter- oder überschreiten. Die Länge sollte jedoch nicht kleiner als 3 Unterrichtsstunden sein.
- Je nach Unterrichtsverlauf und Interesse der Schüler kann die kontextorientierte Unterrichtseinheit zum projektorientierten Unterricht oder zum Projekt ausgedehnt werden.

Wie oben erwähnt, enthält ein guter Unterricht nicht nur derartige kontextorientierte Einheiten, sondern auch Projekte, Projektorientierung und natürlich Problemorientierung. Wir ergänzen daher die obige Definition zu einer für den *kontextorientierten Unterricht*:

**Definition 2** Ein **kontextorientierter Unterricht** wird bestimmt aus kontextorientierten und projektorientierten Einheiten sowie Projekten (siehe Abbildung 5.12). Unterricht, der auf diese Weise nicht gestaltet werden kann, wird überwiegend problemorientiert organisiert.

## 5.5.4 Fazit Informatik und Kontext

In den letzten 30 Jahren ist ein Bestreben nachweisbar, den bisherigen oft fachsystematisch orientierten Unterricht in Mathematik, Naturwissenschaften und Informatik ,näher an den

Schüler' heran zubringen. Die *Lebenswirklichkeit* ist als Leitgedanke in vielen dieser Konzepte enthalten. Dieses folgt aus den konstruktivistischen Positionen, die diesen Ideen gemeinsam als lernpsychologische Grundlage dienen. Die konkrete Umsetzung kann je nach Unterrichtsentwurf unterschiedlich sein. Die ideale Verbindung von Lebenswirklichkeit mit Schule stellt der Projektunterricht dar, der in seiner Durchführung auch die meisten Veränderungen in Organisation von konkretem Unterricht und Schulorganisation bedeutet und von daher sich bis heute in der Schulwirklichkeit nicht hat durchsetzen können.

## 5.6 Ein Blick über die Grenzen

Die Situation für das Schulfach Informatik ist in fast allen Ländern der Erde vergleichbar (schlecht). Nur wenige Länder haben einen verpflichtenden Unterricht in den höheren Jahrgängen, die unserer Sekundarstufe II vergleichbar sind, etabliert [SGEHV05, BAL10, Yah10, S.25-27], [TDJ<sup>+</sup>06, S.2]. Wie in Deutschland ist ein Unterricht in der Sekundarstufe I noch weniger etabliert als in der Sekundarstufe II. Es ist sinnvoll, Inhalte und Strukturen des Unterrichtes in anderen Ländern zu betrachten, um geeignete alternative Ideen für die Weiterentwicklung des Informatikunterrichtes in Deutschland übernehmen bzw. an unsere Verhältnisse anpassen zu können.

### 5.6.1 USA

In den mittleren Jahrgängen vergleichbar unserer Sekundarstufe I sieht es nicht viel besser als in Deutschland aus. Zumeist finden in den USA nur vereinzelt *computer science-course* [SSW10] statt, die zumeist nicht konsistent in einem Curriculum strukturiert sind. In Neuseeland [BAL10] wird derzeit an einem landesweiten Programm zur Einführung von *computer science* gearbeitet.

Die konkrete Unterrichtssituation ist teilweise unabhängig von guten existenten Programmen. Die ACM hat für die USA ein Curriculum *A Model Curriculum for K-12 Computer Science* [TDJ<sup>+</sup>06] entwickelt, das eine wünschenswerte Ausbildung in Informatik in den Schulen beschreibt.

#### Model Curriculum K-12

Als Anspruch für die Zielgruppe wird formuliert:

*“The goals of a K12 computer science curriculum are to:*

- 1. introduce the fundamental concepts on computer science to all students, beginning at the elementary school level.*
- 2. present computer science at the secondary school level in a way that would be both accessible and worthy of a curriculum credit (e.g., math or science).“*

*... [TDJ<sup>+</sup>06, S.2]*

Informatik soll demnach kein Fach für Teilgruppen welcher Art auch immer sein, sondern jede Schülerin und Schüler soll in der Schule Informatikunterricht erhalten. In diesem Curriculum

werden drei Stufen unterschieden. Die unterste Stufe ist in etwa mit den Zielen der ITG zu umschreiben, während die Stufen 2/3 — *level II/III* — mit einem neuen Begriff bezeichnet werden, der *IT fluency*:

*“A recent National Academy study (National Research Council, 1999) defines an idea called IT fluency as something more comprehensive than IT literacy. Whereas IT literacy is the capability to use today’s technology in one’s own field, the notion of IT fluency adds the capability to independently learn and use new technology as it evolves (National Research Council, 1999) throughout one’s professional lifetime. Moreover, IT fluency also includes the active use of algorithmic thinking (including programming) to solve problems, whereas IT literacy is more limited in scope.“ [TDJ<sup>+</sup>06, S.3]*

Ein nicht unerheblicher Teil dieser *IT-fluency* soll bis Ende des Jahrganges 9 [TDJ<sup>+</sup>06, S.11-13] erreicht sein. Das Programm kann umschrieben werden als die schulischen Notwendigkeiten, um dem *computational thinking* nach Wing [Win06] gerecht zu werden.

### **CSTA-Strands**

In einem aktuellen Entwurf für eine überarbeitete Fassung dieses Curriculums wird diese *IT-fluency* in *strands* eingebunden:

*„We address these concerns by distinguishing five complementary and essential strands throughout all three levels in these standards. These strands are: computational thinking; collaboration; computing practice; computers and communication devices; and community, global and ethical impacts. These strands not only demonstrate the richness of computer science but also help organize the subject matter for students so that they can begin to perceive of computer science as more engaging, relevant, and more than a solitary pursuit.“ [Tuc11, S.14]*

Grafisch wird das entsprechend Abbildung 5.13 dargestellt.

Die Strands in diesem Entwurf haben eine gewisse Ähnlichkeit mit den in dieser Arbeit im Kapitel 6 vorgestellten *roten Fäden*. Sie entsprechen allerdings mehr den *Principles* von Denning [Den03], da sie mehr Sichten auf die Informatik darstellen und nicht ein Ordnungs- und Strukturierungsprinzip aus Sicht des Schülers wie die *roten Fäden*.

### **CSTA CS K-8**

Die überarbeiteten Standards haben den Anspruch, dass Informatik in allen Jahrgängen der Schule entsprechend umgesetzt wird. Dies verdeutlicht auch eine Broschüre, die mit dem Schwerpunkt *K-8* Beiträge zum Thema Informatik in den unteren Jahrgängen für Lehrer und Öffentlichkeit zusammenfassend darstellt. *Irene Lee* fordert darin im Sinne der CSTA Schulen, Lehrer und Administration gleichermassen zu einer Umsetzung dieser Standards auf:

*“By implementing these standards, schools can introduce the principles and methodologies of CS to all students at all stages of their learning, whether they are college bound or workplace bound. We envision that teachers of K8 students will use the standards to implement CS activities in the context of other subjects or within stand-alone introductory CS courses. We hope that administrators and policy makers will use the standards to*

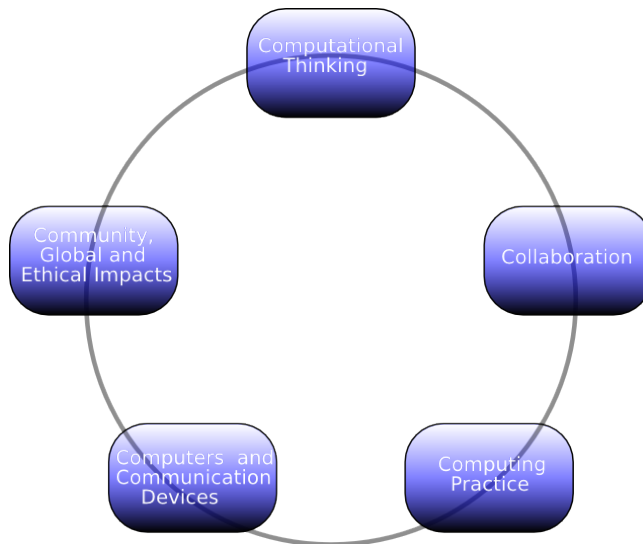


Abbildung 5.13: Die fünf *Strands* des *IT-fluency* im Entwurf für ein Curriculum des CS-TA [Tuc11, S.14]

*understand the importance of CS education as part of the intellectual development of all students and that all who read the standards will see the important linkages between CS and innovation across disciplines.“ [Phi12, S.10]*

## 5.6.2 Grossbritannien

Ähnlich sieht die Situation in Grossbritannien. Auch dort ist Informatik als Schulfach fast unbedeutend. Eine Arbeitsgruppe *Computing at School Working Group* des Fachverbandes *British Computer Society (BCS)* hat im Frühjahr 2012 ein Memorandum *Computer Science as a school subject – Seizing the opportunity* [Com12b] veröffentlicht und die Bedeutung eines Schulfaches 'Computer Science' betont. Dort heisst es: *“Computer Science is the fourth science. The Government is now encouraging every good school to offer Computer Science as part of their curriculum, from primary school onwards.“* [Com12b, S.3]

Inhaltlich berufen sich die Autoren dieses Aufrufes auf das von Wing beschriebene *computational thinking* [Win06]

*“Computational thinking is transforming all aspects of our society. Many subjects teach problem-solving skills to some degree, but Computer Science develops a particularly systematic and deep approach to thinking about complex problems, often called 'computational thinking'. Computational thinking is the process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes. Pupils learn to think about the same problem at many levels of abstraction, and to recognise that a single solution may apply to many other apparently different problems. More profoundly, thinking about other disciplines through a computational lens has radically changed the way those subjects are studied, whether physics or biology, psychology or economics. For example, viewing biological processes*

*as computational systems that process information has led to fundamental new insights in understanding disease that would not have been obtained through traditional thinking. Computational thinking has, indeed, led to whole new disciplines such as bioinformatics, and all scientists now need a core understanding of this kind of thinking.*

*Computational thinking is made concrete in programming. Programming takes computational thinking skills and empowers pupils to take charge of computers and create new software of their own, rather than simply to consume things made by others. [...] To take an analogy from mathematics, every child should understand algebra, be capable of abstracting appropriate problems into algebraic expressions, and be able to solve simple algebraic equations. In the same way, in the computational sphere every child should be able to construct elementary algorithms in programmatic form that encapsulate simple ideas and concepts. Programming is a way of expressing creativity, of communicating and sharing ideas, just as mathematics does in a different area of discourse.“ [Com12b, S.9,10]*

Konsequenterweise wird Informatik für alle Schüler gefordert:

*“Every student should encounter elementary Computer Science concepts, from Key Stage onwards, whether or not they intend to specialise in Computing. [...]*

*There is a strong analogy with the other sciences. We take it for granted that every student should learn the elementary concepts of (say) Physics at primary school and Key Stage 3, after which some will choose to study further at GCSE. Some will choose to take an A-level, and some of them will go on to study Physics at University. Moreover, there is a range of choices at each level, varying in the level of intellectual demand and vocational emphasis. Exactly the same pattern should apply to Computer Science.“ [Com12b, S.12]*

Zusätzlich hat die Arbeitsgruppe noch ein Curriculum für Schulen vorgestellt [Com12a] und unterscheidet bewusst Informatik von der ITG: *“Computer Science and ICT are complementary subjects.“ [Com12b, S.11]*

### 5.6.3 Niederlande

Als Kontrast zu den ‚grossen‘ angloamerikanischen Ländern soll hier noch eine kurze Darstellung der Situation in einem kleineren, bildungsmässig relativ autarkem Land, den Niederlanden, gegeben werden. In den Niederlanden spielte das Schulfach bis Anfang des neuen Jahrtausends praktisch keine Rolle. 1998 wurde es als Wahlfach in den Schulen der Bildungsgänge VWO und HAVO (in etwa vergleichbar mit dem Gymnasium und der Realschule) eingerichtet<sup>18</sup> und kann seit 2007 als ‚Profilwahlfach‘ anstelle von Biologie oder von ‚Natur, Leben & Techniek (NLT)‘ im Umfang von 320 Stunden (HAVO) bzw. 320 Stunden (VWO) gewählt werden<sup>19</sup>. Für beide Schulformen gilt: *„Zowel voor HAVO als VWO is sprake van een kernprogramma van ongeveer 200 studielasturen.“<sup>20</sup>*

<sup>18</sup><http://www.informaticavo.nl/over-het-vak>, letzter Zugriff: 6.8.2012

<sup>19</sup>[www.informaticavo.nl/bestand/pdf/informaticadocenten.doc](http://www.informaticavo.nl/bestand/pdf/informaticadocenten.doc), letzter Zugriff: 6.8.2012

<sup>20</sup>Deutsche Übersetzung: Sowohl für die HAVO als auch für die VWO ist Sprache eines der Kernprogramme von ungefähr 200 Stunden.

Das Schulfach Informatik hat in den Niederlanden nach einer sehr späten Institutionalisierung an Bedeutung und rechtlicher Stellung in der Schullaufbahn gewonnen. Es bleibt zu hoffen, dass die Befürchtungen von *Jan Timmers* vom September 2001 nicht doch noch Realität werden: „[...] *het vak informatiekunde op veel scholen [is] verworpen tot een knoppencursus computergebruik. Het gevolg daarvan is dat de status van het vak tot het nulniveau daalt [...]. Eerst niks doen, dan de conclusie trekken dat er niks wordt gedaan en vervolgens het vak opheffen, zodat er niks meer kan worden gedaan.*

*Als er geen acties ondernomen worden tot het stimuleren van informatica onderwijs op de scholen is datzelfde scenario ook voor het vak informatica te voorzien.*<sup>21 22</sup>

Eine geeignete Unterstützung für Schüler ist als Information des Schulfaches in HVVO und MBO im Internet verfügbar.<sup>23</sup>

## 5.6.4 Weitere Länder

In einer Vergleichsstudie aus dem Jahre 2007 an der TU Dresden<sup>24</sup> wurde untersucht, inwiefern in verschiedensten Ländern *ITK-Lerninhalte* in der Schule integriert sind. Es handelt sich allerdings nur um eine Auswahl von Ländern. So fehlen unter anderen auch die USA und Grossbritannien. Insofern ist die Repräsentativität dieser Studie fraglich. Es wurde zudem nicht wesentlich zwischen *informatischen* und *ITG*-Inhalten unterschieden.

In der Zusammenfassung bezüglich der Sekundarstufen wird dort festgestellt: „Ab der Sekundarstufe werden IKT dann oft auch als eigenes Fach unterrichtet. Weiterhin wird meist vorgeschrieben, dass dies durch den Einsatz im Unterricht in anderen Fächern oder für die Durchführung fächerübergreifender Projekte zu ergänzen ist. Diese Tendenz ist im Sekundarbereich II noch stärker ausgeprägt als im Sekundarbereich I. Nur in 2 europäischen Staaten (Tschechische Republik und Ungarn) werden IKT ausschließlich als eigenes Fach unterrichtet. Im Gegensatz dazu werden sie in Schweden und Norwegen im Sekundarbereich I ausschließlich als Instrument für den Unterricht in anderen Fächern eingesetzt.“

Betrachtet man die darauf folgende Tabelle genauer, so fällt auf, dass diese Aussagen sehr schmeichelhaft sind. Bis auf ganz wenige Ausnahmen wie Litauen, Estland, Polen und Ungarn gibt es keinen verpflichtenden Unterricht. In vielen Ländern werden mehr oder weniger ausgeweitete ITG-Kenntnisse vermittelt und sehr wenig ausgesprochener Informatik-Unterricht angeboten.

<sup>21</sup><http://www.informaticavo.nl/archive/863-Nieuwsbericht+15-10-2001/bestand-downloaden>, letzter Zugriff: 6.8.2012

<sup>22</sup>Deutsche Übersetzung: Das Fach Informationskunde ist an vielen Schulen zu einem Klickkurs Computergebrauch geworden. Erst nichts tun, dann die Schlussfolgerung ziehen, dass nichts getan wurde und darauf folgend das Fach aufgeben, sodass nichts mehr getan werden kann. Wenn keine Aktionen für die Unterstützung der informatischen Unterweisung gestartet werden, ist dasselbe Szenario auch für das Fach Informatik zu erwarten.

<sup>23</sup><https://files.itslearning.com/data/1149/8/>, letzter Zugriff: 6.8.2012

<sup>24</sup>[http://tu-dresden.de/die\\_tu\\_dresden/fakultaeten/fakultaet\\_informatik/smt/dil/ib/laendervergleich/index\\_html](http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_informatik/smt/dil/ib/laendervergleich/index_html), letzter Zugriff: 6. 8.2012

## 5.6.5 Fazit Ein Blick über die Grenzen

Die Ausbildungssituation in Informatik ist bezüglich der vorliegenden Informationen als kritisch zu beurteilen. Es existieren höchstens Ansätze einer zielgerichteten und langfristigen Planung und Realisierung einer informatischen Bildung. Fast überall wird die Bedeutung der Durchdringung der Gesellschaft in Arbeitswelt, Privatbereich und Freizeit mit informatischen Methoden und Produkten erkannt und als notwendige Herausforderung angesehen. Geht es um die Umsetzung derartiger Ansprüche in der Schule, so ist davon letztendlich nicht sehr viel zu sehen. Die Informatikdidaktiker im In- und Ausland stehen daher vor vergleichbaren Aufgaben.

## 5.7 Die Bildungsstandards Informatik

Die Ergebnisse der internationalen empirischen Vergleichsteste waren für die Bundesrepublik ernüchternd [Kli07], [KAH<sup>+</sup>10a]. Als eine Konsequenz wurden auf Bundesebene für die in den vergleichenden Testen untersuchten Fächer sowie zusätzlich für die 1. Fremdsprache Englisch und Französisch bundeseinheitliche *Bildungsstandards* [KMK04a] ... [KMK05c] formuliert.

**Geschichte** Dies führte zu einer sehr umstritten geführten Diskussion unter den im *Königsteiner Kreis* zusammenarbeitenden Fachdidaktikern aus Schulen und Hochschulen, ob es nicht gerechtfertigt und sinnvoll sei, auch für das Fach Informatik Bildungsstandards zu formulieren. Auf der einen Seite wurde bemängelt, dass die Fächer, die nicht mit Bildungsstandards versehen waren, als Fächer zweiter Klasse angesehen werden könnten und dies zumindest bei der Informatik den Stellenwert in der Schule wahrscheinlich noch verschlechtern würde, andererseits wurde auch eine mögliche didaktische Einengung und Normierung für das Schulfach befürchtet. Es gab aber auch Stimmen, die gerade eine stärkere Vereinheitlichung des Faches Informatik in der Sekundarstufe I als Chance ansahen. Schliesslich einigte man sich darauf, den Versuch zu wagen, derartige Bildungsstandards zu formulieren. Es war insofern ein Wagnis, da im Gegensatz zu den etablierten Fächern bis heute kein durchgängig realisiertes Schulfach Informatik in der Bundesrepublik existiert und daher diese Standards eher ein Anspruch für ein künftig zu schaffendes Pflichtfach darstellen konnten. Es war deshalb auch ein Wagnis, weil im Gegensatz zu den etablierten Fächern nicht eine von der Administration eingesetzter Kreis von Experten einen derartigen Entwurf formulierte, sondern in diesem Fall eine doch sehr grosse Gemeinschaft aus Freiwilligen aus Schule und Hochschule aus Deutschland und sogar aus Österreich und der Schweiz über mehrere Jahre an dieser Diskussion teilnahmen und diese *Bildungsstandards Informatik* entwickelten. Die zentralen Diskussionen wurden über mehrere Jahre hinweg auf den jährlich stattfindenden Tagungen des Königsteiner Kreises geführt, die zentrale Koordination — ohne die es bei einem solchen Projekt nicht geht — lag bei *Hermann Puhmann* aus Erlangen, aber es fanden auch Diskussionen ausserhalb des Königsteiner Kreises beispielsweise auf den Informatiktagen in den verschiedenen Ländern statt.



**Empfehlung der GI** Um diesen Bildungsstandards Informatik einen grösseren öffentlichen Stellenwert zuzubilligen, wurden sie der Gesellschaft für Informatik als Beschlussvorlage vorgelegt und dort am 24. Januar 2008 als Empfehlung vom Präsidium der GI beschlossen. Für die schulische und sonstige Öffentlichkeit wurde inzwischen eine Webseite eingerichtet<sup>25</sup>.

**Bildungspolitische Bedeutung** Diese Bildungsstandards Informatik stellen damit ein Unikat in der Bildungslandschaft der Bundesrepublik Deutschland dar. Zum ersten Mal ist gelungen, was von den bildungspolitisch Interessierten und Forschern immer wieder gefordert wird. „In dieser Hinsicht ist Schulentwicklung nicht eine Sache pädagogischer Autoritäten, auch nicht eine Sache der Entscheidung von wenigen über viele: alle Beteiligten haben ein Beurteilungs- und Mitwirkungsrecht.“ [Bik90, S.70]

**Ansatz** Im Gegensatz zu den offiziellen Bildungsstandards in den ‚PISA‘-Fächern formulieren die Bildungsstandards Informatik keinen Regelstandard, sondern einen *Mindeststandard*, den jeder Schüler am Ende der Jahrgangsstufe 10 beherrschen soll [Arb08, S.2]. Begründet wurde dies in den Diskussionen in Königstein, weil damit für den ‚Abnehmer‘ deutlich ist, welche Kompetenzen von einem erfolgreichen Schüler am Abschluss seiner Schullaufbahn erwartet werden können. Bei einem definierten *Regelstandard* ist die eigentlich kritische Frage, welche Kompetenzen von einem Absolventen erwartet werden können, ungeklärt. Damit ist die Aussagekraft eines Abschlusses gefährdet.

Die Standards gehen davon aus, dass ein durchgehender Unterricht in Informatik vom Jahrgang 5 bis zum Jahrgang 10 realisiert ist. Das ist heute noch Utopie. Insofern stellen diese Standards einen noch ungelösten Anspruch dar. Sie bedürfen daher nach praktischer Erprobung in einer ungeklärten Zukunft einer Überarbeitung und Anpassung an die realen Verhältnisse. Dieses wird sicher vonnöten sein, da schon jetzt erkennbar ist, dass in diesen Standards aufgeführten Ziele nicht in einem realisierten Unterricht nicht vollständig erreicht werden können. Die Standards enthalten konkret 151 Einzelpunkte. Bei einem Unterricht von einer Wochenstunde Informatik in 6 Schuljahren bedeutet dies, dass praktisch für jede neue Kompetenz im Schnitt 1,5 Schulstunden zur Verfügung stehen. Auch wenn man konzidiert, dass mehrere Kompetenzen vermittels des konkreten Unterrichtes gleichzeitig in einer Unterrichtsstunde verbessert werden können, so ist das nicht auf den Gesamtunterricht in 6 Jahren zu übertragen.

Die Bildungsstandards orientieren sich an den US-amerikanischen Mathematikstandards des NCTM (National Council of Teachers of Mathematics) [NTC00]. Diese gehen von pädagogischen *principles* aus, die sich aus den Ansprüchen einer demokratischen und menschenrechtsgemässen Gesellschaft ergeben: *Equity, Curriculum, Teaching, Learning, Assessment, Technology* [NTC00, S.11]. Gemeint ist damit, dass allen Schülern chancengleich ein in sich stimmiges Curriculum von ausgebildeten Lehrern mit aus Schülersicht sinnvollen Inhalten unterrichtet wird. Prüfungen sollen kein Faktenwissen, sondern Verstandeswissen testen. Der Mathematikunterricht soll sich dabei nicht der Anwendung moderner Techniken entziehen.

---

<sup>25</sup><http://www.informatikstandards.de/>, letzter Zugriff: 24.7.2012



Abbildung 5.14: Inhalts- und Prozessbereiche der Bildungsstandards Informatik [Arb08, S.11]

**Inhalte und Prozesse** Diese Ansprüche werden auf den Informatikunterricht übertragen und führen zu einer Formulierung und Darstellung der Kompetenzen auf zwei Ebenen: denen der Inhalte und Prozesse. Grafisch kann dies entsprechend Abbildung 5.14 dargestellt werden.

Mit der Ausformulierung dieser Kompetenzen in Form von Inhalten und Prozessen erfüllen diese Bildungsstandards Informatik weitgehend den Ansprüchen von *Wagenschein* an einen *Kanon*. Mit dem Austausch des Faches ‚Physik‘ durch ‚Informatik‘ liest sich das bei *Wagenschein* so:

„Dieser Kanon soll also *nicht ein Lehrbuch für das Kind* sein. Es ist eine Darstellung für den Lehrer, die ihm sagen möchte, wie die Informatik für das Kind von innen ungefähr aussehen könnte, ohne daß das hier Gegebene etwa der Wortlaut sein müßte, wie ihn der Lernende wissen sollte. Es zeichnet das Grundgefüge der — sozusagen — informatischen Landschaft ab, wie sie sich in Kopf und Herz des Lernenden eingelassen und aufgetan haben sollte. [...]

Dieser Kanon soll auch *nicht etwa ein kurzgefaßtes Lehrbuch für den Lehrer* sein, das für sich allein verständlich sein könnte. Er möchte für den schon einigermaßen Kundigen die Linien der schlichtesten Verbindung zeichnen, so wie man wünschen könnte, daß sie sich im Geiste des Lernenden allmählich ausbilden. Dieser Kundige sollte der Abiturient sein. Daß er es heute im allgemeinen nicht ist, hat zum Glück weder informatische noch psychologische Gründe, sondern nur historische. Es kann also noch einmal anders werden.“ [Wag71, S.233]

### 5.7.1 Die Bildungsmatrix

Dieser Kanon enthält neben den inhaltlichen auch prozessuale Ziele. Diese sind einerseits aus den Erkenntnissen der Psychologie bezüglich des Kognitivismus und des Konstruktivismus gefolgert, ergeben sich zusätzlich allerdings noch aus der fachlichen Struktur von Informatik. *Kommunizieren* und *Kooperieren* sind beispielsweise immanente Teile des Softwareentwicklungsprozesses, *Strukturieren* und *Vernetzen* sind notwendige fachliche Erfordernisse

		Prozessbereiche				
		Modellieren und Implementieren	Begründen und Bewerten	Strukturieren und Vernetzen	Kommunizieren und Kooperieren	Darstellen und Interpretieren
Inhaltsbereiche	Information und Daten					
	Algorithmen					
	Sprachen und Automaten					
	Informatiksysteme					
	Informatik, Mensch und Gesellschaft					

Abbildung 5.15: Die Standardmatrix für die Bildungsstandards Informatik [PV09]

		Prozessbereiche				
		Modellieren und Implementieren	Begründen und Bewerten	Strukturieren und Vernetzen	Kommunizieren und Kooperieren	Darstellen und Interpretieren
Inhaltsbereiche	Information und Daten	shell-Skripte				Kommandozeile
	Algorithmen	shell-Skripte				
	Sprachen und Automaten				Installation Fernzugriff	
	Informatiksysteme	Installation			Webserver Installation Fernzugriff	
	Informatik, Mensch und Gesellschaft					Webserver

Abbildung 5.16: Teilausgefüllte Standardmatrix [PV09]

beispielsweise bei der Modellierung eines Datenbanksystems. *Informatisch handeln* und *Informatik lernen* sind also sich ergänzende Tätigkeiten.

Damit liegt es nahe, unterrichtliche Themen nicht nur einzelnen Inhaltskompetenzen oder Prozesskompetenzen zuzuordnen, sondern diese in einer *Bildungsmatrix* entsprechend Abbildung 5.15 einzufügen.

Abbildung 5.16 verdeutlicht dies konkret an dem Thema *Betriebssysteme*. Im Verlaufe des Schullebens in Informatik werden einige Zellen der Bildungsmatrix mit Teilthemen besetzt. Diese Zellen werden teilweise auch von anderen Einheiten gefüllt. Am Ende der Schulzeit sollten alle Zellen zumindest mit einem Teilthema besetzt sein.

Dies bedeutet, dass für den Kanon der Inhalte und Prozesse informatischer Bildung geeignete Themen gesucht werden, die genau diese Zellen füllen. Dies muss und kann nicht nur auf eine Weise geschehen. Damit erhält der Lehrer und die Schüler einen Freiraum für den konkreten Unterricht der Lerngruppe. Dies verdeutlicht den Charakter der Bildungsstandards als Kanon der Kompetenzen und nicht als konkreter Unterrichtsleitfaden.

## 5.7.2 Fazit Bildungsstandards

Als eine Folgerung aus den Ergebnissen der PISA-Untersuchungen wurden in Deutschland in den Fächern Deutsch, 1. Fremdsprache, Mathematik und Naturwissenschaften nationale *Bildungsstandards* formuliert. Diese sind als Regelstandard definiert und sollen am Ende der Sekundarstufe I erreicht werden.

Unter Fachdidaktikern der Informatik setzte darauf hin eine Diskussion ein, ob es sinnvoll ist, auch für das Schulfach Informatik Bildungsstandards für die Sekundarstufe I zu formulieren, um die Relevanz des Faches für die Öffentlichkeit zu dokumentieren. Da es sich nicht um ein Dokument der Schuladministration handelte, sollte dieses Papier von der *Gesellschaft für Informatik (GI)* beschlossen werden, um den fachlichen Anspruch deutlich werden zu lassen. Nach einer etwa fünfjährigen Arbeit wurden diese Bildungsstandards Informatik von der GI im Jahre 2008 beschlossen [Arb08].

Diese Standards sind als Mindeststandards formuliert. Sie enthalten Inhalts- und Prozesskompetenzen, die in einer Bildungsmatrix dargestellt werden können. Mehr als erwartet, wurden diese Bildungsstandards in der Schulöffentlichkeit wahrgenommen, sodass diese heute in Deutschland von vielen Seiten anerkannt, die Basis für die Schulinformatiker darstellen.

## 5.8 Zusammenfassung und Konsequenzen

Informatik als Schulfach beinhaltet als Schulfach die Besonderheit, dass der Schüler auf der einen Seite in einem kommunikativen Prozess mit dem Lehrer wie in jedem anderen Schulfach steht, sich aber zusätzlich in einer Lehrerrolle gegenüber der Maschine befindet, mit der er in einer Sprache kommunizieren muss. In keinem anderen Fach wie Informatik kann sich daher die Schülerin bzw. der Schüler so gut selbst kontrollieren.

In diesem Verhältnis wird auch Informatik als Wissenschaft der Zeichenverarbeitung deutlich. Diese Bedeutung der Informatik wird im Unterricht durch die Verwendung von Programmiersprachen praktiziert. Die Auseinandersetzung mit Programmiersprachen stellt einen Grundpfeiler des Schulfaches dar. Diese kann nicht auf einer visuellen Darstellungsebene verbleiben. Das Erlernen und Praktizieren zumindest einer textuellen Programmiersprache ist daher unumgänglich.

Der Unterricht in Informatik soll sich wie in anderen Fächern an fundamentalen Ideen und Basiskonzepten orientieren. Dieses kann nur exemplarisch erfolgen. Die Relevanz der Informatik ist in der heutigen Gesellschaft so herausragend, dass darauf geachtet werden soll, dass vielfach inhaltliche Gegenstände aus verschiedenen Sichtweisen (Principles) betrachtet werden. Aus konstruktivistischer Sicht ist ein kontextorientierter Unterricht, der je nach Umfang und Bedeutung aus problemorientierten, kontextorientierten und projektorientierten Einheiten sowie aus Projekten gestaltet werden kann, anzustreben. Inhaltliche Basis sind dabei die Bildungsstandards Informatik.

## 5.8.1 Konsequenzen für das Schulfach Informatik

Aus den verschiedensten fachwissenschaftlichen Sichten ist informatisches Wissen sinnvoll. Auch aus bildungswissenschaftlicher Sicht ist ein Schulfach Informatik begründbar. Das *didaktische Dreieck der Informatik* verdeutlicht dabei den besonderen und einzigartigen Charakter des Faches Informatik, das geradezu als Wissenschaft der Zeichenverarbeitung den besonderen Stellenwert von *Sprache* im Denken und Handeln reflektieren kann.

### Folgerungen für die Sekundarstufe I

Nach Bruner ist jeder sinnvoll zu unterrichtende Lerngegenstand in jeder Jahrgangsstufe entsprechend aufbereitet unterrichtbar. Nach van Hiele müssen beim Erschliessen eines Thematik immer eine Folge von Stufen erklommen werden, die alle mit einer dieser Stufe entsprechenden Sprache versehen sind. Diese Stufen können auch nicht dann übersprungen werden, wenn wir den Unterricht in einer lernpsychologisch späteren Phase nach Piaget beginnen. Das bedeutet, dass versucht werden muss, mit Informatik relativ früh zu beginnen. Dafür reichen die bisherigen Angebote wie ITG oder zumeist wahlfreie Arbeitsgemeinschaften nicht aus.

Orientiert man sich an den Naturwissenschaften, bietet sich die Sekundarstufe I für einen derartigen Beginn an. Die naturwissenschaftlichen Fächer starten mit der Jahrgangsstufe 5. In den verschiedenen Schulformen werden diese teilweise fächerintegriert oder getrennt unterrichtet. Eine Fächerverbindung von Informatik mit einem anderen Fach ist alleine deswegen nicht anzuraten, da in einer Fächerintegration eines neuen Faches mit einem tradierten Fach die Ziele des neuen Faches mangels Erfahrung in Inhalten, Prozessen und Methoden die Lehrer dazu neigen, sich im Wesentlichen auf das tradierte Fach zu beziehen. Ein im Land Hamburg teilweise praktiziertes gemeinsames Unterrichten dieser Fachbereiche [Fre11b, Fre11a] in den unteren Jahrgängen 5/6 ist daher mit Aufmerksamkeit zu beobachten.

Anzustreben ist ein zumindest einstündiges verpflichtendes Fach Informatik durchgängig in der gesamten Sekundarstufe I, wie es die Bildungsstandards Informatik [Arb08, S.V] voraussetzen. Es ist sehr fraglich, ob dieser Zeitrahmen für die angestrebten Ziele bei der Vielzahl der in diesen Bildungsstandards aufgeführten Teilziele tatsächlich ausreichen wird. Die Forderung nach einem verpflichtenden Fach Informatik bedeutet, dass an anderer Stelle in der Stundentafel die Verpflichtung entsprechend gekürzt werden muss. Für dieses verpflichtende Fach müssen entsprechende Curricula entwickelt und erprobt werden. Mit den dort gemachten Erfahrungen sollten anschliessend die Bildungsstandards Informatik überarbeitet werden.

### Folgerungen für die Sekundarstufe II

Ein in der Sekundarstufe I begonnene Unterricht in Informatik sollte sinnvollerweise in der Sekundarstufe II fortgesetzt werden. Seit der Reform der gymnasialen Oberstufe 1972 [KMK72] ist Informatik in der Sekundarstufe II zugelassen. In den Bundesländern ist dies im Laufe der Jahre sehr unterschiedlich realisiert worden. In einigen Ländern ist Informatik nur als Grundkurs möglich, in anderen Ländern als Grund- und Leistungskurs [Wee07, Sta10, BOS77] zugelassen. Aufgrund der Bedingungen für das Abitur kann es für Schüler unattraktiv sein, dieses Fach zu wählen. Aufgrund der nachgewiesenen Bedeutung von Informatik für Beruf und Alltags- und Freizeitwelt ist dringend eine Reform nötig.

In ähnlicher Weise wie naturwissenschaftliches Wissen mit einer Mindestanzahl von Kursen nachgewiesen muss, ist dieses auch für informatisches Wissen zu fordern. Da sowohl Informatik wie Mathematik *Strukturwissen* vermittelt, könnte beispielsweise bei einer Verpflichtung von zwei Kursen in Informatik auf zwei Kurse in Mathematik innerhalb einer dreijährigen Sekundarstufe II verzichtet werden. Bei der Festlegung eines Curriculums für einen derartig verpflichtenden Informatikunterrichtes ist darauf zu achten, dass nicht nur algorithmisches Wissen sowie Programmiersprachen behandelt werden, sondern auch andere Themen wie beispielsweise *Netze*, *Datenbanken* und *(semi-)strukturierte Daten* in ausreichendem Umfang unterrichtet werden. Diese Themen sollten sinnvollerweise in *Kontexten* in Form der *roten Fäden* angeboten werden. Da ein verpflichtender Unterricht nach obiger Forderung nur in zwei Kursen stattfindet, muss die Verwebung natürlich bereits dort erfolgen.

# 6 Das Konzept der Roten Fäden

## Überblick

Die Praxis der letzten Jahr(zehnt)e zeigt, dass der Umfang der Inhalte und Kompetenzen, die durch den Informatikunterricht vermittelt werden, immer grösser wurde, der zur Verfügung stehende Zeitraum dagegen nicht grösser wurde. Die zu unterrichtenden Gegenstände wie beispielsweise Programmiersprache, HTML, Verarbeitung grosser Datenmengen in Datenbanken, Datenschutz und Sicherheit, Netze haben aus Sicht der Schüler nur einen recht lockeren Zusammenhang. Diese Ansammlung von Themen suggerierte eine gewisse Beliebigkeit und Austauschbarkeit. Diese Zustand wurde und wird durch aussenstehende Einflüsse und Unsicherheiten und nicht optimale Ausbildung der Kollegen verstärkt.

Es bedarf daher eine System der Strukturierung nicht so sehr — wie im Kapitel 5 ‚Schulfach Informatik‘ beschrieben — aus Sicht der Lehrer, sondern aus Sicht der Schülerinnen und Schüler. Diese müssen die unterschiedlichen Inhalte als einen Zusammenhang erkennen, aus dem sich zum Ende der Schullaufbahn Kompetenzen entsprechend der Bildungsmatrix nach den Bildungsstandards ergeben.

Diese Strukturierung stellen die in diesem Kapitel beschriebenen Roten Fäden dar. In diesen Roten Fäden werden nach unterschiedlichen Kriterien die unterrichtlichen Gegenstände organisiert. Diese Fäden werden im Unterricht nicht einer nach dem anderen systematisch ‚abgearbeitet‘, sondern in den Unterrichtseinheiten sollen nach Möglichkeit Gegenstände unterschiedlicher Fäden erarbeitet werden, die Roten Fäden werden verwoben. Damit wird entsprechend der Idee des Konstruktivismus bei den Schülern ein zusammenhängender ‚Teppich‘ aus Inhalten und Kompetenzen gewebt. Dies gilt auch für den roten Faden Programmiersprache, sodass die unberechtigte Kritik an programmiersprachlichen Kompetenzen im Unterricht der Sekundarstufe I verstummen muss.

Wir haben festgestellt, dass es sinnvoll ist, ein *exemplarisches Lernen* anzustreben. Die dafür auszuwählenden Inhalte und Kompetenzen sollen *fundamentale Ideen* und *Basiskonzepte* repräsentieren. Damit wir die Schüler in ihrer *Lebenswirklichkeit* abholen, ist der Unterricht am Besten in *problem-, kontext- oder projektorientierter* Form durchzuführen. Schliesslich soll noch die *Bildungsmatrix* der *Bildungsstandards* erfüllt werden. Erschwerend kommt noch hinzu, dass auch noch der Schüler einen *Zusammenhang* zwischen den unterschiedlichsten Einheiten im Verlaufe seines Schullebens in einem Fach erkennen soll. Da geht sicher auch manchmal dem besten Lehrer der Überblick verloren, warum er gerade das eine oder andere im Unterricht durchführt oder durchführen soll. Vielfältige Diskussionen über das Curriculum Informatik auch in der Sekundarstufe I in der Fachkonferenz Informatik der Fritz-Steinhoff-Gesamtschule und deren Umsetzung im konkreten Unterricht zeigen in der Tendenz eine immer grössere Abkehr von der ‚Abbilddidaktik‘. Nötig ist daher eine Strukturhilfe, die verhindern soll, dass der Unterricht ins Beliebig abrutscht, sondern die hilft, die Auswahl der Inhalte für die gewünschten Ziele zu begründen.

Aus der Reflektion des Unterrichts der letzten Jahrzehnte kann ein Unterricht mit *roten Fäden* gefolgert werden:

### 6.0.2 Definition der *Roten Fäden*

Eine Möglichkeit zur Strukturierung sind *rote Fäden* [PV09].

**Definition 3** Ein **roter Faden** ist eine Anordnung unterrichtlicher Gegenstände, die den folgenden Kriterien genügt:

- Die unterrichtlichen Gegenstände lassen sich einem gemeinsamen fachinhaltlichen (strukturellen oder thematischen) Zusammenhang zuordnen.
- Der gemeinsame fachinhaltliche Zusammenhang ist zu jedem Zeitpunkt aus Sicht der Schülerinnen und Schüler erkennbar und nachvollziehbar.
- Der gemeinsame fachinhaltliche Zusammenhang wird im Verlauf des Unterrichts aus verschiedenen Blickwinkeln oder in verschiedenen Kontexten dargestellt.
- Die Anordnung der unterrichtlichen Gegenstände durchzieht mehrere Unterrichtseinheiten.

In Abbildung 6.1 werden einige mögliche rote Fäden vorgestellt.

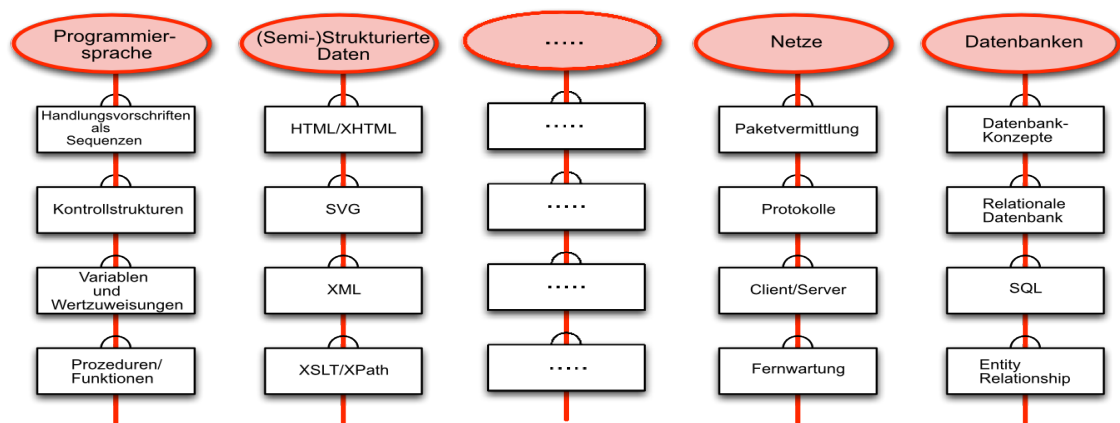


Abbildung 6.1: Beispielhafte *Rote Fäden* im Unterricht der Sekundarstufe I

Weitere Fäden könnten beispielsweise aus dem Bereich Multimedia — als gemeinsamer Faden oder getrennt nach Sound, Video, Animation — oder Grafik stammen oder sich mit Betriebssystemen, Steuern-, Messen und Regeln, Algorithmik, von-Neumann-Rechner etc. beschäftigen. Die Auswahl der konkreten *Roten Fäden* obliegt dabei dem Fachkollegen bei der Erstellung des Curriculums für seinen Kurs. [PV09]

Wenn ein Unterricht an Hand von Fäden realistisch sein soll, dann können und dürfen die Inhalte der Fäden nicht als Selbstzweck nacheinander mehr oder wenig vollständig unterrichtet werden, sondern müssen in konkreten verschiedenen Einheiten, besser noch in Kontexten, eingebunden sein. Ansonsten wären diese *roten Fäden* nichts anderes als Unterrichtseinheiten. Es soll erreicht werden, dass der Schüler erkennt, dass die unterschiedlichsten Themen in der Informatik nicht isoliert nebeneinander stehen, sondern einen Zusammenhang ergeben. Das bedeutet auch, dass die Auswahl eines Softwarewerkzeuges nicht allein von einer Einheit oder einem Kontext abhängig gemacht wird, sondern im Gesamtzusammenhang der verwendeten roten Fäden getroffen werden muss.



Die roten Fäden stellen nicht das einzige strukturelle Konzept dar. Andere Konzepte im Vergleich zu den roten Fäden werden im Kapitel 6.3 diskutiert. Die im Kapitel 5.6.1 vorgestellten *Strands* (Abschnitt: CSTA Strands) aus dem ACM-Curriculum *A Model Curriculum for K-12 Computer Science* [TDJ<sup>+</sup>06] haben zwar einige Ähnlichkeiten mit den roten Fäden, dürfen jedoch nicht mit diesen verwechselt werden.

## 6.1 Beispielhaftes Unterrichten mit *Roten Fäden*

Dieses Vorgehen soll an einem Beispiel verdeutlicht werden, das mehrere Jahre in dieser Weise unterrichtet worden ist. Der hier dargestellte Unterricht existiert nicht mehr, da der Wahlpflichtbereich II in den Jahrgängen 9 und 10 in der Gesamtschule des Landes NRW inzwischen ersatzlos gestrichen wurde. Es wurde daher in den letzten Jahren darauf verzichtet, neue Werkzeuge und neue Umgebungen zu nutzen, da die Abschaffung dieses Unterrichtes bevorstand.

### 6.1.1 Unterrichtliche Gestaltung

Im Anfangsunterricht des Wahlpflichtfaches Informatik im Jahrgang 9 einer Gesamtschule wurde die Installation des verwendeten Sprachensystems *COMAL* [Chr85, Haf88, RR88, Küm90a, Küm90b] durchgeführt. Es handelte sich um ein älteres System, das unter dem Betriebssystem DOS arbeitet und für das es kein automatisches Installationsprogramm gibt. Die Installation dieses Systems wird benutzt, um auf der sogenannten *Kommandozeile* zu arbeiten. Dieses führen die Schüler unter DOS (in einem Emulator unter LINUX) und ebenfalls unter Windows durch. Parallel lernen sie die syntaktisch ähnlichen Anweisungen unter UNIX/LINUX und DOS/Windows kennen. Es werden die dateibezogenen Kommandos einschliesslich grafischer Veranschaulichungen z.B. in einem Baumdiagramm behandelt und die Arbeit unter DOS/Windows mit relativ einfachen Batch-Dateien und unter Linux mit Script-Dateien geübt. Für diese Einheit werden in einem zweistündigen Kurs etwa zwei Monate benötigt<sup>1</sup>.

Benötigt wurden diese Kenntnisse ca. ein halbes Jahr später in einer projektartigen Einheit *Erstellung von HTML-Seiten* (Erstellung von Bewerbungsprofilen in HTML), damit die Dateien auf einen Server übertragen werden konnten. Zusätzlich mussten auf diesem Server die entsprechenden Dienste zur Verfügung gestellt werden. In diesem Zusammenhang wurden die grundlegenden Dienste wie z.B. *ssh* besprochen und praktiziert. Ebenso wurden das Verständnis von Serverdiensten als Prozesse in einem Computersystem erläutert. Ein tieferes Verständnis von Dateisystemen ist dann nötig, wenn die Serverprozesse auf die korrekten Dateiattribute der Dateien achten, die bei den vorhandenen Dateien (teilweise) aus Sicht des Servers nicht korrekt sind. In dieser Einheit sind die Fäden *Betriebssysteme* und *Netze* und *(semi-)strukturierte Daten* verwoben. Für diese stattfindende Einheit wurden etwa zwei bis drei Monate benötigt.

In einer weiteren Einheit *Kryptologie* wurden Verschlüsselungsverfahren und ihre Anwendungen behandelt. Zu den behandelten Anwendungen gehörten auch Fernzugriffstechniken

---

<sup>1</sup>In einem 9. Jahrgang kann ein Unterricht in Informatik auf diese Weise begonnen werden, um die benötigte Umgebung auch zu Hause bereitstellen zu können. In einem 6. Jahrgang ist dieses Vorgehen dagegen nicht anzuraten.

über beispielsweise `ssh`, `telnet`, `ftp`, `scp` und `sftp`. Kenntnisse aus mehreren Fäden wurden hier zum Verständnis nötig bzw. vertieft.

## 6.1.2 Einbettung in die Bildungsstandards

Eine sinnvolle Einbettung dieser Inhalte in die Bildungsmatrix kann nur gelingen, wenn man sie in dem Zusammenhang des konkreten Unterrichts wie oben dargestellt betrachtet. Im konkreten Beispiel des oben beschriebenen Unterrichts ergibt sich die in Abbildung 6.2 dargestellte Einfärbung der Bildungsmatrix. Hierbei füllen unterrichtlich begründete *Inhalte*, die entlang eines *roten Fadens* vermittelt werden, in strukturierter Form Teile der Bildungsmatrix aus.

		Prozessbereiche				
		Modellieren und Implementieren	Begründen und Bewerten	Strukturieren und Vernetzen	Kommunizieren und Kooperieren	Darstellen und Interpretieren
Inhaltsbereiche	Information und Daten	shell-Skripte				Kommandozeile
	Algorithmen	shell-Skripte				
	Sprachen und Automaten				Installation Fernzugriff	
	Informatiksysteme	Installation			Webserver Installation Fernzugriff	
	Informatik, Mensch und Gesellschaft					Webserver

Abbildung 6.2: Inhalts- und Prozessbereiche entlang des roten Fadens *Betriebssysteme* realisiert in verschiedenen Einheiten.

**Fortsetzbarkeit des roten Fadens** Wie bereits bei der Vorstellung der unterrichtlichen Gestaltung angesprochen wurde, orientiert sich der Unterricht nicht ausschliesslich entlang eines einzelnen roten Fadens; vielmehr ist es möglich und sinnvoll, mehrere Fäden miteinander zu verweben.

Diese Verwebung führt in natürlicher Weise dazu, dass der Unterricht einen konkreten Faden zu verschiedenen Zeitpunkten (wieder) aufgreift. Somit ergibt sich ein dem Bruner'schen Spiralcurriculum folgendes Vorgehen (Abbildung 6.3). Solch ein Vorgehen kann am Beispiel des roten Fadens *Betriebssysteme* so ausgeweitet werden, dass sich in der gymnasialen Oberstufe die (vertiefte) Behandlung der Themenbereiche *Client-Server-Strukturen* und *Modellieren von Netzwerkanwendungen* anschliessen lässt. Ähnliche Möglichkeiten der Fortsetzung ergeben sich entlang des roten Fadens (*semi-)*strukturierte Daten, bei dem sich ein Anschluss an die Themenbereiche *Endliche Automaten und Formale Sprachen* oder *Datenstrukturen* anbietet.

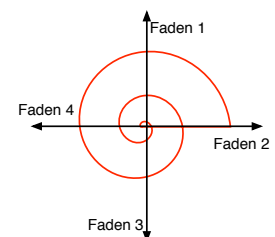


Abbildung 6.3: Spiralförmige Behandlung der roten Fäden.

Im oben beschriebenen Unterricht wurden nicht *kontextorientierte Unterrichtseinheiten* (nach Definition 1) integriert, da diese zur damaligen Zeit noch nicht geläufig waren. Diese sind zweifelsohne denkbar und sinnvoll und haben teilweise in ähnlicher Form stattgefunden. Beispielsweise wurde im Themenbereich der Kontext *Online-Banking* bei einigen Unterrichtsgruppen besprochen. Die Darstellung zeigt, dass es bei einer Gesamtplanung eines kontextorientierten Unterrichtes relativ unerheblich ist, an welcher Stelle konkrete Teile des Curriculums ihre Umsetzung erfahren.

### 6.1.3 Verwobenes Unterrichten

Eine der wesentlichen Gedanken des Arbeitens mit den roten Fäden besteht darin, dass in einer konkreten Unterrichtseinheit Elemente verschiedener Fäden enthalten sind. Erst am Ende der Schullaufbahn sind die Fäden insgesamt erarbeitet und damit auch die Zellen in der Bildungsmatrix gefüllt. Dies führt uns zu einer weiteren Definition:

**Definition 4** *Verwobenes Unterrichten* ist das Unterrichten eines gegebenen Curriculums durch eine Auswahl an Roten Fäden, die so häufig wie möglich in Unterrichtseinheiten oder Konzepten (teilweise) gemeinsam betrachtet und untersucht werden. [PV10]

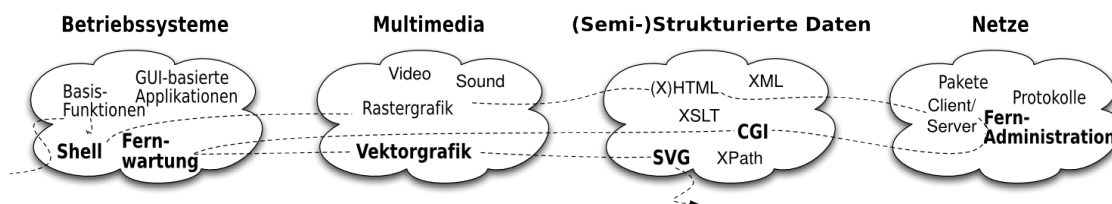


Abbildung 6.4: Der verwobene Rote Faden *Programmiersprache* in einem Projekt

Abbildung 6.4 zeigt dies grafisch am Beispiel des roten Fadens *Programmiersprache*, der mit anderen Fäden verwoben wird. In einem Kontext oder Projekt wird ein System erstellt, das auf einem Webserver aus den Angaben von Benutzern SVG-Grafiken erstellt und diese in einer XML- oder HTML-Seite dem Browser sendet. Die Verarbeitung der Daten erfolgt durch Skripte auf dem Server. Der Server, der nicht als ‚fertiges Produkt‘ auf einem Server existent ist und sofort genutzt werden kann, wird im Rahmen von Fernwartung teilweise mit Skripten gestartet und administriert. Das Programmieren und die erstellten Programme sind dabei nicht Selbstzweck, sondern werden benötigt, um das gewählte Ziel zu erreichen. Im Rahmen der Erstellung dieser Programme werden eventuell noch fehlende Kompetenzen in der Programmierung erworben. In diesem Beispiel werden Kompetenzen aus den roten Fäden Betriebssysteme (Nutzung der Shell bei der Fernwartung), Multimedia (Erstellung von Vektorgrafiken im SVG-Format), (semi-)strukturierte Daten (als HTML, XML-Dokument angefordert per CGI mit SVG-Grafiken), Netze (mit einer Client/Server-Struktur, gewartet durch Fernadministration) benötigt, angewandt und vertieft. Während des Unterrichtens der Elemente des (hier gestrichelt dargestellten) roten Fadens Programmiersprache werden unterschiedlichste andere Fäden benutzt und bearbeitet.

Programmiersprache wird daher nicht in einer oder mehreren aufeinander folgenden Einheiten in einer geschlossenen Struktur behandelt, sondern je nach Auswahl der Einheiten, Kontexte und Projekte entsprechend der dort vorhandenen Anforderungen betrachtet (ausführlicher im Kapitel 6.2). Durch diese Art des Unterrichtens ist auch gewährleistet, dass nach den Vorstellungen von Bruner immer wieder dieselben fachlichen Gegenstände und Prozesskompetenzen unter unterschiedlichen Bedingungen und Schwierigkeitsgraden angesprochen werden.

Durch die roten Fäden erreichen wir, dass die Wahrscheinlichkeit für *kumulatives Lernen* zunimmt. Die Sinnhaftigkeit der Vernetzung besteht auch in anderen Fächern. So berichtet *Merzyn* von einem Kollegen, der bereits 1991 für den Physikunterricht beklagt: „Die systematische Gliederung der Physik in Teilgebiete wie Elektrik, Optik und in deren Unterabschnitte zerteile den Unterricht in Episoden des Lernens, die beziehungslos zueinander ständen. Immanentes Wiederholen, Vertiefen, Belohnen von bereits erbrachten Lernleistungen finde nicht statt.“ [Mer08, S.110]

## 6.2 Programmiersprache als *Roter Faden*

An verschiedenen Stellen, insbesondere im Kapitel 5.1, wurde auf die Bedeutung von Sprache im Rahmen des Schulfaches Informatik hingewiesen. Diese Bedeutung realisiert sich in der Behandlung und Erschließung von Programmiersprachen. Im üblichen Informatikunterricht nimmt das Erlernen der Programmiersprache einen nicht unerheblichen Zeitanteil in Anspruch. Dieses Erlernen wird abgesehen von der Algorithmik und Teilen der Modellierung zumeist unabhängig von den anderen Themen der Informatik behandelt. Es interessiert daher natürlich, wie sich das Erlernen einer Programmiersprache mit dem Konzept der roten Fäden verträgt.

In den Bildungsstandards werden an den verschiedensten Stellen zum Thema **Algorithmen** Forderungen aufgestellt. So wird unter anderem im Kapitel *Kompetenzen über alle Jahrgangsstufen* formuliert:

„Schülerinnen und Schüler aller Jahrgangsstufen

- kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten und lesen und interpretieren gegebene Algorithmen,
- entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar“ [Arb08, S.13].

Im Abschnitt 5.2.3 wurde aus Untersuchungen berichtet, die zeigen, dass ein Algorithmus zumeist in letzter Konsequenz erst dann voll verstanden ist, wenn er (zumindest potentiell) programmiert werden kann oder programmiert worden ist. *John A. N. Lee* meint entsprechend:

*“In the beginning, the initial idea is often expressed in such complex terms that it is overlooked by the majority of readers. The one or two who can see through the veil of complexity simplify the concept, and express it in understandable terms, that in computer science can lead to implementation and application.” [Lee96].*

Sind *Algorithmen* Bestandteil eines Curriculums, dann ist auch *Programmiersprache* Bestandteil dieses Curriculums, denn diese Algorithmen sollen zumeist auch sprachlich dargestellt, also auch programmiert werden. *Juraj Hromkovič* schreibt dazu: „Die historisch wichtigen Begriffe, welche die Informatik zur selbstständigen Disziplin gemacht haben, sind die Begriffe *Algorithmus* und *Programm*. Und wo könnte man die Bedeutung dieser Begriffe besser vermitteln als beim Programmieren“ [Hro08, S.11]? Damit folgt aus den Bildungsstandards Informatik:

Programmierung in einer Programmiersprache ist ein unverzichtbarer Teil des Informatikunterrichtes in der Sekundarstufe I.

Eine Programmiersprache zu erlernen, ist zweifelsohne eine intellektuelle Herausforderung. Dieses gilt natürlich erst recht für Schülerinnen und Schüler in der Sekundarstufe I. Die Bildungsstandards verstehen sich als Standards für alle Schüler. Daraus folgt, dass auch die Grundzüge in einer Programmiersprache in diesem Rahmen für alle Schüler erlernbar sein müssen. Ein Curriculum darf nicht an besonders begabten und interessierten Schülern ausgerichtet werden.

Für heutige Schülerinnen und Schüler erfolgt aus der Beschäftigung mit Computern oder allgemein informatischen Systemen nicht, dass es zwingend notwendig oder sinnvoll ist, sich auch mit Programmierung zu beschäftigen. Sehr treffend wird die Rolle und die Aufgabe der Programmierung im Schulunterricht im amerikanischen Modellcurriculum 'A Model Curriculum for K12 Computer Science' beschrieben:

*“While programming is a central activity in computer science, it is only a tool that provides a window into a much richer academic and professional field. That is, programming is to the study of computer science as literacy is to the study of literature“* [TDJ<sup>+</sup>06, S.2].

Dieses Werkzeug soll dann auch als Werkzeug erfahren werden und nicht durch ein *Lernen auf Vorrat* in Form eines Lehrganges welcher Programmiersprache auch immer. Die Programmierung in dieser Sprache kann dann auch kein Selbstzweck sein. Sollen die Schüler erkennen, dass dieses Werkzeug auf dem Computer genutzt werden kann, darf die Bedeutung dieses Werkzeuges nicht auf eine sogenannte *Mikrowelt* reduziert werden. Zumindest müssen die Erkenntnisse auf universell anwendbare Sprachen übertragen werden. Und letztlich möchten wir, dass die in der Bildungsmatrix angegebenen Ziele verflochten werden, sodass das Erlernen einer Programmiersprache im Unterricht als ein *Roter Faden* angesehen werden und entsprechend konzipiert werden muss.

## 6.3 *Rote Fäden* im Vergleich gegenüber anderen Konzepten

In diesem Kapitel soll das Konzept der roten Fäden mit anderen Strukturierungs- bzw. Klassifikationskonzepten verglichen werden. Ziel ist es, zu zeigen, dass das Konzept über die bisherigen Ansätze hinausgeht und gleichzeitig mit diesen kompatibel bleibt. Grundsätzlich gilt, dass die roten Fäden die fachlichen Strukturen aus Sicht des Schülers betrachten, während viele Strukturierungskonzepte in erster Linie eine Hilfestellung für den unterrichtenden Lehrer darstellen.

### 6.3.1 Fundamentale Ideen

Im Abschnitt über fundamentale Ideen wurde bereits angemerkt, dass fundamentale Ideen keine konkreten fachlichen Inhalte und Prozesse beschreiben, sondern dass ein Unterrichtender mit einem geeigneten Curriculum mit dem Wissen über diese Ideen diese auch vermitteln kann. Eine Kollektion roter Fäden mit den einzelnen Elementen stellt dagegen einen Kanon im Sinne von Wagenschein allerdings in einer strukturierten Form für den Unterricht dar.

Eine fundamentale Idee der Informatik ist ein fachinhaltliches ‚Denk-, Handlungs-, Beschreibungs- oder Erklärungsschema‘ [Sch93], das verschiedene Anforderungen hinsichtlich seiner Anwendbarkeit und Vermittelbarkeit, seiner historischen Bedeutung und seines lebensweltlichen Bezugs aufweist. Der oben vorgestellte (fachinhaltliche) rote Faden (*semi-*)*strukturierte Daten* kann mit der fundamentalen (Master-)Idee *Strukturierte Zerlegung* zwar inhaltlich zur Deckung gebracht werden, die Orientierung entlang des roten Fadens besteht jedoch beispielhaft in der *wiederholten* Umsetzung der fundamentalen Idee in eine Sprache der XML-Familie.<sup>2</sup> Eine solche Übereinstimmung muss jedoch nicht zwangsläufig vorliegen. Als Beispiel hierfür kann der (thematische) rote Faden *Betriebssysteme* gesehen werden, der keiner fundamentalen Idee zuzuordnen ist. Als Konsequenz ist es also möglich, bei der Verfolgung eines thematisch ausgerichteten roten Fadens mehrere fundamentale Ideen zu *besuchen*. Die Abgrenzung kann so verfeinert werden, dass eine fundamentale Idee einen zu vermittelnden Fachinhalt aus Lehrersicht, ein Unterricht entlang eines roten Fadens diesen Inhalt jedoch aus Schülersicht strukturieren soll.

### 6.3.2 Bildungsstandards

**Inhaltsbereiche** Wie bereits im vorigen Absatz festgestellt, ist es möglich, dass ein roter Faden fachinhaltlich ausgerichtet ist. Es ist daher zu prüfen, in welchem Maße sich das Konzept der roten Fäden von den in den Bildungsstandards [Arb08] vorgeschlagenen Inhaltsbereichen abgrenzen lässt. Der zuvor als mit einer fundamentalen Idee wesentlich übereinstimmend identifizierte rote Faden (*semi-*)*strukturierte Daten* kann hier als Beispiel dafür dienen, dass ein roter Faden durch mehrere Inhaltsbereiche verlaufen kann: Es ist möglich und sinnvoll, bei der Verfolgung dieses roten Fadens die Inhaltsbereiche *Information und Daten*, *Sprachen und Automaten* sowie ggfs. *Algorithmen* zu berühren. Auf der anderen Seite ist es aber auch möglich, einen roten Faden so anzulegen, dass er im Wesentlichen im Inneren eines Inhaltsbereiches verläuft. Beispielhaft sei hier der rote Faden *Betriebssysteme* genannt, der dem Inhaltsbereich *Informatiksysteme* zugeordnet werden kann.

**Prozessbereiche** Da die Prozessbereiche im Wesentlichen die im Unterricht zu erlangenden Prozesskompetenzen abbilden, ist eine Übereinstimmung mit den fachinhaltlich bzw. thematisch orientierten roten Fäden nicht zu erwarten. Der Vollständigkeit halber sei jedoch festgehalten, dass entlang eines roten Fadens die Vermittlung von Kompetenzen verschiedener

---

<sup>2</sup>An dieser Stelle sei erwähnt, dass der Begriff *roter Faden* in der Fachdidaktik der Mathematik, beispielsweise in den Rahmenrichtlinien des Bundeslandes Niedersachsen [Der03], synonym mit dem Begriff *fundamentale Idee* verwendet wird.

Prozessbereiche möglich ist. Als Beispiel hierfür kann wiederum der rote Faden *Betriebssysteme* dienen, entlang dessen Kompetenzen des Prozessbereiche *Strukturieren und Vernetzen, Kommunizieren und Kooperieren* und *Darstellen und Interpretieren* vermittelt werden können (vgl. Abbildung 6.2).

### 6.3.3 Great Principles

Die von Denning definierten Fenster auf die Informatik [Den03] *Computation, Communication, Coordination, Automation, Recollection* sind Sichtweisen, unter denen Inhalte und Prozesse der Informatik betrachtet werden können. So werden die Elemente des roten Fadens *Netze* aus Sicht des Fensters 'Communication' gesehen, in einem anderen Faden wie beispielsweise *Betriebssysteme* kann es sich bei demselben Element um das Fenster *Coordination* handeln. Unterrichten nach dem Konzept der roten Fäden gewährleistet, dass die 'Windows' im Sinne Dennings benutzt und eingeübt werden.

### 6.3.4 Didaktische Linsen

Ähnlich verhält es sich mit den *didaktischen Linsen* nach Magenheimer, Schulte und Dörge [MS06, SD08]. Dieser Ansatz ist zudem eingeschränkt auf die Betrachtung *soziotechnischer Informatiksysteme* und *digitaler Artefakte*. Das Arbeiten mit den roten Fäden beschränkt sich nicht auf das Arbeiten mit derartigen Systemen. Bei der Betrachtung der Systeme und Artefakte ist in den verschiedenen Fäden die Linse unterschiedlich. Beispielsweise kann die im Faden *Netze* die Frage der Verschlüsselung unter technischen Gesichtspunkten unter der didaktischen Linse *Vernetzung und Kooperation* erfolgen und im Faden *(semi-)strukturierte Daten* bei der Besprechung von HTML-Seiten für das Online-Banking mit der Linse *Normen und Regulierung* oder *soziale und ethische Aspekte* betrachtet werden. Das Unterrichten nach dem Konzept der roten Fäden realisiert auch hier, dass verschiedenste didaktische Linsen fast zwangsläufig benutzt werden.

### 6.3.5 Informatikturm

Der Informatikturm [Nie95] (siehe Abbildung 6.5 wurde 1995 von Nievergelt vorgestellt, um die verschiedenen Schichten informatischer Bildung für verschiedene Gruppen der Bevölkerung zu untersuchen. Unter einer sehr breiten Plattform der reinen Anwender oben auf dem Turm sind mehrere immer schmaler werdende Schichten bis zum Fundament der Theorie der Informatik angeordnet. Diese Schichten liegen orthogonal zu den Fäden. Die Summe der Elemente der roten Fäden geben den Gesamtkanon (aus den Bildungsstandards) an, der im Laufe der Sekundarstufe I erreicht werden soll. Dabei steigen die Schüler spiralförmig entlang der verwobenen Fäden im Laufe der Schulzeit immer tiefer in den Informatikturm hinunter. Sie erreichen sicherlich in der Sekundarstufe I nicht das Fundament dieses Turmes. Das Modell des Informatikturms und die roten Fäden ergänzen sich in der Weise, da das Ziel des Unterrichtes mit den roten Fäden ist, die Schüler weit hinter die Oberfläche zu führen.

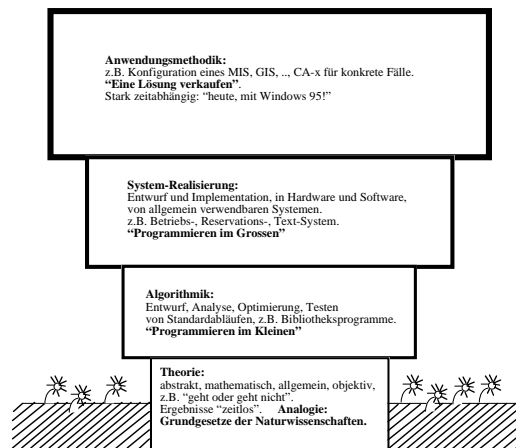


Abbildung 6.5: Der Informatikturm [Nie95]

## 6.4 Zusammenfassung

Die *roten Fäden* stellen ein geeignetes Strukturierungsmittel zur Organisation des Unterrichtes dar. Mit ihnen können gezielt die Kompetenzen, die in den Bildungsstandards Informatik niedergelegt sind, im Laufe der Sekundarstufe I aufbereitet werden. Sie verhindern, dass der Schüler aus seiner Sicht zusammenhanglose Teilgebiete der Informatik kennenlernt. Auch das Erlernen einer Programmiersprache muss entsprechend in Form eines roten Fadens Programmiersprache erfolgen.

Dabei werden ausgehend von fachlichen Orientierungen für die Schülerinnen und Schüler erkennbar wiederkehrende Strukturen ausgewählt, die dann verschiedene Kontexte mehrfach erkennbar durchziehen.

In einem auf roten Fäden aufbauenden Curriculum ist unter anderem eine Strukturierung entlang von Kontexten möglich, jedoch immer der fachinhaltlichen Strukturierung nachgeordnet. Diese Priorisierung führt unter anderem dazu, dass durch die Verwendung eines aus mehreren roten Fäden bestehenden Netzes eine Abdeckung der in den Bildungsstandards vorgesehenen Inhalts- und Prozessbereiche einfach hergestellt und verifiziert werden kann.

Sie ergänzen andere Strukturierungsmittel wie beispielsweise *Fundamentale Ideen*, *Great Principles*, *Didaktische Linsen*. Die *roten Fäden* können nicht durch diese anderen Strukturierungsmittel ersetzt werden und führen in der praktischen Umsetzung zu keinen wesentlichen Wechseln im methodischen Vorgehen.

Der wesentliche Vorteil für die konkrete Unterrichtsgestaltung besteht darin, dass sich durch die Verfolgung roter Fäden auf natürliche Weise die Möglichkeit ergibt, ein Vorgehen nach dem Bruner'schen Spiralcurriculum durchzuführen. Ein derartig auf roten Fäden aufbauendes Curriculum verhindert, dass der Informatikunterricht die von Baumann [Bau96, S.120] zu Recht kritisierte Unterteilung der Schüler in *Nutzer und Betroffene* (Sekundarstufe I) bzw. *Entwickler und Gestalter* (Sekundarstufe II) durchführt. Eine solche Unterteilung jedoch widerspricht jedem Selbstverständnis der Informatik, ein inhaltlich kohärent unterrichtbares Schulfach mit Anspruch auf festen Platz in den Stundentafeln zu sein.



Mit den *roten Fäden* soll erreicht werden, dass die Schüler im *Informatikturm* [Nie95] nicht oben auf der Plattform verbleiben, sondern möglichst tief in diesen Turm hinabsteigen.



# 7 Unterrichtsprojekt an der FSG

## Überblick

An der Fritz-Steinhoff-Gesamtschule Hagen wurde zum Schuljahr 2010/2011 ein Wahlpflichtkurs Informatik/Physik im 6. Schuljahr eingerichtet. Ein Wahlpflichtkurs hat Hauptfachcharakter und wird bis zum Ende des 10. Schuljahr unterrichtet. Dementsprechend werden in einem derartigen Kurs auch schriftliche Arbeiten geschrieben. Damit eignet sich dieser Kurs, die gestellten Ansprüche an das Fach Informatik empirisch zu überprüfen. Die Zusammensetzung und die Grösse des Kurses entspricht weitgehend der Zusammensetzung der Schulpopulation und den üblichen Schulbedingungen, sodass die Ergebnisse stellvertretend für die Schülerschaft sprechen können.

In diesem Kapitel wird das erste Jahr dieses Wahlpflichtkurses vorgestellt. Beschrieben wird der Prozess der Installation dieses Kurses am Ende des Schuljahres, die Planung und Durchführung des Unterrichts im ersten Jahr sowie auch die schriftlichen Überprüfungen. In diesem Kurs wurden der Ansatz eines Unterrichtes mit roten Fäden realisiert, sodass dieses Konzept in der Praxis überprüft werden konnte.

In den vorigen Kapiteln wurde ausführlich dargestellt, dass Informatik heute ein verpflichtendes Schulfach sein müsste. Diese Verpflichtung sollte für die Sekundarstufe I und auch Sekundarstufe II gelten, damit jede Schülerin und jeder Schüler eine ihrem bzw. seinem Alter und Entwicklungsstand entsprechende informatische Bildung aufbauen kann. Von einer Realisierung dieser Ansprüche sind wir in der Bundesrepublik, aber auch in den meisten anderen Ländern dieser Welt weit entfernt.

Erschwerend kommt hinzu, dass es neben den begründeten Forderungen nach einem Pflichtfach Informatik kein für die gesamte Schullaufbahn geführter Nachweis existiert, dass ein solcher Unterricht tatsächlich zum Erfolg führt. Mit Erfolg ist hier gemeint, dass Schüler in einer normal zusammengesetzten Schülergruppe gemeinsam Informatikunterricht erfahren haben und dabei weder unter- noch überfordert werden. Dabei müssen sie selbstverständlich ihre schulischen Laufbahnziele erfüllen können. In einem derartigen Unterricht sollen natürlich die informatischen Kompetenzen entsprechend der Bildungsstandards entwickelt werden und schlussendlich sollen die Schülerinnen und Schüler Interesse am Unterrichtsgeschehen und der Informatik entwickeln und auch möglichst viel, wenn auch nicht immer Spass dabei haben.

Diese Situation entspricht dem bekannten ‚Henne und Ei‘- Problem. Da es keinen verpflichtenden Unterricht in Informatik gibt, gibt es in der Sekundarstufe I auch teilweise keinen derartigen obig beschriebenen Unterricht [Wee07, Sta10], weil aufgrund der vollen Stundentafel kein zusätzlicher freiwilliger Unterricht über einen längeren Zeitraum möglich ist. Die in den verschiedensten Schulen existierenden Arbeitsgemeinschaften oder Kurse, die eine Einführung in konkrete Applikationen oder in das (Schul-)Netz darstellen, sind kein Ersatz für einen solchen Unterricht. Und da kein Nachweis über einen derartigen erfolgreichen Unterricht existiert, kann ein solcher Unterricht dann auch nicht verpflichtend eingeführt werden.

Ansatzweise können je nach Schulform ab dem 6. oder 7. Schuljahr in NRW im Wahlpflichtbereich Kurse mit Informatikinhalten gewählt werden. Diese erfüllen zumeist nicht die

Anforderungen eines Informatikunterrichtes nach den Bildungsstandards<sup>1</sup>.

Um dem Anspruch nach einem Pflichtfach Informatik Nachdruck verleihen zu können, ist es äusserst wichtig, zumindest ein Beispiel für einen erfolgreichen verpflichtenden Informatikunterricht nach den Bildungsstandards beizubringen.

## 7.1 Bildung des WP-Faches NW (If/PH)

Entsprechende Voraussetzungen für einen derartigen Beispielkurs konnten an der *Fritz-Steinhoff-Gesamtschule Hagen (FSG)* geschaffen werden.

Die Fritz-Steinhoff-Gesamtschule gehört zur den ersten Gesamtschulen in Nordrhein-Westfalen und besteht seit 1975 und liegt im Norden von Hagen zwischen den Stadtteilen Boele und Helfe. Sie hat (derzeit) 7 parallele Züge in der Sekundarstufe I und insgesamt ca. 1500 Schüler. Die Oberstufe hat je nach Jahrgang drei bis vier Züge. Schon Ende der siebziger und Anfang der achtziger Jahre setzten sich einige informatikinteressierte Mathematik-Lehrer für den Aufbau eines Faches Informatik und die Anschaffung von Computern in der Schule ein. Die Schule beteiligte sich unter anderem Anfang der 80er Jahre am Schulversuch zur Einführung eines Leistungskurses Informatik. Grundkurse in Informatik hat es in den letzten 30 Jahren bis auf wenige Ausnahmen immer gegeben.



Abbildung 7.1: Die FSG in Hagen

### 7.1.1 Informatik als Wahlpflicht II - Fach

Anfang der 80er Jahre wurde der erste (Apple-II)-Computer beschafft und im Wahlpflichtbereich II (im 9. und 10. Jahrgang) Informatik als Fach installiert. Bis zum Ende der Existenz des Wahlpflichtfaches II in NRW im Jahre 2010 wurde Informatik als Fach durchgängig angeboten und auch teilweise mit bis zu drei Gruppen parallel unterrichtet. Im Laufe der Zeit wurde der Lehrplan mehrfach überarbeitet. Dabei ist auffällig, dass die Anzahl der Themen entsprechend Abbildung 7.2 immer grösser wurde, die Unterrichtszeit aber konstant blieb.

Diese Themen wurden in den letzten Jahren immer mehr zumindest ansatzweise verzahnt unterrichtet, dieses erfolgte allerdings nicht aus einer konkreten entsprechenden curricularen

<sup>1</sup>Beispiele für Curricula mit Informatikinhalten: <http://www.rsamrhein.de/unterricht/fachbereiche/83-informatik.html>  
[http://www.realschulefinnentrop.de/mediapool/85/851450/data/pdf-Kopi/Informatik\\_RS\\_Finnentrop.pdf](http://www.realschulefinnentrop.de/mediapool/85/851450/data/pdf-Kopi/Informatik_RS_Finnentrop.pdf),  
letzter Zugriff: 8.8.2012

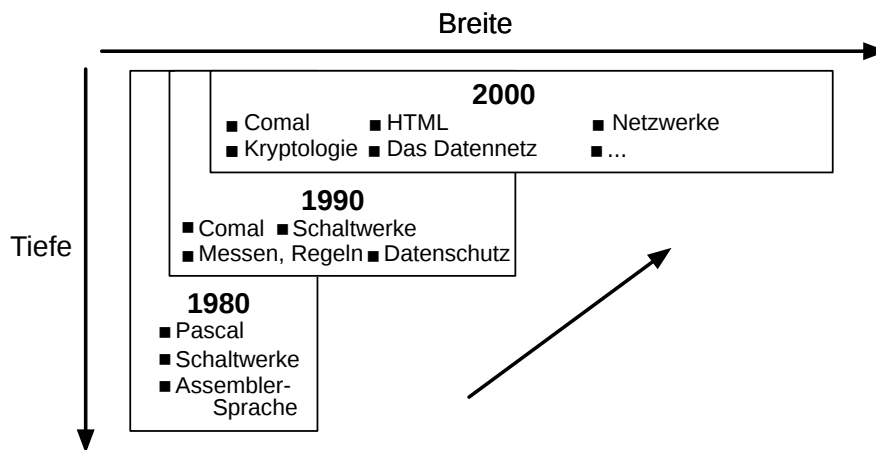


Abbildung 7.2: Themenentwicklung in WPII-Informatik 1980-2010

Planung und Zielsetzung. Erfahrungen aus diesem Unterricht haben zur Entwicklung des Konzeptes der roten Fäden entscheidend beigetragen.

Im Verlaufe der Umstrukturierung der gymnasialen Oberstufe in NRW wurde an den Gesamtschulen das verpflichtende Fach WP II abgeschafft. Schulen können entsprechend ihrer Lehrerkapazitäten ein freiwilliges Angebot schaffen. Dieses Angebot ist im Allgemeinen gegenüber der früheren Struktur sehr reduziert. Entsprechend bietet die Fritz-Steinhoff-Gesamtschule pflichtgemäß nur eine Fremdsprache an. Aufgrund der Abschaffung des Wahlpflichtfaches II ist an vielen Schulen in NRW eine systematische Unterweisung in Informatik selbst für einen Teil der Schüler in der Sekundarstufe I nicht mehr existent.

Mit der Abschaffung dieses Wahlpflichtbereiches ist ein wichtiger Teil Gesamtschule abgeschafft worden. Die Gesamtschule ist als Schulform aus den Schulreformbestrebungen der 60er Jahre entstanden. Sie konnte an der bestehenden Fächerstruktur — wie Heymann entsprechend angemerkt hat [Hey96, 29] — wenig ändern und war somit nur in der organisatorischen Form eine konsequente Umsetzung der Ideen der Schulreform. Die inhaltlichen Reformideen konnten fast nur in der ‚Nische‘ Wahlpflichtbereich II teilweise erhalten und gepflegt werden.

### 7.1.2 Informatik als Wahlpflicht (I) - Fach

Eine Alternative bietet sich im Rahmen der Stundentafel nur im Wahlpflichtbereich (I) an, der nach Umstellung bereits im Jahrgang 6 statt im Jahrgang 7 beginnt. An den Gesamtschulen ist das Fächerangebot im Wahlpflicht-Bereich I gegenüber dem bisherigen Wahlpflichtangebot II laut ‚Ausbildungs- und Prüfungsordnung für die Sekundarstufe I‘ reduziert:

„§ 19 Gesamtschule: [...] (2) Der Wahlpflichtunterricht umfasst ab Klasse 6 die zweite Fremdsprache, den Lernbereich Arbeitslehre und den Lernbereich Naturwissenschaften. Der Lernbereich Darstellen und Gestalten kann nach Entscheidung der Schulkonferenz zusätzlich angeboten werden.“ [NRW10]

Diese Einschränkung begründet sich im ‚Hauptfachcharakter‘ des WP (I)-Faches. Die sogenannten ‚Hauptfächer‘ Deutsch, Mathematik, Englisch und WP (I) haben einen erheblichen Einfluss auf den möglichen Abschluss der Sekundarstufe I und damit auch für den Übergang in die Sekundarstufe II. Das WP II-Fach war dagegen ‚nur‘ ein ‚Nebenfach‘ und damit nur bedingt abschlussrelevant.

Interessanterweise unterscheidet sich das Wahlpflichtangebot der Realschule von dem der Gesamtschule. Hier sind auch technische Fächer zugelassen:

„§ 15 Realschule: [...] (3) Im Wahlpflichtunterricht ab Klasse 7 kann die Schule neben der fortgeführten zweiten Fremdsprache Schwerpunkte in Naturwissenschaften/Technik, Sozialwissenschaften und Musik/Kunst mit jeweils einem Schwerpunktfach anbieten.,, [NRW10]

Auf den Internetseiten des Kultusministeriums wird dies speziell für die Realschule noch genauer spezifiziert: „Wahlpflichtunterricht [...] Hinzu kommen je nach Möglichkeiten der Schule [...] ein naturwissenschaftlich-technischer Schwerpunkt mit den Fächern Biologie, Chemie, Physik, Technik oder Informatik<sup>2</sup>.“ Warum gerade die ursprüngliche Reformschulform Gesamtschule gegenüber einer Schulform des klassischen dreigliedrigen Schulsystems in curricularer pädagogischer Arbeit benachteiligt ist, ist nicht nachvollziehbar.

Ausgehend von diesen nicht optimalen Voraussetzungen für die Gesamtschule wurde ein Kurs konzipiert, unter dem Dach des Bereiches Naturwissenschaften trotzdem eine Kombination aus Physik und Informatik anzubieten. In diesem Kurs sollten dann die in den vorigen Kapiteln vorgestellten Ideen für den Anfangsunterricht in einem 6. Jahrgang konkretisiert, ausgeführt, empirisch begleitet und ausgewertet werden.

### 7.1.3 Hypothesen

Dieser Kurs, der vom 6. bis zum 10. Schuljahr durchgeführt wird, wurde im 1. Unterrichtsjahr untersucht und am Ende dieses 1. Jahres wurde überprüft, ob der Ansatz gerechtfertigt ist. Es interessierten uns folgende Fragestellungen:

---

<sup>2</sup><http://www.schulministerium.nrw.de/BP/Schulsystem/Schulformen/Realschule/>, letzter Zugriff: 9.8.2012

### Hauptfragen

- Können wir bereits in den Anfangsjahren der Sekundarstufe I Informatik als vollwertiges Fach unterrichten?
- Können wir Informatik sogar als Hauptfach unterrichten?

Darüberhinaus wollten wir klären:

### Spezielle Fragen

- Erkennen alle Beteiligten Informatik als ein vollwertiges Fach an? (Akzeptanz)
- Unterscheiden sich die Noten in Informatik von denen in Mathematik & Naturwissenschaften? (Noten)
- Unterscheiden sich die Inhalte in Informatik von denen in einem ITG-Kurs? (Inhalte)

Die Berechtigung des Schulfaches Informatik ist prinzipiell unabhängig von einer ‚informations- und kommunikationstechnologischen Grundbildung (ITG)‘, trotzdem ist es interessant, ob über einen Informatikunterricht die in der ITG gewünschten Ziele ebenfalls erreicht werden und wenn ja, in welchem Umfang.

## 7.1.4 Vorstellung

Die Eltern wählen in Kooperation mit ihren Kindern den geeigneten WP-Kurs für ihr Kind. Damit diese Wahl auf einer gesicherten Grundlage durchgeführt wird, wird den Eltern einige Wochen vor der Wahl eine kleine Broschüre übergeben, in der sich die zur Wahl stehenden Fächer auf ca. zwei bis vier Seiten selber vorstellen können (siehe Anhang 11.1). Ebenso wird jedes Jahr ein Elternabend durchgeführt, bei dem die Eltern eine Präsentation der geplanten Fachkurse besuchen können. Da das geplante Angebot *WP – Physik/Informatik* aussergewöhnlich war, war eine derartige Vorstellung sehr notwendig.

Bei der Fachvorstellung auf dem Elternabend liessen sich etwa 60 Eltern sich über das Angebot informieren. Es war auffällig, dass sehr viele Eltern das Fach für sehr wichtig fanden und daher das Angebot begrüßten. Teilweise hatten sie Befürchtungen, welche Konsequenzen es für ihr Kind haben könnte, wenn sich im Verlaufe des Versuches herausstellen sollte, dass es bei möglichen Konzeptionsfehlern Probleme mit den Noten geben könnte. Ihnen konnte versichert werden, dass mögliche Fehlentwicklungen nicht zu Lasten ihrer Kinder gehen würden. Zusätzlich machte sich bemerkbar, dass inhaltliche Nachfragen sehr schwer zu beantworten waren, da die Antworten trotz aller Bemühungen der Vereinfachung und Fachwortvermeidung bei den Eltern ein Basiswissen weitergehender informatischer Begriffe und Zusammenhänge verlangt hätte, über das sie aufgrund des Fehlens einer informatischen Bildung in unserer Gesellschaft nicht verfügten. Daher blieb am Ende dieses Elternabends ein etwas unzufriedenes Gefühl zurück, da letzten Endes nicht abschätzbar war, wie diese Vorstellung bei den Eltern angekommen war. Dieses unguete Gefühl konnte auch nicht durch die teilweise anwesenden

Kollegen (z.B. Klassenlehrer des betreffenden Jahrganges) beseitigt werden, die die Vorstellung positiv rückmeldeten.

Es gibt keine Garantie für die Einrichtung eines Wahlpflichtkurses und zum Zeitpunkt des Elternabends war es völlig unklar, ob dieser Kurs überhaupt zustande kommen würde. Es stellte sich einige Wochen später heraus, dass sich endgültig genügend Interessenten für dieses Fach angemeldet hatten.

## 7.2 Darstellung der gebildeten Gruppe

Mit dem Beginn des Schuljahres 2010/2011 startete die *WP-Gruppe Naturwissenschaften (Physik/Informatik)* ihren Unterricht. Acht Mädchen und zwanzig Jungen, also insgesamt 28 Schülerinnen und Schüler waren letztendlich diesem Kurs zugewiesen worden. Damit erfüllte der Kurs eine wesentliche Grundlage für aussagekräftige Ergebnisse: Der Kurs hatte Klassenstärke<sup>3</sup>. Damit war eine Voraussetzung für die Übertragung der Aussagen in der geplanten Untersuchung auf andere ‚normalen‘ Gruppen in Schulen erfüllt. Die Schülerinnen und Schüler entstammen sechs von sieben Klassen des Jahrganges, aus einer Klasse hatte sich also kein Schüler für dieses Fach interessiert, aus drei Klassen stammten jeweils mindestens fünf Schüler.

Es handelt sich aufgrund des Wahlcharakters dieses Faches nicht um einen Pflichtunterricht, wie er von uns gefordert wird. Dadurch ergeben sich teilweise Unterschiede in der Zusammensetzung dieses Kurses gegenüber einer repräsentativen Klasse an der FSG. Im Folgenden wird beispielhaft überprüft, ob sich der Kurs bezüglich des Geschlechtes und des Ausländer- bzw. des Migrantenanteils von der Schulpopulation unterscheidet.

Derzeit besuchen 757 Schüler und 629 Schülerinnen die Schule. Damit hat die FSG eine Gesamtschüleranzahl von 1386 bei einem Mädchenanteil von 45,3%. Im untersuchten Jahrgang sind 103 Jungen und 83 Mädchen, der Mädchenanteil ist daher mit 44,6% etwas geringer als in der gesamten Schule. Im gebildeten WP-Kurs ist der Mädchenanteil mit 28,6% etwa 15%-Punkte unterhalb des Jahrganganteiles. Damit berechnet sich ein  $p$ -Wert nach dem  $\chi^2$ -Test von 0,1625. Die Abweichung der Zusammensetzung des Kurses bezüglich der Geschlechter ist damit statistisch nicht signifikant.

An der gesamten Schule gibt es 241 Schüler mit nicht-deutscher Staatsangehörigkeit, das entspricht 17,4% der Schülerinnen und Schüler. Im WP-Kurs haben 4 Schüler, also 14,2% einen nicht-deutschen Pass. Aus der Staatsangehörigkeit alleine lassen sich nur wenig soziale Eigenschaften ableiten. Die Anzahl der Schüler mit Migrationshintergrund<sup>4</sup> beträgt im Jahrgang 107, der prozentuale Anteil ist damit 57,5%. Im WP-Kurs haben mit 10 Schülern nur 35,7% einen Migrationshintergrund. Damit berechnet sich ein  $p$ -Wert nach dem  $\chi^2$ -Test von 0,05023. Dieser Wert liegt bei einer gesetzten Irrtumswahrscheinlichkeit von 5% fast auf der

---

<sup>3</sup>Bei der Bildung neuer Klassen im 5. Jahrgang besteht die Eingangsgröße für eine Klasse 27 Schülerinnen und Schüler

<sup>4</sup>Nach Definition des statistischen Bundesamtes hat eine Person einen *Migrationshintergrund*, wenn sie selber nach Deutschland eingewandert ist oder Nachkomme einer eingewanderten Person ist. Quelle: [https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2012/09/PD12\\_326\\_122.html](https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2012/09/PD12_326_122.html), letzter Zugriff: 16.11.2012



Notenintervall	1.00 – 2.99	3.00 – 3.33	3.34 – 6.00
gesamter Jahrgang	54 28,4%	81 42,6%	55 28,9%
WP-Kurs If/Ph	9 32,1%	10 35,8%	9 32,1%

Tabelle 7.1: Verteilung der Schüler auf Notenintervalle entsprechend der Durchschnittsnote im Abschlusszeugnis der Grundschule

Empfehlung	(bedingt) Gymnasium	(bedingt) Realschule	Hauptschule
gesamter Jahrgang	8 4,4%	88 47,4%	90 48,2%
WP-Kurs If/Ph	3 10,7%	18 64,3%	7 25%

Tabelle 7.2: Verteilung der Schüler entsprechend der Empfehlungen der Grundschule für die verschiedenen Schulformen

Grenze zum Ablehnbereich der Nullhypothese.

Auch wenn die Abweichungen statistisch noch akzeptabel sind, zeigt sich als Trend, dass einzelne Schülergruppen, hier Mädchen und Migranten, trotz zumindest in diesem Fall positiver pädagogischer Beratung unterrepräsentiert sind. Bei vergleichbaren Kursen sollte daher überprüft werden, ob sich dieser Trend bestätigt oder nicht.

Bei der Aufnahme der Schüler in die Fritz-Steinhoff-Schule wird die Durchschnittsnote auf den Abschlusszeugnissen der Grundschulen erfasst. Diese Durchschnittsnote ist ein Indiz, ob die Zusammensetzung des WP-Kurses leistungsmässig mit dem Jahrgang übereinstimmt. Aus der Verteilung der Schülerinnen und Schüler auf die erfassten Notenintervalle entsprechend Tabelle 7.1 ergibt sich bei einem  $p$ -Wert nach dem  $\chi^2$ -Test von 0.78, dass die WP-Gruppe bezüglich der Grundschulabschlussdurchschnittsnote statistisch nicht vom Jahrgangsschnitt abweicht. Sowohl im oberen und unteren Intervall gibt es eine gleich grosse leichte Erhöhung gegenüber dem Jahrgang, sodass diese Veränderungen sich wieder ausgleichen. Betrachtet man das affine Fach Mathematik, so hatten zum Ende der Jahrgangsstufe 5 mit 16 Schülerinnen und Schüler 57,1% eine Zeugnisnote, die besser oder gleich befriedigend (3) war, im gesamten Jahrgang waren es mit 106 Schülerinnen und Schülern genau 57%.

Tabelle 7.2 gibt die Verteilung der Schülerinnen und Schüler entsprechend der Schulformen des dreigliedrigen Schulsystems an, wie sie am Ende der Jahrgangsstufe 4 von den Grundschulen vorgenommen wurde. Die Angaben für den gesamten Jahrgang sind summarisch erfasst worden. Es existieren keine Aufzeichnungen für die einzelnen Schüler, sodass die Kursschüler befragt worden sind. Diese Angabe kann daher im Einzelfall aus der Erinnerung der Schüler heraus nicht korrekt sein. Der  $p$ -Wert nach dem  $\chi^2$ -Test liegt bei 0,04286. Damit ergibt sich interessanterweise eine statistisch relevante Abweichung vom Jahrgangsschnitt nach oben. Diese Abweichung überrascht, da die Leistungsdaten, die in den Notenschnitten der Abschlussklasse der Grundschule bzw. der Mathematiknote nach Jahrgang 5 statistisch weitgehend zwi-

schen Kurs und Jahrgang weitgehend übereinstimmen. Eine mögliche Ursache könnte sein, dass sich in den Zuweisungen zu den Schulformen seitens der Grundschule mehr als gewollt unbewusst die Einstellungen der Lehrer und Eltern bezüglich der Erwartungen an ihre Kinder niederschlagen, die schlussendlich auch bei der Wahl zu einem derartigen neuen, von interessierten Eltern als sinnvoll angesehenen Faches entsprechend zur Wirkung kommen<sup>5</sup>.

## 7.2.1 Eingangspopulation

Diese Gruppe entspricht in der sozialen Zusammensetzung nicht ganz dem Jahrgangsstruktur. Bezüglich der Leistungen in Mathematik allerdings entspricht diese Gruppe genau dem Jahrgangsschnitt. Sie ist damit keine mathematisch-naturwissenschaftlich-technische ‚Elite‘-Gruppe. Besonders beachtenswert ist, dass der Anteil der Schüler mit gymnasialer Empfehlung relativ gering ist. Aufgrund des Wahlverhaltens der Eltern bezüglich der FSG und der benachbarten Gymnasien und Realschulen ist dieser Anteil in allen Jahrgängen recht niedrig.

Die Ergebnisse dieser Untersuchung an einer Gesamtschule, deren Schülerinnen und Schüler trotz aller Einschränkung aus allen sozialen Schichten kommen, die die verschiedensten Bildungsvoraussetzungen mitbringen und weitgehend auch alle Begabungsprofile abdecken, können daher deutlich machen, dass ein Informatikunterricht stattfinden kann, der im Sinne der Bildungsstandards Informatik informatische Bildung für alle ermöglicht und nicht nur für potentielle Gymnasiasten.

## 7.2.2 Unterrichtliche Voraussetzungen

Die Schüler hatten bisher alle keinen Unterricht in Informatik erhalten. An der Fritz-Steinhoff-Schule werden allerdings im 5. Jahrgang im Rahmen sogenannter verpflichtender Zertifikatskurse alle Schülerinnen und Schüler in die Grundlagen der Arbeit mit Textverarbeitung und Tabellenkalkulation eingeführt. Dies dient auch dazu, die Schülerinnen und Schüler mit dem schuleigenen Netz vertraut zu machen, das gelegentlich im Fachunterricht genutzt wird. Dieses arbeitet mit dem Betriebssystem LINUX. An den über 250 Arbeitsplätzen in sieben Computerräumen und teilweise in Klassen und Fachräumen wird an den Computern immer dieselbe Oberfläche präsentiert. Die Schüler haben klassen- und kursweise Zugriffsberechtigungen und können an jedem beliebigen Computer der Schule arbeiten. Damit ist gewährleistet, dass die Schülerinnen und Schüler nicht über einen eigenen häuslichen Computer verfügen müssen oder an dem der Eltern arbeiten müssen.

Zusätzlich stand auf der Webseite der Schule der Gruppe von Beginn an ein geschützter Bereich zur Verfügung, über den die Schüler – aber auch deren Eltern – alle Materialien zur Verfügung gestellt bekommen haben, die im Rahmen dieses Unterrichtes benutzt wurden. Der Umgang mit diesem Medium wurde im Verlauf dieses ersten Jahres behutsam eingeführt und das System entsprechend genutzt.

Der Kurs erhält ab dem 6. bis zum 10 Schuljahr in diesem Kurs in etwa gleichviel Physik wie Informatikunterricht. Da im Schuljahr 2010/2011 eine Renovierung eines Grossteils der

---

<sup>5</sup>Sollte diese Vermutung zutreffen, sind die gegenüber dem Jahrgangsschnitt geringeren Anteile von Mädchen und Migranten im Kurs erklärbar.

naturwissenschaftlichen Räume durchgeführt wurde, bot es sich an, in diesem ersten Jahr ausschliesslich Informatik zu unterrichten und damit die Raumproblematik etwas zu entspannen. Für diesen Unterricht wurde der ‚grosse Computerraum‘ mit 30 Schülerarbeitsplätzen zur Verfügung gestellt. Gelegentlich musste der Raum beispielsweise der Oberstufe für Klausuren überlassen werden. Dann konnte der Unterricht in einem anderen Computerraum der Schule mit jeweils 14 Schülerarbeitsplätzen durchgeführt werden.

## 7.3 Planung des Unterrichts für das erste Jahr (Jahrgangsstufe 6)

Die Schülerinnen und Schüler haben bis zum Ende der Sekundarstufe I im Rahmen dieses Kurses insgesamt drei Jahre Unterricht in Physik/Informatik mit jeweils zwei Wochenstunden à 60 Minuten. Das entspricht in etwa den Anforderungen der Bildungsstandards Informatik. Es wird sich im Laufe des Verlaufs des gesamten Projektes herausstellen, welche der möglichen roten Fäden auch in Zusammenarbeit mit dem geplanten Physikunterricht für diese Gruppe geeignet sind. Gerade aus dem Blickwinkel der Physik bietet es sich an, die roten Fäden aus dem *Multimedia*-Bereich intensiv einzubeziehen. So können im Physikunterricht beispielsweise im Rahmen der Optik und der Akustik die physikalischen Eigenschaften vorbereitet werden, die in der Informatik in den entsprechenden roten Fäden *Sound* und *Grafik* aufgegriffen werden können und müssen. Auf jeden Fall sollen die roten Fäden *Betriebssysteme* und *Netze* im Verlaufe der Jahre bearbeitet werden. Zwingend werden auch die Fäden (*semi*-)strukturierte *Daten* und zu einem späteren Zeitpunkt *Datenbanken* angesehen. In Anbetracht der doch relativ geringen Gesamtzeit für den Unterricht dürfte darüber hinaus nicht mehr viel zu leisten sein. Dieses muss dann entsprechend in den späteren Jahren angepasst werden.

Es ist leicht einzusehen, dass ein kontextorientiertes Unterrichten auf der Basis schon erlernter Grundkenntnisse leichter zu realisieren ist als zu Beginn des Unterrichts in Informatik. *Leon E. Winslow* vertritt in *Programming Pedagogy – A Psychological Overview* [Win96] mit Bezug auf *Hubert L. und Stuart E. Dreyfus* [DD87, S.43/44] die Position, dass ein Novize überhaupt nicht kontextorientiert vorgehen kann: *“Learns objective facts and features and rules for determining actions based upon these facts and features. (Everything they do is context free).“* Gemeint ist damit nicht, dass die eigentliche Lernmotivation nicht aus einem Kontext erfolgt bzw. erfolgen soll, sondern dass neben der aus dem Kontext bezogenen Motivation einen grossen Aufwand für den Lernenden erfordert, sich die Anfänge eines neuen Gebietes anzueignen, sodass zu Beginn des Lernprozesses der Kontext normalerweise nicht betrachtet wird. Ein Fahranfänger, der seine Motivation aus dem täglichen Kontext der Mobilität erfährt, lernt zu Beginn kontextfrei einige technische Grundregeln wie das Schalten des Getriebes [DD87, S.40]. Auch unseren Lernanfängern müssen wir solche Lernphasen zustehen. Weiter stellt Winslow fest:

- *“Given a new, unfamiliar language, the syntax is not the problem, learning how to use and combine the statements to achieve the desired effect is difficult. [...]*
- *Learning the concepts and techniques of a new language requires writing programs in that language. [...]*

- *Problem solution by analogy is common at all levels; choosing the proper analogy may be difficult [...]* [Win96]

### 7.3.1 Verwobenes Unterrichten

Im Folgenden werden die Unterrichtseinheiten beschrieben, die im 1. Jahr Unterricht Informatik im 6. Jahrgang stattgefunden haben. Die Beschreibung der Einheiten zeigt, dass schon von Beginn an die unterrichtlichen Gegenstände in Kontexte eingebunden sind, die von mehreren roten Fäden durchzogen werden. Speziell handelte es sich im Kursverlauf des ersten Jahres um folgende fünf rote Fäden (siehe auch 6.1):

**Programmiersprache:** Anweisungen, Sequenzen, Kontrollstrukturen, Variable und Zuweisungen, Prozeduren, Funktionen, ...

**(Semi-)Strukturierte Daten:** HTML, XHTML, SVG, XML, XSLT/XPath, ...

**Datentypen:** Verwaltung und Organisation durch Typen, insbesondere Klassen in Programmiersprachen, oder Datentypen in einem Betriebssystem.

**Multimedia:** Rastergrafiken, Vektorgrafiken, Sound, ...

**Betriebssystem:** Elementare Funktionen, GUI-basierte Applikationen, Shell, Remote Access, ...

Ein *verwobenes Unterrichten* findet damit bereits im ersten Unterrichtsjahr statt. Mit zunehmenden Kenntnissen in den weiteren Jahren wird es leichter sein, dieses Konzept zu praktizieren. Abbildung 7.3 verdeutlicht das verwobene Unterrichten. Beim Nachvollziehen der weiter unten beschriebenen Einheiten sollte diese Struktur immer mitbedacht werden.

	Einheit 1	Einheit 2	Einheit 3	Einheit 4	Einheit 5	Einheit 6
Faden Pr						
Faden Da						
Faden Ty						
Faden MM						
Faden BS						

(Faden Pr: Programmiersprache, Faden Da: (semi-)strukturierte Daten,

Faden Ty: Datentypen, Faden MM: Multimedia, Faden BS: Betriebssystem

Abbildung 7.3: Unterrichtseinheiten und rote Fäden

## 7.4 Der rote Faden *Programmiersprache*

Eine besondere Aufmerksamkeit verlangt der rote Faden *Programmiersprache* aufgrund der Bedeutung algorithmischer, aber auch programmiersprachlicher Ideen und Konstrukte in der Informatik. Wie im Kapitel 6 ‚Rote Fäden‘ dargelegt, werden die Elemente des roten Fadens

*Programmiersprache* nicht mehr in einem geschlossenen Unterrichtsgang über die Sprache eingeführt, sondern an Hand der Notwendigkeiten im kontextorientierten Unterricht. Selbstverständlich richtet sich die Auswahl einer Programmiersprache zudem nach dem Alter und den Vorerfahrungen der Schüler, in diesem Fall Schülerinnen und Schüler des Jahrganges 6.

### 7.4.1 Auswahl einer Programmiersprache

Daher scheidet für diesen Versuch alle Programmiersprachen aus, die innerhalb einer Modellwelt ihre Stärken, aber oft ausserhalb dieser Modellwelt ihre Schwächen haben. Aus praktischen Gründen ist eine interpretative Sprache vorzuziehen, da mit einer derartigen Sprache ein interaktives Arbeiten leichter möglich ist. Damit scheidet Sprachen wie *Java* und *Pascal* aus. Zugleich soll aufgrund der möglichen Vernetzung mit den roten Fäden *Betriebssysteme* und auch *Netze* die Arbeit in einer begleitenden *Shell* eingeübt werden. So kann den Schülern deutlich werden, dass ein Computer nacheinander *Anweisungen* oder *Kommandos* ausführt, die eventuell auf bestimmte *Parameter* zurückgreifen. Damit scheidet webbasierte Sprachen wie *Javascript* oder *PHP* aus. Mögliche Programmiersprachen für unsere Ansprüche sind unter anderen *Logo*, *Python* oder *Tcl/Tk*, für die alle frei nutzbare Versionen unter den verschiedenen Betriebssystemen vorhanden sind.

**Logo** Die Sprache *Logo* basiert semantisch auf *funktionalen* Prinzipien, auch wenn ihr gerade im Umfeld der *Turtle-Grafik* imperative Elemente hinzugefügt worden sind. Die Sprache *Logo* ist von *Seymour Papert* in erster Linie entworfen worden, um mit erweiterten Ideen von *Piaget* Mathematik zu betreiben. In seinem grundlegenden Buch *Gedankenblitze*<sup>6</sup> schreibt *Papert* entsprechend: „Die Idee, mit einem Computer ‚Mathematik zu sprechen‘, kann zu der Betrachtungsweise verallgemeinert werden, daß Mathematik in ‚Mathematikland‘ gelernt wird, das heißt in einem Kontext, der für das Lernen von Mathematik dasselbe ist wie Leben in Frankreich für das Französischlernen.“ [Pap85, S.14] Die grundlegende Struktur der Programmiersprache *Logo* ist aus informatorischer Sicht<sup>7</sup> für Schüler im 6. Jahrgang nicht die geeignete Wahl, da das funktionale Denken beispielsweise im Mathematikunterricht bisher höchstens rudimentär entwickelt ist und wir in der Informatik zu einem solch frühen Zeitpunkt des Unterrichtens darauf nicht zurückgreifen können. Wir können *Papert* aber grundsätzlich zustimmen, wenn er feststellt: „In vielen Schulen bedeutet der Ausdruck ‚computerunterstützter Unterricht‘, daß der Computer das Kind unterrichtet. Man könnte sagen: *Der Computer wird benutzt, um das Kind zu programmieren.* In meiner Vorstellung *programmiert das Kind den Computer*, und dadurch erwirkt es nicht nur ein Gefühl der Souveränität gegenüber einem Produkt der modernsten und leistungsfähigsten Technologie, es stellt auch eine persönliche Beziehung zu einigen der tiefgreifendsten Ideen aus dem Bereich der Naturwissenschaften, der Mathematik und der Kunst des geistigen Modellbaus her.“ [Pap85, S.13]

**Python** Die Programmiersprache *Python* wurde von *Guido van Rossum* im Umfeld der Entwicklung des Betriebssystems *Amoeba* unter *Tanenbaum* Anfang der 90er Jahre entwi-

<sup>6</sup>Die englische Originalausgabe trägt den Namen: ‚Mindstorms‘.

<sup>7</sup>Die Eignung von *Logo* für mathematische Lernziele steht hier nicht zur Diskussion

ckelt. In ihr wurden Testprogramme geschrieben und so wurde auf leichte Erweiterbarkeit wert gelegt [LF97, S.1]. Diese Kurzbeschreibung weist darauf hin, dass in und mit der Programmiersprache Python auch ein Arbeiten in und mit den roten Fäden Betriebssysteme und Netze möglich ist. Sie kann mit Grafikbibliotheken wie beispielsweise Tk aus dem Umfeld von Tcl/Tk (siehe Abschnitt 7.4.2 umgehen. Python eignet sich daher gut als Anfängersprache (siehe dazu auch [KL10]). Es existiert eine grosse Entwicklergemeinschaft, eine sehr umfassende Literatur, es gibt sogar Lehrwerke, die nicht nur ausschliesslich die Syntax und Semantik der Sprache Python vermitteln, sondern die Sprachvermittlung in Kontexte einbinden [GE09, ML10]. Im Internet gibt es umfassende Hilfen, unter anderem über die ‚Heimatseite‘ <http://www.python.org>.

Der Unterschied zwischen der Sprache *Python* und der dann später im Unterricht verwendeten Sprache *Tcl/Tk* wird an folgenden ‚Codeschnitzeln‘ deutlich. In beiden Varianten wird eine Leinwand in gelb mit einer blauen Linie gezeichnet:

#### Beispielcode in Python:

```
from Tkinter import *
grafikfenster = Tk()
leinwand = Canvas(grafikfenster, width=350, height=500, background="yellow")
leinwand.pack()

leinwand.create_line (20, 30, 200, 350, fill="blue")
```

#### Beispielcode in Tcl/Tk:

```
canvas .leinwand -width 350 -height 500 -background yellow
pack .leinwand

.leinwand create line 20 30 200 350 -fill blue
```

Der Code ist in beiden Sprachen sehr ähnlich, in Tcl/Tk allerdings noch etwas kürzer und eleganter. Das ist aber nicht das eigentliche Problem. In der Shell der Entwicklungsumgebung der Sprache Python werden deutliche Unterschiede zwischen Sprachelementen von Python und Betriebssystemkommandos gemacht. In Tcl/Tk gibt es keinen derartigen Unterschied. Der Charakter eines Betriebssystem-Kommandos oder einer Sprachanweisung kann vom Lehrer nach praktikablen Gesichtspunkten erläutert bzw. eingeführt werden. Dieser Unterschied wird an folgenden beiden Beispielen mit derselben Wirkung deutlich:

#### Beispielcode in der Python-Shell:

```
from os import *
chdir("F")
chdir("erwin")
...
system("import -window tk -pause 10 plan_erwin.jpg")
...
system("opera plan_erwin.jpg")
```

#### Beispielcode in der Tcl-Konsole:

```
cd F
cd erwin
...
import -window plan3.tcl -pause 10 plan_erwin.jpg
...
opera plan_erwin.jpg
```

Eine Alternative ist, die Tätigkeiten, die nicht zwingend in Python-Skripts erledigt werden müssen, in einer `bash`-Sitzung auszuführen. Das bedeutet allerdings das frühzeitige Arbeiten in zwei verschiedenen Shell-Umgebungen mit unterschiedlicher Syntax. Vor allem wird es problematisch, wenn Programme zwecks Automatisierungsvorgängen aus einem Python-Skript heraus aufgerufen werden sollen und dort die Syntax von der `bash`-Syntax relativ stark abweicht.

Dies Situation ist im Anfangsunterricht problematisch. Die daraus zu erwartenden Schwierigkeiten führten dazu, dass in unserem Fall Python nicht als Schulsprache ausgewählt wurde.

Im Anhang sind zur Veranschaulichung Beispiele für Arbeitsblätter angefügt, bei denen als Sprache Python statt Tcl/Tk verwendet wird. Im Arbeitsblatt 11.6 der Abwandlung des Arbeitsblattes ‚Übung 2‘ wird angenommen, dass die Schüler als Basis ihrer Arbeit ein Terminal `bash` benutzen und von dort ihre Python-Programm starten. Ebenso können sie von dort jedes beliebige Unix-Kommando ausführen. In einer weiteren Abwandlung des Arbeitsblattes ‚Übung 2‘ 11.6 wird angenommen, dass die Schüler als Basis ihrer Arbeit die interaktive Python-Kommandozeile benutzen und von dort ihre Python-Programm starten. Ebenso können sie von dort jedes beliebige Unix-Kommando mit Hilfe des Moduls `os` ausführen. Ein weiteres Beispiel ist die Abwandlung des Arbeitsblattes ‚Übung 3‘ 11.6, das aufgrund rein innersprachlicher Anwendungen relativ unproblematisch ist.

**DrScheme** Eine interessante Programmiersprache mit durchdachtem didaktischen Konzept ist *DrScheme* aus der Arbeitsgruppe von *Matthias Felleisen* [FFFK01]<sup>8</sup>. Dieses Sprachkonzept hat seinen Schwerpunkt in der Entwicklung von Sprach- und Entwicklungskonzepten und ist daher nicht besonders gut für jüngere Schüler geeignet. Zudem erfüllt *DrScheme* als Sprache mit funktionalem Paradigma wie Logo nicht die gewünschten Anforderungen.

**Andere Sprachen** An dieser Stelle soll abgesehen von der unten folgenden Darstellung der Sprache *Tcl/Tk* keine weitere Analyse anderer Programmiersprachen erfolgen, denn die Sprachdarstellung an dieser Stelle soll die sehr heftige Diskussion um die richtige Programmiersprache aus den 70er und 80er Jahren nicht fortsetzen. Die Sprache ist nur ein Werkzeug, allerdings gibt es gute und schlechte Werkzeuge und jeder Lehrer soll und muss sich Gedanken um das geeignete Werkzeug für seinen Unterricht machen. Die angestrebten Kompetenzen durch den Informatikunterricht lassen sich zweifelsohne mit unterschiedlichen Sprachwerkzeugen erreichen.

## 7.4.2 Werkzeug Programmiersprache Tcl/Tk

Die Programmierung soll als *Roter Faden* durchgeführt werden, und damit muss die Sprache in den verschiedensten Kontexten wie ein *Klebstoff* gute Dienste leisten. „*Scripting languages are sometimes referred to as glue languages or system integration languages*“ [Ous97], schreibt *Ousterhout* über derartige Sprachen.

---

<sup>8</sup>Das Buch und weitere Informationen sind unter <http://www.htdp.org/> (letzter Zugriff: 11.8.2012) im Internet zu erhalten

Die von ihm in den 80er Jahren initiierte und später von einer sehr grossen Entwicklergemeinschaft ausgebauten und gepflegten Sprache [Ous95], [OJ09] erfüllt in vielfältiger Hinsicht die Ansprüche, die an eine Unterrichtssprache in der Sekundarstufe I weiter oben beschrieben worden sind. Die wesentlichen Eigenschaften seien hier aufgelistet:

### Eigenschaften der Sprache Tcl/Tk

- Die Aus- und Eingabe ist sowohl text- als auch grafikorientiert.
- Der Grafik, baukasten' Tk ist objektorientiert.
- Sie enthält eine eigene Shell:  
Betriebssystemkommandos sind dadurch Teil des Sprachumfangs.
- Es lassen sich CGI-Skripte als Tcl-Programme realisieren.
- Aus HTML-Seiten heraus können Tcl-Skripte wie Java-Applets ausgeführt werden.

Hinzu kommen noch einige Punkte, die aus pädagogischer Sicht von Interesse sind:

### Pädagogische Eignung der Sprache Tcl/Tk

- Tcl/Tk ist (relativ) einfach zu erlernen, obwohl sie nicht didaktisch geprägt ist.
- Mit der Sprache kann interaktiv gearbeitet werden.
- Aufgrund der eigenen Shell existiert keine Trennung von Arbeiten mit dem Computer (-betriebssystem) allgemein und der Programmiersprache mehr.
- Aus dieser Shell sind u.a. (Remote-) Steuerungen von Computern möglich.
- Die CGI-Skripte lassen sich sehr einfach erstellen.

Die Arbeit mit dieser Sprache wird vor allem für den Lehrenden durch die sehr umfangreichen Informationen im Internet unterstützt<sup>9</sup>. Es gibt zusätzlich eine sehr umfangreiche Literatur für diese Sprache, von der hier nur wenige angegeben werden können [Web00, Rai98, RT99, Ous95, HM98, BBWJH03, OJ09, Wal07, Fly12, Har97, Mau96]. Es ist daher sicher nicht übertrieben, dass *Tcl/Tk als ein programmiersprachliches Universalwerkzeug* im gesamten Informatik-Unterricht der Sekundarstufe I angesehen werden kann.

## 7.4.3 Zwei kleine Beispiele in Tcl/Tk

Folgende zwei Beispiele können einen kleinen Eindruck in die Arbeit mit Tcl/Tk geben. Im ersten Beispiel wird nur mit den textbasierten Ein- und Ausgabemöglichkeiten gearbeitet. Dieses Programm kann in einem Editor geschrieben und später ausgeführt werden oder es wird direkt in der Tcl-Shell wie links in der Abbildung 7.4 gezeigt eingegeben. Die Prozedur ergänzt dann die bisher bekannten Kommandos in der Shell um das Kommando *addiere*<sup>10</sup>

```
...
proc addiere {summand1 summand2} {
    return [expr $summand1 + $summand2]
}
```

<sup>9</sup>Internationale Gemeinschaft: <http://www.tcl.tk>, deutschsprachige Entwicklungsgemeinschaft: <http://www.self-tcl.de> letzter Zugriff: 11.8.2012

<sup>10</sup>Diese Beispiele dienen hier nur zur Demonstration der Sprache Tcl/Tk und werden so nicht im kontextorientierten Unterricht eingesetzt.



```

% more addition1.tcl
# Ein kleines Demo-Programm in TCL/TK

proc addiere {summand1 summand2} {
    return [expr $summand1 + $summand2]
}

puts "Bitte Summand 1 eingeben:"
gets stdin summand1
puts "Bitte Summand 2 eingeben:"
gets stdin summand2

set summe [addiere $summand1 $summand2]
puts "Die Summe ist: $summe"

% source addition1.tcl
Bitte Summand 1 eingeben:
3
Bitte Summand 2 eingeben:
4
Die Summe ist: 7
%

```

```

# Ein zweites Demo-Programm in TCL/TK

proc addiere {summand1 summand2} {
    return [expr $summand1 + $summand2]
}

label .info_summand1 -text "Summand 1:"
entry .summand1 -textvariable summand1
label .info_summand2 -text "Summand 2:"
entry .summand2 -textvariable summand2

label .info_summe -text "Die Summe betraegt:"
label .summe -textvariable summe

button .addiere -text "Addiere" \
    -command {set summe [addiere $summand1 $summand2]}

pack .info_summand1
pack .summand1
pack .info_summand2
pack .summand2
pack .info_summe
pack .summe
pack .addiere

% source addition2.tcl
%

```

Abbildung 7.4: Text- und GUI-basiertes Arbeiten in Tcl/Tk

```

}

puts "Bitte Summand 1 eingeben:"
gets stdin summand1
puts "Bitte Summand 2 eingeben:"
gets stdin summand2

set summe [addiere $summand1 $summand2]
puts "Die Summe ist: $summe"
...

```

Tcl verlangt eine strikte syntaktische Trennung zwischen Angabe der Variablenbezeichnung und der Bezeichnung des Wertes der Variablen. Dieses hilft den Schülern bei der doch recht schwierigen Entwicklung eines korrekten Variablenbegriffes.

Dasselbe Beispiel mit der Ein- und Ausgabe im Grafikfenster sieht folgendermassen aus:

```

...
proc addiere {summand1 summand2} {
    return [expr $summand1 + $summand2]
}

label .info_summand1 -text "Summand 1:"
entry .summand1 -textvariable summand1
label .info_summand2 -text "Summand 2:"
entry .summand2 -textvariable summand2

label .info_summe -text "Die Summe betraegt:"
label .summe -textvariable summe

button .addiere -text "Addiere" \
    -command {set summe [addiere $summand1 $summand2]}

pack .info_summand1
pack .summand1
pack .info_summand2
pack .summand2
pack .info_summe
pack .summe
pack .addiere
...

```

Als Shell darf jetzt nicht mehr `tcl` verwendet werden. Diese kennt die Erweiterung `Tk` nicht. Die grafische Variante ist `wish`. Die Bezeichnung *wish* steht für *window shell*. Neben der Konsole aus der textbasierten `tcl` wird ein zweites Fenster für die Grafikausgabe entsprechend rechts in der Abbildung 7.4 bereitgestellt und genutzt.

Mit der erfolgten Auswahl einer Programmiersprache, die im gesamten Unterrichtsverlauf bis einschliesslich Jahrgangsstufe 10 verwendet werden soll, konnte die konkrete Unterrichtsplanung durchgeführt werden.

## 7.5 Unterrichtsverlauf Informatik Jahrgangsstufe 6

Schüler erfahren heute den Computer im Wesentlichen als *Black-Box*. Zudem ist der Zugang zum Computer normalerweise mit einer grafischen Oberfläche verdeckt. Der Computer als zeichenverarbeitende Maschine wird nicht mehr deutlich und den Schülern dadurch auch nicht mehr vermittelt.

Aus Sicht des *Roten Fadens Programmiersprache* ist daher genau dieses Bewusstsein zu schaffen. Schülerinnen und Schüler sollen erfahren, dass der Computer *Kommandos*, die als *Text* geschrieben werden, nacheinander abarbeitet und dies auch dann passiert, wenn sie als Benutzer auf der grafischen Oberfläche agieren.

### 7.5.1 Einheit 1: Buttons und Labels

*Dauer: 5 Wochen*

**Inhalt** Es liegt auf der Hand, diese beiden Argumentationsstränge als Ausgangspunkt der Arbeit mit den Schülern zu nehmen. Die Schülerinnen und Schüler sollen sogenannte *Buttons* erzeugen und mit diesen Buttons irgendwelche Programme starten. Genau das ist ihre tägliche Praxis. Mit diesen ersten, sehr kleinen und einfachen Anweisungen werden sie in die Arbeitsweise mit `Tcl/Tk` eingeführt und sie erkennen, dass die Arbeit mit `Tcl/Tk` und ihre tägliche Beschäftigung mit dem Computer nicht auseinander fallen.

Nach einer kurzen Vorstellung probieren die Schüler die ersten Anweisungen in der `Tcl`-Konsole direkt aus:

```
button .opera starten -text "Opera starten" -command opera
pack .opera
```

Entsprechend lassen sich jetzt schnell weitere Buttons für andere Programme gestalten. Die verwandten Objekte aus der Klasse `Label` werden dabei sehr schnell ebenfalls genutzt. Am Material erhalten die Schüler relativ einfach gestaltete und didaktisch reduzierte Informationen über die `Tk`-Klassen `Label` und `Button` (siehe im Anhang unter Materialien 11.3).

Mit relativ wenig Mühe gestalten die Schüler damit eine Oberfläche, die im Grafikfenster einer üblichen Oberfläche eines heutigen Computer nicht unähnlich ist.

Der Bezug zu den Bildungsstandards ist mehrfach vorhanden. Wir lesen unter anderem: „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] lesen und verstehen Handlungsvorschriften für das Arbeiten mit Informatiksystemen, interpretieren Handlungsvorschriften korrekt und führen sie schrittweise aus, [...] entwerfen Handlungsvorschriften als Text oder mit formalen Darstellungsformen, entwerfen und testen einfache Algorithmen.“ [Arb08, S.15/16]

**Übungen** Informatik-Unterricht leidet oft daran, dass nicht genügend geeignetes Übungsmaterial angeboten wird. Es verwundert dann nicht, wenn relativ schnell einige Schüler den Anschluss an den Unterricht verlieren und das Fach Informatik und speziell das Programmieren den Stempel der ‚schwierigen‘ Materie erhalten.

Die Schülerinnen und Schüler müssen gerade bei ihren ersten ‚Gehversuchen‘ unterstützt und angeleitet werden. Wie lange dauert es, bis auch (fast) alle verstanden haben, dass Leerzeichen an den richtigen Stellen wichtig sind, dass ein Bindestrich vor den Attributen notwendig ist und dass zwischen Bindestrich und Attributname kein Leerzeichen stehen darf etc.. Entsprechend der Arbeitsweise beispielsweise in den Fremdsprachen muss dies immer wieder geübt werden, bis der Aufbau und der Sinn eines programmiersprachlichen Textes total verinnerlicht ist.

Als Übungsmaterial wurde daher ein Stundenplan entsprechend Abbildung 7.5 aus ‚Labeln‘ oder auch ‚Buttons‘ produziert. Das vollständiges Listing befindet sich im Anhang unter Materialien 11.3.

```
puts "Wir basteln uns unseren Stundenplan"
...

frame .montag
pack .montag -side left
label .montag.tag -text "Montag" -width "15" -height "2"
pack .montag.tag
label .montag.stunde1 -text "DE" -background "red" -width "15"
pack .montag.stunde1
...

label .freitag.stunde5 -text "AS" -background "purple" -width "15"
pack .freitag.stunde5
label .freitag.stunde6 -width "15"
pack .freitag.stunde6
```

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
DE	GL	EN	MA	DE
MA	SP	MA	GL	WP
MA	DE	SP	DE	EN
EN	KU	WP	NW	SP
AS	M-FÖ	OS	ZK	AS
MU		KU	REL	

Abbildung 7.5: Ein Stundenplan erstellt in Tcl/Tk

Als Kritik muss angemerkt werden, dass diese Übungen zwar ausreichend umfangreich waren und die Schüler auch genügend Zeit zur Umsetzung ihres eigenen Stundenplanes hatten, aber nicht genügend Abwechslung boten. Hier müssen noch mehr alternative Beispiele entwickelt werden, die auch mehr den verschiedenen Anforderungsniveaus gerecht werden.

## 7.5.2 Einheit 2: Bilder und Webseiten

Dauer: 5 Wochen

**Bilder exportieren** Im weiteren Unterrichtsverlauf wurde der Stundenplan als Bild in eine Webseite integriert. Dazu musste zuerst aus dem Grafikfenster in Tcl/Tk ein Bild in einem gängigen Format produziert werden.

Da die Tcl/Tk-Konsole nicht nur eine Arbeitskonsole für die zu interpretierende Sprache darstellt, können von hier aus auch Betriebssystemkommandos aufgerufen werden. Je nach verwendetem Betriebssystem und den dort installierten Programmen kann nun entsprechend gehandelt werden. Bei unserem Linux-System ist u.a. auch das Grafikpaket *ImageMagick*<sup>11</sup> installiert. Mit dem dort enthaltenen Tool `import`<sup>12</sup> kann dieser Export realisiert werden.

```
import -pause 10 stundenplan.jpg
oder
import -window grafikfenster stundenplan.jpg
```

erzeugt aus dem Grafikfenster eine Grafikdatei `stundenplan.jpg`. (Ohne zu merken, haben wir damit ein kleines Mosaiksteinchen aus dem *Roten Faden Betriebssystem* bearbeitet.)

Dieses Bild kann dann anschliessend mit einem Grafikprogramm oder Grafikbetrachter oder auch in einem Browser angesehen werden. Die besonderen Eigenschaften einer solchen Pixelgrafik werden aber erst später angesprochen. Die Schülerinnen und Schüler lernen in diesem Zusammenhang allerdings, dass die Endung einer Datei etwas Wichtiges aussagt (bzw. aussagen sollte).

**Die erste Webseite** Das Bild im Browser sollte mit Überschriften und eventuell mit Kommentaren versehen werden. Daher wurde eine erste kleine HTML-Datei erzeugt. Diese wurde natürlich vorgegeben und mit den Schülern besprochen und analysiert. Allerdings konnten anschliessend die Schüler mehr oder weniger gezielte Veränderungen vornehmen, sodass sie den Aufbau einer HTML-Datei kennen und auch zumindest teilweise schon verstehen gelernt haben.

Die HTML-Datei sieht folgendermassen aus:

```
<html>
  <head><title>Stundenplan von Emil</title></head>
  <body>
    <center>
      <h1>Stundenplan von Emil</h1>
      
    </center>
  </body>
</html>
```

Diese HTML-Datei konnten die Schüler anschliessend im Browser betrachten und gegebenenfalls je nach Zeit noch etwas verändern und nach ihren Vorstellungen ‚verschönern‘ (siehe dazu die Übungsblätter unter 11.4). Interessant war, dass die Schüler zumeist den Browser aus der Tcl-Konsole mit dem Dateinamen als Parameter aufgerufen haben und nicht durch ‚Klicken‘ auf den auf der grafischen Oberfläche vorhandenen Button für Opera gestartet haben.

Mit diesem ersten Exkurs zu HTML hatten die Schüler den *Roten Faden (semi-)Strukturierte Daten* aufgegriffen.

<sup>11</sup>Dieses Programmpaket gibt es auch für das Betriebssystem Windows.

<sup>12</sup>Warum das Tool zum *Exportieren* nun gerade `import` heisst, ist unverständlich.

Der Bezug zu den Bildungsstandards ergibt sich durch: „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] erstellen Dokumente (z. B. Grafik- und Textdokumente, Kalkulationstabellen) und nutzen die Strukturierungsmöglichkeiten für die jeweilige Dokumentenart angemessen.“ [Arb08, S.15] „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] bezeichnen Dateien problemadäquat und ordnen gängigen Dateinamenserweiterungen passende Anwendungen zu.“ [Arb08, S.16] „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] bearbeiten Dokumente mit ausgewählten Anwendungen, arbeiten in Netzen, [...] wissen, dass digitale Daten leicht manipulierbar sind.“ [Arb08, S.17/18]

### 7.5.3 Einheit 3: Dateiorganisation

*Dauer: 3 Wochen*

Weil der ‚grosse‘ Computerraum beispielsweise während der Klausurphasen gelegentlich durch Kursgruppen der Oberstufe belegt war, ergab sich der Zwang, ab und zu in anderen Computerräumen arbeiten zu müssen. Damit wir an den Dateien entsprechend Einheit 2 weiterarbeiten konnten, wurde diese Situation zum Anlass genommen, die Schüler in die Strukturen unseres Schulnetzes einzuweisen, die verschiedenen physikalischen Speicherorte im Netz und ihre Repräsentation im System zu analysieren und zu besprechen. Dabei wurde die Organisation und die Darstellung von Dateien und Dateibäumen auf einem einzelnen Computer und in Netzen behandelt. Besonderen Wert wurde auf die Darstellung von Bäumen gelegt. Dies finden wir in den Bildungsstandards wieder: „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] kennen und verwenden Baumstrukturen am Beispiel von Verzeichnisbäumen.“ [Arb08, S.14] Und wenig später heisst es: „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 kennen die Navigations- und Änderungsmöglichkeiten für Verzeichnisbäume und deuten sie in Beispielen inhaltlich, [...] navigieren in Verzeichnisbäumen und verändern Verzeichnisbäume sachgerecht“. [Arb08, S.15]

Die Schüler mussten verschiedene Repräsentationen zeichnerisch darstellen und diese in textuelle Kommandos auf der Betriebssystemebene übertragen und auch ausprobieren (siehe Übungsblatt unter 11.4).

<pre>mkdir daten mkdir texte mkdir tabellen  cd daten nedit verkauf10.dat nedit verkauf11.dat nedit verkauf12.dat cd ..  cd texte nedit protokoll1.txt nedit protokoll2.txt cd ..  cd tabellen nedit geld5.tab nedit geld6.tab</pre>	<pre>.  -- daten      -- verkauf10.dat      -- verkauf11.dat     `-- verkauf12.dat  -- tabellen      -- geld5.tab     `-- geld6.tab `-- texte      -- protokoll1.txt     `-- protokoll2.txt</pre>
--	---

*BS-Kommandos für Dateien in der Shell*

*Ausgabe mit tree*

Für dieses gegebene Beispiel erhalten wir Abbildung 7.6 als grafische Baumdarstellung.

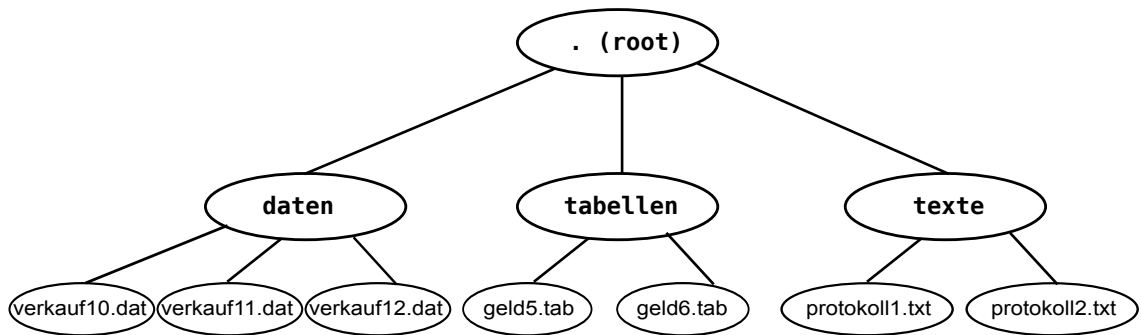


Abbildung 7.6: Beispielhafter Dateibaum

Eines der Ziele war, dass die Schüler in der Lage sind, zu entscheiden, an welcher Stelle sie welche Daten — im Netz oder lokal auf ihrem Arbeitscomputer — sinnvollerweise ablegen. In dieser Einheit befanden wir uns weitgehend im Faden *Betriebssysteme*, auch teilweise im Faden *Datentypen*, da auch über die Eigenschaften von Dateien und ihre Bezeichnungen reflektiert wurde.

#### 7.5.4 Einheit 4: Zeichnungen in Tcl/Tk

*Dauer: 8 Wochen*

Diese Einheit dient als zentrale Vorübung zu algorithmischen Denken. Die Schülerinnen und Schüler müssen zu gegebenen Angaben in bildlicher oder sprachlicher Form die dazugehörigen Anweisungen in Tcl/Tk erstellen oder ein Script in Tcl/Tk in ein Bild umsetzen.

In Tcl/Tk erfolgt dies durch die Nutzung der Klasse `canvas` der Grafikkbibliothek Tk. Das kleine Script:

```

canvas .leinwand -width 350 -height 500 -background yellow}
pack .leinwand

create line 20 30 200 350 -fill blue

```

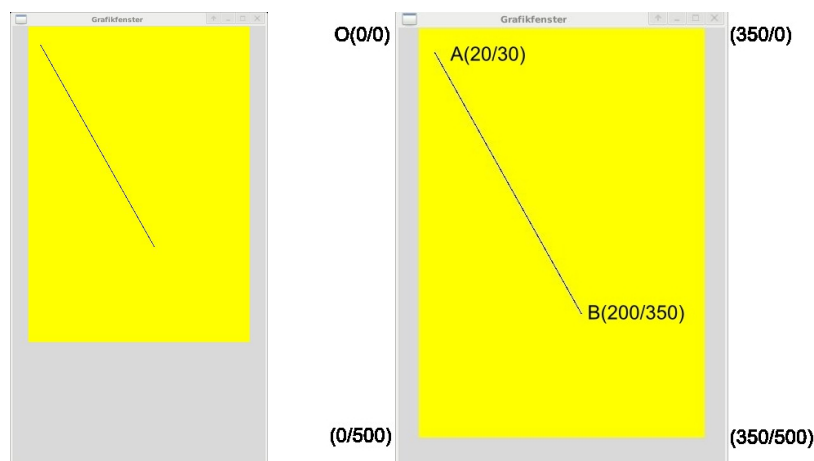


Abbildung 7.7: Leinwand mit Strecke im Grafikfenster

erzeugt im Grafikfenster eine Leinwandfläche mit dem Namen *.leinwand* entsprechend Abbildung 7.7. Diese Leinwandfläche ist 350 Pixel breit und 500 Pixel hoch. Die Hintergrundfarbe ist gelb. Eine Strecke vom Punkt  $A(20/30)$  bis zum Punkt  $B(200/350)$  wird in der Farbe *blau* gezeichnet.

Zur Erledigung der Aufgaben ist es notwendig, dass analysiert wird, aus welchen Komponenten — in der Anfangsphase ausschließlich verschiedenste Strecken — das Bild besteht und diese in eine Sequenz von fast immer gleichartigen Anweisungen übersetzt werden. Damit erzeugen die Schüler *Vektorgrafiken*. Der Fachbegriff wurde hier allerdings noch nicht eingeführt. Nebenbei wurde auch das Arbeiten in Koordinatensystemen aus der Mathematik wiederholt und vertieft. Es zeigte sich dabei, dass einige Schüler zur Verfestigung dieser mathematischen Kenntnisse intensive Übungen nötig hatten. Entsprechend wurden viele Trainingsaufgaben zur Verfügung gestellt, sodass alle Schüler ausreichend Erfolgserlebnisse erzielen konnten. Es wurde dabei darauf geachtet, dass die Schüler immer vor der Eingabe am Computer eine Zeichnung inklusive Koordinatenangaben in ihrem Heft anfertigten. Die Aufgaben entsprechend der Übungsblätter 11.4 hatten etwa folgenden Charakter:

*Zeichne auf der Leinwand zwei Strecken:*

*Die erste soll eine senkrechte Strecke mit dem Pixelabstand 42 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 42 von der rechten Leinwandkante sein.*

Einfache Modellierungsansprüche werden bei freien Zeichnungen gefordert beispielsweise in:

*Zeichne auf der Leinwand ein beliebiges Dreieck oder in Zeichne das Haus vom ‚Nikolaus‘.*

Der Bezug zu den Bildungsstandards ist vielfältig: „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] lesen und verstehen Handlungsvorschriften für das Arbeiten mit Informatiksystemen, benutzen die algorithmischen Grundbausteine zur Darstellung von Handlungsvorschriften, interpretieren Handlungsvorschriften korrekt und führen sie schrittweise aus, [...] entwerfen Handlungsvorschriften als Text oder mit formalen Darstellungsformen, entwerfen und testen einfache Algorithmen.“ [Arb08, S.15/16] „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] identifizieren Objekte in Informatiksystemen und erkennen Attribute und deren Werte“ [Arb08, S.19]

## **7.5.5 Einheit 5: Vektor- und Pixelgrafik**

**Dauer: 6 Wochen**

Die Bildungsstandards schreiben dazu: „Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 [...] unterscheiden die Darstellung von Grafiken als Pixelgrafik und Vektorgrafik.“ [Arb08, S.14] Diese Unterscheidung erfährt man nicht, indem zwei verschiedene Anwendungsprogramme genutzt werden, mit denen grafische Objekte produziert und verändert werden können und die eine Anwendung die Objekte als Pixelgrafiken und die andere als Vektorgrafiken bearbeitet und speichert. Wird der Unterschied nicht *sichtbar* im wahrsten Sinne des Wortes, bleibt dieser Anspruch nur abstrakt und kann kaum verstanden werden.

Der hier vorgeschlagene und durchgeführte Weg ist ein anderer. Vektorgrafiken haben die Schüler bereits in der vorherigen Einheit erstellt. Es sind Tcl-Scripte, die irgendwelche grafischen Objekte auf die Leinwand zeichnen. Solche Objekte lassen sich auch als Pixelgrafik konvertieren. Auch mit Pixelgrafiken haben die Schüler bereits früher gearbeitet. Sie haben aus

dem Tcl/Tk-Grafikfenster eine `jpg`-Datei erzeugt und diese im Browser angezeigt bzw. diese in eine HTML-Seite integriert. In dieser Einheit werden die Schüler genauer mit dem Aufbau einer Pixeldatei vertraut gemacht. Dazu eignet sich das `jpg`-Format jedoch nicht gut. Ideal ist das `xpm`-Format, das eine Präsentation der Pixel im ASCII-Format enthält. Diese Dateien können im Editor manipuliert werden und anschließend in einem Grafik-Betrachter angesehen werden. Zu einem späteren Zeitpunkt können die gängigen Formate `gif` und `jpg` zusätzlich erläutert werden. Mit dieser Einheit befinden wir uns damit im *Roten Faden Multimedia oder Grafik*.

Die Schüler erzeugten eine kleine Grafik in Tcl/Tk und exportierten diese wie schon bekannt in eine Grafik, diesmal aber in das `xpm`-Format. Die Arbeitsweise wird anhand folgender Aufgabe aus dem Arbeitsblatt 4 (siehe Anhang 11.4) deutlich:

1. Wie lauten die zugehörigen Anweisungen in *Tcl/Tk* für folgende Beschreibung:
  - Erstelle eine Leinwand der Breite 20 und Höhe 10 Pixel. Die Leinwand soll einen grünen Hintergrund haben.
  - Zeichne ein Kreuz bestehend aus zwei Diagonalen auf diese Leinwand: Eine Diagonale soll von der Ecke oben rechts zur Ecke unten links verlaufen und die andere von oben links nach unten rechts. Die Diagonalen sollen rot sein.
2. Erstelle eine Zeichnung in Deinem Heft. Zeichne sie farbig. (1 Kästchen pro Pixel)
3. Diese in Tcl/Tk beschriebene *Vektorgrafik* wurde in das Pixel-Format `xpm` übertragen. Der Anfang dieser Datei sieht folgendermassen aus:

```
/* XPM */
static char *aufgabe_a[] = {
/* columns rows colors chars-per-pixel */
"22 12 3 1",
"o c green",
". c red",
"X c blue",
/* pixels */
"XXXXXXXXXXXXXXXXXXXXXXXXX",
"X...oooooooooooooooo...oX",
```

Schreibe den Inhalt der gesamten Datei ohne Kopf auf Dein Blatt. Färbe Deinen Dateitext.

In einer späteren Einheit wird dieses Thema aufgegriffen werden müssen, um Maßstabveränderungen bei Pixel- und Vektorgrafik zu betrachten. Am Ende dieser Einheit waren die Schüler inzwischen recht gut geübt, je nach Zweck das jeweils richtige Programm zum Betrachten oder Bearbeiten der verschiedenen Dateitypen zu verwenden.

Entsprechend dem Konzept der verwobenen roten Fäden fand in den bisherigen Einheiten keine konkrete Einführung in eine Programmiersprache statt, sondern dort, wo es sinnvoll und angemessen war, haben die Schüler Kommandos in die Tcl-Konsole geschrieben oder diese als Skript mit einem Editor erstellt und dann ausgeführt. Es handelte sich bisher ausschliesslich um lineare Sequenzen von Anweisungen. Dieses war kein Zufall, denn alle Kontrollstrukturen benötigen zumindest ein Vorverständnis des *Variablenbegriffes*. Dieses kann man von den Schülern in diesem Alter nicht erwarten.



Ab einem Alter von etwa 11 Jahren können Kinder das Konzept der Variablen verstehen, wie *Jean Piaget* nachgewiesen hat. Piaget [Pia00, S.167] erläuterte, dass Kinder die Stufe der *formalen Operationen* in diesem Alter erklimmen und nur in dieser ist ein formaler Begriff wie der der Variablen versteh- und anwendbar. Unsere Schüler sind am Ende der Jahrgangsstufe 6 11 Jahre oder ein bisschen mehr alt. Die Stufe der formalen Operationen wird nicht auf ‚einen Schlag‘ erobert, sondern wird individuell und nicht zeitgleich von allen Schülern erklimmt. Daher bietet es sich an, die Einführung des formalen Begriffes der Variablen noch an konkrete Objekte möglichst aus der realen Welt zu binden. Damit erreichen wir auch die Schüler, die noch länger in der Phase der *konkreten Operationen* verweilen.

## 7.5.6 Einheit 6: Variablen

*Dauer: 5 Wochen*

Aus den vorigen Einheiten existieren inzwischen eine Menge an Bildern, die sie zum Teil selbst gezeichnet, verändert und entwickelt haben. Zusätzlich erhielten sie noch eine weitere Menge an Bildern. Diese Bilder waren alle in einem Verzeichnis auf dem Dateiserver gespeichert. Die Schüler erfuhren nun, wie sie in Tcl/Tk ein Bild in einer Variablen speichern konnten und wie man anschließend damit operieren kann. Kinder malen und zeichnen gerne. Sie erhielten daher Aufgaben, in denen sie Bilder beispielsweise von Dorfstrassen mit Autos, Bäumen und Häusern kreieren sollten.



Abbildung 7.8: *Konkrete Vorübungen zum Begriff Variable*

Bevor diese Bilder für den Computer konstruiert werden, haben sich die Schüler ohne Computer die Begrifflichkeiten und Handlungen visualisiert. Mehrere Din-A4-Seiten repräsentierten den

externen Speicher mit Dateien, den Arbeitsspeicher mit Variablen und den Bildschirmspeicher mit der lokalen Präsenz von Bildern an bestimmten Koordinaten. Die Schüler erhielten Abdrucke der verschiedenen Bilder entsprechend Abbildung 7.10 und mussten die besprochenen und entwickelten Anweisungen mit ihrem Papiercomputer nachspielen (siehe Abbildung 7.9).

Folgende Ideenkette wurde dabei angewandt:

- Rastergrafiken befinden sich als Dateien in einem Verzeichnis.
- Der Inhalt dieser Dateien wird im Arbeitsspeicher in eine sogenannte Variable mit einem Namen kopiert.
- Der Wert dieser Variablen kann (evtl. mehrfach) in den Bildschirmspeicher kopiert werden.

Diese Ideenkette muss (mehrfach) unabhängig vom Computer ‚gespielt‘ werden (siehe Abbildung 7.8). Erst nachdem an Hand der konkreten Objekte die Vorgänge veranschaulicht und verstanden waren, wurden die Skripte für den Computer erstellt und ausprobiert.

Diese Einheit verwob die beiden Fäden *Multimedia* und *Programmiersprache*. Im nächsten Schuljahr werden darauf aufbauend Animationen und Videos unter Konstruktion von Bildern auf einer Leinwand in Tcl/Tk erstellt werden. Dabei werden die Wiederholungsstrukturen

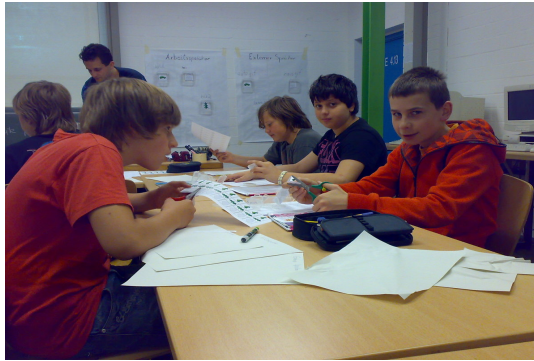

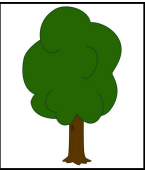





Abbildung 7.9: Die Schüler *konstruieren* sich den Begriff *Variable*

```

canvas .leinwand -width 400 -height 500 -background yellow
pack .leinwand

set auto_links [image create photo -file auto.gif]

.leinwand create line 0 180 400 180 -fill black
.leinwand create line 0 320 400 320 -fill black

.leinwand create image 60 290 -image $auto_links

set baum [image create photo -file Bilder/baum.gif]

.leinwand create image 350 330 -image $baum

set villa [image create photo -file Bilder/villa.gif]

.leinwand create image 150 135 -image $villa

```

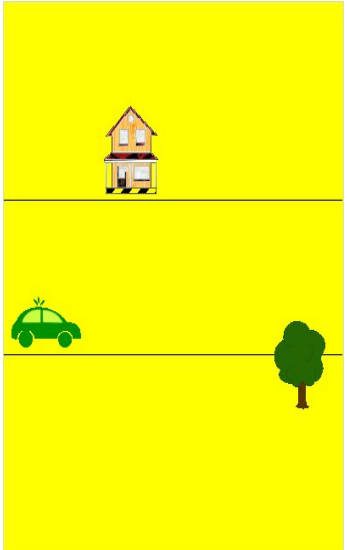


Abbildung 7.10: Bildervorlage und erstellte Leinwand

eingeführt werden. Wir haben entsprechend dem Spiralprinzip nach Bruner die weiter oben schon erwähnten Punkte der Bildungsstandards zu algorithmischen Grundstrukturen und zu Grafiken aufgegriffen.

In den Bildungsstandards ist der Variablenbegriff erst der Jahrgangsstufe 8 zugeordnet: „Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10 stellen die algorithmischen Grundbausteine formal dar, verwenden Variablen und Wertzuweisungen, [...]“ [Arb08, S.16] Hier bei uns handelte es sich um Vorübungen, die in zukünftigen Jahren vertieft werden. Auch die mathematischen Kompetenzen dürften durch dieses *sanfte Vorgehen*, wie *Sherry Turkle* es nennen würde, bezüglich des Variablenbegriffes profitieren. Im Gegensatz zu dem üblichen formalen Weg wurde hier spielerisch und grafisch ein erster anschaulicher Schritt zum Verständnis dieser für Schüler oft schwierigen mathematisch-informatischen Idee gegangen. Bei einer vergleichbaren Situation beschreibt das *Turkle* im Buch *Die Wunschmaschine* folgendermassen: „Im Laufe eines langen Nachmittags hat Ronnie gelernt, mit einem kleinen formalen System zu arbeiten, einem System, dessen Prinzipien andere im Algebra-Unterricht

lernen, wenn ‚Geschwindigkeit, Zeit und Entfernung‘ auf dem Lehrplan steht. Aber Ronnie wäre vielleicht nie zu bis zu diesem Punkt vorgestoßen, da auf dem normalen Weg zum Verständnis der Algebra viele qualvolle Stunden liegen, in denen eine andere Art von Betätigung gefordert wird: still am Tisch sitzen, Zahlen in Kästchen schreiben, Gleichungen auf einem Blatt Papier lösen.“ [Tur84, S.149]

## 7.5.7 Zusammenfassung

Nach einem Jahr Informatikunterricht im Jahrgang 6 ergibt sich aus Lehrersicht folgendes Bild, das aus dem konkreten Unterrichtsgeschehen abgeleitet ist:

- Ein kontextorientierter Unterricht mit fachlichem Schwerpunkt ist möglich.
- Eine Binnendifferenzierung für schwächere Schüler und auch leistungsstarke Schüler ist möglich.
- Der Unterricht ist sprachlich nicht zu überfrachtet.
- Es lassen sich genügend Übungsaufgaben erstellen.
- Die Schüler sind im Unterricht zumeist motiviert.

Natürlich gibt es auch noch Probleme, über die weiter nachgedacht werden muss: Es ist schwierig, alle Schüler zu betreuen, wenn sie einzeln an einem Computer arbeiten. Auf der einen Seite ist es positiv, dass sich alle Schüler bei den Problemen weitgehend selbst ‚durchbeissen‘ müssen und sich nicht hinter einem Partner ‚verstecken‘ können. Immerhin können sie jederzeit ihren Nachbarn fragen, wenn es nicht weitergeht. Allerdings benötigen sie verständlicherweise Unterstützung seitens des Lehrers und sei es nur, dass sie in ihrer Ungeduld aufgemuntert werden müssen. Es ist eine Überlegung wert, die Praxisphasen gezielter als Partnerarbeit mit zwei Schülern pro Computer oder als Einzelarbeit zu organisieren. Diese Frage stellt sich allerdings nur, wenn die Gruppe über den doch aus schulischer Sicht Luxus verfügt, jedem Schüler einen Computer zur Verfügung stellen zu können. Selbst die Situation mit zwei Schülern pro Computern ist gelegentlich mit Schwierigkeiten verbunden, da bei einer Gruppe in Klassenstärke bereits etwa 15 Arbeitsgruppen zu betreuen sind. Der Lehrer ist aufgefordert, hier genügend Zeit ‚in Reserve‘ für Praxisphasen einzuplanen, damit die Schüler trotz gelegentlicher Wartephase ihre Arbeit auch zu Ende führen können.

Eine Unterstützung seitens des Elternhaus ist oft nicht möglich. Dies hat unterschiedlichste Gründe. Ein Grund ist die Schwierigkeit, die von uns verwendete Software zu installieren, obwohl sie für alle Betriebssysteme zur Verfügung steht. Sofern bei der Installation einige Fragen in englischer Sprache gestellt werden, ist eine eigenständige Installation für viele Schüler und deren Eltern trotz Erklärungen in der Schule nicht möglich. Einige Schüler durften zudem die Software zu Hause auf dem häuslichen Computer nicht installieren.

Diese Situation ist ein Problem und eine Chance zugleich. Aufgaben, die zu Hause erledigt werden sollten, wurden immer so gestellt, dass sie ohne Computer zu erledigen waren. In der Schule wurden sie dann besprochen und gegebenenfalls in den Computer eingegeben. Dadurch erhält der Lehrer zwangsläufig eine bessere Übersicht über die Erledigung der Hausaufgaben, als wenn er sich auf die nicht kontrollierbare Computerarbeit stützen müsste. Aufgrund dieser Form der Hausaufgaben wurde kein Schüler benachteiligt bzw. bevorteilt.

## 7.6 Kursarbeiten

Das Wahlpflichtfach ist an der Gesamtschule neben Deutsch, Englisch und Mathematik Hauptfach. Dementsprechend müssen auch Kursarbeiten — in der Gesamtschule *Test* genannt — geschrieben werden. Dieses stellt eine besondere Herausforderung dar, da es kaum Erfahrungen gibt, wie denn nun solche Klassenarbeiten in Informatik bei Schülern in diesem Alter auszusehen haben.

Eine besondere Schwierigkeit liegt darin, dass die Schüler auf der einen Seite mit programmiersprachlichen Texten – sei es Tcl/Tk, HTML oder auch Betriebssystemkommandos – umgehen können und diese auch anwenden sollen. Gleichzeitig müssen sie diese umgangssprachlich beschreiben. Dabei haben insbesondere leistungsschwache Schülerinnen und Schüler häufig sprachliche Defizite im mündlichen und schriftlichen Bereich. Dies legt nahe, in Klassenarbeiten auf Aufgabentypen zu verzichten, die im Wesentlichen aus einer Umsetzung von einer in die andere Sprachebene bestehen.

In der Informatik wird der Computer als eine Maschine verstanden, die Zeichenketten manipuliert. Oft ist das Ziel, visuelle Bilder oder Vorstellungen beim Benutzer zu produzieren. Diese dem Text zugeordnete visuelle Darstellung kann nicht nur sprachlich, sondern auch bildlich dargestellt werden. Also wurden in den Klassenarbeiten zumeist Texte in Bilder umgesetzt oder umgekehrt. Da Kinder in diesem Alter gerne zeichnen und malen, entsprachen diese Anforderungen in den Klassenarbeiten dem kindlichen Entwicklungsstand. Dieses Vorgehen wird auch gestützt durch die Aussagen von *Hans Aebli*, die im Kapitel 5.2 über Programmiersprachen vorgestellt worden sind: „Die Sprache baut die Vorstellung der Sache im Geist des Hörers dadurch auf, daß sie Wörter aneinanderreihet, Wortketten (strings of words) erzeugt. [...] Sprache wird damit in einem zweiten, tieferen Sinn zu einem *stellvertretenden Handlungssystem*. [...] Wir behandeln nun das Wort, wie wenn es die Sache wäre. [...] Das Verstehen von Sprache erfolgt also durch Nachkonstruktion.“ [Aeb76, S.54–59] Dieses *stellvertretende Handlungssystem* ist in einfachster Form der auf der Computerebene der über den Computermonitor aufgenommene visuelle Eindruck.

Der Mathematik-Didaktiker *Walther Lietzmann* überträgt diese Einsicht auf die Erziehung und Bildung: „Sehenlernen ist ein wichtiges Erziehungsziel; es ist nicht so, daß da gar nichts zu lernen, daß diese Fähigkeit eben angeboren sei. [...] Ein weiterer Schritt ist es dann von der bloßen Anschauung zu der Fähigkeit, sich von dem Gesehenen in Zeichnung oder Modell ein Bild zu machen, und noch ein weiterer Schritt, die Dinge zu beschreiben. [...] Grundvoraussetzung ist aber, daß man zumindest von dem, was man beschreiben will, eine deutliche Anschauung oder Vorstellung hat.“ [Lie49, S.73]

Generell muss berücksichtigt werden, dass diese Schülergruppe keine nach Leistung ausgewählte Gruppe ist. Die Schülerinnen und Schüler haben den Kurs aus Interesse gewählt. Auch leistungsschwache Schüler müssen eine realistische Chance haben, zumindest ein ‚ausreichend‘ in den schriftlichen Tests zu erreichen. In unserem Fall ist dies umso bedeutsamer, weil für die Schüler dieser Kurs ein Hauptfach ist und damit mit entscheidend für den Abschluss am Ende des 10. Schuljahres wirkt.

Der Anspruch, dass Informatik ein allgemeinbildendes Fach als Pflichtfach werden soll, bedingt, dass an die Anforderungen keinerlei ‚Elite‘-Ansprüche gestellt werden dürfen. Wir befinden uns mit dieser Problematik an der Wegescheidung, an der die naturwissenschaftli-

chen Fächer in den letzten Jahrzehnten zumindest teilweise in die falsche Richtung gegangen sind und derzeit mit Konzepten wie *Chemie/Physik/Biologie im Kontext* abmildern wollen. Aus vielen Untersuchungen bei Schülern und Lehrern folgert der Physik-Didaktiker *Merzyn*:

„Der enge innere Zusammenhang zwischen den vier Aspekten Schwierigkeit - Stofffülle - Lernerfolge - Zensuren liegt auf der Hand. [...] die Abstraktheit von Inhalten, die Atemlosigkeit beim Voranschreiten, das Gefühl eigenen Ungenügens und Nichtverstehens. Die Häufigkeit von Misserfolgserlebnissen, stärker als in anderen Fächern, ist offenbar eine zentrale Ursache dafür, dass sich so viele Schüler vom Physik- und Chemieunterricht abwenden. Die Erinnerung an solche Misserfolge ist verbreitet und zum Teil noch Jahrzehnte nach Ende der Schulzeit in Befragungen nachweisbar.“[Mer08, S.135/136]

Diese Fehler der naturwissenschaftlichen Fächer (sowie auch des Faches Mathematik) sollte die Informatik nicht wiederholen.

Im Anhang 11.5 sind die Teste vollständig dokumentiert. Der aufmerksame Leser wird feststellen, dass in den späteren Testen immer eine Aufgabe aus dem vorigen Test enthalten ist. Dies hat sich in langen Jahren der Unterrichtstätigkeit des Verfassers als sinnvoll erwiesen, da mit diesem Verfahren die Schüler bei der Anfertigung einer Berichtigung einen Sinn über die reine Korrektur der Fehler in einem zurückgegebenen Test sehen. Die Schüler wissen dabei nicht, welche Aufgabe aus dem letzten Test wiederholt gestellt wird. Zumeist erhalten die Schüler zudem als Hausaufgabe, nochmal den ganzen oder zumindest Teile des Testes im Hausheft anzufertigen, damit sie sich noch einmal mit der Materie beschäftigen. Beide Massnahmen verstärken die Nachhaltigkeit des Lernens. Schüler erfahren dadurch, dass man zumindest nicht für einen Test lernt.

Die Schüler haben bei der Rückgabe eine vom Lehrer erstellte Lösung des Testes (siehe Anhang 11.5) erhalten. Damit können sie bei Unklarheiten nachsehen, wie denn nun die gewünschte Fragestellung im Test tatsächlich beantwortet werden konnte. Selbst wenn ein Schüler bei der Korrektur statt versuchter Eigenarbeit nur von dieser Vorlage abschreibt, ist dies besser, als wenn er oder sie mit wenig Einsatz doch wieder Falsches als Korrektur aufschreibt. Da im Vergleich zu den ‚klassischen‘ Fächern wie beispielsweise Mathematik und Physik die Eltern keinerlei Zugang zu den Aufgaben in den Kursarbeiten haben, ist es ihnen damit möglich gewesen, mit ihren Kindern über den Test zu reden und vielleicht auf Wunsch auch etwas Hilfestellung bei der Abfassung der Berichtigung oder Neufassung zu geben.

## 7.6.1 Aufgabenbeispiele

Der Charakter der Aufgaben in den vier Testen wird an folgenden Beispielen deutlich:

### Test 1/Aufgabe 1: *Buttons* und *Labels*

a) Folgende Kommandos werden in der Konsole eingegeben:

```
label .info1 -text "Ganz aktuell:"  
pack .info1  
button .knopf1 -text "Drueck mich" -command opera  
pack .knopf1
```

Zeichne, was Du im Grafikfenster siehst!

Was kann man dort machen und was passiert dann?

b) Du möchtest einen weiteren Knopf haben, der den Editor `nedit` startet. Was musst Du eingeben.

### Beispielhafte Schülerlösungen

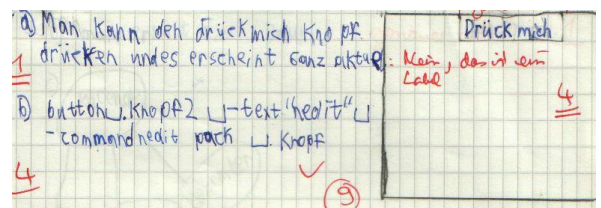
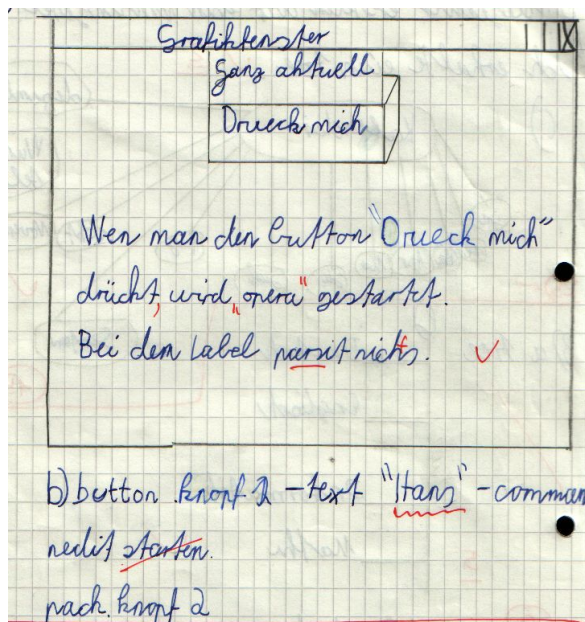
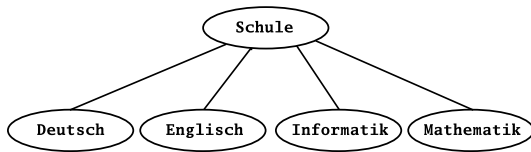


Abbildung 7.11: Beispielhafte Schülerlösungen zu Aufgabe 1 im Test 1



### Test 1/Aufgabe 3: Dateibäume



- Du möchtest diesen Dateibaum in der Konsole erzeugen. Schreibe die dazu notwendige Befehlsfolge auf.
- Wie sieht die Ausgabe aus, wenn Du jetzt das Kommando `tree` eingibst?
- Was musst Du eingeben, wenn Du im Verzeichnis `Deutsch` noch Unterverzeichnisse für `Januar` und `Februar` anlegen möchtest.

### Beispielhafte Schülerlösungen

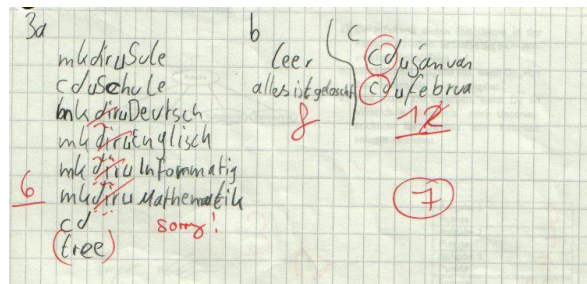
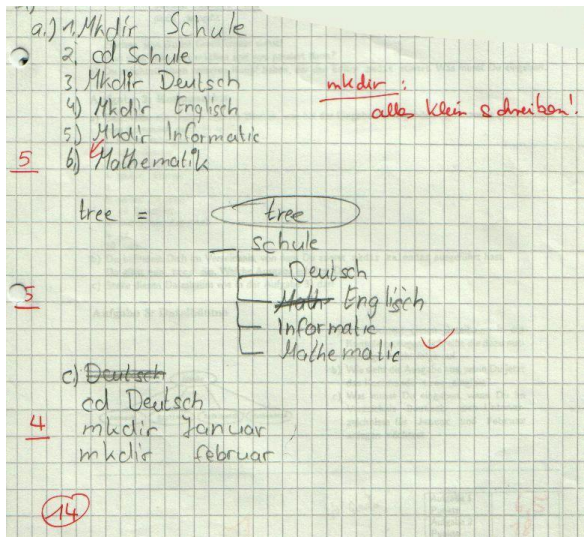





Abbildung 7.12: Beispielhafte Schülerlösungen zu Aufgabe 3 im Test 1

## Test 2/Aufgabe 1: Webseite von Emil

Folgende Bilddateien hat Emil erstellt bzw. erhalten:

		
gesicht.jpg	rad.jpg	eisenbahn.jpg

Diese verwendet er in seiner neuen Webseite. Wie sieht diese Webseite im Browser aus?

```
<html>
  <head><title>Webseite von Emil</title></head>
  <body>
    <center>
      <h1>Webseite von Emil Saueressig</h1>
      Hier seht ihr meine aktuelle Webseite.<br>
      Zuerst ein Bild von mir:<br>

      <br>

      <h2>Meine Hobbys:<br>
      
      
      <br>
      Radfahren und Eisenbahn.
    </center>
  </body>
</html>
```

## Beispielhafte Schülerlösungen

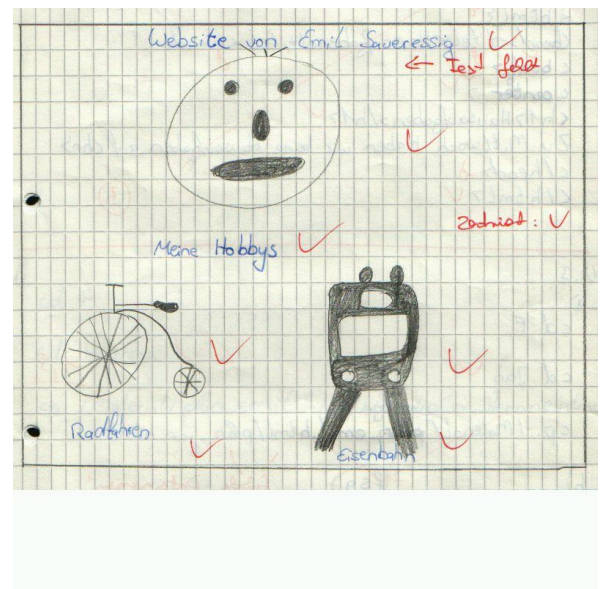
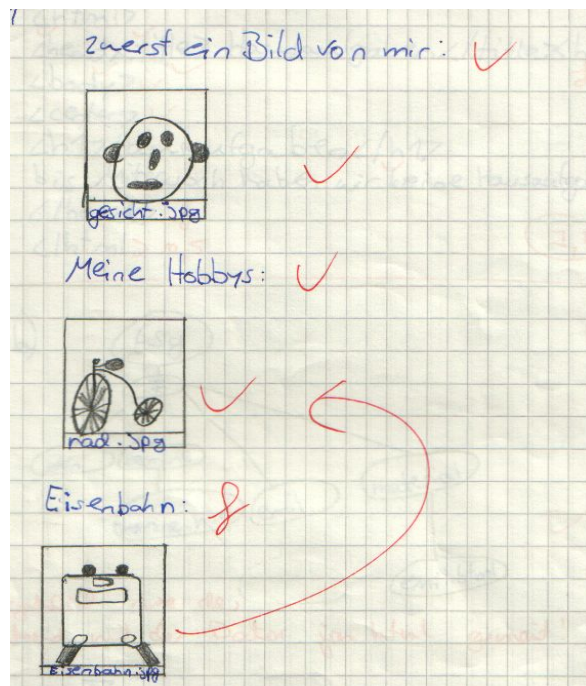


Abbildung 7.13: Beispielhafte Schülerlösungen zu Aufgabe 1 im Test 2



**Test 2/Aufgabe 2: HTML** Marvin hat im Unterricht nicht gut aufgepasst und anschließend schlecht abgeschrieben:

```
{html}
<hut><titel>Hausaufgaben</titel><hut>
<buddy>
  <Zentrum>
  <h1>Hausaufgaben</d1>
  Zu Mittwoch haben wir keine Hausaufgaben auf.
</head>
</htm>
```

Schreibe diese HTML-Datei ohne Fehler neu auf.

### Beispielhafte Schülerlösungen

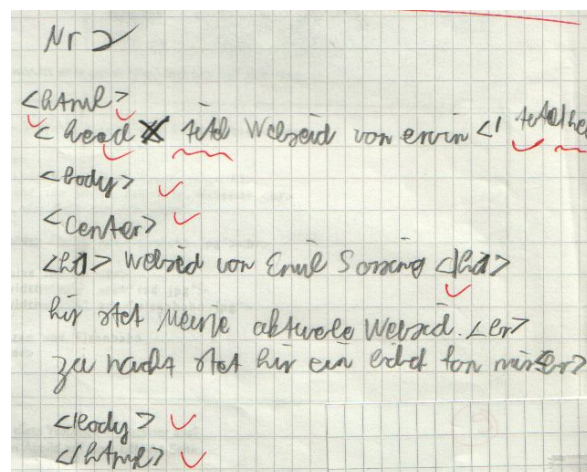
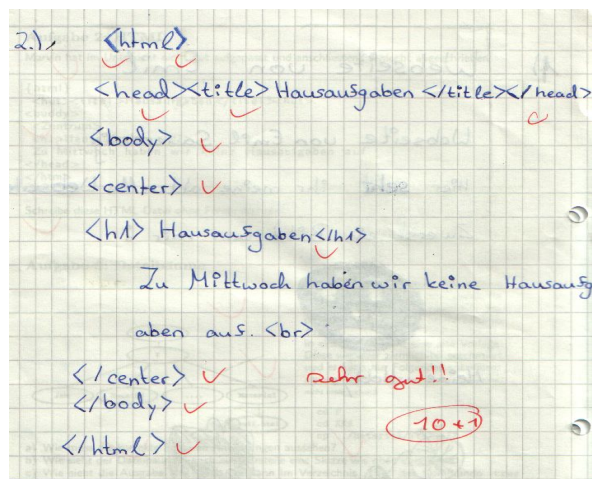


Abbildung 7.14: Beispielhafte Schülerlösungen zu Aufgabe 2 im Test 2

**Test 3/Aufgabe 2: Strecken** Auf einer Leinwand soll eine Strecke gezeichnet werden, die von ganz rechts nach ganz links genau in der Mitte der Leinwand verläuft. Zusätzlich soll auf der Leinwand eine zweite Strecke erscheinen, die von ganz oben nach ganz unten ebenfalls genau in der Mitte der Leinwand verläuft.

- Erstelle eine Zeichnung!
- Wie lauten die zugehörigen Anweisungen in *TCL/TK*?

**Beispielhafte Schülerlösungen**

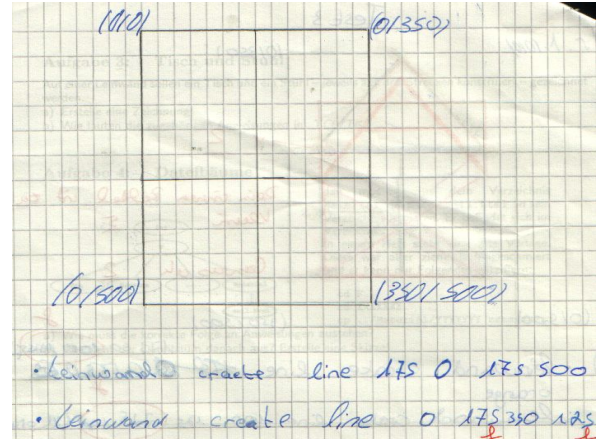
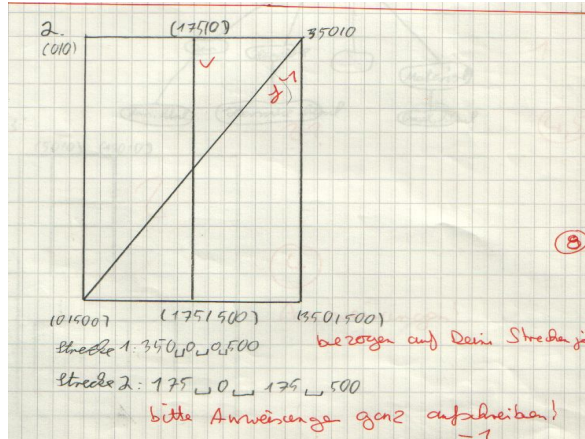


Abbildung 7.15: Beispielhafte Schülerlösungen zu Aufgabe 2 im Test 3

**Test 3/Aufgabe 3: Tisch und Stuhl** Auf einer Leinwand sollen ein Tisch und ein Stuhl – jeweils aus drei Strecken konstruiert – gezeichnet werden.

- Erstelle eine Zeichnung!
- Wie lauten die zugehörigen Anweisungen in *Tcl/Tk*?

**Beispielhafte Schülerlösungen**

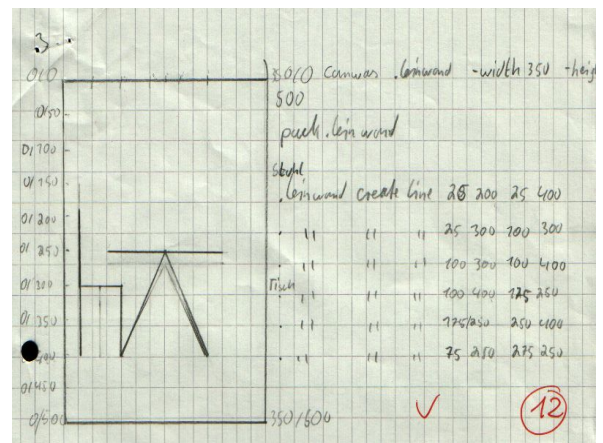
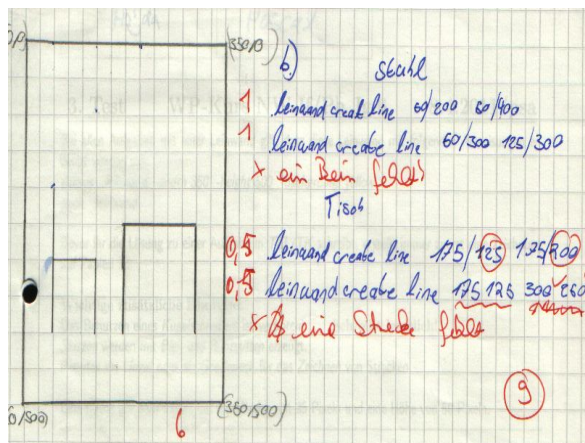


Abbildung 7.16: Beispielhafte Schülerlösungen zu Aufgabe 3 im Test 3

**Test 4/Aufgabe 2: Von Tcl/Tk zu XPM** Folgende Anweisungen werden in der TCL-Konsole eingegeben:

```

canvas .leinwand -width 7 -height 5 -background yellow
pack .leinwand

.leinwand create line 0 3 7 3 -fill blue
.leinwand create line 4 0 4 5 -fill red
    
```

- a) Wie gross ist die erstellte Leinwand?
- b) Was sehen wir auf dieser Leinwand? Erstelle eine Zeichnung! (Wähle 1 Kästchen für 1 Pixel)
- c) Wie üblich, wurde eine XPM-Datei aus diesem Fenster erzeugt. Diese Datei wird mit einem Editor geöffnet. Kopf und Anfang der Datei sehen aus wie folgt:

```

/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"9 7 4 1",
"u c blue",
". c red",
"X c yellow",
"o c gray",
/* pixels */
"oooooooo",
"oXXX.XXXo",
    
```

- d) Wie wird diese Datei dargestellt, wenn der Kopf folgendermassen aussieht, der Rest der Datei aber unverändert bleibt:

```

/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"9 7 5 1",
"X c blue",
". c red",
"u c yellow",
"+ c gray",
"o c orange",
/* pixels */
    
```

Zeichne das entsprechende Bild in Dein Heft!

**Beispielhafte Schülerlösungen**

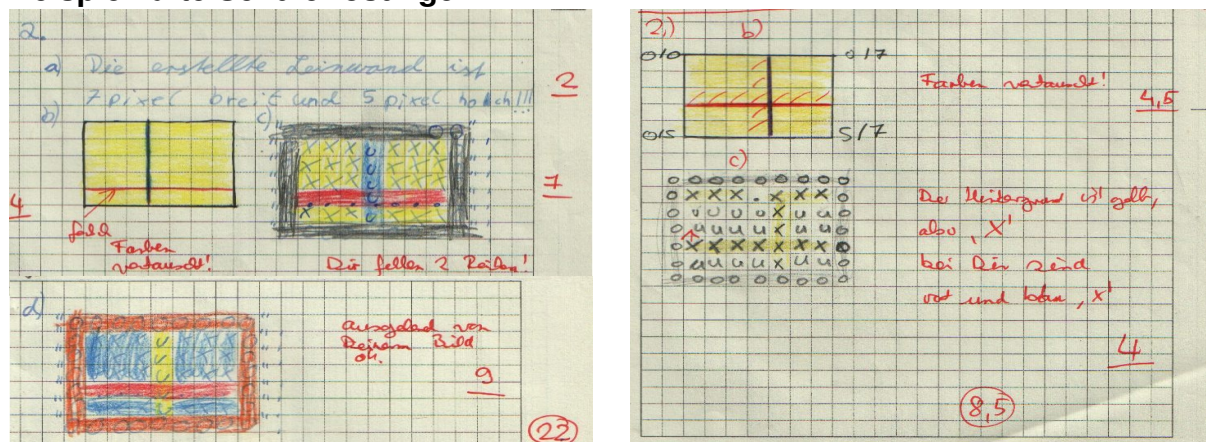


Abbildung 7.17: Beispielhafte Schülerlösungen zu Aufgabe 2 im Test 4

## 7.6.2 Zusammenfassung

Schriftliche Kursarbeiten in einem so frühen Jahrgang wie hier im Jahrgang 6 sind in der Geschichte des Unterrichtsfaches Informatik ungewöhnlich. Die Besonderheit des Hauptfaches erforderte eine grosse Vorsicht, um auch den Schullaufbahninteressen der Schüler gerecht zu werden. Diese Arbeiten bestanden daher im Wesentlichen aus einer Kombination kindgerechter zeichnerischer Darstellung informatischer Fragestellungen sowie teils aus einfachen textuell-formalen Aufgaben. Die Schüler konnten in diesen Aufgaben sehr gut ihr Verstehen, Wissen und Können in Informatik beweisen. Dieser Weg ist erfolgversprechender als die in den Naturwissenschaften üblichen Leistungskontrollen wie es Merzyn kritisiert: „Erfasst wird so vornehmlich kurzzeitiges Behalten und begrenztes Verstehen; Flexibilisierung und Verankerung des Neugelerten im Frühgelernten treten zurück. Derartige Leistungskontrollen begünstigen Selbsttäuschungen darüber, was der Unterricht erreicht.“ [Mer08, S.111] Diese Erkenntnisse aus den Kursarbeiten in Informatik sind auch auf den physikalischen Teil des Unterrichts zu übertragen, da dort die Gefahr besteht, die tradierten Formen der Lernerfolgskontrollen unbewusst zu übernehmen.

# 8 Evaluation

## Überblick

In diesem Kapitel wird die Evaluation des im Kapitel 7 dargestellten Unterrichtsprojekts beschrieben. Der Erfolg oder Misserfolg eines Unterrichtes hängt von verschiedenen Aspekten ab. Ein erster Erfolg war zweifelsohne das Zustandekommen des Kurses an sich. Da es sich um einen Kurs mit Hauptfachcharakter gehandelt hat, bedarf es des Vertrauens von Schüler und Eltern in das Projekt und in die darin involvierten Personen, dass die Teilnahme über viele Jahre für die Schüler zu keinem Nachteil wird. Umso erfreulicher war es, dass dieser Kurs in Klassenstärke zustande kam und damit eine Evaluation eines Informatikunterrichtes unter schulischen Normalbedingungen ermöglicht hat.

Aus Sicht von Schülern sind die schriftlichen Leistungskontrollen kritische schulische Elemente. Diese mussten in diesem Kurs durchgeführt werden, da es sich für die Schüler um ein Hauptfach handelt. Insgesamt vier Kursarbeiten wurden erstellt und stellten eine besondere Herausforderung dar, da es bisher kaum Erfahrungen mit solchen Arbeiten in Informatik in jungen Jahren gibt.

Ein wesentlicher Kern der Evaluation besteht in der Auswertung eines dreiteiligen Fragebogens mit insgesamt über 50 Fragen. Dieser Fragebogen wurde zu Beginn, zur ‚Halbzeit‘, am Ende und erneut im neuen Schuljahr an alle Schüler ausgeteilt und ausgewertet. Verglichen wurden diese Gruppe mit insgesamt fünf Gruppen an einer Realschule und einer Hauptschule, die einen ITG-ähnlichen Unterricht erhalten haben. In der FSG gab es mit einer Klasse ohne jeglichen Schüler im Informatikkurs eine Kontrollgruppe.

Die Selbstauskünfte wurden durch mehrfach durchgeführte Think-Aloud-Untersuchungen überprüft. Ein Ausblick mit einem Vergleich zwischen dem Informatikunterricht und ITG-Unterricht rundet diese Untersuchung ab.

## 8.1 Ergebnisse der Kursarbeiten

Wie schon öfter erwähnt, spielen die Kursarbeiten in diesem Kurs eine wichtige Rolle, sowohl als Indikator für Schüler wie Lehrer bezüglich des augenblicklichen Standes des Gesamtkurses wie auch des einzelnen Schülers als auch über das Jahr hinaus bezüglich des zukünftigen Schulabschlusses. Als Anspruch an das Ergebnis von Kursarbeiten wurde gestellt, dass es für Schülerinnen und Schüler möglich sein muss, gute und auch sehr gute Leistungen zu erzielen. Ebenso sollte ein Ergebnis einer Arbeit auch die Leistungsbreite in einem solchen Kurs widerspiegeln. Denn gerade an einer Gesamtschule mit einer heterogenen Schülerschaft ist nicht davon auszugehen, dass alle Schüler die gestellten Aufgaben gleich gut bewältigen können.

Es wurden im Laufe des Jahres vier Arbeiten geschrieben, zwei in jedem Halbjahr. Die erzielten Punkte in den Arbeiten sind in Abbildung 8.1, die erzielten Noten sind in Abbildung 8.2<sup>1</sup> sehen.

Bei einer ausgeglichenen heterogenen Kursgruppe und eine auf diese Gruppe abgestimmten Kursarbeit ist in der Punkteverteilung etwa mit einer Normalverteilung nach Gauß zu rechnen. Geht man von dieser Annahme aus, so ist zu erkennen, dass im ersten Test eine Lücke zwischen ‚guten‘ und ‚nicht so guten‘ Schülern erkennbar ist. Dies ist nicht ganz ungewöhnlich,

---

<sup>1</sup>Die Notenintervalle sind nicht unbedingt gleich gross, da die Grenzen zwischen den Noten u.a. nach pädagogischen Gesichtspunkten (z.B. lokales Minimum in der Punkteverteilung) im Einzelfall gesetzt werden.



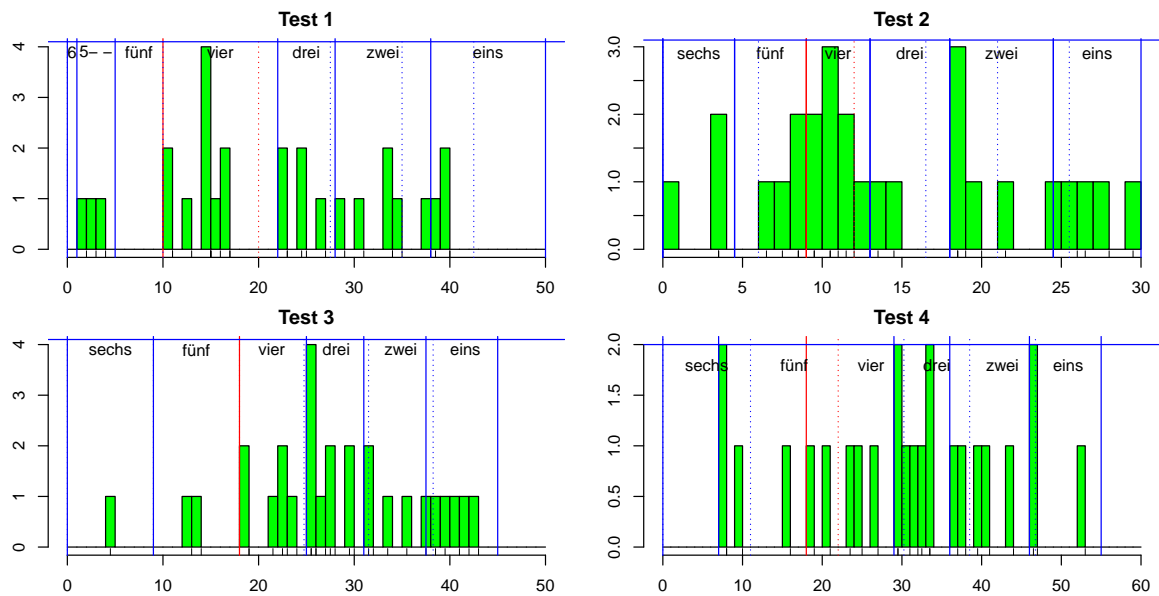


Abbildung 8.1: Histogramme der erzielten Punkte in den Kursarbeiten

(x-Achse: Punkte, y-Achse: Anzahl Schüler)

da manche Schüler zu Beginn eines neuen Faches einige Schwierigkeiten mit den neuen Anforderungen haben. Gelegentlich unterschätzen einige Schüler diese Anforderungen, gerade wenn die ersten Stunden aus ihrer Sicht ‚doch so leicht‘ erscheinen. Diese Lücke wird im zweiten Test noch grösser. Die Gruppe drohte in zwei Teilgruppen zu zerbrechen und die schlechtere davon eventuell den Anschluss an den Unterricht zu verlieren.

Dies wurde sicher gefördert durch die Problematik der Haus- und Wochenplanaufgaben in der FSG als ein typisches Problem einer Gesamtschule. Die Schülerinnen und Schüler haben im Rahmen des Ganztagesunterrichtes zwei Arbeitsstunden für die in allen Fächern gestellten Aufgaben. Da sie nach dem Schulschluss um 15:30 oft erst eine bis  $1\frac{1}{2}$  Stunden später zu Hause sind, werden dort oft nicht in ausreichendem Maße die restlichen Aufgaben erledigt. Daher müssen im normalen Unterrichtsverlauf ausreichend Aufgaben gestellt und bearbeitet werden. Dies galt in unserem Fall noch mehr, da Hilfen von Eltern bei Aufgaben im Fach Informatik kaum zu erwarten waren und auch nicht gesichert war, dass die Schüler zu Hause an einem Computer arbeiten konnten. Ebenso stand in den Klassenräumen in den Arbeitsstunden kein Computer zur Verfügung.

Die Übungsarbeit wurde auf zwei Arten verbessert. Haus- und Wochenplanaufgaben wurden mehr als vorbereitende Aufgaben in schriftlicher Form erstellt, die zur Erledigung keinen Computer benötigten und im Unterricht selber wurden noch mehr Übungsphasen eingerichtet.

Offensichtlich haben diese Maßnahmen ihr Ziel nicht verfehlt. Der dritte und erst recht der vierte Test zeigen eine deutliche Entwicklung zum gewünschten Ergebnis. Auch die Boxplots (Abbildung 8.3) zeigen, dass der negative Trend ab dem 3. Test korrigiert werden konnte.

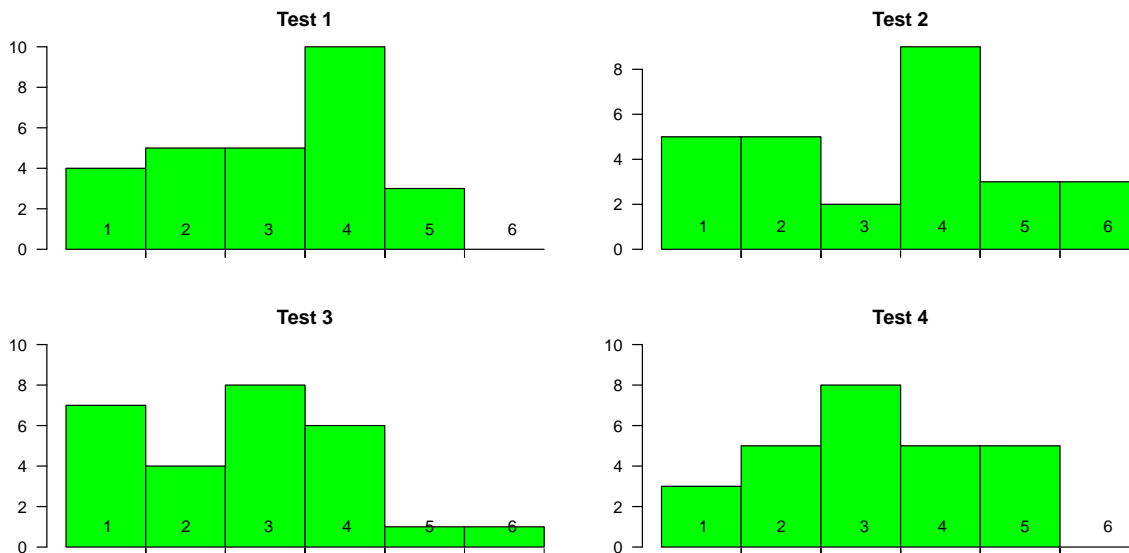


Abbildung 8.2: Histogramme der erzielten Noten in den Kursarbeiten

(x-Achse: Noten, y-Achse: Anzahl Schüler)

### 8.1.1 Zeugnisnoten

Die Berechtigung des Anspruches eines ‚normalen Hauptfaches‘ wird unter anderem durch die Zeugnisnoten nachweisbar. Die Dichteverteilung der Noten der Schüler des Wahlpflichtkurses in den Fächern Mathematik, Naturwissenschaften<sup>2</sup> und Informatik am Ende des 6. Schuljahres entsprechend Abbildung 8.4 zeigt, dass die Noten in Mathematik fast genau der Gaußschen Normalverteilung entsprechen. Von den diesen Fächern hat Mathematik als Fach die längste Tradition, aber auch den grössten Stundenumfang. Es existieren offensichtlich ausreichend Erfahrungen, welche Anforderungen von den Schülern in einer heterogenen Gruppe (auch Mathematik wird im 6. Jahrgang noch undifferenziert unterrichtet) mit befriedigenden Ergebnissen erlernt werden können. Dies gilt auch oder gerade deswegen, weil in dieser Verteilung der Mathematik-Noten die Schüler aus 6 Klassen mit 5 unterschiedlichen Lehrern wiedergegeben sind.

Sowohl die naturwissenschaftlichen als auch die informatischen Noten zeigen zwei Maxima. Die in Informatik liegen deutlich weiter auseinander als in den Naturwissenschaften, bei der ein Maximum ziemlich genau mit dem Maximum in Mathematik zusammenfällt. Berücksichtigt man die Erfahrungen mit dem Erfolg des gesteigerten Übungsanteil in Informatik nach der 2. Kursarbeit, bei der sich eine noch schärfere Zweiteilung andeutete, so kann daraus gefolgert werden, dass sich eine derartige Verteilung wie in Naturwissenschaften und Informatik durch ausreichende Übungen gerade auch für die schwächeren Schüler vermeiden lässt. Mathematik hat als Schulfach generell einen Vorlauf von mehreren Jahren, da es mit dem 1. Schuljahr beginnt, und hat teilweise einen fast doppelt so grossen Stundenanteil wie Natur-

<sup>2</sup>Naturwissenschaften wird an der FSG an Stelle der drei Fächer Biologie, Chemie und Physik vom Jahrgang 5 bis Jahrgang 7 fächerintegriert im Umfang von 2 Wochenstunden à 60 min als Pflichtfach unterrichtet

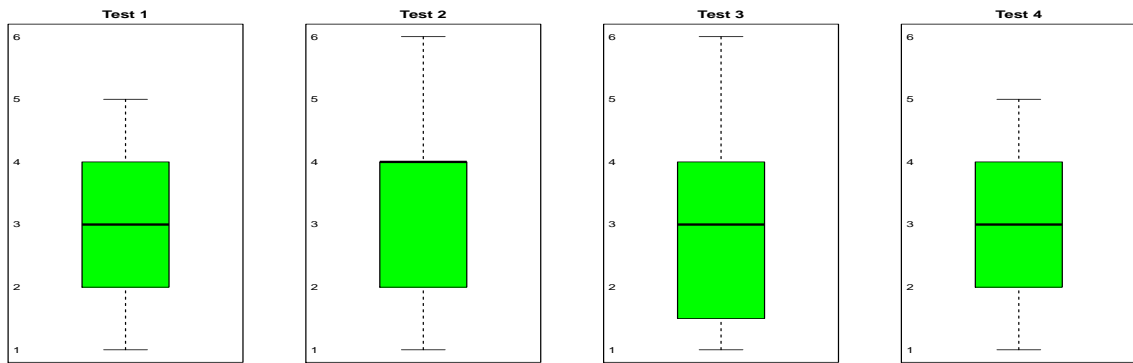


Abbildung 8.3: Boxplots der erzielten Noten in den Kursarbeiten

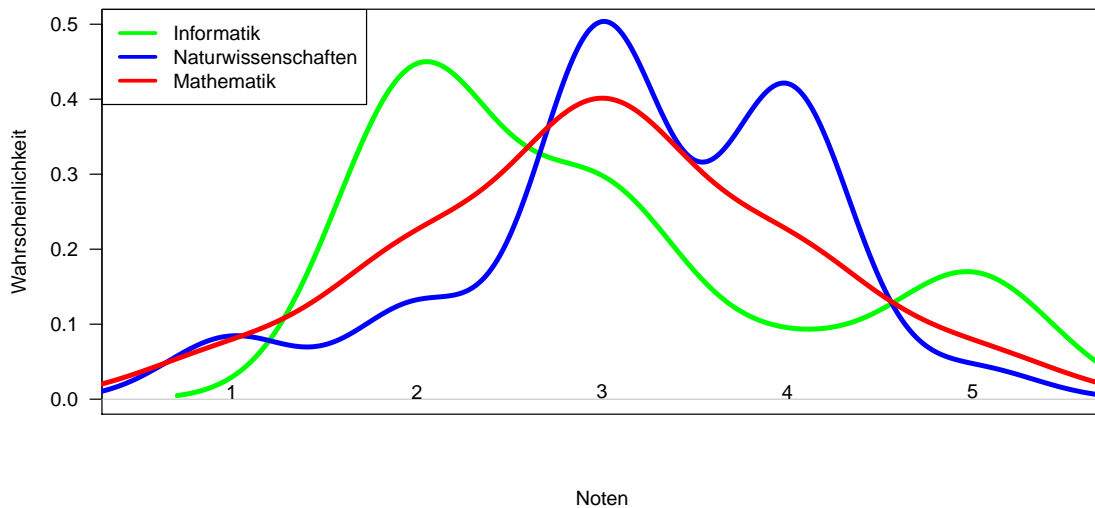


Abbildung 8.4: Dichte der Zeugnisnotenverteilung am Ende der Jahrgangsstufe 6

wissenschaften und Informatik.

Der Ausfall der Noten in Informatik zeigt auch, dass die sicher berechtigten Ängste der Eltern zu Beginn des Schulversuches, ob ihre Kinder eventuell die geforderten Leistungen in einem doch neuen Schulfach nicht erbringen können, nicht eingetreten sind. Nur wenige Schüler konnten zumindest kein ‚ausreichend‘ erzielen.

### 8.1.2 Lieblingsfach

Am Ende des Jahres wurden die Schülerinnen und Schüler befragt, wie sie die drei Schulfächer *Mathematik*, *Naturwissenschaften* und *Informatik* für sich als Lieblingsfach sortieren würden. Sie vergaben je eine ‚eins‘, ‚zwei‘ und ‚drei‘. Abbildung 8.5 zeigt das Ergebnis. Mathematik ist danach mit etwa 50% Zustimmung eindeutig das Lieblingsfach. Das entspricht



allen bekannten Untersuchungen über die Beliebtheit und Unbeliebtheit von Mathematik als Schulfach [Mer08, S.15]. Naturwissenschaften folgt, wobei zu berücksichtigen ist, dass nach den Aussagen einiger Schüler sie den Platz 1 für Naturwissenschaften aufgrund der häufigen Arbeit mit Präsentationen am Computer in Physik begründeten. Dies ist sicher nicht eine typische Unterrichtsform für Naturwissenschaften. Mathematik und Informatik sind etwa gleich oft beliebter als Naturwissenschaften. Insgesamt zeigt sich, dass in dieser Gruppe Informatik ein gleich geachtetes Fach unter anderen geworden ist und damit auch keinen ‚Exoten‘-Charakter trägt.

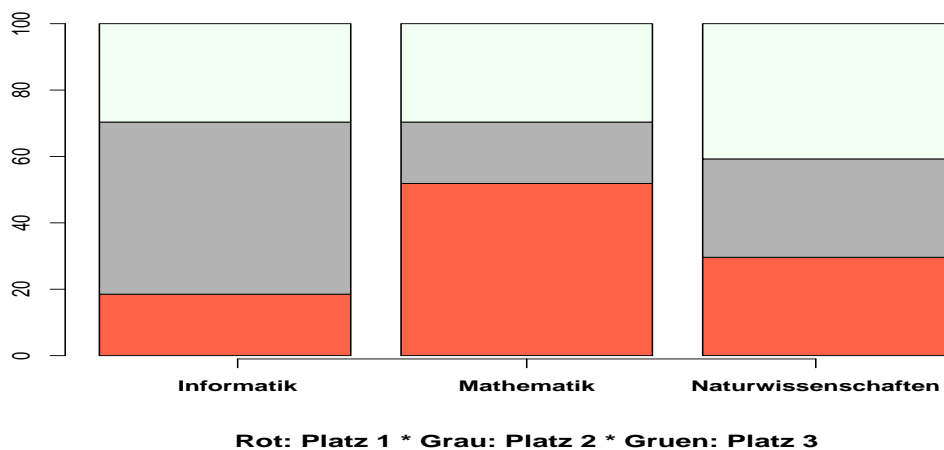


Abbildung 8.5: Lieblingsfach nach einem Jahr Informatikunterricht

### 8.1.3 Zusammenfassung Ergebnisse der Kursarbeiten

Es zeigt sich, dass in einem 6. Jahrgang schriftliche Arbeiten gestellt werden können, die einerseits Seite genügend differenziert Aussagen über das Leistungsvermögens der Schüler ermöglichen und andererseits den Schülern ermöglichen, ihr eigenes Leistungsvermögen einzuschätzen. Ebenso zeigen die Zeugnisnoten am Ende der Jahrgangsstufe 6 im Vergleich zu Mathematik und den Naturwissenschaften, dass ein relativ breites Notenspektrum erzielt wurde. Sowohl die Naturwissenschaften als auch die Informatik weichen (derzeit noch) von der ‚Idealkurve‘ in Mathematik ab. Die Mathematik profitiert offensichtlich von ihrer jahrzehntelangen Praxis als zensurengabendes Fach in schriftlicher und mündlicher Form. Bei der Befragung nach der Beliebtheit zeigt sich, dass Informatik ein normales Fach geworden ist, dass von einigen Schülern sehr geschätzt wird und von manchen weniger. In der Beliebtheit erreicht es aber nicht die Mathematik.

## 8.2 Empirische Untersuchung des Unterrichtsprojekts

Das in den vorigen Kapiteln beschriebene Unterrichtsprojekt ist die Folge aus der Analyse des Unterrichts vor allem des Autors, aber natürlich auch des Unterrichtes anderer Kolleginnen und Kollegen, den man durch mehr oder weniger intensive Zusammenarbeit oder Diskussion zumindest teilweise kennengelernt hat.

Eine Neukonzeption bedarf einer genauen Analyse des bisherigen Unterrichtes. Damit ist aber noch lange nicht garantiert, dass der neu konzipierte Unterricht mehr Erfolg als der bisherige Unterricht erzielt. Neben der üblichen Analyse durch den Unterrichtenden, die natürlich immer persönlich gefärbt ist, ist die *Empirie* ein Mittel, um den durchgeführten Unterricht hinsichtlich seiner Ziele und Ergebnisse einer Messung zugänglich zu machen. "Inhaltliche Theorien und Hypothesen wären gehaltlos, wenn man nicht zeigen könnte, dass es tatsächlich stichhaltige, beobachtbare Fakten gibt, die für diese oder gegen diese Theorien und Hypothesen sprechen.,, [RFHN10, S.VI] formulieren *Björn Rasch* et al. in ihrem Buch *Quantitative Methoden* zur Bedeutung von statistischen Untersuchungen.

Dieses Unterfangen ist jedoch mit einigen Schwierigkeiten verbunden. Eine Messung ist nur dann tauglich, wenn festgestellt werden kann, dass auch das wirklich gemessen wurde, was gemessen werden sollte. Im Gegensatz zu einem klassischen naturwissenschaftlichen Versuch, bei dem beispielsweise die stofflichen Veränderungen nach einem konkreten Ereignis analysiert werden, ist dies bei einer Gruppe von Schülern nicht ganz so einfach. Wer kann schon mit absoluter Bestimmtheit sagen, dass die gemessene Veränderung tatsächlich aufgrund eines konkreten Vorgehens im Unterricht erfolgt ist. Zu viele Einflüsse stürmen auf die Schülerinnen und Schüler ein, sodass hier eine klare Aussage nicht ganz einfach zu treffen ist. Dieser Schwierigkeit kann begegnet werden, indem zumindest mehrere Gruppen von verschiedenen Lehrern nach einem Verfahren unterrichtet werden, andere dem oder den bisherigen Verfahren unterworfen werden und bzw. oder gar keinen Unterricht in diesem Bereich erfahren werden. Durch Vergleiche dieser Gruppen lassen sich zumindest mit grosser Sicherheit Fremdeinflüsse ausschliessen.

Im *Schulfach Informatik* ist ein solches Vorgehen kaum durchzuführen. Hierzu müsste man die entsprechende Anzahl von Gruppen zur Verfügung haben, die – man wie oben beschrieben – einteilen können müsste. Eine solche Anzahl von Gruppen, vor allem noch in der Sekundarstufe I, gibt es aber gar nicht. Es ist die reine Ausnahme, dass überhaupt eine Gruppe Informatik-Unterricht vergleichbar mit anderen Fächern hat. Selbst wenn es solche Gruppen gäbe, wäre aufgrund eines fehlenden gemeinsamen Curriculums eine vergleichende Untersuchung erschwert.

In dieser Untersuchung wird gezeigt, in welcher Form Veränderungen durch den Unterricht in Informatik im Vergleich zu einer Gruppe, die an derselben Schule *keinen* Unterricht in Informatik hat, eintreten. In einem weiteren Vergleich wird untersucht, inwieweit sich messbare Unterschiede im Vergleich zu Gruppen (an anderen Schulen) ergeben, die einen Unterricht im weitesten Sinne einer *Informations- und kommunikationstechnologischen Grundbildung* erfahren haben. Diese Gruppen gehören zu einer Hauptschule in Hagen-Vorhalle und zu einer Realschule in Hagen-Halden. Damit kann zumindest ansatzweise überprüft werden, in welchem Umfang Schüler dieser unterschiedlichen Gruppen informatische Bildung durch ihren jeweiligen Unterricht erhalten oder auch nicht. Ebenso kann ansatzweise gezeigt werden, ob

ein Informatik-Unterricht auch Teile des ITG-Kanons unterrichtlich erreicht oder nicht.

## 8.3 Beteiligte Schüler-Gruppen

An der gesamten Untersuchung sind sieben Schülergruppen beteiligt:

- Wahlpflichtkurs Informatik/Physik an der FSG (8 Schülerinnen und 20 Schüler)
- Klasse 6.2 der FSG als Kontrollgruppe ohne Informatikunterricht (12 Schülerinnen und 15 Schüler)
- Eingangsklasse 5a der Hauptschule Vorhalle (10 Schülerinnen und 8 Schüler)
- Klasse 6a der Hauptschule Vorhalle (10 Schülerinnen und 11 Schüler)
- Klasse 6b der Hauptschule Vorhalle (8 Schülerinnen und 12 Schüler)
- Klasse 6a der Realschule Halden (13 Schülerinnen und 12 Schüler)
- Klasse 6c der Realschule Halden (12 Schülerinnen und 15 Schüler)

### 8.3.1 Gruppen an der Fritz-Steinhoff-Gesamtschule

#### Wahlpflichtkurs Informatik/Physik an der FSG

Der Wahlpflichtkurs Informatik/Physik des Jahrganges 6 entstand Anfang des Schuljahres 2010/2011 aus 8 Schülerinnen und 20 Schülern. Sie entstammten aus allen Klassen des Jahrganges bis auf die Klasse 6.2. Drei Klassen sind in dieser Gruppe recht stark vertreten, während aus den anderen drei Klassen jeweils nur zwei Schüler entstammen. Eine ausführliche Präsentation der Gruppe wurde im Kapitel 7.2 vorgenommen.

#### Klasse 6.2

Da aus der Klasse 6.2 kein Schüler Informatik/Physik gewählt hatte, wurde diese Klasse als Kontrollgruppe (Gruppe ohne Informatik-Unterricht) ausgewählt. In dieser Klasse sind 12 Schülerinnen und 15 Schüler.

### 8.3.2 Gruppen an der Hauptschule Vorhalle

An der Hauptschule in Vorhalle wurden alle Klassen (eine Klasse im Jahrgang 5 und zwei Klassen im Jahrgang 6) im 5. und 6. Jahrgang in die Untersuchung einbezogen. Alle Gruppen hatten in der Woche 2 Stunden Unterricht im Computerraum der Schule.

#### Schulcurriculum an der Hauptschule Vorhalle

Für die Jahrgangsstufen 5 und 6 gibt an der Hauptschule Vorhalle ein einheitliches Curriculum. Dieses Curriculum aus dem Bereich der *ITG* besteht nach einführenden Grundlagen im Wesentlichen mit der Erarbeitung der Textverarbeitung und je nach Klasse etwas Einarbeitung mit

der Tabellenkalkulation<sup>3</sup>. Teilweise wurde dieser Unterricht mit dem Unterricht in Mathematik und Deutsch verzahnt.

Wesentliche Bestandteile des Curriculums für den 5. und 6. Jahrgangs sind<sup>4</sup>:

- Betriebssystem:  
Erlernen der Windows-Oberfläche  
Rechner/Programme starten und herunterfahren  
Grundbegriffe von Windows Bilder (Dokumente) speichern, öffnen und drucken
- PC-Arbeitsplatz:  
Hauptbestandteile eines Computers
- Textverarbeitung:  
Texteingabe und -korrektur  
Zeichenformatierungen  
Absatzformatierungen (Textausrichtung)  
Texte umstellen  
Einfache Tabellen erstellen (z. B. Stundenplan)
- Internet

## **Klasse 5a**

In der Hauptschule gab es in Schuljahr 2010/2011 nur eine Klasse im 5. Jahrgang. Als Besonderheit waren in ihr 3 Schüler enthalten, die im Rahmen eines besonderen Förderbedarfes – sei es durch Verhalten oder durch Lernschwierigkeiten – eine besondere Betreuung erfahren. Daher waren neben dem Lehrer gelegentlich mehrere erwachsenen Personen im Unterrichtsraum anwesend. Die Klasse war relativ klein, in ihr waren zu Beginn nur 10 Schülerinnen und 8 Schüler vertreten. Es zeigte sich dann im Verlaufe des Schuljahres, dass eine im Vergleich zu üblichen Klassen relativ hohe Fluktuation der Schüler vorhanden war. Es gingen sowohl Schüler zu anderen Schulen als auch andere Schüler kamen neu in die Klasse.

## **Klassen 6a und 6b**

Im 6. Jahrgang der Hauptschule Vorhalle gab es zwei Klassen, die beide an der Untersuchung teilnahmen. Die Klasse 6a bestand aus 10 Schülerinnen und 11 Schülern, die Klasse 6b aus 8 Schülerinnen und 12 Schülern. Die Fluktuation war geringer als im Jahrgang 5. Schüler mit besonderem Förderbedarf gibt es nicht in diesen beiden Klassen.

### **8.3.3 Gruppen an der Realschule Halden**

In der Realschule Halden gibt es ein hausinternes Curriculum für den Unterricht in Informatik<sup>5</sup>. Die Schülerinnen und Schüler müssen zum Jahrgang 7 ein Wahlpflichtfach wählen. Zur Wahl stehen neben Französisch auch Informatik und Technik. Damit die Schüler eine rationale

---

<sup>3</sup>Die Arbeit mit einem Tabellenkalkulationsprogramm ist nach dem schulinternen Curriculum eigentlich erst nach Jahrgangsstufe 6 vorgesehen.

<sup>4</sup>Ein Auszug des Lehrplanes Informatik an der Hauptschule Vorhalle befindet sich im Anhang unter 11.13.1

<sup>5</sup>Ein Auszug des Lehrplanes Informatik an der Realschule Halden befindet sich im Anhang unter 11.13.2

Entscheidung fällen können, wird im 6. Jahrgang ergänzend zum Pflichtunterricht Informatik und Technik je 1-stündig unterrichtet.

## Schulcurriculum an der Realschule Halden

Das Curriculum in Informatik umfasst im Wesentlichen Themen des üblichen ITG-Unterrichts, aber auch einige Aspekte, die etwas darüber hinausgehen:

- Sich anmelden und sich abmelden
- Ordner anlegen und Dateien verschieben
- Bildverarbeitung
- Lernen im Internet
- Texte schreiben
- Tabellenkalkulation
- Homepage

Nach Aussagen der Kollegen an der RS Halden zeigt die Praxis allerdings, dass der konkrete Unterricht sehr von dem eigenen Vorwissen und der Vorbildung der jeweilig Unterrichtenden abhängt. In der aktuellen Klasse 6c unterrichtete ein Technik-Kollege die Klasse, in der aktuellen Klasse 6a eine Deutsch-Lehrerin. In der Klasse 6a wurden daher nach eigenen Aussagen der Lehrer auch wesentlich mehr Aspekte des klassischen ITG-Unterrichtes unterrichtet.

## Klassen 6a und 6c

Die Klasse 6a bestand aus 12 Jungen und 13 Mädchen, die 6c aus 15 Jungen und 12 Mädchen. Im Laufe fand eine geringe Fluktuation statt. In der Nähe der Realschule Halden gibt es wenige soziale Brennpunkte. Dieses wirkt sich auf die Schülerpopulation aus<sup>6</sup>.

## 8.4 Untersuchungsmethodik

Der Untersuchungsgegenstand – Unterricht nach dem Prinzip der *verwobenen Roten Fäden* – entwickelte sich in theoretischen Überlegungen aus der langjährigen Praxis des Informatikunterrichtes an der FSG und entsprechender Literatur- und Forschungsarbeit. Zum Informatikunterricht in der Sekundarstufe I allgemein gibt es vielfältige Erfahrungen und Positionen. Dies gilt, auch wenn man konzedieren muss, dass es keinen flächendeckenden Unterricht in Informatik vor allem in der Sekundarstufe I gibt. Somit handelt es sich bei dem Untersuchungsgegenstand nicht um ein neues Forschungsgebiet, für das eine *explorative Untersuchung* [Bor06, S.50] als Voruntersuchung zur begründeten Problemfindung notwendig wäre. Das aus dem bisherigen Unterricht in Theorie und Praxis abgeleitete Konzept wird daher in einer sogenannten *explanativen Untersuchung* [Bor06, S.50] überprüft. Der aus den konzeptionellen Überlegungen gefolgerte Unterricht wurde anschliessend in der Fritz-Steinhoff-Gesamtschule in einer *quasiexperimentellen* Umgebung [Bor06, S.114ff] untersucht. Ob der Informatikunterricht einen Effekt auf das informatische Wissen und Denken der Schüler hat, wird diese Gruppe

---

<sup>6</sup>Die Klasse 6b nahm nicht an der Untersuchung teil.

im Rahmen eines *Zweiggruppenplanes* [Bor06, S.115] mit einer Kontrollgruppe verglichen, in denen kein Schüler an einem Informatikunterricht teilnimmt. Da die Zugehörigkeit zur ersten Gruppe aufgrund der Eigenschaft einer *Wahlpflichtgruppe* nicht nach dem Randomisierungsprinzip [Bor06, 54] erfolgen konnte, ist besonders darauf zu achten, welche Veränderungen in der Beobachtungszeit in den jeweiligen Gruppen erkennbar sind.

Die Schüler der Gruppen in der Realschule Halden und der Hauptschule Vorhalle erhalten alle einen Unterricht, der wesentliche Teile einer *informations- und kommunikationstechnischen Grundbildung* enthält. Die in den Befragungen erzielten Erkenntnisse werden mit beiden Gruppen der Fritz-Steinhoff-Gesamtschule verglichen. Auf diese Weise kann erkannt werden, ob und wie weit der ITG-Unterricht gleiche, ähnliche oder nicht ähnliche Ergebnisse bezüglich des informatischen Denkens erzielt wie ein Informatik-Unterricht. Vergleichbare Gruppen wie in der Realschule Halden und in der Hauptschule Vorhalle mit einem ITG-Unterricht gibt es an der Fritz-Steinhoff-Gesamtschule nicht. Das erschwert natürlich eine vergleichende Untersuchung. Wie bei einer *Feldbeobachtung* [Bor06, S.50] üblich, sind in der normalen sozialen Umgebung viele Faktoren aktiv, die eine direkte Vergleichbarkeit erschweren. Daher ist auch hier den Veränderungen in den jeweiligen Gruppen besondere Aufmerksamkeit zu widmen.

## 8.5 Untersuchungsverfahren

Um eine möglichst genaue Analyse der Schülerinnen und Schüler vorzunehmen, werden alle Schüler befragt. Die Untersuchung besteht aus einem Schülerfragebogen sowie sechs Aufgaben aus dem Biber-Wettbewerb<sup>7</sup> der letzten Jahre. Die Aussagen der Schüler auf den Schülerfragebogen wurden mit *Think-Aloud-Interviews* [Häd10, Kon10] überprüft.

### 8.5.1 Schülerfragebogen

Die Befragung erfolgt mit einem Fragebogen, der aus drei Teilen besteht ist:

In einem ersten Teil wird nach Begriffen und Fakten gefragt. Eine der Fragen heisst beispielsweise: ‚Ich weiss, was eine *Pixelgrafik* ist‘. Diese 29 Fragen werden mit einer Skala *Ja/Nein/Weiss nicht* beantwortet.

In einem zweiten Teil geht es um Einschätzungen von Aussagen bzw. eigener Fähigkeiten. Beispiel: ‚Programmieren ist sehr leicht‘. Diese 20 Fragen können in einer Likert-Skala [May08, S.87ff] der Art: *stimmt gar nicht/stimmt nicht/weder ja noch nein/stimmt/stimmt völlig/weiss nicht* beantwortet werden.

Ebenfalls mit einer derartigen Likert-Skala werden 21 Fragen zu Einstellungen der Schülerinnen und Schüler erfragt. Ein Beispiel: ‚Ein Informatiker sollte ein guter Teamworker sein.‘.

Um Zufallsantworten auszuschliessen, sind zu vielen Fragen Kontrollitems in den Bögen enthalten. Die Fragebögen enthalten Fragen zu Themen der ITG, wie beispielsweise ‚Ich kenne mich sehr gut mit Textverarbeitung aus‘ als auch Fragen, die eher der klassischen Informatik zuzurechnen sind wie beispielsweise ‚Ein Programmtext sollte sehr gut lesbar sein‘ oder ‚Ich weiss, was eine Variable ist.‘

---

<sup>7</sup><http://www.informatik-biber.de>, letzter Zugriff: 24.10.2011

Zusätzlich wurde den Schülern des Wahlpflichtkurses an der FSG am Ende des Schuljahres schriftlich die Frage gestellt, in welcher Reihenfolge sie die Unterrichtsfächer *Mathematik*, *Naturwissenschaften* oder *Informatik* präferieren würden.

## 8.5.2 Überprüfung mit *Think Aloud*-Interviews

Die Gefahr bei Fragebögen mit dem Ziel einer Selbsteinschätzung seitens von Schülerinnen und Schüler ist, dass sie selbst gar nicht wissen können, ob ihre gegebene Antwort korrekt ist oder nicht. Schülerinnen und Schüler wissen auch nicht nach einer Arbeit genau, ob die erstellten Lösungen korrekt sind.

Aus diesem Grunde wurden am Ende des Schuljahres an allen drei Schulen sogenannte *Think aloud - Interviews* [Häd10, Kon10] durchgeführt. Ein Teil der Schüler wurde in kleinen Interviews von ca. 5 Minuten zu zentralen Begriffen gefragt. Diese Interviews wurden anschliessend transskribiert und die Aussagen der Schülerinnen und Schüler tabellarisch analysiert. Auf diese Weise ist es möglich, die Selbsteinschätzungen der Schüler zu überprüfen. Die Anzahl der Interviews war in den verschiedenen Klassen unterschiedlich. Die meisten Interviews wurden in der Gruppe durchgeführt, die den originären Unterricht in Informatik hatte.

## 8.5.3 Aufgaben aus dem *Biber*-Wettbewerb

Der *Informatik-Biber* [PSH09] ist ein Wettbewerb, der unter der Schirmherrschaft der *Gesellschaft für Informatik* von der Initiative *Bundesweit Informatiknachwuchs fördern (BWINF)* als Einstiegsprojekt seit 2007 in Deutschland jeweils im November durchgeführt wird. Er ist der deutsche Ableger des internationalen Wettbewerbs *Beaver-Contest* [HCD11]<sup>8</sup>.

Dieser Wettbewerb wurde erstmals 2004 in Litauen durchgeführt. In den folgenden Jahren schlossen sich immer mehr Länder dieser Idee an. Die Fragen in jedem Jahr werden für ganz Europa einheitlich erarbeitet. Die Schüler bearbeiten die Aufgaben am Computer und gehen ihre Lösungen online ein. Am Ende des Wettbewerbes erhält jeder Teilnehmer eine Urkunde. Dieser Wettbewerb wird in verschiedenen Alterstufen durchgeführt und hat die Aufgabe, Jungen und Mädchen auf die Informatik als Berufsfeld aufmerksam zu machen. Die Idee eines solchen Wettbewerbs ist allerdings nicht originär im Umfeld der Informatik entstanden, schon länger gibt es beispielsweise einen vergleichbaren Wettbewerb in der Mathematik, den *Känguru*<sup>9</sup>. Im Unterschied zur Informatik findet der Mathematik-Wettbewerb nicht online statt.

Da Informatik-Unterricht zumindest in den unteren Klassen nicht vorausgesetzt werden kann, setzen die Aufgaben keine spezifischen Informatikkenntnisse voraus. Die Aufgaben sind in drei Kategorien – leicht \* mittel \* schwer – eingeteilt.

Aus den Wettbewerben der Jahre 2007 bis 2011 wurden sechs Aufgaben ausgewählt. Sie wurden so ausgewählt, dass sowohl verschiedene Schwierigkeitsgrade [Sch09b, Sch10] als

<sup>8</sup><http://www.bebbras.org>, letzter Zugriff: 24.10.2011

<sup>9</sup><http://www.mathe-kaenguru.de/>, letzter Zugriff: 24.10.2011

auch verschiedene Themenbereiche – Algorithmik, (semi-)strukturierte Daten, Betriebssysteme, Multimedia – vertreten waren. Zu jeder Aufgabe wurden die auch im Wettbewerb angegebenen Lösungsvorschläge vorgegeben.

Als Beispiel möge die Aufgabe **A6: Film digital** aus dem Multimedia-Bereich dienen:

*Bei digital gespeicherten Filmen spielt es eine Rolle, wie stark sich aufeinander folgende Bilder des Films unterscheiden.*

*Der Unterschied zwischen zwei aufeinander folgenden Bildern A und B lässt sich wie folgt berechnen:*

*Die Anzahl der Objekte in Bild A, die nicht in Bild B sind,*

*plus die Anzahl der Objekte in Bild B, die nicht in Bild A sind.*

*Die Unterschiedssumme eines Films ist dann die Summe aller Unterschiede zwischen je zwei aufeinander folgenden Bildern.*

*Was ist die Unterschiedssumme der folgenden sechs Bilder?*

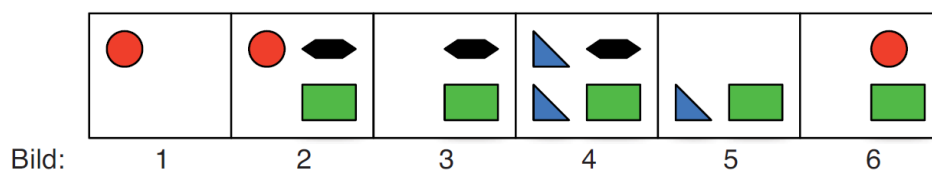


Bild:

1

2

3

4

5

6

8

9

11

13

Schon bei der als erste Gruppe in der Hauptschule Vorhalle befragten Klasse 5 wurde deutlich, dass viele Schülerinnen und Schüler mit diesen auch recht textlastigen Aufgaben grosse Schwierigkeiten hatten und keine Lösung ankreuzten oder sehr unruhig waren.

Daher wurden den Schülern zwei weitere Antwortmöglichkeiten angeboten:

- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.*
- Ich habe die Aufgabenstellung nicht verstanden.*

Da die Schüler am Anfang der Umfrage jeweils darauf hingewiesen wurden, dass es sich bei diesen Umfragen um keine Klassenarbeit handeln würde und auch viele Fragen in den Bögen enthalten seien, die sie noch nicht beantworten könnten, waren sie mit diesen zusätzlichen Möglichkeiten einverstanden.

## 8.6 Auswertung der Fragen aus dem *Biber-Wettbewerb*

Wie oben schon erwähnt, sind die Aufgaben aus dem Biber-Wettbewerb für viele Schüler allein textlich schon sehr schwierig gewesen. Dieses war dem Autor schon ein Jahr zuvor in seiner eigenen Klasse des damaligen Jahrganges 5 aufgefallen. Einige Schüler waren dabei sogar so wenig des Lesens mächtig, dass sie die Aufgaben des damals aktuellen Biber-Wettbewerbes



	A1	A2	A3	A4	A5	A6
HSV Pretest	5.8	14.3	18.9	15.6	13.3	25.0
HSV Posttest	5.0	13.6	10.0	8.3	8.5	10.3
RSH Pretest	11.8	39.2	32.7	30.6	20.4	10.0
RSH Posttest	31.6	65.9	34.1	16.7	39.0	17.1
FSG-KG Pretest	14.3	39.1	36.4	33.3	22.7	22.2
FSG-KG Posttest	16.0	52.0	29.2	17.4	56.5	25.0
FSG-IF Pretest	3.8	74.1	50.0	39.3	14.3	28.6
FSG-IF Posttest	7.7	46.2	48.1	38.5	32.0	48.0

Tabelle 8.1: Richtige Lösungen bei den Fragen aus dem Biberwettbewerb in Prozent

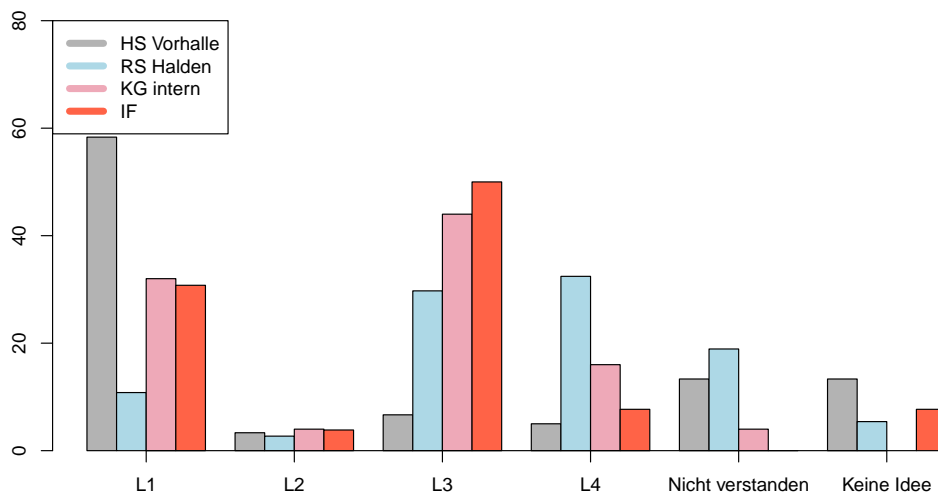
auf Nachfragen völlig falsch und unsinnig wiedergaben<sup>10</sup>. Die weitere Herausforderung war, aus der Frage heraus auf eine Lösungsidee zu kommen und entsprechend dieser Idee eine Lösung zu erarbeiten. Dies erforderte eine hohe Konzentration über einen längeren Zeitraum, die von vielen Schülern noch nicht aufgebracht werden konnte. Sie hatten zwar genügend Zeit, die Aufgaben durchzuarbeiten und sie wurden auch aufgefordert, eine Aufgabe mehrfach zu lesen, wenn sie vorschnell die Aufgabe als 'Nicht-Verstanden' kennzeichnen wollten, aber manche Schüler wollten oder konnten dies nicht tun. Einige Schüler verdeutlichten auch sehr ihren Unmut, derartige Aufgaben lösen zu sollen. Es gab allerdings auch Schüler, die sehr lange und intensiv mit den einzelnen Aufgaben beschäftigt waren.

Es zeigte sich bei der Auswertung, dass das Unwohlsein der Kinder bei der Beantwortung der Aufgaben eine Grundlage hatte. Richtige Antworten waren die Ausnahme. Die Abbildungen (11.8) und die Tabelle 8.1 verdeutlichen dies.

Es ist zu erkennen, dass richtige Antworten teilweise mehr im Pre-, teilweise mehr im Posttest vorhanden waren. Manchmal sind die Schüler an der Realschule Halden erfolgreicher, mal die Schüler an der FSG, aber dort nicht immer die Informatikgruppe. Die Schüler der Hauptschule Vorhalle hatten ganz grosse Probleme, diese Aufgaben zu lösen. Sie schnitten im Vergleich zu allen anderen Gruppen sehr schlecht ab. Bei fast allen Aufgaben liegen die richtigen Antworten der Hauptschüler unter 10%. Dabei fällt der Posttest noch negativer auf. Alle Aufgaben haben die Schüler schlechter gelöst als im Pretest. Bei allen anderen Gruppen ist bis auf Ausnahmen eine Erhöhung der Anzahl der richtigen Lösungen zu erkennen. Da dies auch für die Kontrollgruppe an der FSG gilt, ist davon aus zu gehen, dass die korrekte oder falsche Beantwortung dieser Aufgaben mit dem Informatikunterricht nur bedingt etwas zu tun hat. Vor allem das schlechte Abschneiden der Hauptschüler lässt vermuten, dass bei diesen Aufgaben das allgemeine Leseverständnis mehr ausschlaggebend ist.

Betrachtet man beispielsweise den Ausfall einer konkreten Aufgabe entsprechend Tabelle 8.2 und Abbildung 8.6, so ist erkennbar, dass die Schüler ihre Antworten breit streuen. Auffällig sind hier auch wieder die Antworten vieler Hauptschüler, dass sie die Aufgabe nicht verstanden oder lösen konnten.

<sup>10</sup>Diese Schwierigkeit auf der Schülerseite fällt im Online-Wettbewerb nicht weiter auf, da dann zumeist seitens des Schülers eine (falsche) Antwort angekreuzt wird.



A1: Bibers Geheimcode (Loesung: L4)

Abbildung 8.6: Lösungsangaben zu Biberaufgabe 1 im Posttest

## 8.7 Auswertung des Fragebogens ‚Wissen‘

In der Umfrage wurden Fragen nach Inhaltsbereichen gestellt, die teilweise von einigen oder sogar allen Schülern nicht beantwortet werden konnten, weil die Themen entweder in der jeweiligen Gruppe nicht im Unterricht besprochen wurden oder sogar im Unterricht keiner Gruppe besprochen wurden. Auf diese Weise ist erkennbar, ob für bestimmte oder sogar viele Themen Unterricht notwendig ist und das Wissen darüber nicht per Alltagswissen vom Schüler aufgenommen wird. Da ein Unterricht im 6. Jahrgang in dieser Form noch nicht durchgeführt worden ist, ergaben sich dadurch auch Freiheiten im konkreten Unterrichtsverlauf. Die erfragten Inhalte sind jedoch alle Bestandteile eines Informatikunterrichts in den Anfangsjahren entsprechend der Bildungsstandards [Arb08]. Ergänzende quantitative Untersuchungen finden sich im Kapitel 8.12.1.

### 8.7.1 Programmiersprachliche Konzepte

#### Die bedingte Anweisung

In einem Unterricht, der sich an den Zielen der *ITG* orientiert, sind programmiersprachliche Konzepte nebensächlich. Aber auch im Informatik-Unterricht wurden im ersten Jahr nur Ansätze angewandt, da im Informatik-Unterricht Elemente der Programmierung und von Programmiersprachen vor allem zu Beginn der Unterweisung in Informatik nicht im Vordergrund stehen. In den verschiedenen Unterrichtseinheiten wurden jedoch vorbereitend Grundlagen für das algorithmische Denken vor allem von Sequenzen gelegt und teilweise deren sprachliche

	L1	L2	L3	L4	Nicht ver- standen	Keine Idee
HSV Pretest	46.2	1.9	15.4	5.8	3.8	26.9
HSV Posttest	58.3	3.3	6.7	5.0	13.3	13.3
RSH Pretest	39.2	2.0	23.5	11.8	7.8	15.7
RSH Posttest	10.5	5.3	28.9	31.6	18.4	5.3
FSG-KG Pretest	47.6	0.0	33.3	14.3	0.0	4.8
FSG-KG Posttest	32.0	4.0	44.0	16.0	4.0	0.0
FSG-IF Pretest	38.5	7.7	46.1	3.8	3.8	0.0
FSG-IF Posttest	30.8	3.8	50.0	7.7	0.0	7.7

Tabelle 8.2: A1: Bibers Geheimcode (Lösung: L4)

Formulierung auf Scriptebene praktiziert.

Kontrollstrukturen haben die Schüler in der Informatikgruppe und erst recht die Schüler der anderen Gruppen noch nicht kennengelernt. Daher müssten praktisch alle Schülerinnen und Schüler die Frage entsprechend Abbildung [8.7] nach der *bedingten Anweisung* mit *Weiss nicht* beantworten. Es beantworten aber ca. 30% aller Schüler, dass sie diesen Begriff kennen würden.

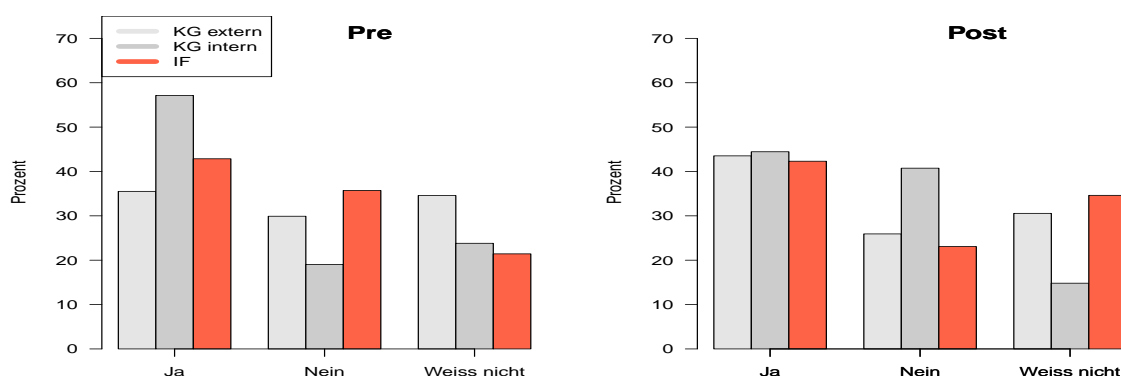


Abbildung 8.7: Ich weiss, was eine *bedingte Anweisung* ist.

Bei der Kontrollfrage zur bedingten Anweisung wird jedoch deutlich, dass der hohe zustimmende Prozentsatz bei der Frage nach dem Begriff sich nicht inhaltlich begründen lässt. Abbildung [8.8] zeigt, dass doch weit mehr als die Hälfte aller Schüler erkennen, dass sie diese Frage nicht beantworten können.

Diese algorithmischen Grundstrukturen werden in den Bildungsstandards [Arb08, S.15ff] im Inhaltsbereich *Algorithmen* für die Jahrgangsstufen 7 bis 10 verlangt.

Auffallend ist, dass die Schüler der Informatik-Gruppe deutlicher erkennen, dass sie diese Frage nicht sinnvoll beantworten können.

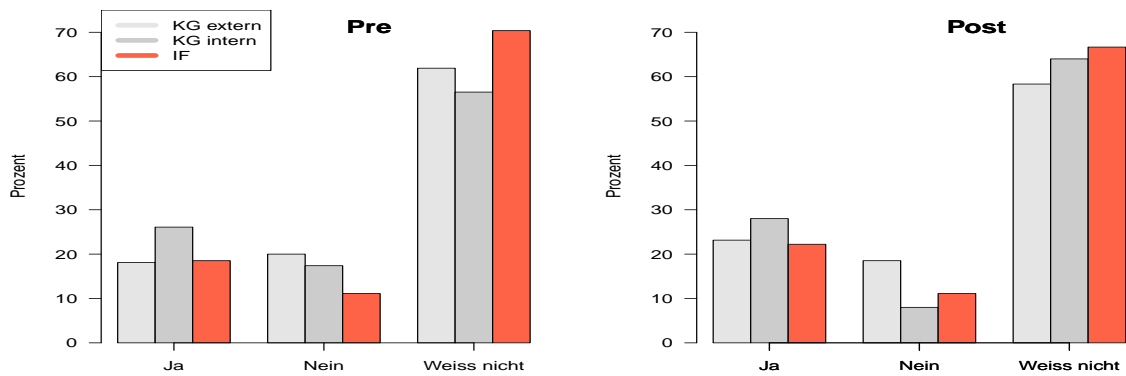


Abbildung 8.8: Mit Hilfe einer *bedingten Anweisung* kann der Computer beispielsweise unterschiedlich reagieren, wenn eine eingegebene Zahl grösser oder kleiner bzw. gleich 7 ist.

## Wiederholungsanweisungen

Auch Wiederholungsanweisungen sind im ersten Jahr noch nicht besprochen worden. Entsprechend Abbildung [8.9] ist jedoch in allen Gruppen die Mehrheit der Meinung, dass diese Antwort richtig oder falsch ist, die Unentschlossenen sind deutlich in der Minderheit. Offensichtlich ist die Bezeichnung *Wiederholungsanweisung* für die Schülerinnen und Schüler *sinnstiftend*, sodass sie hier glauben, antworten zu können. Auch bei dieser Fragestellung sind die Schülerinnen und Schüler der Informatik-Gruppe in der Beantwortung der Frage vorsichtiger.

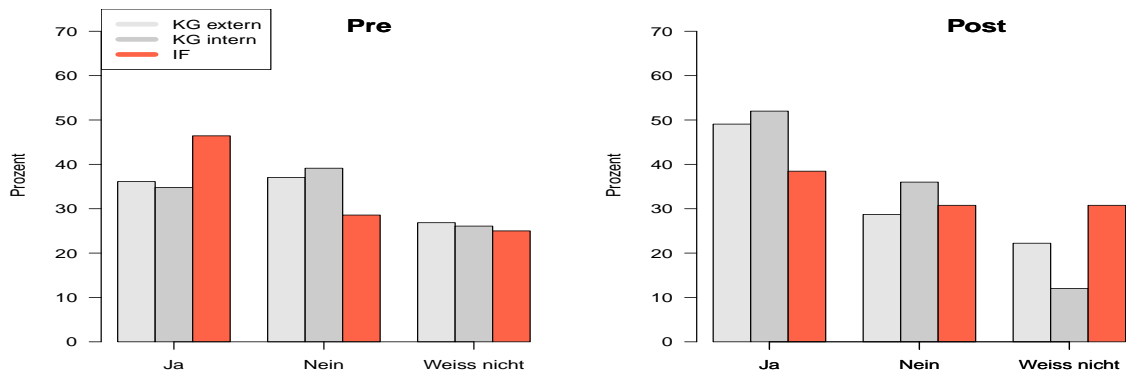


Abbildung 8.9: Ich weiss, was eine *Wiederholungsanweisung* ist.

Die Abbildung [8.10] zeigt jedoch, dass die Kontrollfrage, in der der Begriff mit seinem Inhalt verknüpft wird, von den meisten Schülern zu Recht mit *Weiss nicht* beantwortet wird. Die Schüler aus der Informatik-Gruppe sind hierbei wieder die selbstkritischsten Schüler.

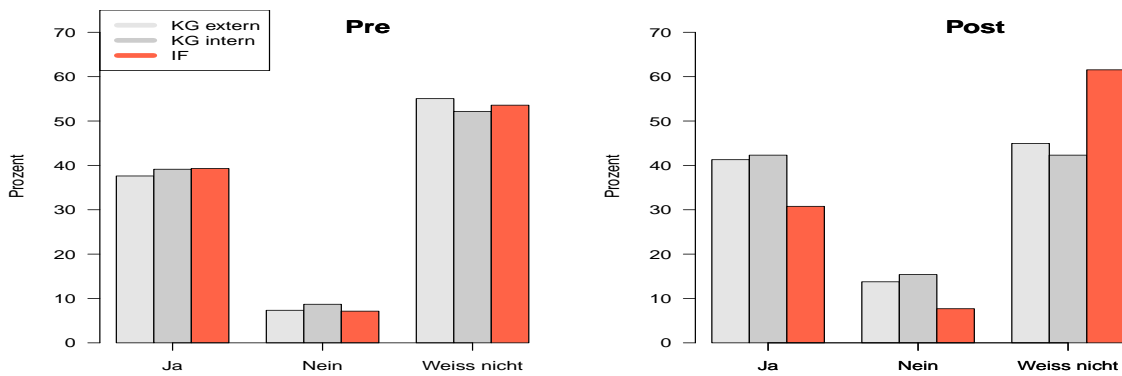


Abbildung 8.10: Um die Zahlen 1 bis 100 zu summieren, kann man im Programm eine *Wiederholungsanweisung* verwenden.

### Algorithmusbegriff

Erst ansatzweise haben die Schüler Handlungsvorschriften kennengelernt und ausprobiert. Erst wenn das Konzept, solche Handlungsvorschriften und ihre Eigenschaften im Unterricht generalisiert worden ist, rechtfertigt sich aus Schülersicht ein neuer Begriff. Die Schüler der Kontrollgruppen kennen unterrichtlich diesen Begriff und die dahinter stehenden Konzepte erst recht nicht. Auch im Mathematik-Unterricht wurde dieser Begriff nicht thematisiert, aber auch kein konkreter Algorithmus zur Berechnung einer Grösse eingeführt und verwendet.

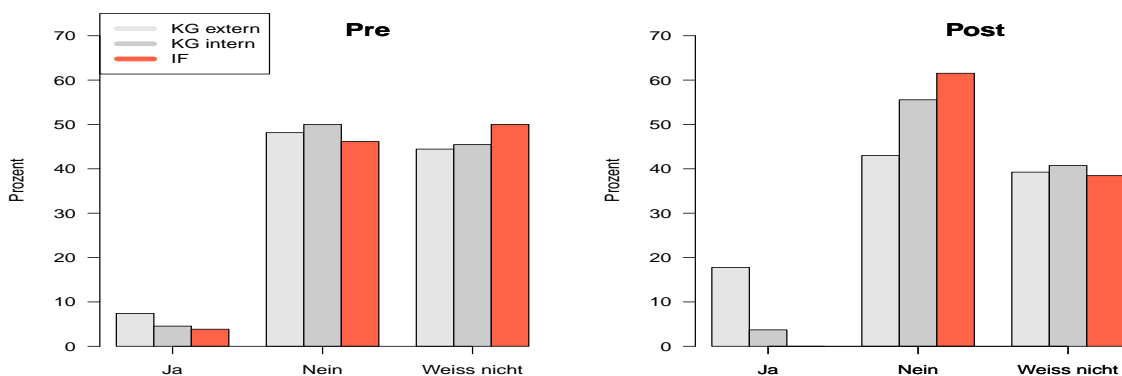


Abbildung 8.11: Ich weiss, was ein *Algorithmus* ist.

40% aller Schüler sagen zu Recht, dass sie den Begriff nicht kennen, 60% meinen allerdings, sie wüssten, was dieser Begriff bedeutet. Dabei meinen sie fälschlicherweise in der überwiegenden Zahl, dass der Begriff Algorithmus keine eindeutige Handlungsanweisung darstellt.

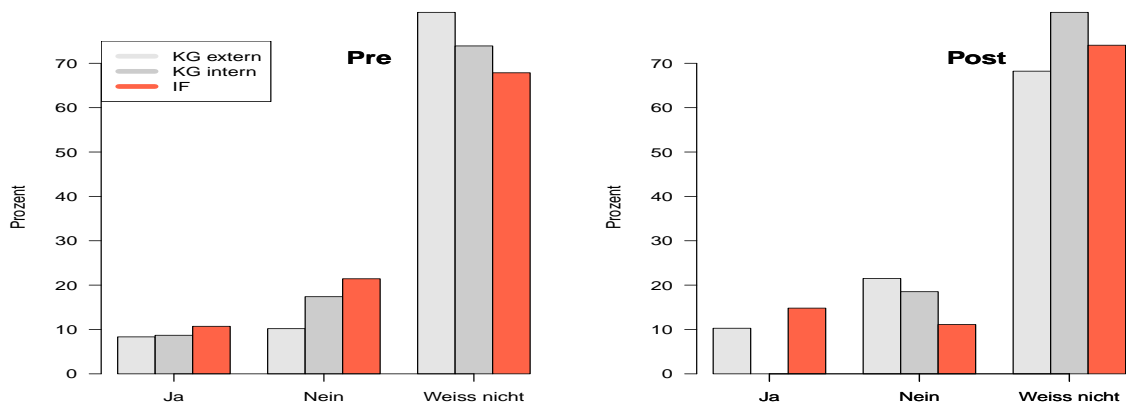


Abbildung 8.12: Eine eindeutige Handlungsvorschrift wird als Algorithmus bezeichnet.

Die Kontrollfrage entsprechend Abbildung [8.12] zeigt dagegen erwartungsgemäss, dass die Schülerinnen und Schüler in ihrer grossen Mehrheit den Begriff tatsächlich inhaltlich nicht zuordnen können. Sie erkennen dieses auch, indem ca. 70% entsprechend *Weiss nicht* angeben.

## Prozeduren

Auch *Prozeduren* und *Funktionen* sind im Informatik-Unterricht noch nicht eingeführt worden. Es verwundert daher nicht, dass Schüler aller Gruppen überwiegend angeben, dass sie den Begriff *Prozedur* entsprechend Abbildung [8.13] nicht kennen. Am Ende des Jahres antworten die verschiedenen Schülergruppen nachvollziehbar vergleichbar, während zu Beginn des Jahres die Informatik-Schüler eher eine nicht begründbare positive Antwort abgeben.

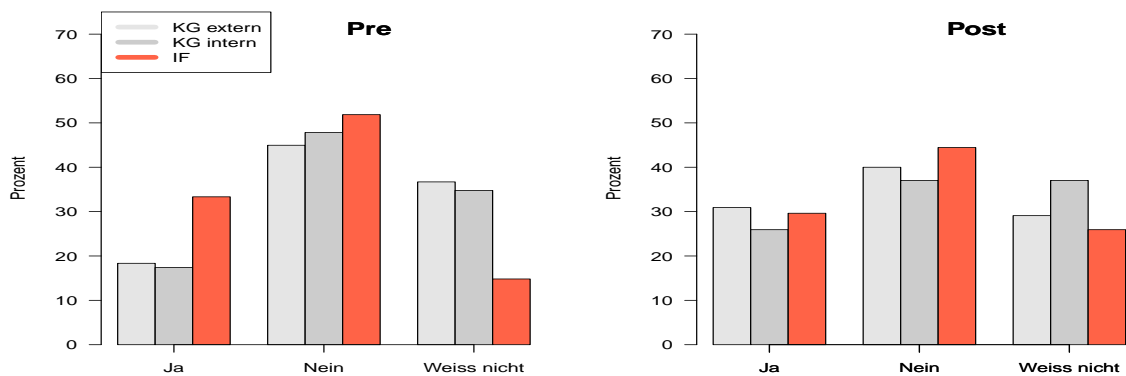


Abbildung 8.13: Ich weiss, was eine *Prozedur* ist.

Allerdings haben die Schüler der Informatik-Gruppe öfters beispielsweise in *Tcl/Tk* im Editor Anweisungsfolgen erstellt und diese dann ausprobiert oder sogar nach einer Woche weiterbearbeitet. Diese sind noch nicht als *Prozedur* formuliert worden. Daher wird die Frage

in Abbildung [8.14] von ca. 50% der Schüler korrekt beantwortet. Da sie schon häufig Anweisungsfolgen als Datei gesichert haben, können einige auch dieses Vorgehen mit der Frage verbunden haben. Die Schüler der Kontrollgruppen haben in ihrem Unterricht derartiges Vorgehen noch nicht praktiziert und müssten mit *Weiss nicht* antworten. Aber auch von ihnen antworten ca. 50% mit *Ja*. Es ist daher fraglich, ob die Schüler der Informatik-Gruppe ihre teilweise korrekte Antwort aus obigen Überlegungen gefolgert haben.

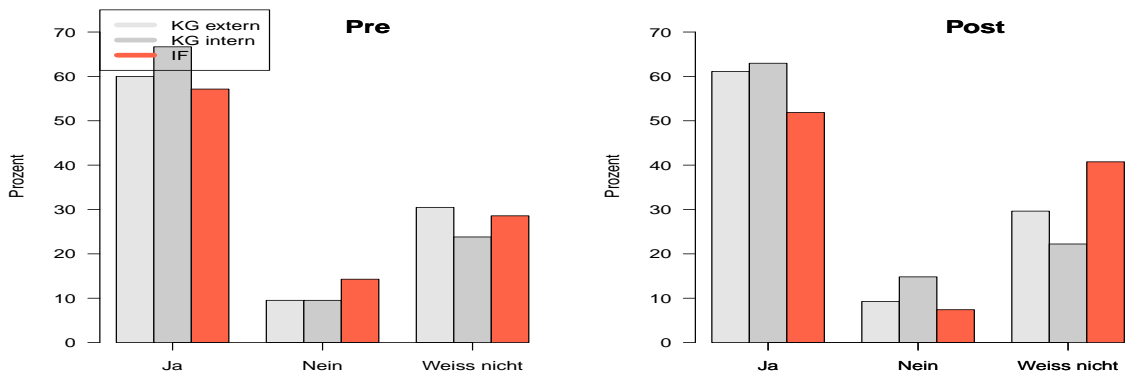


Abbildung 8.14: Mehrere Anweisungen können in einem Computerprogramm unter einem Namen zusammengefasst werden.

Es fällt auf, dass beim Pretest die Anzahl der unsicheren Schüler etwa gleich ist, während beim Post-Test der Anteil der unsicheren Schüler in der Informatik-Gruppe am grössten ist. Das lässt darauf schliessen, dass diese Schüler Frage genauer lesen und verstehen können und ihr eigentliches Defizit besser als die anderen Schülergruppen einschätzen können.

## Zusammenfassung

In den obigen Abschnitten wurden die Fragen ausgewertet, die sich auf programmiersprachliche Konzepte beziehen, die im bisherigen Unterricht aller Gruppen nicht unterrichtet worden sind. Es zeigt sich, dass die dahinter stehenden Konzepte nicht bekannt sind. Die Antworten der Schüler aus den verschiedenen Gruppen unterscheiden sich dabei kaum.

Wird nach der Kenntnis eines Begriffes gefragt, sind die Schüler teilweise bereit, anzugeben, dass sie den erfragten Begriff kennen würden, auch wenn sich in der inhaltlichen Kontrollfrage oft zeigt, dass sie diesen Begriff tatsächlich inhaltlich nicht füllen können. Tendenziell neigen die Schüler der Informatik-Gruppe eher dazu, ihr Wissen bzw. Nicht-Wissen besser einschätzen zu können.

## Variablenbegriff

Am Ende des Schuljahres wurde in der Informatik-Gruppe das Konzept einer *Variablen* behandelt und eingeführt. In den Kontrollgruppen spielte dieser Begriff keine Rolle. In den Bildungsstandards [Arb08, S.16] wird dieser Begriff jedoch erst ab Jahrgangsstufe 8 verlangt.

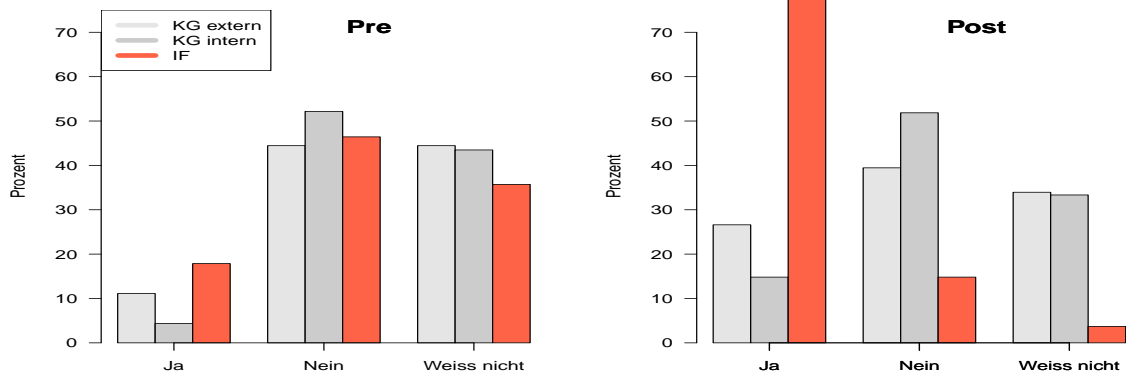


Abbildung 8.15: Ich weiss, was eine Variable ist.

Abbildung [8.15] zeigt, dass der Begriff von ca. 85% der Informatikschüler erinnert wird. Auch wenn der Begriff in den letzten Wochen jede Stunde thematisiert worden ist, gibt es einige wenige Schüler, die diesen Begriff nicht kennen.

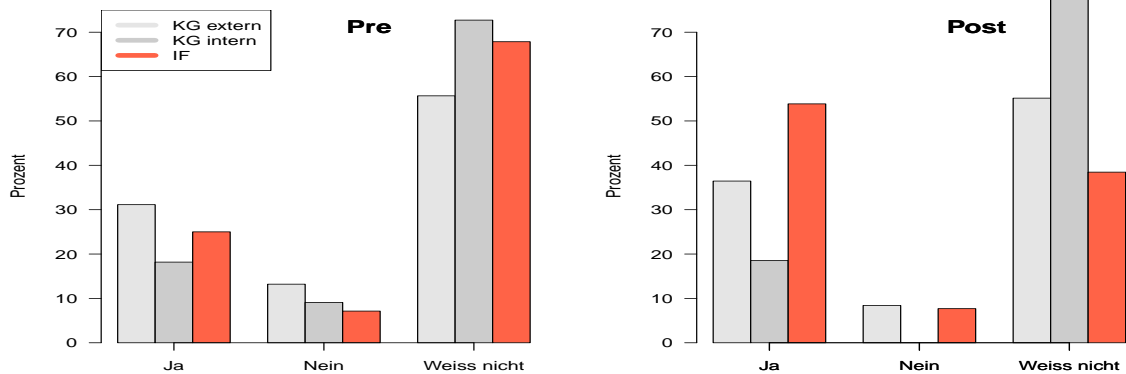


Abbildung 8.16: Ein Wert kann in einer Variablen gespeichert werden.

In der Kontrollfrage zur inhaltlichen Bedeutung des Variablenbegriffes entsprechend Abbildung [8.16] wird deutlich, dass nicht alle Schüler diesen Begriff voll erfasst haben. Allerdings geben über 50% eine korrekte Antwort. Da dies eine erste Einführung dieser Thematik war, kann behauptet werden, dass im weiteren Verlauf des Unterrichtes in den nächsten Jahren zu erwarten ist, dass viele Schülerinnen und Schülern ein umfassendes und korrektes Verständnis dieses wichtigen Begriffes erlangen werden.



## 8.7.2 Graphik

In den verschiedensten Einheiten im Verlaufe des Jahres wurden Grafiken erzeugt und bearbeitet. Sie wurden beispielsweise durch ein Tcl/Tk-Script als Vektorgrafik produziert, in ein Pixelformat konvertiert und anschliessend weiter bearbeitet. Dieses entspricht den Forderungen der Bildungsstandards im Inhaltsbereich *Information und Daten* [Arb08, S.14].

### Das Pixelformat

Entsprechend kann ein Verständnis über den Begriff einer *Pixelgrafik* und seiner inhaltlichen Bedeutung erwartet werden. So zeigt Abbildung [8.17], dass der schon zu Beginn recht hohe Bekanntheitsgrad des Begriffes nach dem Unterricht des ersten Jahres bei allen Schülern zugenommen hat. Auch im Unterricht der Kontrollgruppen in der Hauptschule Vorhalle und Realschule Halden ist im Rahmen der Arbeit mit Textverarbeitungssystemen mit Grafiken umgegangen worden, sodass ein Teil der Schüler zumindest diesen Begriff kennt. Vor allem bei den Informatik-Schülern ist der Begriff praktisch allen geläufig.

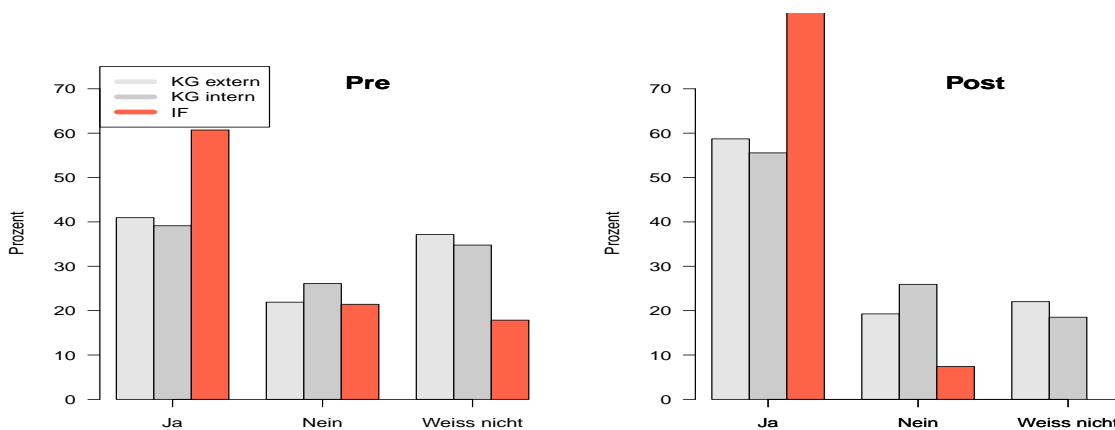


Abbildung 8.17: Ich weiss, was eine *Pixelgrafik* ist.

In der Informatik-Gruppe ist besprochen worden, dass neben dem behandelten `xpm`-Format weitere Pixel-Formate existieren. Diese sind auch im Rahmen anderer Einheiten konkret verwendet worden. Dabei handelt es sich vor allem um die Formate `jpg` und `gif`. Erwartungsgemäss kennen daher die Schüler der Informatik-Gruppe zum überwiegenden Teil diese Formate. Auch die Schüler der externen Kontrollgruppen kennen die konkreten Formate besser als zu Beginn des Jahres, aber in deutlich abgeschwächter Form. Nur die Schüler der internen Kontrollgruppe haben grössere Schwierigkeiten mit diesen Formaten (Abbildung [8.18] und Abbildung [8.19]).

### Das Vektorformat

Im Unterricht der Informatik-Gruppe ist schon früh das Vektorformat eingeführt worden. Die Formulierung eines Skriptes in Tcl/Tk, das ein vektorisiertes Bild beschreibt, macht diese

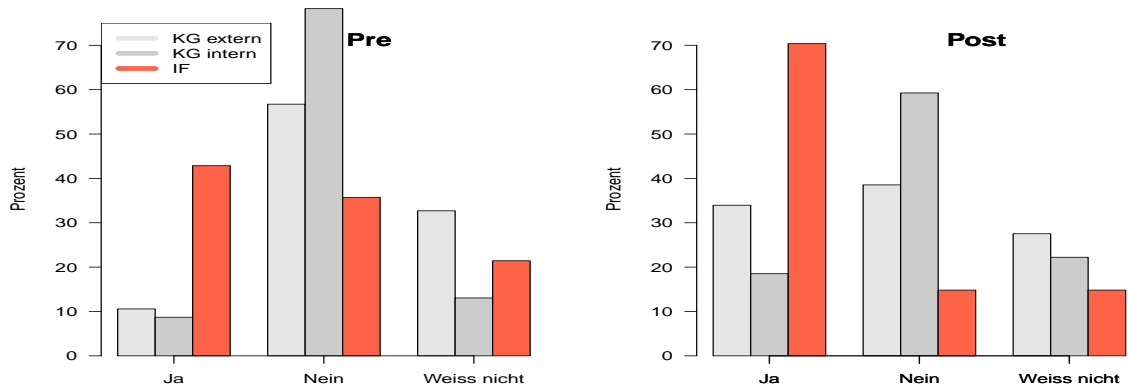


Abbildung 8.18: Ich kenne das *JPG-Format*.

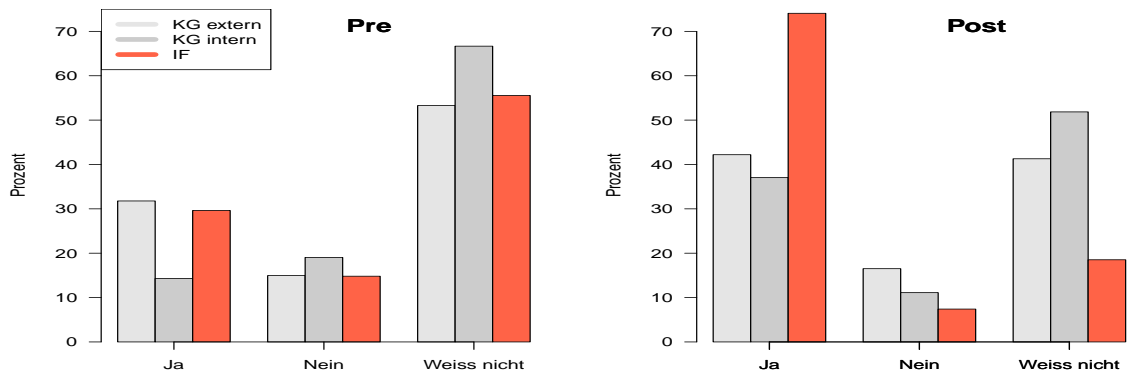


Abbildung 8.19: Ich kenne das *GIF-Format*.

strukturell andere Darstellung bereits offensichtlich. Natürlich sind nach einem Jahr noch nicht alle Unterschiede, Gemeinsamkeiten, Vor- und Nachteile der Formate den Schülern bewusst, jedoch verfügen sie schon über ein Basiswissen, das in den nächsten Jahren ausgebaut werden wird. Abbildung [8.20] zeigt, dass die Schüler aus der Informatikgruppe dieses auch entsprechend beantworten. Die Schüler der Kontrollgruppen zeigen auch einen höheren Kenntnisgrad als zu Beginn des Schuljahres, das aber deutlich von dem der Informatikgruppe abweicht.

In den *Think-Aloud* Gesprächen wurde zudem getestet, in wie weit die Kenntnis der Bildformate auch inhaltlich unterlegt ist. Es zeigte sich dabei, dass das Vektorformat inhaltlich noch einer Reihe von Schüler Schwierigkeiten bereitet, die in den nächsten Jahren aufgearbeitet werden müssen.

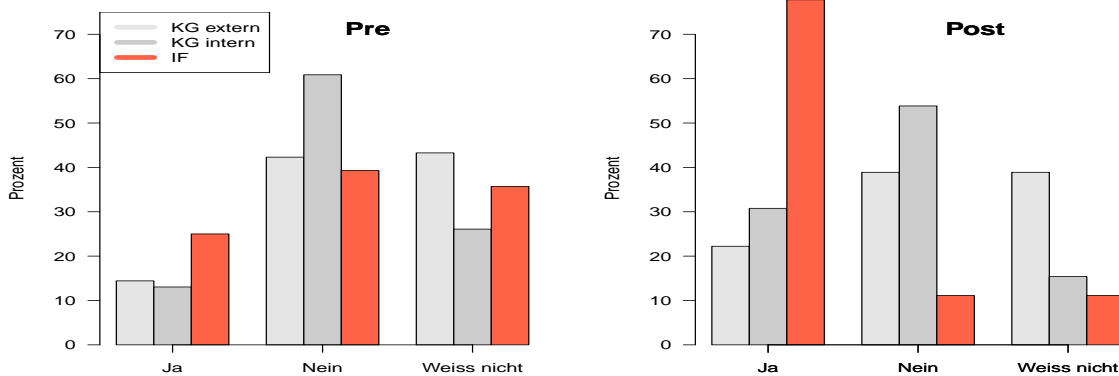


Abbildung 8.20: Ich weiss, was eine *Vektorgrafik* ist.

## Zusammenfassung

Es ist bei allen Gruppen festzustellen, dass die Begriffe *Pixelformat* und *Vektorformat* nach einem Jahr vielen Schülern bekannter sind als zu Beginn der Untersuchung. Auch die konkreten Pixelformate *gif* und *jpg* sind auch bekannter geworden, wenn auch teilweise nicht so deutlich. Der Unterricht der Informatikgruppe, der diese Formate an den verschiedensten Stellen behandelt hat, hat einen enorme Steigerung in der Kenntnis verursacht. Aber auch die externen Kontrollgruppen, die teilweise mit Grafiken gearbeitet haben, verzeichnen einen Kenntniszuwinn. An der internen Kontrollgruppe wird deutlich, dass auch die Alltagserfahrungen der Schüler in diesem Alter zu einer Zunahme im Wissen um diese Formate führen, auch wenn die Mehrheit mit den Formaten nichts anfangen kann. Ein Unterricht, der die Eigenschaften und Bedeutung der Formate aufgreift, ist daher gerechtfertigt.

### 8.7.3 Netze

In den Bildungsstandards [Arb08, S.17] wird von Schülerinnen und Schüler der Jahrgangsstufe 5 bis 7 verlangt, dass sie „*unterscheiden lokale von globalen Netzen*“ und sie „*arbeiten in Netzen*“. Dazu müssen sie einige Begriffe und Fakten aus dem Themengebiet Netze kennen und verstanden haben. Dies erfährt die heutige Jugend sicher teilweise schon durch das alltägliche Handeln zu Hause und in der Freizeit. Alle Schüler haben zusätzlich in der Schule teilweise unabhängig von dem Unterricht in Informatik gelernt, im Netz der jeweiligen Schule zu arbeiten. Insofern ist die Beantwortung der Frage nach Netzwerken entsprechend Abbildung [8.21] nicht überraschend. Überraschend dagegen ist sicher die ‚vorsichtigere‘ Beantwortung seitens der Informatikgruppe. Vielleicht sind einige Schüler durch den Ausdruck *Netzwerke* statt *Netze* verunsichert worden, da sie schon gelernt haben, informatische Fragen differenzierter als die Kontrollgruppen zu lesen.

Dies zeigt sich bei den beiden Fragen zur Eigenschaft des *Internets* als Zusammenfassung von Netzwerken. Fast alle Schüler erkennen, dass das Internet aus vielen einzelnen Netzen besteht.

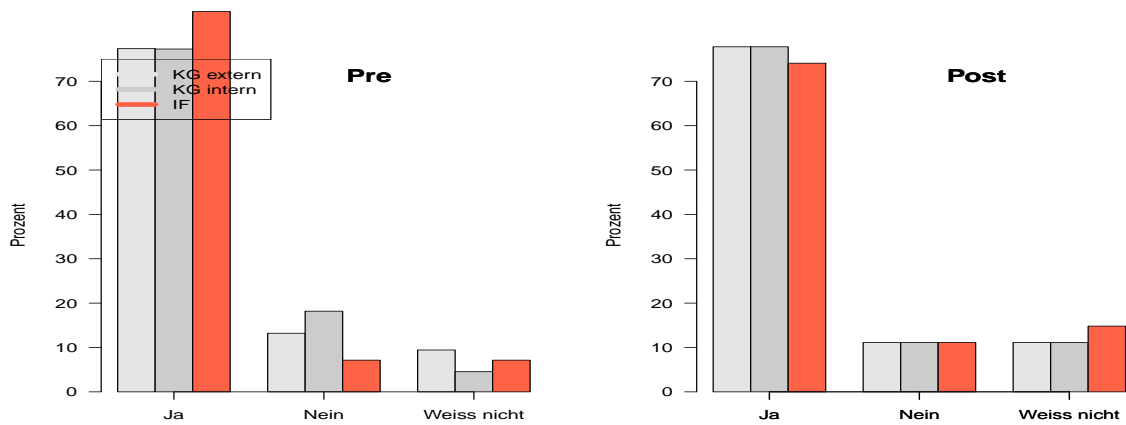


Abbildung 8.21: Ich weiss, was *Netzwerke* sind.

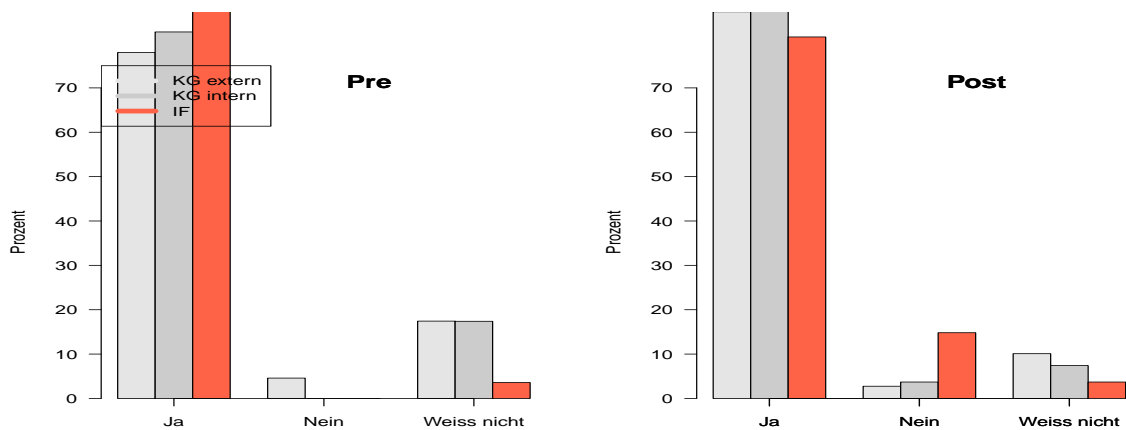


Abbildung 8.22: Das *Internet* ist eine Zusammenfassung von vielen Netzwerken.

Trotzdem sind einige der widersprechenden Meinung, das das Internet nur aus einem einzelnes Netzwerk besteht. Die Schüler der Informatikgruppe verneinen diese Frage in dem Umfang, wie sie die vorhergehende Frage bejaht haben (Abbildungen [8.22] und [8.23]).

### 8.7.4 Betriebssysteme

In den Bildungsstandards [Arb08, S.17] werden Inhalte aus dem Themengebiet *Betriebssysteme* unter dem Inhaltsbereich *Informatiksysteme* aufgeführt. Beispielhaft wird dort verlangt: Schülerinnen und Schüler „*unterscheiden Betriebssystem und Anwendersoftware*“ und „*verwenden Dateien und verwalten sie in Verzeichnissen*“.

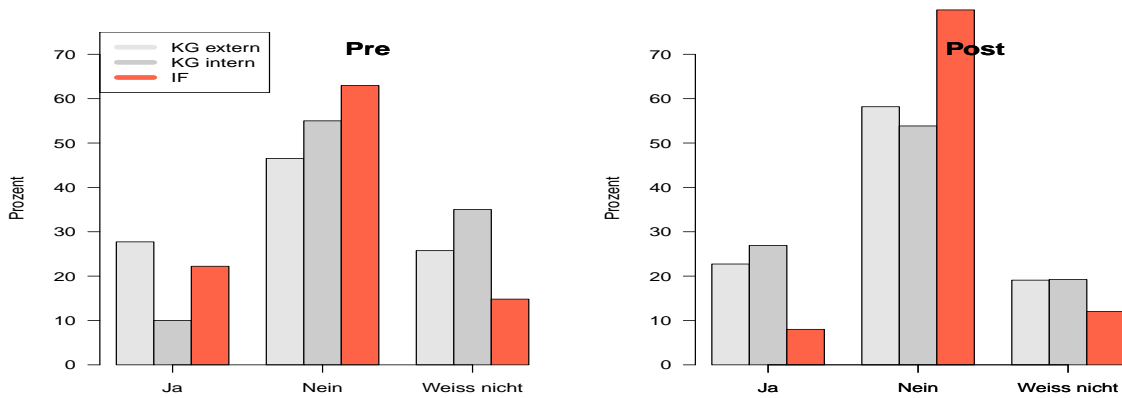


Abbildung 8.23: Das Internet ist ein einzelnes Netzwerk.

## Softwaresysteme

Die grosse Mehrheit aller Schüler, erst recht der Schüler in der Informatikgruppe, gibt an, zu wissen, was ein Betriebssystem ist, aber den konträren Begriff der *Anwendersoftware* kennen nur ca. 50%. Es ist bei allen Gruppen zum Ende des Schuljahres ein Zuwachs zu erkennen, die Unterschiede zwischen den Gruppen bleiben aber bestehen (Abbildungen [8.24] und [8.25]).

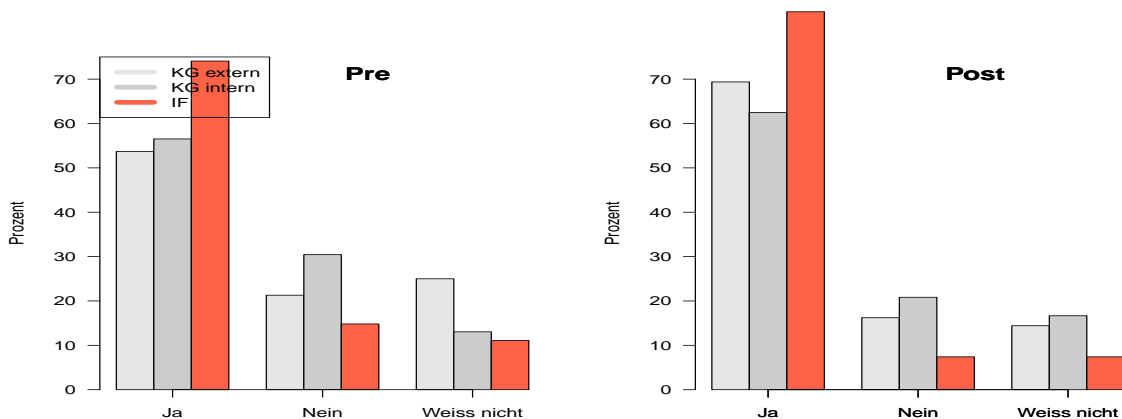


Abbildung 8.24: Ich weiss, was ein Betriebssystem ist.

Um festzustellen, ob die Fragen korrekt beantwortet wurden, wurde konkrete Beispiele angegeben. Es wurde nach den Betriebssystemen *Windows XP*, *Linux* und dem Anwendungsprogramm *Word* gefragt. Die Antworten in den Abbildungen [8.26], [8.27] und [8.28] machen deutlich, dass in allen Gruppen noch Bedarf besteht, diese Begriffe mit Inhalt zu füllen und voneinander abzugrenzen. Es wird deutlich, dass dies seitens der Lehrer ein bewusstes Handeln verlangt, da diese Begrifflichkeiten nicht nebenbei aus dem Unterricht heraus gebildet werden.

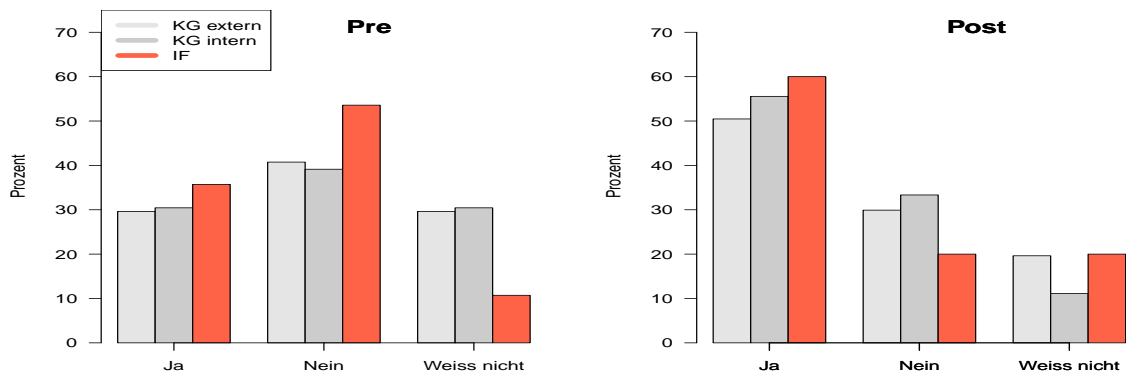


Abbildung 8.25: Ich weiss, was *Anwendersoftware* ist

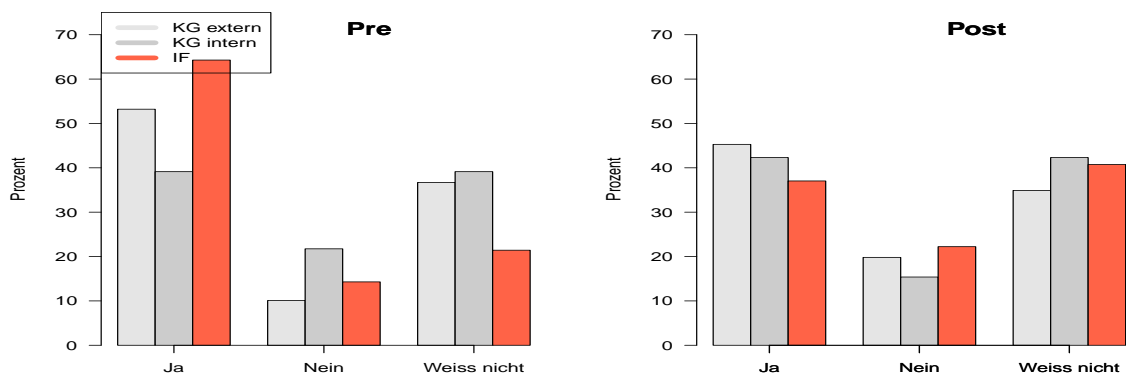


Abbildung 8.26: *Windows XP* ist eine *Anwendersoftware*.

Auf der einen Seite wurde nach zwei Betriebssystemen gefragt, aber auch die Interpretation einer Applikation als Betriebssystem ist nicht unüblich. Da eine Textverarbeitung für viele die Hauptanwendung an einem Computer darstellt, lag es nahe, dieser den Betriebssystemcharakter entsprechend Frage [8.28] zuzuordnen.

Bei fast allen Gruppen und Fragen schärft sich im Laufe dieses Jahres in dieser Frage das Bewusstsein. Sehr bedenklich erscheint aber, dass die Anwendungssoftware *Word* nach einem Jahr Unterricht in den externen Kontrollgruppen, die u.a. sehr viel mit dieser Textverarbeitung gearbeitet haben, von vielen Schülern in den Rang eines Betriebssystems erhoben wird. Hier scheint der Unterricht kontraproduktiv bezüglich der untersuchten Fragestellung zu sein. Das über die Hälfte der Schüler in den externen Kontrollgruppen mit dem Begriff *Linux* nichts anfangen kann, ist sicher dem Umstand zu verdanken, dass an diesen Schulen nur das Betriebssystem *Windows* verwendet wird, während an der Fritz-Steinhoff-Gesamtschule im pädagogischen Netz *Linux* verwendet wird, sodass hier die Schüler in der internen Kontrollgruppe wie auch in der Informatikgruppe durch die praktische Nutzung in der Schule einen Vorsprung

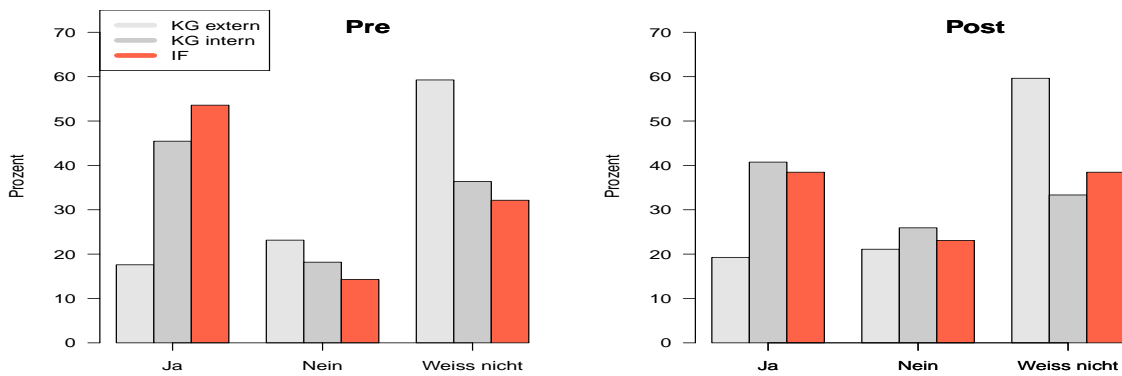


Abbildung 8.27: *LINUX* ist eine Anwendersoftware.

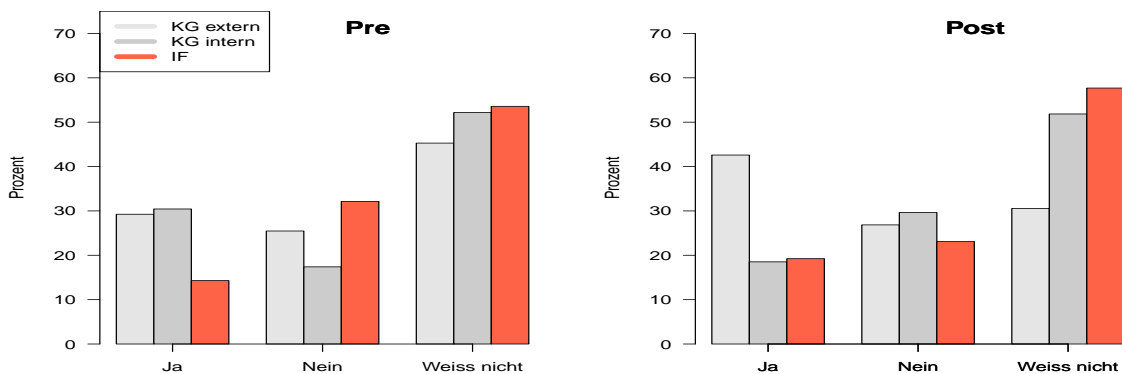


Abbildung 8.28: *WORD* ist ein Betriebssystem.

haben.

Dass die Schülerinnen und Schüler der Kontrollgruppe entsprechend Abbildung [8.29] trotzdem angeben, dass sie mehrere Betriebssysteme kennen würden, ist wohl eher dem Umstand zu verdanken, dass sie zwischen Betriebssystem und Anwendersoftware nicht sinnvoll unterscheiden können.

## Benutzeroberflächen

Alle heute benutzten Betriebssysteme werden zumeist über eine grafische Oberfläche, der sogenannten *GUI* bedient. Ergänzend dazu ist es jedoch gelegentlich nützlich, das System mit Hilfe der sogenannten Kommandozeile, der *Shell*, zu bedienen. Explizit werden diese Begriffe nicht eingeführt, jedoch nach unterrichtlichem Geschehen indirekt thematisiert. Die Schüler der Informatikgruppe kennen eine Kommandozeile aus der Arbeit mit `Tcl`, da sie die Eingaben direkt in die dort integrierte Shell eingeben. Die Schüler der Kontrollgruppen haben

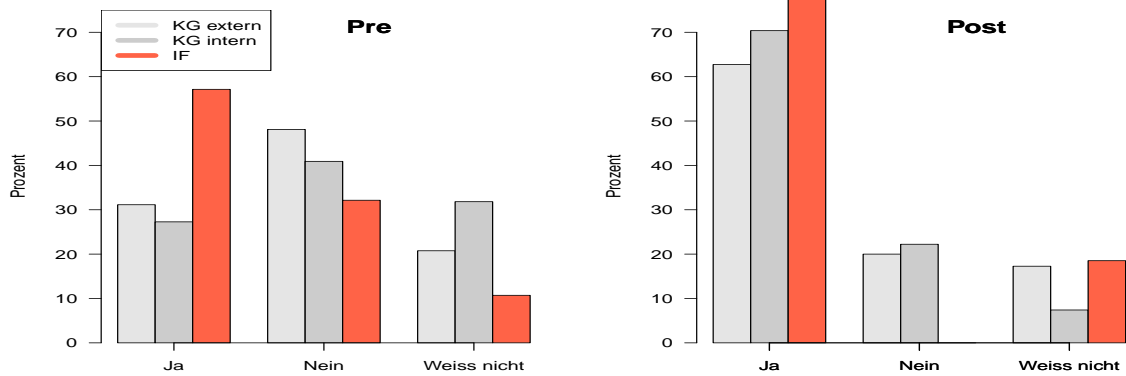


Abbildung 8.29: Ich kenne *mehrere* Betriebssysteme.

unterrichtlich noch nie mit einer Kommandozeile gearbeitet.

Die Bildungsstandards [Arb08, S.17] fordern für die Jahrgangsstufe 5 bis 7: „Schüler und Schülerinnen [...] arbeiten mit grafischen Benutzungsoberflächen“ und für die Jahrgangsstufen 8 bis 10 „... benutzen das Betriebssystem zweckgerichtet“. Diese zweckgerichtete Nutzung ist im Rahmen des Unterrichts mit Tcl/Tk für die Schüler der Informatikgruppe bereits früher zwingend gewesen.

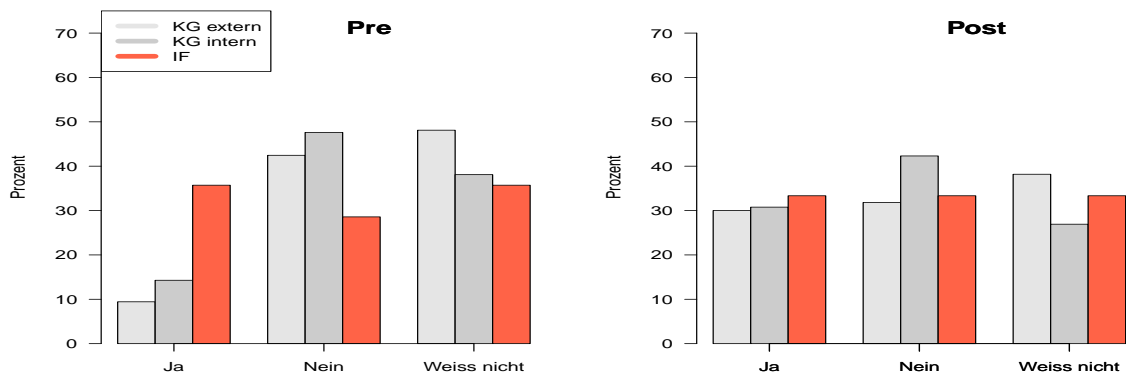


Abbildung 8.30: Ich weiss, was eine *Shell* ist.

Die Begriffe *Shell* und vor allem *GUI* sind entsprechend Abbildungen [8.30], [8.31] bisher den Schülern relativ unbekannt. Der Begriff *Kommandozeile* Abbildung [8.32] ist allein aus sprachlichen Gründen den Schülern offensichtlich näher. Die deutlich stärkere Bejahung der Frage nach der *Kommandozeile* seitens der Informatikgruppe dürfte darin begründet sein, dass die Schüler eine derartige Kommandozeile als Arbeitsumgebung und darüber als Begriff kennen, auch wenn die Eigenschaft als Kommandozeile noch nicht thematisiert wurde.



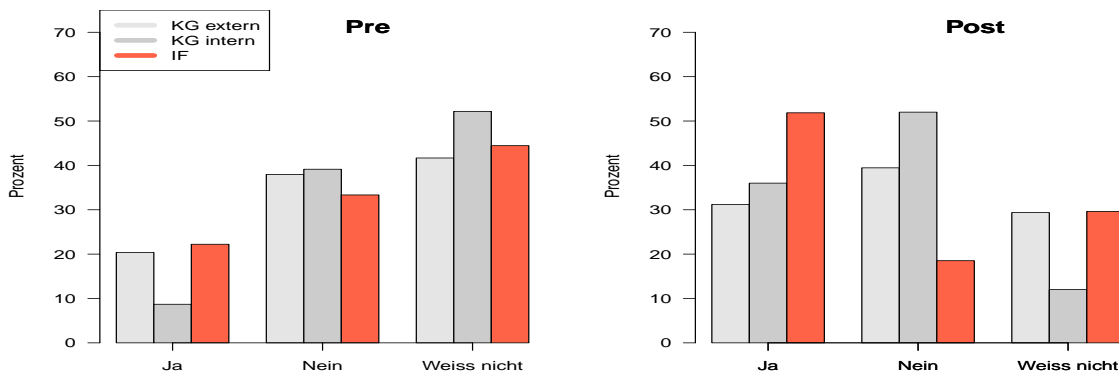


Abbildung 8.31: Ich weiss, was eine *Kommandozeile* ist.

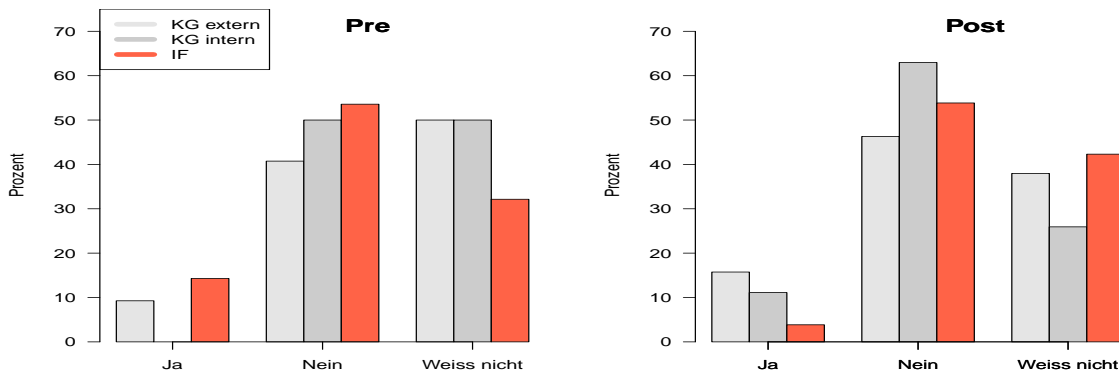


Abbildung 8.32: Ich weiss, was *GUI's* sind.

## Verzeichnisstrukturen

Die Arbeit an und mit dem Computer wird sicher nicht wesentlich davon beeinträchtigt, ob jemand die Begriffe *Datei*, *Verzeichnis*, *Ordner* und *Verzeichnisbaum* kennt und damit umgehen kann. Das ist bei *Verzeichnis* und *Verzeichnisbaum* nicht selbstverständlich. Sinnvolle Verzeichnisstrukturen sind bei der grossen Menge an Dateien, die viele Nutzer heute gestalten und nutzen, sinnvoll und hilfreich. Daher fordern die Bildungsstandards [Arb08, S.17] auch: „Schüler und Schülerinnen [...] verwenden Dateien und verwalten sie in Verzeichnissen“. An anderer Stelle fordern die Bildungsstandards: „Schülerinnen und Schüler [...] kennen die Navigations- und Änderungsmöglichkeiten für Verzeichnisbäume und deuten sie in Beispielen inhaltlich“ [Arb08, S.15].

Entsprechend haben die Schüler der Informatikgruppe schon recht früh die Verzeichnisstrukturen unter anderem im Schulnetz kennengelernt und thematisiert. Sie haben über die Kommandozeile der Sprachumgebung `Tcl` Verzeichnisbäume angelegt und genutzt. In den

Klassenarbeiten wurden diese Kenntnisse abgefragt. Auch in den externen Kontrollgruppen ist das Thema Verzeichnisse angesprochen worden.

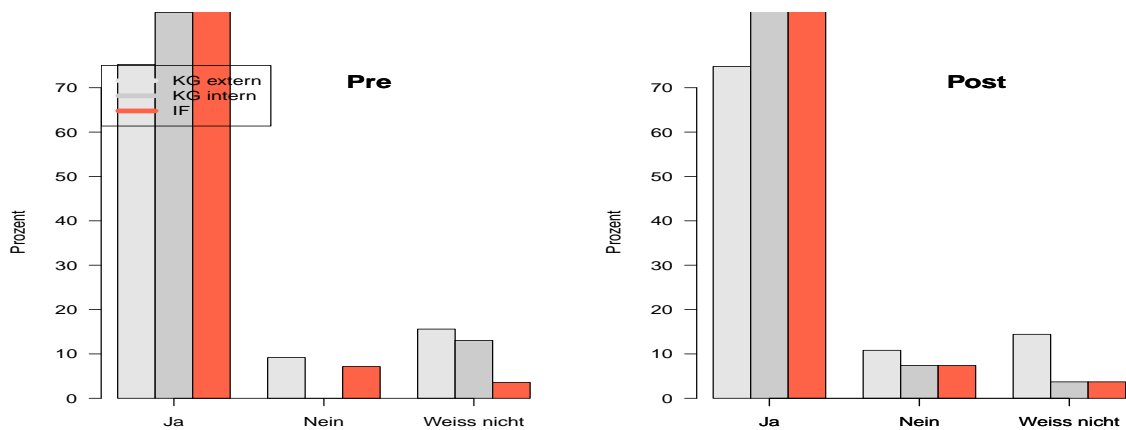


Abbildung 8.33: Ich weiss, was ein *Verzeichnis* bzw. ein *Ordner* ist.

Sehr deutlich ist in Abbildung [8.33] zu erkennen, dass schon zu Beginn des Unterrichtes im untersuchten Zeitraum der Begriff *Verzeichnis* und noch mehr *Ordner* aus der Alltagswelt in die Begriffswelt der Schüler integriert worden ist. Hierzu muss die Schule keinen zusätzlichen Beitrag leisten.

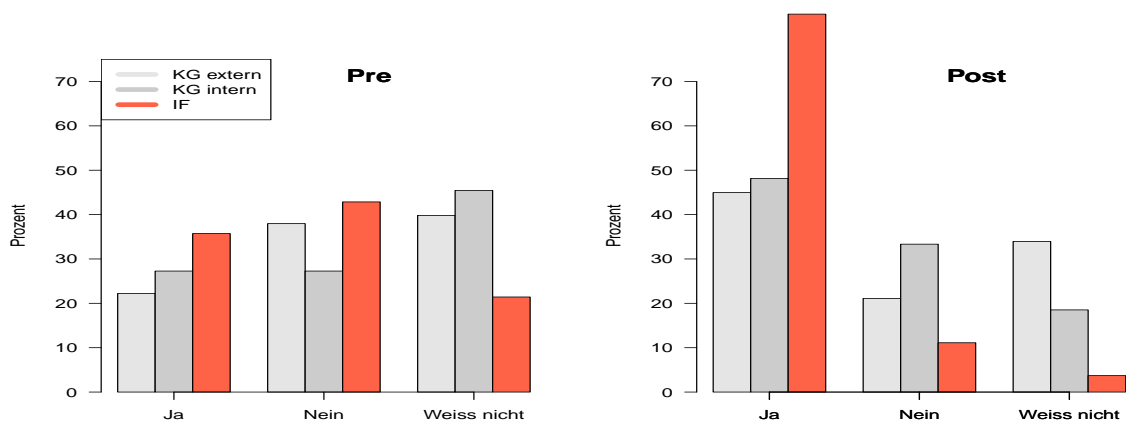


Abbildung 8.34: Ich weiss, was ein *Verzeichnisbaum* ist.

Über die flache Hierarchie eines oder mehrerer Verzeichnisse hinaus, ist nur relativ wenig Schülern eine Struktur bewusst. Dieses wird erst durch Unterricht möglich, wie sich an Abbildung [8.34] ablesen lässt. Ca. 80% der Schüler kennen einen Verzeichnisbaum, während in den Kontrollgruppen dies nur ca. 40% der Schüler von sich behaupten. Diese Verhältnisse bestätigten sich in den *Think-Aloud*-Interviews am Ende des Schuljahres in den verschiedenen Gruppen.

## Zusammenfassung

In diesem ersten Jahr sind Themen aus dem *Roten Faden Betriebssysteme* im geringem Umfang gelegentlich indirekt und nicht explizit angesprochen worden. Einzelne Aspekte wurden ausschliesslich in anderen Zusammenhängen eingebracht. Das kann das Erlernen von Fachausdrücken und Begriffen von den Schülern erschweren.

In der Informatikgruppe wurde jedoch der Umstand ‚ausgenutzt‘, dass aufgrund von regelmässigem Raumwechsel eine sinnvolle Verzeichnisstruktur für die bearbeiteten Dateien sehr schnell notwendig wurde. Verzeichnisse und Verzeichnisstrukturen wurden an dieser Stelle umfassend diskutiert und auch zeichnerisch dargestellt.

Aus den Fragen in der Untersuchung wird deutlich, dass sich viele der in diesem Umfeld auftretenden Begriffe und Inhalte nicht von selbst und auch nicht automatisch korrekt bei den Schüler verankern. Dabei ist die Beschäftigung mit dem Computer für ein Kind und einen Jugendlichen heute eine alltägliche Angelegenheit. Aber schon die Grundbegriffe *Betriebssystem* und *Anwendersoftware* können nicht sicher voneinander getrennt werden. Damit ist es dem Jugendlichen aber verwehrt, eine Diskussion um Computer und seine Komponenten richtig und umfassend zu verstehen und dann auch mündig beispielsweise bei einem Kauf entscheiden zu können. Zweifelsohne ist in der Zukunft noch zu überlegen, auf welche Art und Weise ein systematisches Wissen aufgebaut werden soll oder muss oder ob diese Begrifflichkeiten sich im Laufe der Jahre quasi von selbst korrekt angelegt werden.

Bei den Fragen zu *Verzeichnissen* und *Verzeichnisstrukturen* wird deutlich, welcher Einfluss Unterricht hat. Der eine Begriff *Verzeichnis* wird aufgrund der ‚Alltagsnutzung‘ seitens der Schüler schon in der frühen Kindheit erlernt, die systematische und strukturelle Erweiterung auf *Verzeichnisbäume* ergibt sich damit aber nicht von selbst. Da junge Menschen im Laufe ihres Lebens auf vielfältigste Art mit Dateien und deren Speicherung konfrontiert werden, ist hier sicher Wert auf ein sinnvolles Vorgehen zu legen.

## 8.8 Auswertung des Fragebogens zu Aussagen über die Informatik

Im zweiten Teil der Umfrage wurde nach Einschätzungen der Schülerinnen und Schüler zu Aussagen über das Fach Informatik und zur eigenen Kompetenz gefragt. Solche Fragen lassen sich nicht eindeutig mit einem *Ja/Nein-Schema* beantworten. Daher wurde für diese Fragen eine *Likert-Skala* [May08, S.87ff] gewählt, in der die Schüler ihre Antworten von ‚*stimmt gar nicht*‘ bis ‚*stimmt völlig*‘ in einer 5-stufigen Skala ein gruppieren konnten. Da auch in diesem Teil des Fragebogens einige Fragen von den Schülern aufgrund ihres bisherigen Wissensstandes noch nicht beantwortet werden konnten, wurde zusätzlich ‚*weiss nicht*‘ als Antwort-Möglichkeit angeboten.

Ein Teil der Fragen befasste sich mit Vorstellungen über das, was im Fach Informatik wichtig ist oder nicht. Dabei wurden Items untersucht, die in der Lebenswirklichkeit der Schüler wie beispielsweise *Webseiten* bekannt sind als auch Fragestellungen, die eher aus der Innensicht eines Informatikers wie beispielsweise die *Rolle von Programmiersprachen* stammen.

## Bedeutung in der Informatik

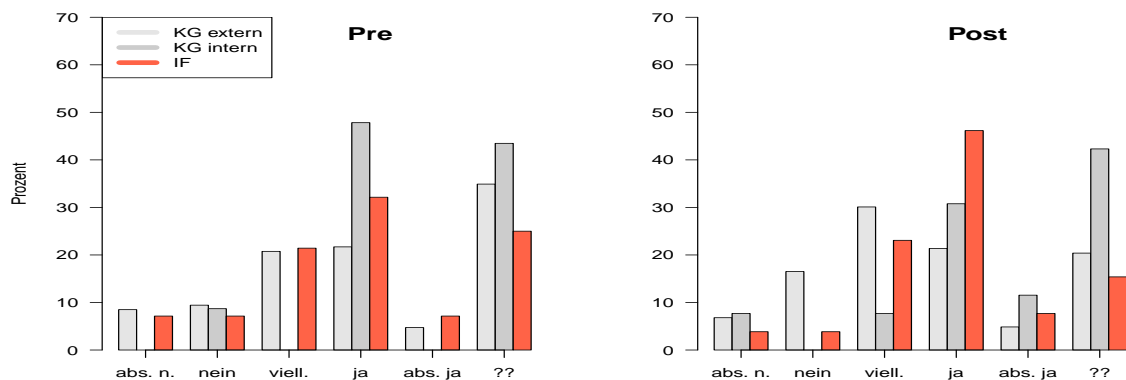


Abbildung 8.35: In der Informatik ist Webseiten-Gestaltung sehr wichtig.

Jeder Schüler und jede Schülerin kennt aus der eigenen Anschauung und Nutzung Webseiten. Abbildung [8.35] zeigt, dass die Schüler mit ca. 30% meinen, dass Webseiten eine Aufgabe der Informatik sind, aber genauso viele Schüler sind sich zumindest sehr unschlüssig. Die Position verändert sich auch nicht zum Ende des Jahres bis auf die Position der internen Kontrollgruppe, die am Ende des Jahres der Meinung der anderen Gruppe angeglichen ist. Alle anderen Gruppen haben am Anfang des Jahres wie am Ende des Jahres eine weitgehend unveränderte Position. Da die Auszeichnungssprache *HTML* in diesem Schuljahr nur punktuell am Rande angesprochen wurde, ist dieses Ergebnis nicht weiter überraschend.

## Programmierung

Im Unterricht in Informatik ist das Programmieren als Tätigkeit mit diesem Begriff noch nicht eingeführt worden. Kleine Sequenzen an Anweisungen in *Tcl/Tk* sind aber bereits erstellt und genutzt worden. Die Kontrollgruppen haben dagegen nichts Entsprechendes geleistet.

Zu diesem Thema wurden verschiedenste Fragen gestellt:

### Programmieren ist wichtig

Schon zu Beginn des Schuljahres haben die Informatikschüler ein deutlicheres Bewusstsein für das Programmieren als eine Tätigkeit von Informatikern. Dieses ist am Ende des Jahres noch ausgeprägter. Auch die Schüler der externen Kontrollgruppen erfahren unbewusst(?), dass Programmieren in der Informatik nicht unbedeutend ist. Nur die Schüler der internen Kontrollgruppe ändern ihre Meinung nicht. Offensichtlich ist die intensivere Beschäftigung mit dem Computer bzw. deren Erzeugnissen hier entscheidender als das genaue Verständnis davon, was eigentlich ein Programm ist.

Die entsprechende Kontrollfrage wird von den Schülern entsprechend adäquat beantwortet.

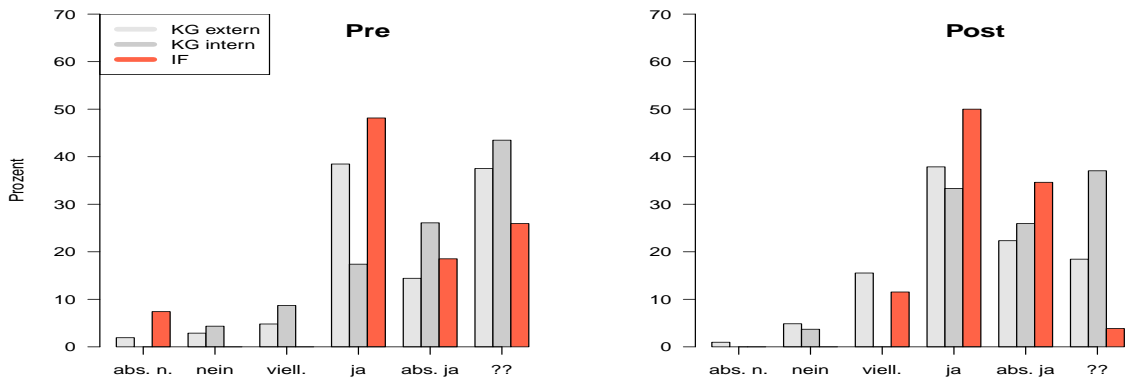


Abbildung 8.36: Programmierung ist in Informatik sehr wichtig.

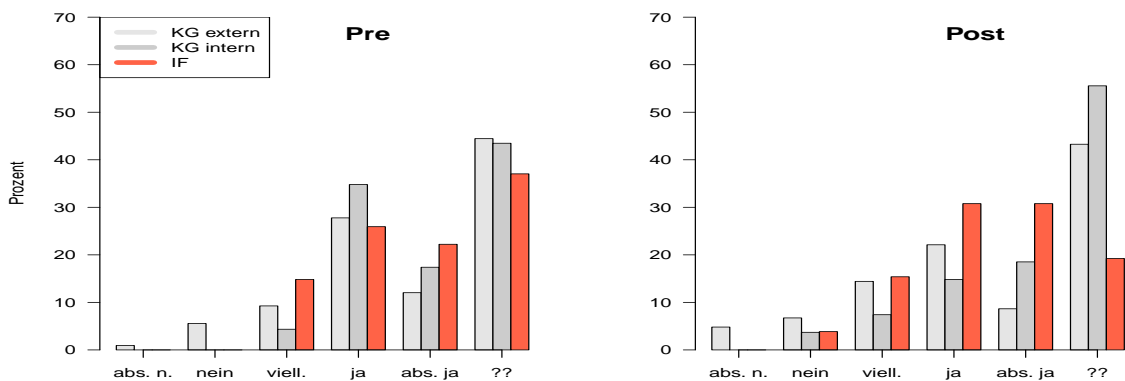


Abbildung 8.37: In der Informatik sind Programmiersprachen sehr wichtig.

### Eigenschaften von Programmen

Erstaunlicherweise meinen die internen Kontrollschüler deutlich mehr als Schüler aus den anderen Gruppen, dass es für eine Aufgabe mehr als ein Programm geben kann. Hier sind die Schüler aus der Informatik-Gruppe und den Informatik-nahen Gruppen vorsichtiger. Dabei haben sie teilweise doch schon mehrere Programme für ein und denselben Zweck kennengelernt wie beispielsweise mehrere Editoren, Webbrowser etc..

Die Kontrollfrage, bei der viele Schüler erwartungsgemäss mit *Weiss nicht* antworten, zeigt, dass sie diese Frage tatsächlich noch nicht beantworten können.

Ähnlich verhält es sich mit der Frage zur Lesbarkeit von Programmtexten. In der vorsichtigeren Aussage der Informatik-Schüler und der externen Kontrollgruppe wird eher deutlich, dass sie sich bei dieser Frage etwas unwohl fühlen, weil sie eigentlich über das Aussehen von Programmen noch gar nichts wissen. Die internen Kontrollschüler antworten wohl aus dem Bewusstsein über Texte jeglicher Art, für die eine solche Aussage über die Lesbarkeit sicher

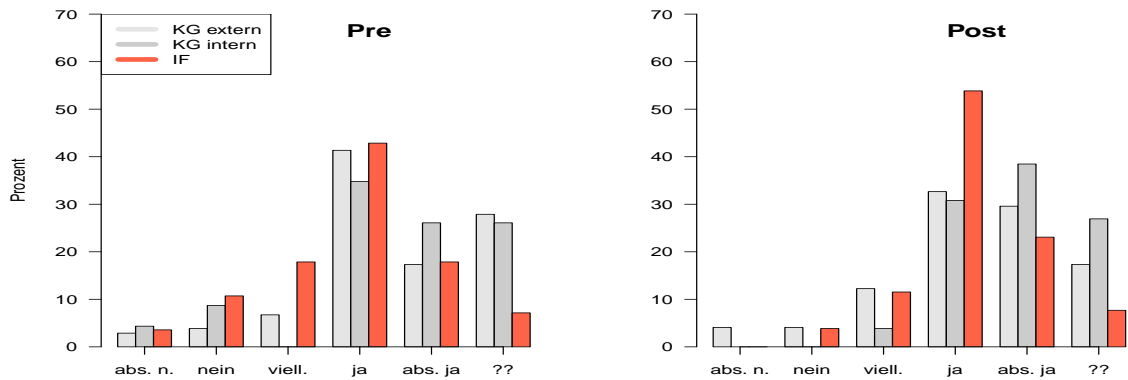


Abbildung 8.38: Für viele Aufgaben gibt es mehr als ein Programm.

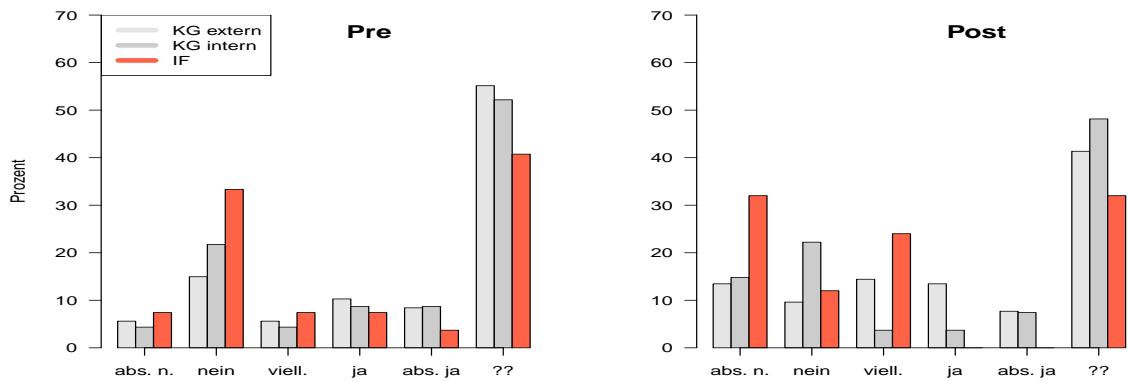


Abbildung 8.39: Für ein Problem gibt es nur ein mögliches Programm.

ebenfalls zu Recht besteht.

### Schwierigkeiten beim Programmieren

Trotzdem sind viele Schüler aller Gruppen im Pre- und Posttest der Meinung, dass sie gut programmieren können. Es halten sich Ja- und Nein-Stimmen in etwa die Waage, aber es gibt nur relative wenige Stimmen mit *Weiss nicht*. Was die Schüler hier unter Programmieren verstehen, bleibt unklar.

Zwar antworten die Schülerinnen und Schüler die Frage nach der Leichtigkeit des Programmierens mit etwas mehr Vorsicht, aber auch hier hätte man deutlich mehr eine Aussage des Nichtwissens erwartet.

Andererseits beantworten sie die Frage nach der Schwierigkeit der Erstellung von Software kritischer. Die Antworten zu diesem Fragenkomplex zeigen, dass die Schüler zwar manchmal glauben, sie könnten schon etwas zu diesem Thema sagen, die von ihnen gemachten Aussagen

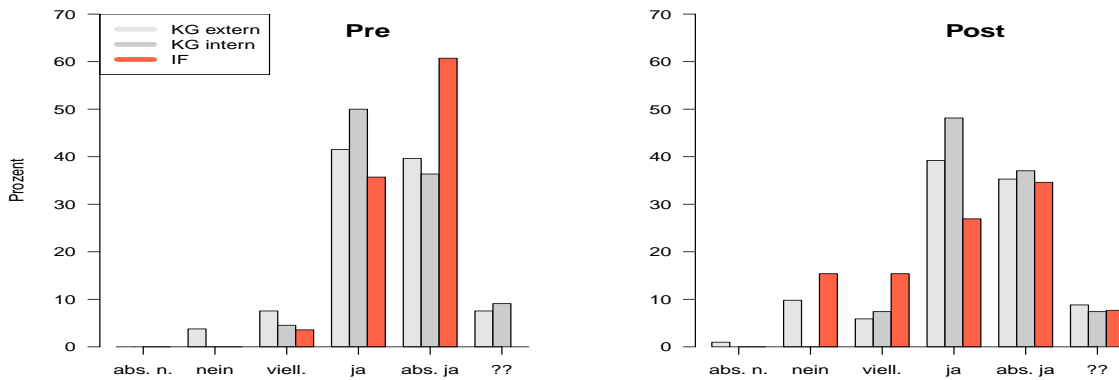


Abbildung 8.40: Ein Programmtext sollte sehr gut lesbar sein.

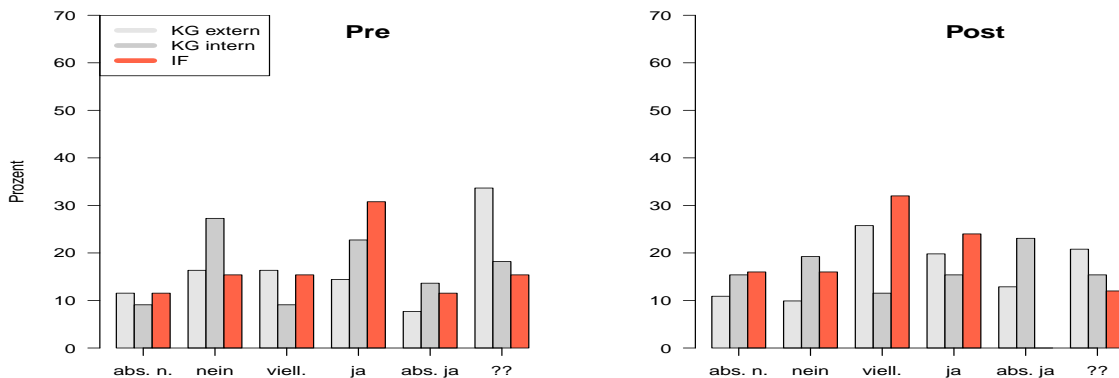


Abbildung 8.41: Ich kann sehr gut programmieren.

sind aber widersprüchlich und noch wenig durchdacht.

Auch die Frage nach Funktionalität und Art der Konstruktion eines Programmes lässt sich erst beantworten, wenn man Programme als solche kennengelernt hat und erfahren hat, dass verschiedene Programme eventuell verschieden gut ein Problem lösen. Entsprechend antworten ca. 40% aller Schüler, dass sie diese Frage nicht beantworten können. Die Informatikschüler beantworten im Post-Text zu 30% diese Frage mit *vielleicht* und nur der Rest macht eine deutlicheren Aussage. Auch hier zeigt sich offensichtlich, dass bereits ein in den Anfängen aufgebautes Fachverständnis zu einem erhöhten Bewusstsein über das eigentliche noch fehlende Wissen führt.

## Der Algorithmus-Begriff

Der Fachausdruck *Algorithmus* ist den Schülern in diesem Jahrgang noch nicht geläufig. Daher verwundert es nicht, dass die entsprechende Frage von der grössten Gruppe entsprechend

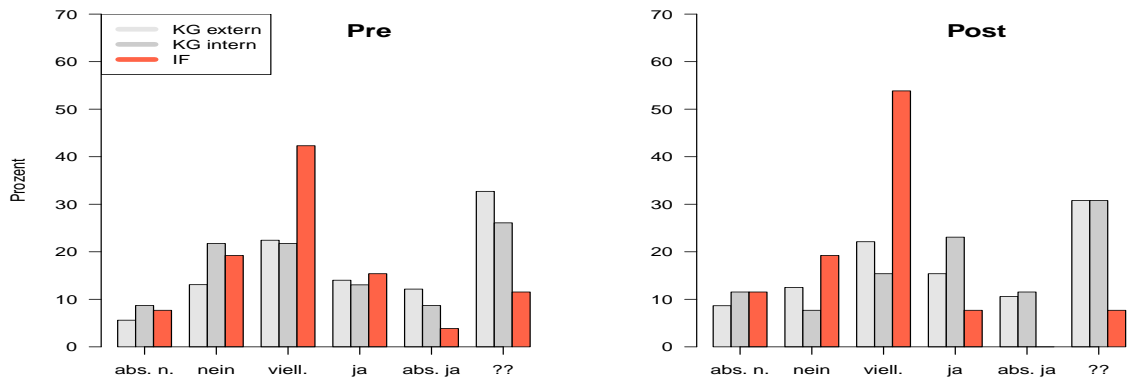


Abbildung 8.42: Programmieren ist sehr leicht.

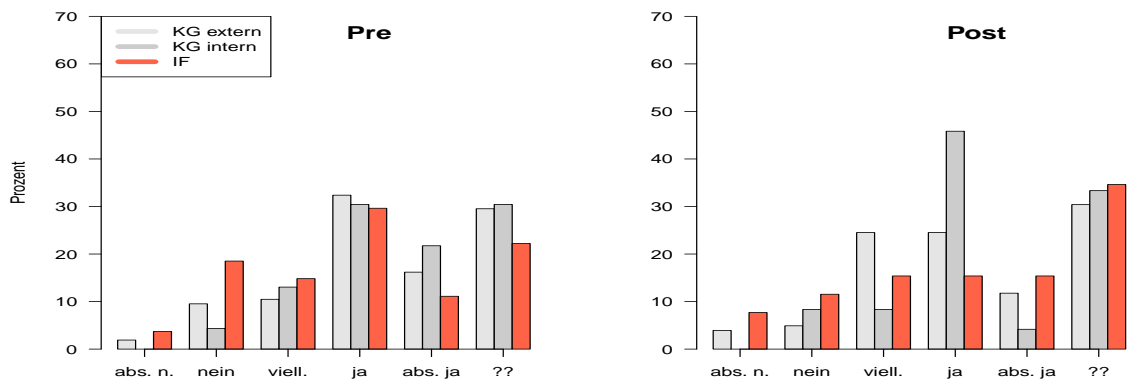


Abbildung 8.43: Ein Computerprogramm ist oft nicht einfach zu erstellen.

beantwortet wird. Es ist in Abbildung [8.45] erkennbar, dass die Schüler der Informatikgruppe nach einem Jahr wesentlich unschlüssiger antworten als am Anfang des Jahres. Ihnen ist offensichtlich bewusst, dass sie diese Frage eigentlich nicht beantworten können. Schüler der anderen Gruppen neigen dazu eher, die Beantwortung mit einer klaren Aussage zu versehen.

Ähnlich skeptisch beantworten die Schüler die Frage nach dem eigenen Handeln beim Programmieren. Da sie selber bis auf das Erstellen von kleinen Skripten noch nicht programmieren, verwundert dies nicht. Nach einem Jahr ist allerdings wieder ein verschärftes Bewusstsein entsprechend Abbildung [8.46] vor allem bei den Teilnehmern der Informatikgruppe erkennbar. Ihre zu Beginn des Jahres stärker vorhandene Ausprägung zu klaren Aussagen weicht einer sehr deutlichen vorsichtigen Haltung, indem sie vermehrt *Weiss nicht* oder *Vielleicht* angeben, während die Schüler der anderen Gruppen in etwa ihre Position vom Anfang des Jahres beibehalten.



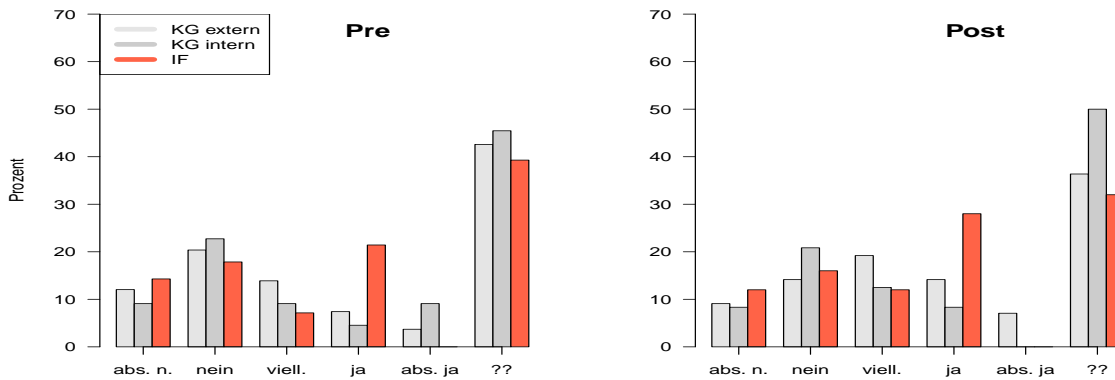


Abbildung 8.44: Wenn ein Programm das tut, was es soll, ist es egal, wie es konstruiert wurde.

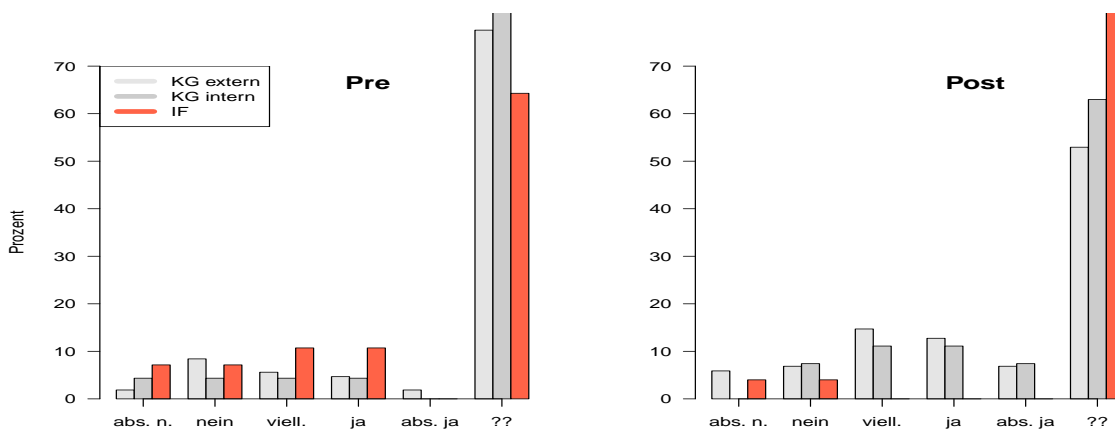


Abbildung 8.45: Es geht in der Informatik sehr oft um Algorithmen.

## Berufswelt und Informatik

Interessant ist die Beantwortung der Frage nach der Berufswelt der Informatiker in Abbildung [8.47]. Ein Einblick in diese Berufswelt ist für Schüler in diesem Alter kaum möglich. Also enthalten sich sehr viele Schüler bei der Frage nach der Tätigkeit eines Programmierers. Erstaunlich ist, dass bei der Informatik-Gruppe die Meinung nach der Tätigkeit eines Informatikers kippt. Während zu Beginn des Jahres die Schülerinnen und Schüler der Informatikgruppe der Aussage mehr zustimmten als ablehnten, lehnten am Ende des Jahres deutlich mehr Schüler diese Aussage ab. Ebenso stieg die Zahl derer an, die sich nicht sicher waren und mit *vielleicht* geantwortet haben. Bei den beiden Kontrollgruppen veränderten sich die Meinungen wenig. Die Mehrheit der Schüler in den Kontrollgruppen antworteten mit *Weiss nicht*.

Die Schülerinnen und Schüler der Informatikgruppe identifizieren den Informatik-Unterricht mit der ihnen nicht bekannten Berufswelt. Offensichtlich hat dieses eine Jahr Unterricht zu ei-

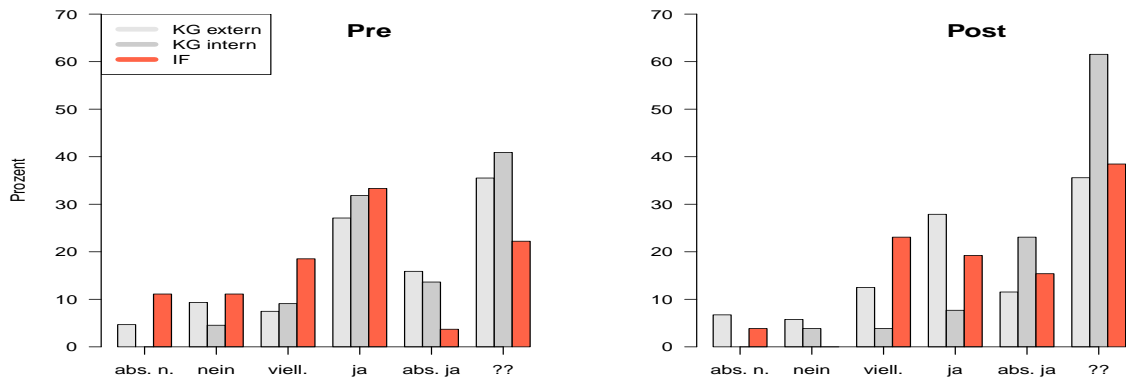


Abbildung 8.46: Bevor ich an den Computer zum Programmieren gehe, durchdenke ich das Problem erst vollständig.

ner Verschiebung der Wahrnehmung geführt. Dabei ist natürlich zu berücksichtigen, dass die Vorstellungen vom *Programmieren* noch sehr diffus sind.

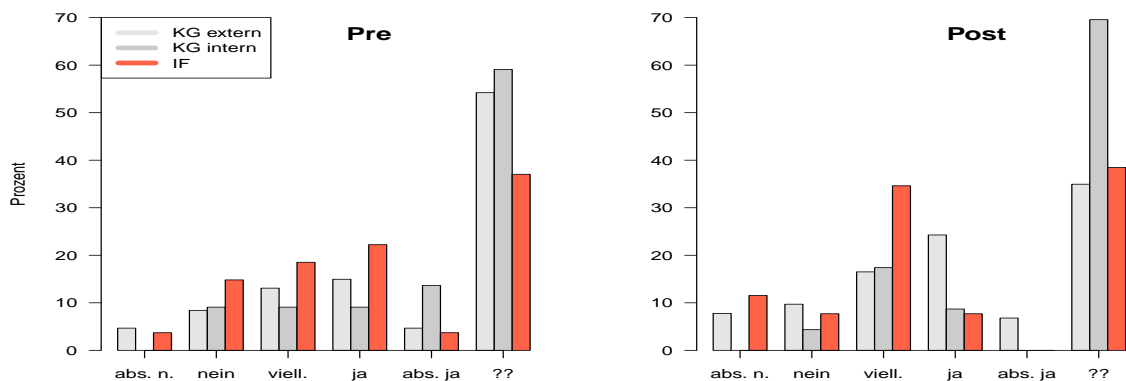


Abbildung 8.47: Informatiker muessen die meiste Zeit programmieren.

## Programmversionen

In zwei Fragen wurden die Schüler befragt, wie wichtig für sie die Installation der jeweils aktuellen Version eines konkreten Programmes ist. Wir wollten wissen, ob die Installation eines Programmes aus ihrer Sicht überhaupt ein Teil von Informatik ist. Etwa 50% der Schülerinnen und Schüler aller Gruppen sehen darin eine Tätigkeit der Informatiker. Am Ende des Jahres gibt es dabei wenig Differenzen zwischen den Gruppen. Zu Beginn des Jahres war die Gruppen noch etwas unterschiedlicher Meinung.

Ergänzend wurden die Schüler befragt, inwieweit sie selber für sich die jeweils aktuelle

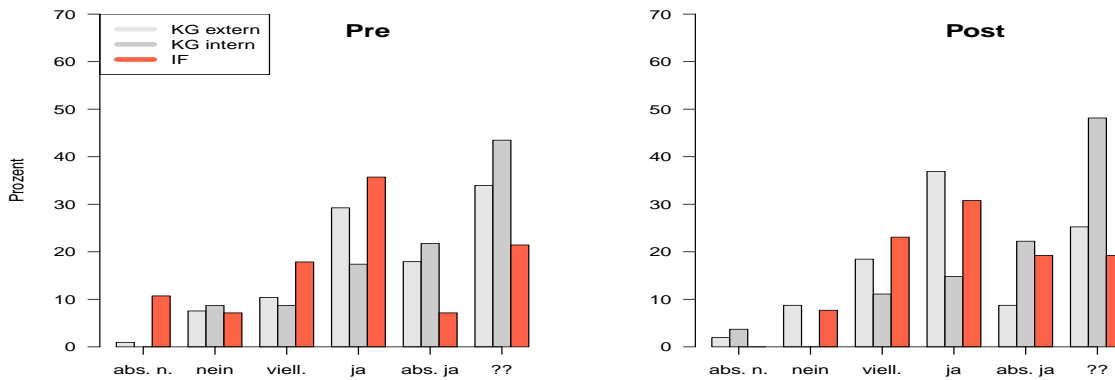


Abbildung 8.48: Installieren von Software ist ein sehr wichtiger Teil von Informatik.

Softwareversion als notwendig erachten. Auch hier liegen die Meinungen der unterschiedlichen Gruppen nicht entscheidend auseinander. Über die Hälfte der Schülerinnen und Schüler antworten entweder mit *Weiss nicht* oder *Vielleicht*. Sie haben offensichtlich in der Mehrheit noch keine ausgeprägte Meinung zu diesen Fragestellungen.

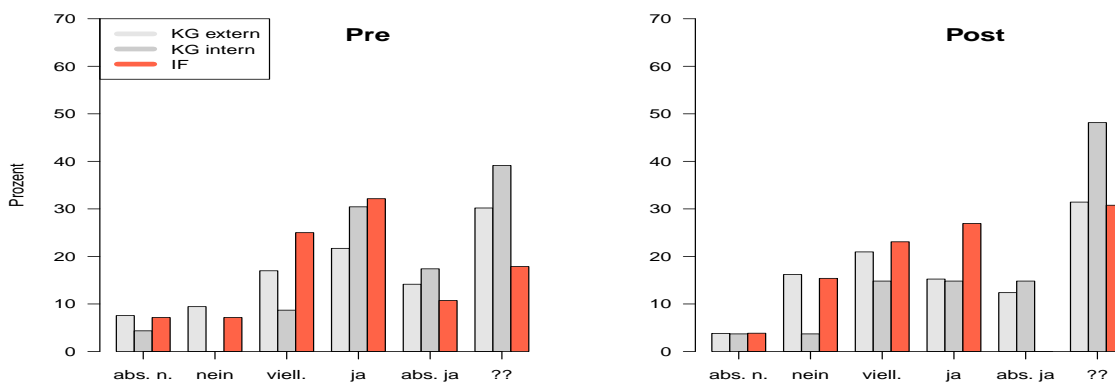


Abbildung 8.49: Es ist bei jedem Programm sehr wichtig, die aktuellste Softwareversion zu haben.

## Anwendungssoftware

In einem weiteren Fragenkomplex wurde nach den klassischen Anwendungen im privaten und beruflichen Umfeld gefragt. Dabei ging es um die sogenannten Officeprogramme wie Textverarbeitung und Tabellenkalkulation, aber auch altersgemäss um Spielprogramme. Auch nach der Bedeutung von Handbüchern wurde gefragt.

Schülerinnen und Schüler verbringen heutzutage einen nicht unerheblichen Teil ihrer Frei-

zeit mit dem Computer [Med10],[Med09] und Computerspielen<sup>11</sup>. Daher ist die Ergebnis in der Abbildung [8.50] auf die Frage, ob er bzw. sie sich mit Computerspielen auskennt schon beinahe wie ein ‚Eulen nach Athen bringen.‘. Natürlich kennen sich über 80% der Schüler mit Computerspielen aus. Interessant ist, dass hier die interne Kontrollgruppe, die unterrichtlich von allen Gruppen gar nicht mit Computern arbeitet, die grösste Kenntnis von Computerspielen angibt.

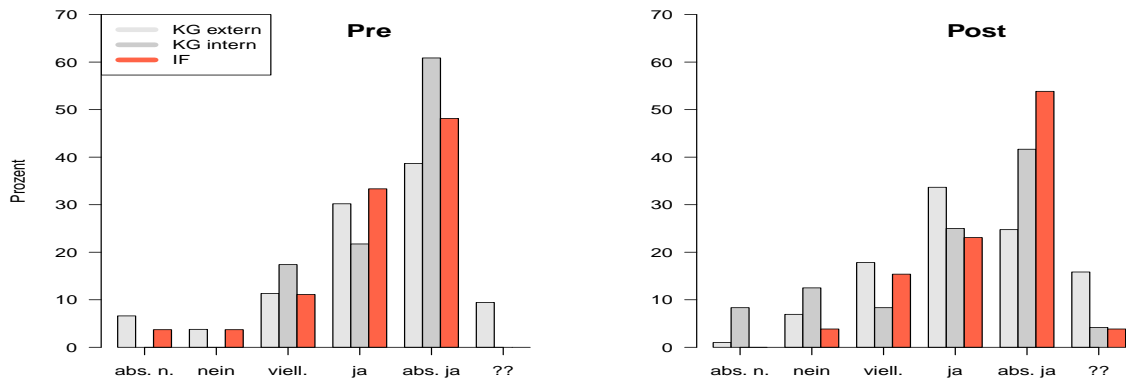


Abbildung 8.50: Ich kenne mich sehr gut mit Computer-Spielen aus.

Ein intensiver Umgang mit Textverarbeitung und vor allem Tabellenkalkulation seitens der Schülerinnen und Schüler ist nicht unbedingt als selbstverständlich anzunehmen. Nicht jeder Schüler hat Spass am Erstellen von Texten und nicht jede Schule (Grund- und auch weiterführende Schule) führt intensiv in dieses Metier ein. Alle befragten Schülerinnen und Schüler sind zumindest im früheren oder aktuellen Unterricht mit Textverarbeitung in Berührung gekommen. An der Hauptschule Vorhalle und der Realschule Halden ist die Beschäftigung mit Office-Software ein wesentlicher Bestandteil des Informatik-Unterrichtes. Die Schülerinnen und Schüler an der Fritz-Steinhoff-Gesamtschule haben dies in einem kleinen Kurs à 6 Wochen geübt. Dementsprechend ist das Ergebnis nach Abbildung [8.51] auch nicht weiter überraschend. Überraschend ist allerdings die starke Zunahme des Selbstvertrauens in die eigene Meinung der internen Kontrollgruppe, die gerade im letzten Jahr schulisch überhaupt nicht mit der Textverarbeitung gearbeitet hat. Die Informatik-Gruppe ist auch hier nach einem Jahr etwas vorsichtiger in der eigenen Bewertung.

Das Internet ist für die heutigen Kinder und Jugendlichen eine Institution, mit der sie aufgewachsen sind. Im Netz bewegen sie sich völlig souverän, sei es schulisch oder privat. Auch hier ergibt sich ein ähnliches Bild entsprechend Abbildung [8.52]. Auf der einen Seite sind es über 70% der Schülerinnen und Schüler, die sich nach eigenen Angaben *gut* bzw. *sehr gut* im Internet auskennen. Es gibt jedoch Ausnahmen, die sich *wenig* oder *sehr wenig* im Internet zu Hause fühlen. Sie gehören fast alle der Informatik-Gruppe an. Da sie allerdings im Unterricht

<sup>11</sup>Laut Zeit-Online verbringen 15-jährige Jugendliche entsprechend einer Studie des ‚Kriminologischen Forschungsinstitut Niedersachsen (KFN)‘ ca. 6 bis 7 Stunden täglich am Computer und davon ca. 2,5 Stunden mit Computerspielen(<http://www.zeit.de/online/2009/12/computerspielsucht-jugend-2>, Zugriff: 5.12.2011).

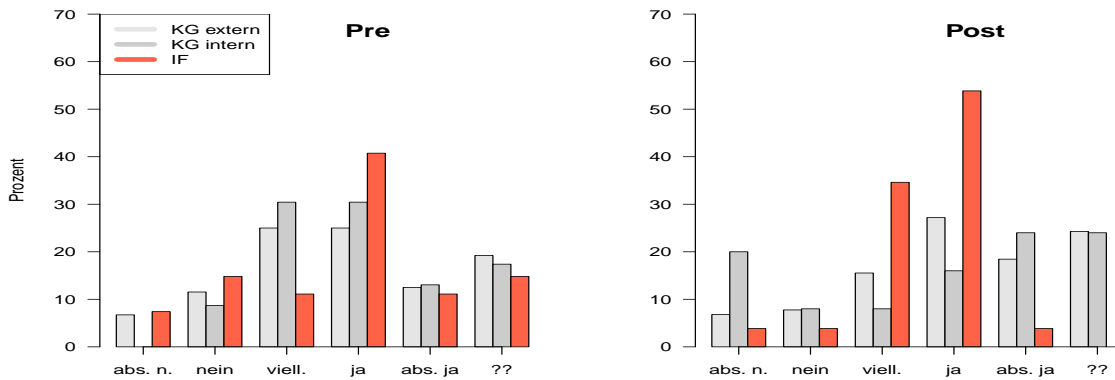


Abbildung 8.51: Ich kenne mich sehr gut mit Textverarbeitung aus.

schon mit und im Internet gearbeitet haben, haben sie offensichtlich hohe Erwartungen und Anforderungen an ihr eigenes *Know-How*.

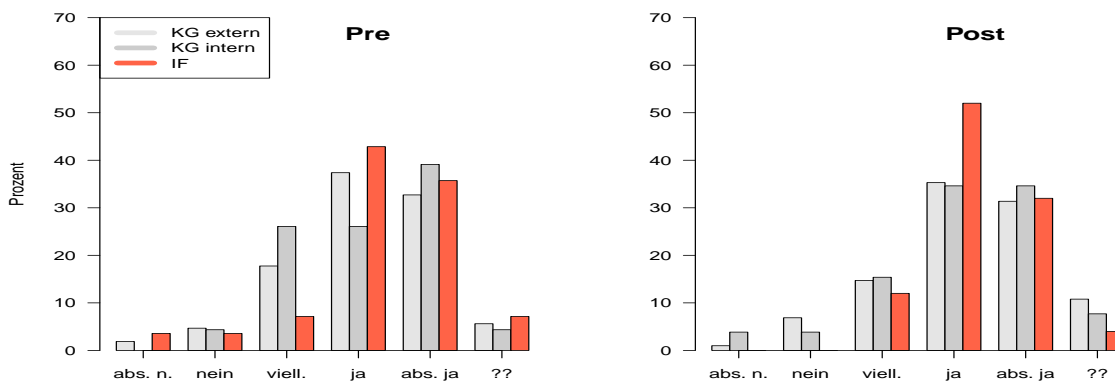


Abbildung 8.52: Ich kenne mich sehr gut mit dem Internet aus.

Wie anzunehmen, ist die Kenntnis der Tabellenkalkulation nicht ganz so hoch wie die zur Textverarbeitung, aber noch ca. 60% bis 70% stimmen zu, dass sie sich damit gut oder sehr gut auskennen. Auch hier ist die interne Kontrollgruppe ohne Unterricht mit diesen Medien die Gruppe mit der grössten Zustimmung. Beide anderen Gruppen zeigen erhebliche Veränderungen im Verlaufe des Jahres. Nicht unwesentlich viele Schüler waren sich zu Beginn des Jahres unschlüssig über die Tabellenkalkulation, während am Ende des Jahres eine Zustimmung zu Kenntnissen über diese Softwareklasse angegeben wird. Das gilt in erster Linie für die externe Kontrollgruppe, die im Unterricht konkret mit dieser Software gearbeitet hat, aber auch für die Informatikgruppe, die unterrichtlich noch nicht mit einer Tabellenkalkulation gearbeitet hat.

Viele Benutzer kommen ohne Computer-Handbücher aus. Doch manche Features lassen sich ohne Informationen aus Handbüchern oft nicht finden und beschreiben. Immerhin sind

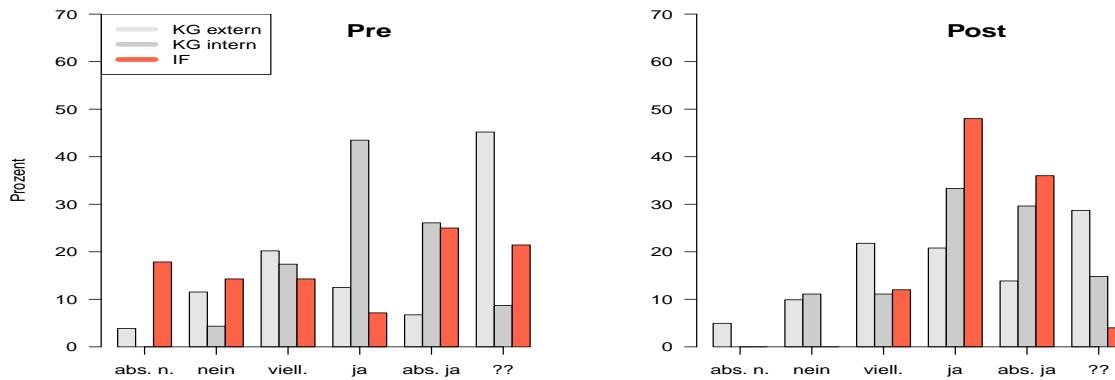


Abbildung 8.53: Ich kenne mich sehr gut mit Tabellenkalkulation aus.

etwa 50% der Schüler derselben Meinung und weitere ca. 30% sind sich nicht sicher. Ob die Schülerinnen und Schüler die Handbücher tatsächlich nutzen, kann daraus nicht gefolgert werden.

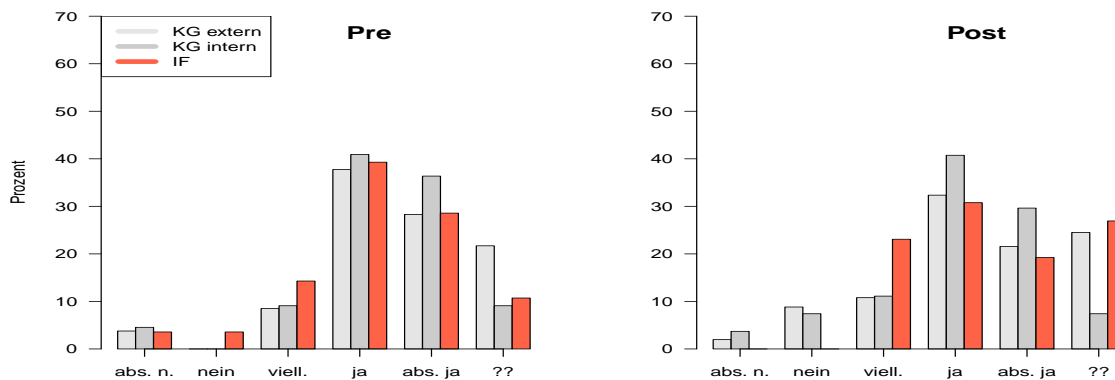


Abbildung 8.54: Computer-Handbücher sind gelegentlich sehr nützlich.

## 8.8.1 Zusammenfassung

In diesem Fragenkomplex wurden die Schülerinnen und Schüler nach ihrer Einschätzung und Vorstellung zu Informatik sowie zur Bedeutung der Informatik, zur Programmierung und Software gefragt. Dabei interessiert natürlich, in wieweit der Informatikunterricht Veränderungen in der Wahrnehmung des Faches Informatik unterstützt. Auch wenn nach einem Jahr Anfangsunterricht noch nicht grosse Veränderungen erwartet werden können, lassen sich Tendenzen erkennen. Die Schülerinnen und Schüler der Informatikgruppe sind sehr häufig deutlich vorsichtiger in der Beantwortung der Fragen. Gerade die interne Kontrollgruppe, die ausser den

Einführungskursen von ca. 6 Wochen in der Schule keinen Computer verwendet, wagen deutlich häufiger eine Aussage. Die Reflektionsfähigkeit der Schüler und Schülerinnen in der Informatikgruppe entwickelt sich sichtbar besser.

Ein Bewusstseinswandel über die Berufswelt der Informatiker hat schon eingesetzt, wie die Beantwortung zur Frage ‚*Informatiker müssen die meiste Zeit programmieren*‘ in Abbildung [8.47] zeigt. Dieser Bewusstseinswandel geht in die richtige Richtung und kann bei dieser Gruppe nur durch den Unterricht erfolgt sein, da die beiden anderen Kontrollgruppen sich völlig anders verhalten.

Viele Fragen, die den bisherigen Unterricht noch nicht tangieren, beantworten die Schüler zumeist mit *Weiss nicht*. Das deutet darauf hin, dass die beantworteten Fragen auch tatsächlich stimmig beantwortet worden sind. So geben ca. 80% der Schüler zu Recht an, dass sie den Begriff *Algorithmus* nicht kennen.

## **8.9 Auswertung des Fragebogens zu Einstellungen, Meinungen und Verhalten**

In einem dritten Abschnitt wurden die Schülerinnen und Schüler zu ihren Einstellungen und Meinungen gefragt. Es sollte festgestellt werden, ob sich diese schon nach kurzer Zeit ändern, wenn die Schüler einen Informatik-Unterricht erhalten. Natürlich darf auch hier nicht erwartet werden, dass ein Schulunterricht in diesem für die Schüler noch neuem Gebiet nach bereits einem Jahr erhebliche Meinungsänderungen produziert.

### **8.9.1 Beschäftigung mit dem Computer**

Zunächst wollten wir wissen, wie oft sich die Schülerinnen und Schüler mit dem Computer beschäftigen. Dabei spielte es keine Rolle, ob dies im Rahmen von Freizeitbeschäftigungen erfolgt oder ob ein schulischer Zusammenhang besteht. Wir fragten nach der täglichen [8.56] und wöchentlichen [8.55] Beschäftigungszeit mit dem Computer.

Aus den Zahlen ist ablesbar, dass mehr als 50% der Schülerinnen und Schüler täglich mit dem Computer arbeiten oder spielen. Nur sehr wenige verneinen dies. In allen Gruppen nimmt die Beschäftigung im Laufe des Schuljahres zu, wenn auch bei den Schülern der externen Gruppen deutlich geringer. Die Schüler der Informatik-Gruppe beschäftigen sich zu Beginn der Jahres am meisten mit dem Computer, werden aber zum Ende des Schuljahres von den Schülern der internen Kontrollgruppe überholt. Der Computer ist ein täglicher Begleiter in der kindlichen und jugendlichen Welt geworden. Diese nicht gerade neue Kenntnis wird noch eindeutiger, wenn man bedenkt, dass zu diesen Zeiten auch noch die Zeiten hinzugezählt werden könnten, in denen sich die Kinder und Jugendlichen mit dem Smartphone oder ähnlichen Geräten beschäftigen.

Naheliegender ist in diesem Zusammenhang das Interesse und die Beschäftigung mit Computerspielen.

Abbildung [8.57] zeigt das Interesse für Computerspiele. Die sehr ähnlichen Zahlen für das Computerspielinteresse und der täglichen Beschäftigung [8.56] mit dem Computer lässt den

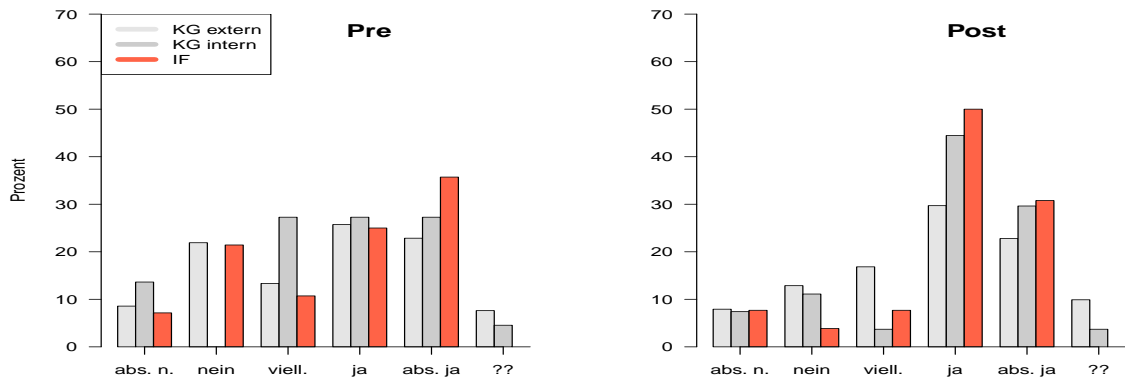


Abbildung 8.55: Ich beschäftige mich mehrmals wöchentlich mit Computern

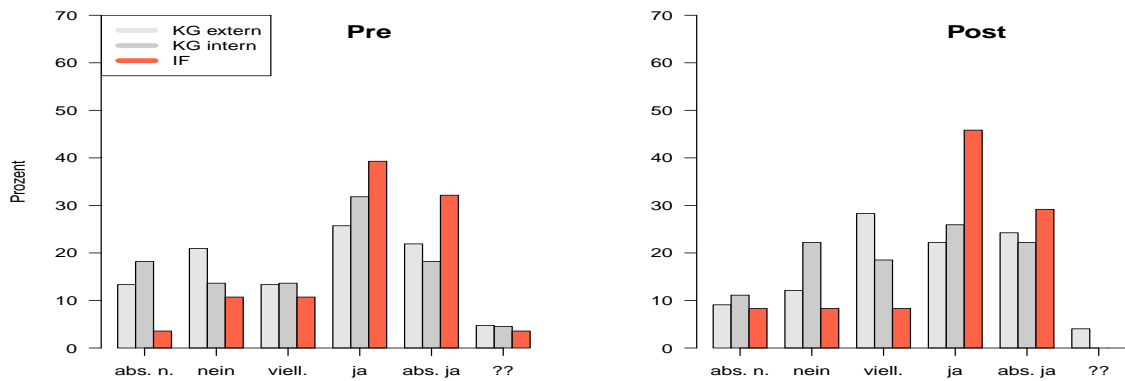


Abbildung 8.56: Ich beschäftige mich täglich mit Computern

Schluss zu, dass am Computer weitgehend gespielt wird.

Es wäre aber voreilig zu schliessen, die Schülerinnen und Schüler wollten nur am Computer spielen. Bei allen Gruppen ist feststellbar, dass die meisten den Computer nicht nur als Spielmaschine sehen. Sie wollen mehr können als nur spielen. Dies zeigt Abbildung [8.58] eindeutig. Dementsprechend hat auch das Bildungssystem entsprechende Angebote zu schaffen, denn eine sinnvolle Beschäftigung mit dem Computer ist nicht von allen Schülerinnen und Schüler ohne Hilfe zu erwarten.

## Programmieren

Dies wird bei der Beantwortung zur Frage nach den *Basiskenntnissen im Programmieren* deutlich. Auch wenn die Informatikgruppe erst rudimentäre Ansätze des Programmieren erlernt hat, werden von den Schülern deutlich höhere Ansprüche gestellt als in den anderen Gruppen [8.59]. Während in den Kontrollgruppen im Pretest fast 70% und im Posttest fast 50% der



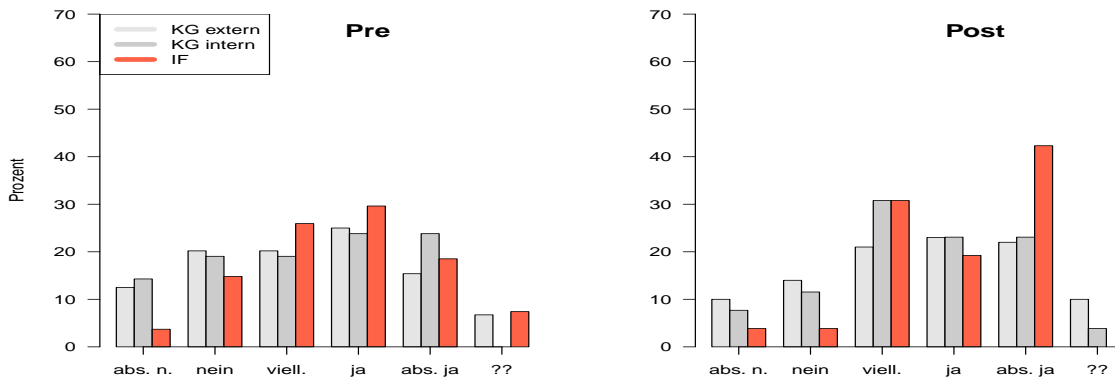


Abbildung 8.57: Ich interessiere mich für Computer-Spiele.

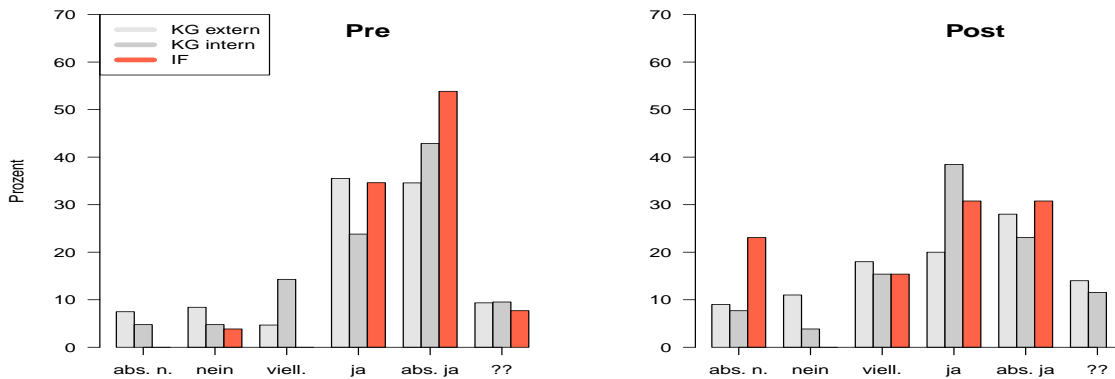


Abbildung 8.58: Ich möchte mehr können als nur am Computer zu spielen.

Schülerinnen und Schüler sich der Meinung enthalten, steigt die Zustimmung zur Notwendigkeit von Basiskenntnissen vom Programmieren in der Informatikgruppe von unter 40% auf über 60%. Bei den Kontrollgruppen im Pretest beträgt die Zustimmung unter 20%, steigt dann jedoch auf ähnlich hohe Werte an. Allerdings sind in der Informatikgruppe 30% der Schüler der Meinung, dass die Programmierung vielleicht wichtig sein könnte, während sich die Schüler in den Kontrollgruppen im Zweifel mit über 50% (interne Kontrollgruppe) und über 30% (externe Kontrollgruppe) völlig enthalten. Es ist zu erwarten, dass die vielen *vielleicht*-Stimmen im Laufe der Jahre, in denen an den verschiedensten Stellen in Kontexten das Programmieren eine wichtige Hilfsfunktion einnehmen wird, eine deutlichere Position beziehen werden.

Dies bedeutet nicht, dass alle Schülerinnen und Schüler bei dieser Tätigkeit Spass haben. Die Zustimmung fällt tendenziell vorsichtiger aus, auch wenn die Mehrheit positive Antworten gibt. Es fällt auf, dass auch die Schülerinnen und Schüler der Kontrollgruppe, in denen nicht die Grundlagen für das Programmieren gelegt worden sind, teilweise positive Antworten

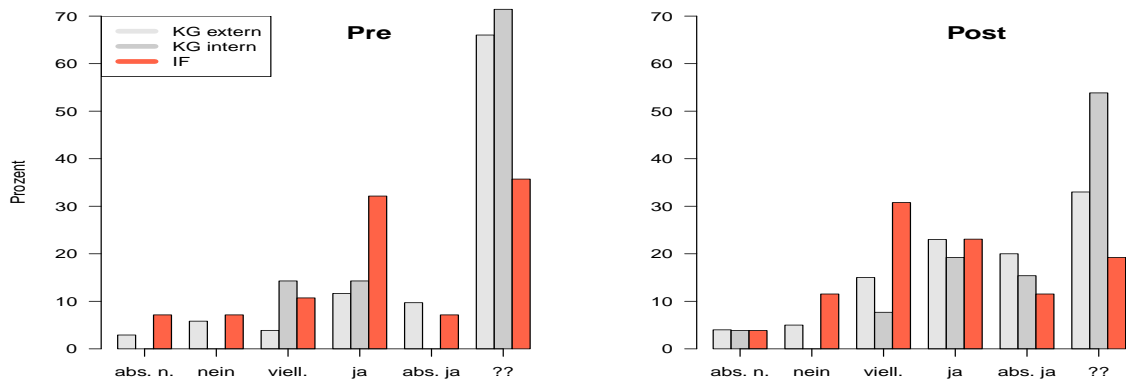


Abbildung 8.59: Es ist wichtig, Basiskenntnisse in Programmieren zu haben.

ten geben. Die positiven Antworten entsprechen in der Grössenordnung denen aus der Frage nach den Basiskenntnissen im Programmieren. Offensichtlich wird das Programmieren hier mit Arbeit mit dem Computer gleichgesetzt.

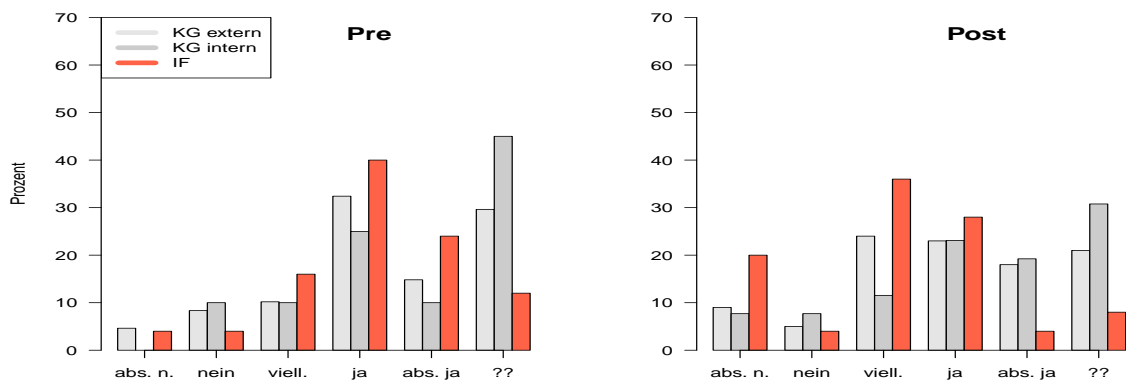


Abbildung 8.60: Die Arbeit am Computer beim Programmieren macht Spass.

## Lernen am und mit dem Computer

Unabhängig von den konkreten Unterrichtsinhalten stellt sich die Frage, ob das Lernen am und mit dem Computer als schwer und anstrengend empfunden wird und wie gelernt wird. Es ist naheliegend, dass der Computer als Werkzeug im Lernprozess in Informatikunterricht eine nicht unerhebliche Rolle spielt. Daher ist es im Fach Informatik leicht, die Schülerinnen und Schüler zu einer Eigentätigkeit an zuleiten. Ob das aus Sicht der Schüler auch sinnvoll ist, wurde mit der nächsten Frage [8.61] erfragt.

Sehr auffällig ist, dass diese Aussage von über 60% der Schüler der Informatikgruppe unterstützt wird und nur von unter 20% abgelehnt wird. Bei allen Gruppen wird schon beim

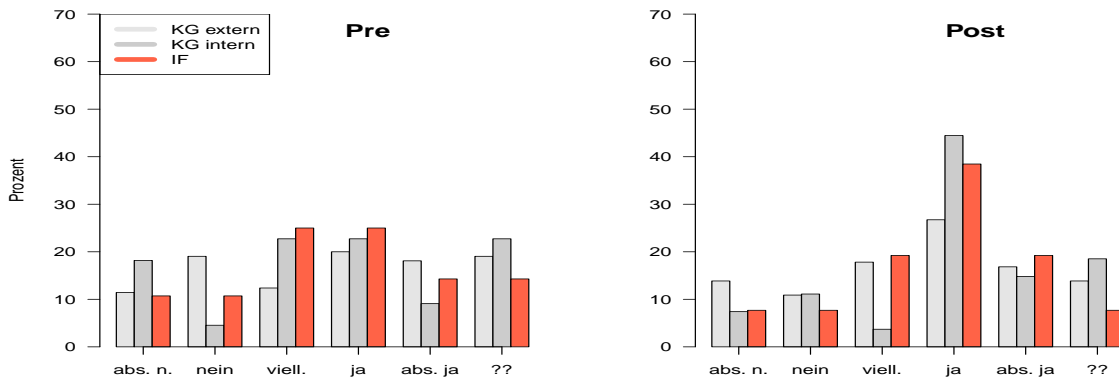


Abbildung 8.61: Ich lerne die meisten Sachen erst, wenn ich sie am Computer ausprobiert habe.

Pretest diese Aussage mit ca. 30% bejaht. Bei der internen Gruppe gibt es auch einen Zuwachs von ca. 10%. Da die Schüler gelegentlich auch im Fachunterricht in einen der Computerräume gehen, könnte dieser Zuwachs mit den dort gemachten Erfahrungen erklärt werden. Auffällig ist zudem, dass es bei den externen Kontrollgruppen mit einem GRIN-Curriculum praktisch keine Veränderung in der Zustimmung zu der Aussage zum handelnden Lernen gibt.

Informatik hat in der Öffentlichkeit einen Ruf, ein Fach mit hohen Anforderungen zu sein. Entsprechend Abbildung [8.62] erwarten 80% der Schülerinnen und Schüler am Anfang des Jahres, dass sie jede Stunde aufmerksam sein müssen. Auch die Schüler in den Kontrollgruppen sehen das ähnlich. Am Ende des Jahres ist zwar immer noch eine knappe Mehrheit dieser Meinung, aber ca. 15% meinen, nicht alle Stunden aufmerksam sein zu müssen. Zu Beginn des Jahres war kein(!) Schüler der Informatikgruppe der Meinung, in Unterrichtsstunden in Informatik nicht aufmerksam sein zu dürfen. Diese *Normalisierung* der erwarteten Ansprüche ist sicher positiv zu werten.

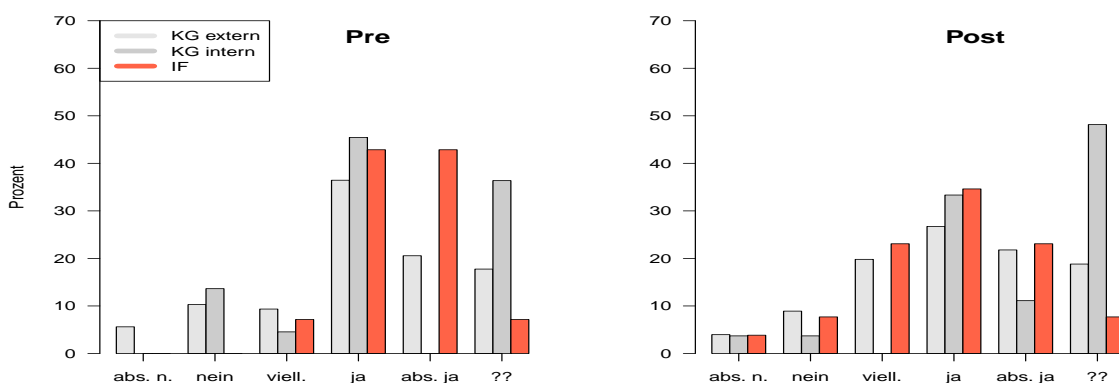


Abbildung 8.62: Informatik kann man nur lernen, wenn man jede Stunde aufmerksam ist.

Eine etwas differenzierte Position wird bei der Frage nach der *Schwierigkeit beim Programmieren* entsprechend Abbildung [8.63] deutlich. Zu Beginn des Jahres enthält sich ca. die Hälfte aller Schüler und Schülerinnen in allen Gruppen der Meinung. Am Ende des Jahres gilt das weiterhin für die Mitglieder der Kontrollgruppen. In der Informatikgruppe hat sich das Bild aber total gewandelt. Nur noch eine kleine Minderheit möchte sich zu dieser Frage nicht äussern. 20% der Schüler finden Programmieren nicht anstrengend, 30% der Schüler dagegen sehr wohl und 40% antworten mit *Vielleicht*. Diese Antworten basieren auf den Tätigkeiten im ersten Jahr in der *Tcl/Tk*-Umgebung. Die Antworten machen deutlich, dass die grosse Mehrheit der Schülerinnen und Schüler sich nicht überfordert fühlt. Auch die Schüler, die diese Aussage nach der *Schwierigkeit* bejahen, empfinden das zwangsläufig nicht unbedingt als negativ.

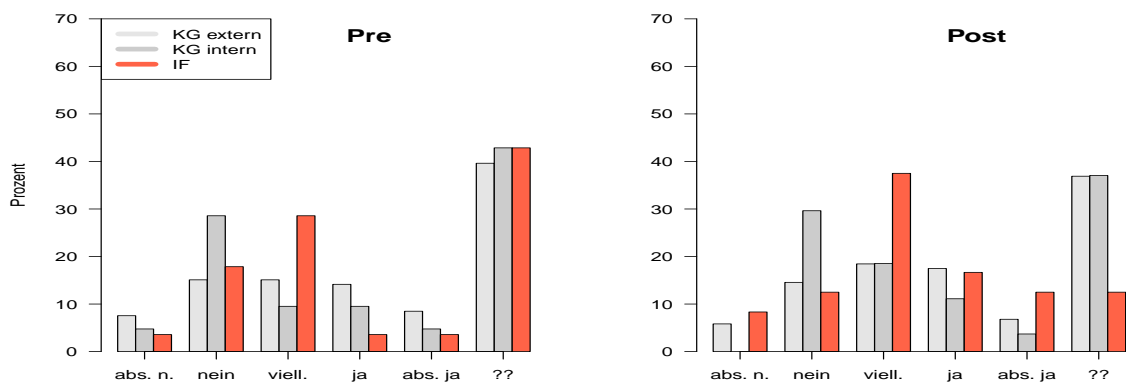


Abbildung 8.63: Die Arbeit am Computer beim Programmieren ist schwierig und anstrengend.

## Berufswelt

Für Schülerinnen und Schüler im 6. Jahrgang ist die Zeit nach der Schule ‚noch praktisch unendlich weit‘ weg. Trotzdem machen sich manche schon Gedanken um ihre berufliche Zukunft. Zumindest die Eltern wissen, dass diese Frage relativ schnell auf die Kinder und Jugendlichen zukommt. Dies hat natürlich Auswirkungen auf die Gespräche innerhalb der Familien. Es ist verständlich, dass viele Schüler noch keine klaren Vorstellungen von ihrer Berufswahl haben. Falls schon Vorstellungen vorhanden sind, wechseln diese auch häufig. Es überrascht daher nicht, dass die Antworten im Pre- und im Posttest entsprechend der Abbildung [8.64] differieren. Auffällig ist allerdings, dass die Schüler der Informatikgruppe eine extrem starke Veränderung zur Unsicherheit über ihre eigene Berufswahl haben. 55,5% gegenüber 32% sagen, dass ihre bisherigen Berufsvorstellungen, sofern bereits vorhanden, überdacht werden. Bei der internen Kontrollgruppe steigt dagegen diese Zahl nur von 31% auf 37%. Bei der externen Gruppe sinkt sie dagegen von 49,5% auf 44,5%. Die Schülerinnen und Schüler in der FSG denken offensichtlich über ihre möglichen zukünftigen Berufe anders nach als die Schüler in den anderen Schulen.

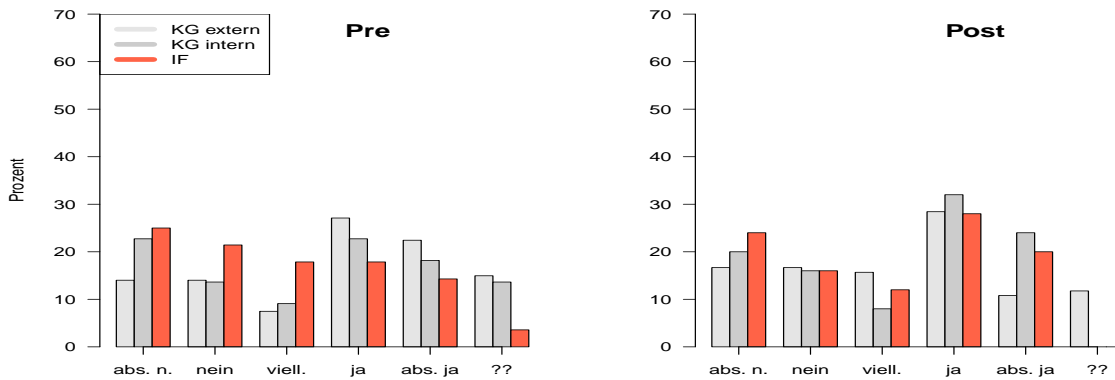


Abbildung 8.64: Ich weiss noch nicht, was ich später machen will.

In einer weiteren Frage wurde dann konkret nach möglichen Berufswünschen im Berufsfeld der Informationstechnologie nachgefragt. Informatik ist ein Berufsfeld mit Zukunft. Eine Motivation für das Fach Informatik seitens der Schüler und vor allem auch der Eltern kann daher sein, in der Schule schon für eine mögliche Berufstätigkeit einen Grundstein zu legen. Dies ist zweifelsohne eine Aufgabe der Schule, wie es beispielsweise Heymann [Hey96, S.51] beschreibt. Auch bei dieser Frage wird deutlich, dass die Berufswelt für die Schüler

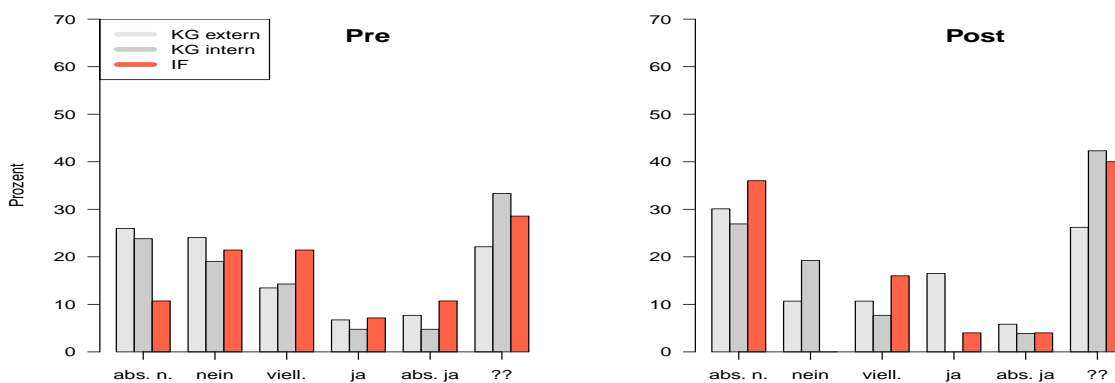


Abbildung 8.65: Ich möchte später irgend etwas mit Informatik oder Computern machen.

im Jahrgang 6 noch sehr weit entfernt ist, aber sie spielt gelegentlich doch schon eine Rolle. Abbildung [8.65] zeigt diese Unsicherheit im grossen Anteil der noch nicht Entschiedenen. Interessanterweise steigt dieser Anteil bei der internen Kontrollgruppe um fast 20% von 33% auf 52%, während er bei der externen Kontrollgruppe nur um 7% zunahm und der Informatikgruppe mit mit einem Anstieg um 2% auf 30,7% fast konstant bleibt. Ein leichter Anstieg ist bei der Informatikgruppe zudem bei den positiven Aussagen zur möglichen beruflichen Beschäftigung mit Computern festzustellen. Dieser Anstieg ist bei der externen Gruppe nicht

festzustellen. Aber auch nach einem Jahr gibt es fast genauso viele Schülerinnen und Schüler, die später nicht mit Computern arbeiten möchten. Der generell positive Trend lässt jedoch erwarten, dass bei weiterem Unterricht noch mehr Schülerinnen und Schüler einen Beruf im IT-Umfeld anstreben werden.

## Berufsbild

Ob Schülerinnen und Schüler Interesse für ein Berufsfeld entwickeln, hängt nicht nur von den Chancen in diesem Feld ab, sondern auch, welches Bild sie von den Berufen und von den dort arbeitenden Menschen haben. Dieses Bild muss nicht unbedingt mit der Realität übereinstimmen. Dies gilt natürlich erst recht von den informatiknahen Berufen, da sie für viele aussenstehende Personen mangels eigener schulischer Erfahrung kaum durchschaubar sind. Das Bild wird zudem nicht unwesentlich von den Medien beeinflusst.

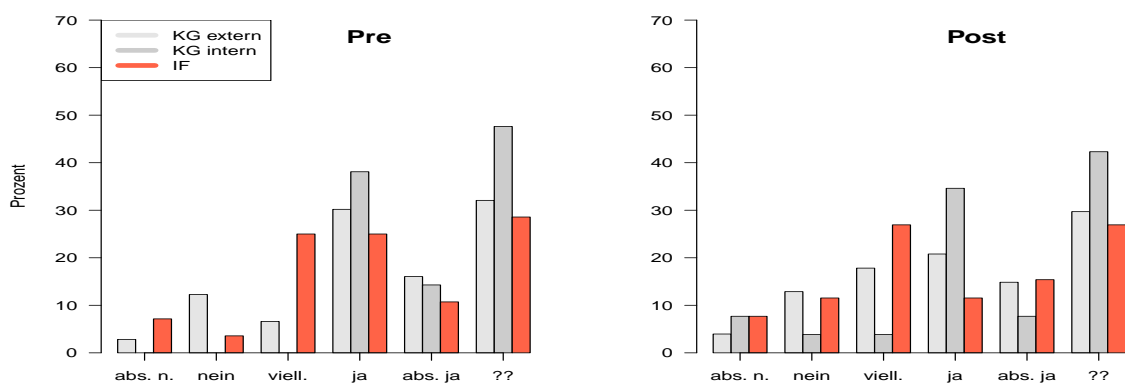


Abbildung 8.66: Ein Informatiker sollte ein guter Team-Worker sein.

Aus der Abbildung [8.66] kann entnommen werden, dass die Schülerinnen und Schüler, die einen ‚computernahen‘ Unterricht erfahren haben, deutlich mehr der Meinung zustimmen, dass Informatiker gute Teamworker sind bzw. sein sollen. Die Schülerinnen und Schüler aus der ‚computerlosen‘ internen Kontrollgruppe enthalten sich dagegen mit fast 60% der Teilnehmer. Es zeigt sich aber auch, dass eine kleine Minderheit aus der Informatikgruppe der Meinung ist, dass Teamarbeit nicht so entscheidend ist. Dies könnte vielleicht dadurch erklärt werden, dass die Schülerinnen und Schüler in bestimmten Phasen des Unterrichtes allein an je einem Computer sitzen und versuchen müssen, das gerade behandelte Programm bzw. Skript zu erstellen. Dabei ist der Nachbar zur Linken und Rechten nicht immer so eine Hilfe, wie mancher sie im kritischen Fall bräuchte.

Deutlicher differieren die Aussagen bei der Frage, ob der Informatiker ein sogenannter ‚Fachidiot‘ ist (Abbildung [8.67]). Im Pretest sind fast alle Antworten bis auf Ausnahmen im *Nein-Bereich* bei allerdings einer sehr hohen Enthaltung von praktisch 50% der internen Kontrollgruppe. Die Enthaltung ist bei der externen Kontrollgruppe nur etwa halb so gross und liegt bei der Informatikergruppe bei ca. 15%. Das Bild wandelt sich erheblich zum Posttest. Die Enthaltungen sind grössenordnungsmässig etwa gleich geblieben. Allerdings gibt es

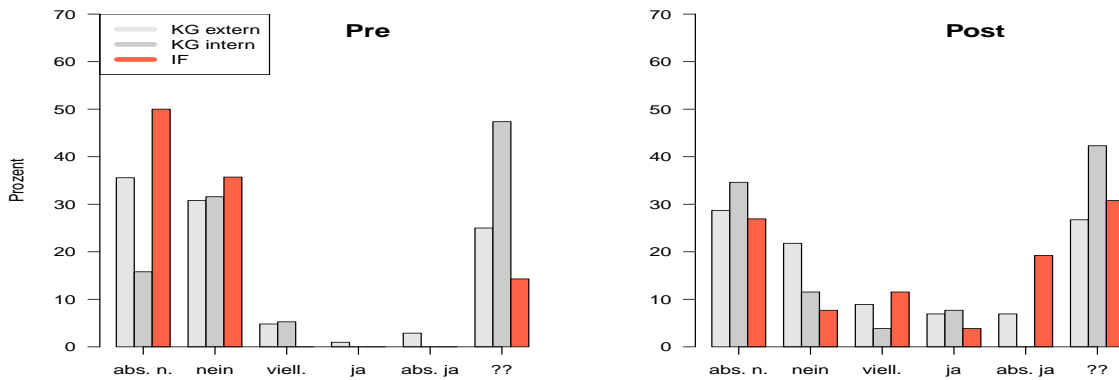


Abbildung 8.67: Ein Informatiker ist ein Fachidiot.

jetzt eine Gruppe von fast 30% der Informatiker und etwa 20% der externen Kontrollgruppe, die nun der Position zustimmen, dass Informatiker *Fachidioten* seien. Hier dürften die gelegentlichen Frustrationen im laufenden Unterricht bei konkreten Aufgabenstellungen, die nicht gelingen wollen, eine Rolle spielen. Und zweifelsohne dürfte der manchmal höhere Abstraktionsgrad im Informatik-Unterricht im Gegensatz zum ITG-Unterricht in den externen Kontrollgruppen zu mehr Misserfolgserlebnissen geführt haben.

Dies entspricht auch der Beobachtung im Unterricht. Auf der einen Seite ist es sicher oft für den Schüler positiv, wenn er bzw. sie alleine vor einem Computer sitzen kann und die Aufgaben versucht zu bewältigen. Allerdings fehlt dann die direkte Hilfe eines Mitschülers, wenn es – wie nicht selten – nicht auf Anhieb klappt. Der Lehrer kann bei einer Klassengröße von fast 30 Schülern nur sehr bedingt helfen. Das führt natürlich zu Frustrationserlebnissen. Auch auf der Lehrerseite ist es oft unbefriedigend, bei manchen Problemen nicht so schnell helfen zu können, wie es wünschenswert wäre.

Die Einstellung, Informatiker seien Fachidioten, ist sicher nicht positiv. Es ist daher überlegenswert, mehr als bereits gelegentlich im Untersuchungsjahr praktiziert, mit zwei Schülern pro Computer zu arbeiten, auch wenn dann gelegentlich sich mancher Schüler geistig der Arbeit entziehen kann. Probleme können dann besser gemeinsam angegangen werden und der Lehrer ist bei nur halb so vielen Arbeitsgruppen schneller ‚greifbar‘. Zudem verstärkt dies den Teamgeistgedanken positiv. Das optimale Verhältnis zwischen diesen Arbeitsformen muss noch experimentell bestimmt werden<sup>12</sup>.

## Genderproblematik

Informatik hat leider den Ruf eines *Jungenfaches*. Dementsprechend ist auch der Mädchenanteil in den Informatikkursen zu gering. Im Schuljahr 2010/2011 waren gerade einmal 4,4% der Schülerinnen in der Qualifikationsphase der gymnasialen Oberstufe in NRW in einem

<sup>12</sup>Dieses Problem stellt sich natürlich nur, wenn man wie an der FSG über den seltenen Luxus einer 1:1-Situation Schüler zu Computer verfügt.

Informatikkurs. Einen Leistungskurs belegten sogar nur 0,15% der Schülerinnen<sup>13</sup>. [Eng11] Aus der Genderforschung ist bekannt, dass die Einstellung zum häufig erwarteten Mädchen- und Frauenverhalten bis zur einsetzenden Pubertät geändert werden muss, wenn die Mädchen nicht doch überwiegend die tradierten Berufsfelder anstreben sollen [Bud08, S.21], [Wen07, S.19]. Je früher die Mädchen konkrete Erfahrungen mit eher untypischen Fächern machen können, desto mehr kann sich ihre Einstellung gegen andere Positionen in ihrer persönlichen Umgebung wie auch in der allgemeinen gesellschaftlichen Struktur durchsetzen. Dies ist auch eine wichtige Begründung für einen in der Schule in frühen Jahren einsetzenden Informatikunterricht, wie er hier in dem Forschungsvorhaben umgesetzt wurde.

Daher war es natürlich von besonderem Interesse, wie die Schülerinnen und Schüler Fragen zu Mädchen und Jungen bezug auf das Fach Informatik beantworten.

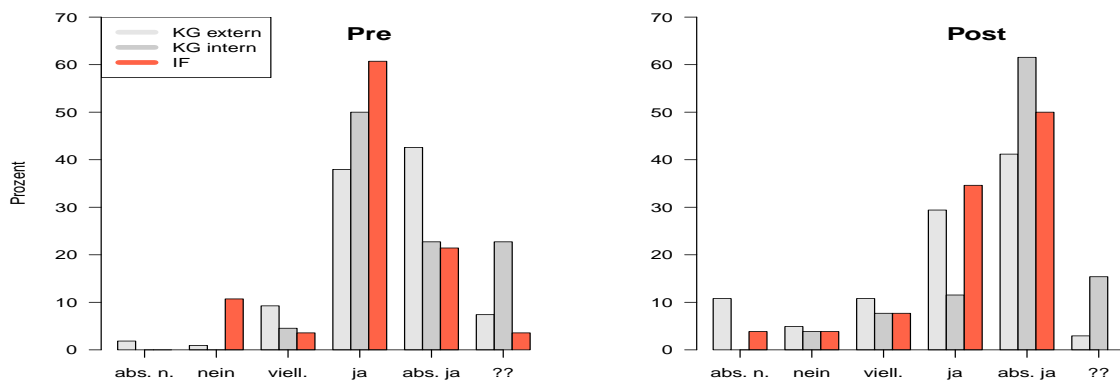


Abbildung 8.68: Informatik ist für Jungen und Mädchen genauso gut geeignet.

Die Aussage, Informatik ist für Jungen und Mädchen gleichermaßen geeignet, wird mit einer hohen Zustimmung von über 70% in allen Gruppen bejaht (Abbildung [8.68]). Es gibt in der Informatikgruppe einen Anteil von 10,7% im Pre- und 12,7% im Posttest, die dies allerdings anders sehen. In der internen Kontrollgruppe bleibt diese Quote unter 5%. Bei der externen Kontrollgruppe steigt der Anteil allerdings von 2,7% auf 12,7%. Berücksichtigt man, dass in der FSG sehr stark auf gleiche Behandlung von Mädchen und Jungen geachtet wird, könnte sich in der deutlich gestiegenen Ablehnung dieser Aussage in der Kontrollgruppe andeuten, dass die geschlechterspezifische Ablehnung von Informatik langsam einsetzt.

Differenzierter ist die Beantwortung der Frage ‚Jungen sind in Informatik besser als Mädchen‘ entsprechend Abbildung [8.69] Die Mehrheit in allen drei Gruppen lehnt diese Aussage ab, jedoch sinkt diese in den beiden Gruppen mit Informatikbezug um jeweils ca. 3% ab, während sie in der internen Kontrollgruppe sogar um fast 10% zunimmt. Gleichzeitig steigt in allen Gruppen die Zustimmung zu dieser Aussage von ursprünglich sehr niedrigen Werten auf immerhin 22,2% in der Informatik-Gruppe und 18,1% in der ITG-Gruppe. Die Beschäftigung mit Informatik-spezifischen Themen in jungen Jahren führt also nicht automatisch dazu,

<sup>13</sup>[http://www.schulministerium.nrw.de/BP/Schulsystem/Statistik/2010\\_11/StatUebers373.pdf](http://www.schulministerium.nrw.de/BP/Schulsystem/Statistik/2010_11/StatUebers373.pdf), S.82 (letzter Zugriff: 16.1.2012)



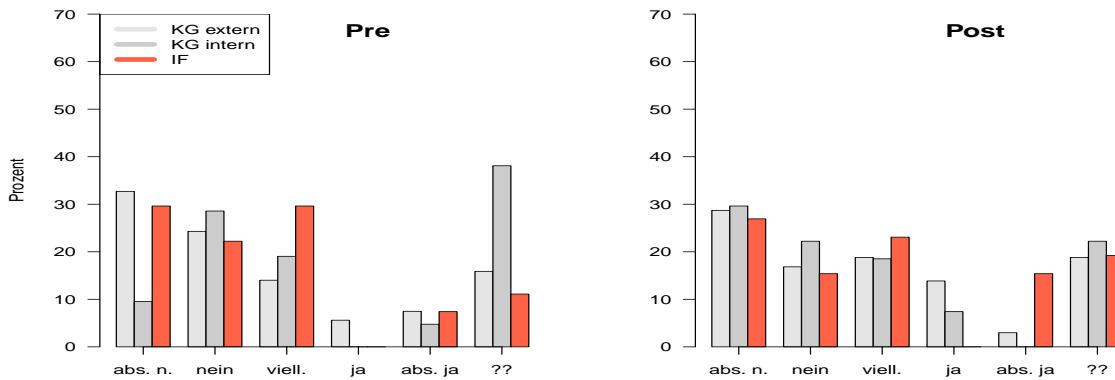


Abbildung 8.69: Jungen sind in Informatik besser als Mädchen.

dass die Schüler und Schülerinnen die Ergebnisse geschlechterneutral betrachten. Auffällig ist jedoch, dass im Pretest mit 29,6% der Schüler und Schülerinnen deutlich mehr als in den Kontrollgruppen mit 19% bzw. 14% sich der Stimme enthielten. Im Posttest sank bei allen Gruppen dieser Wert unter 20%. Diese zuvor Unentschlossenen dürften für den hohen Anstieg der Zustimmung verantwortlich sein.

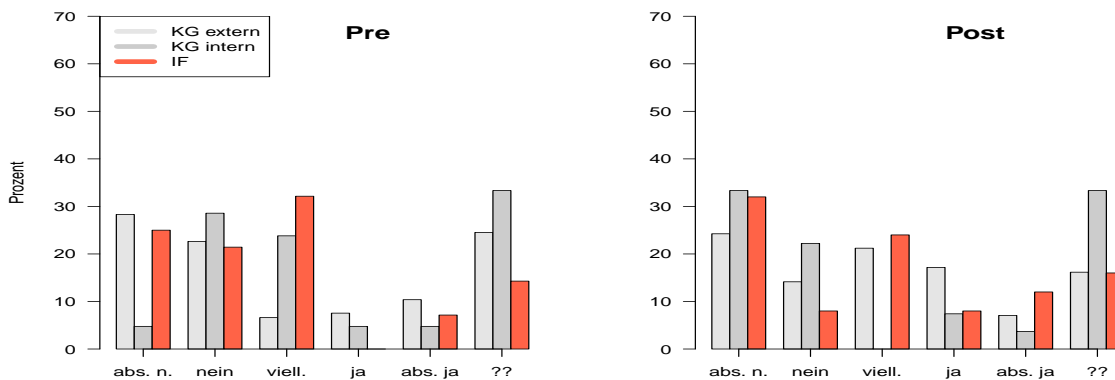


Abbildung 8.70: In Informatik haben Jungen mehr Erfolg als Mädchen.

Die Antwort, ob ‚Jungen auch einen höheren Erfolg in Informatik‘ haben, werden entsprechend Abbildung [8.70] ganz ähnlich beantwortet. Auch hier nimmt in allen Gruppen die Zahl der Unentschlossenen ab. Gleichzeitig steigen sowohl die Ablehnung als auch die Zustimmung zu dieser Aussage. Überraschend steigt bei der internen Kontrollgruppe die Ablehnung um fast 30% und sinkt sogar die Zustimmung um mehr als die Hälfte auf unter 4%.

Die Kontrollfrage, ‚Mädchen sind genauso erfolgreich wie Jungen in Informatik‘, wird entsprechend Abbildung [8.71] ähnlich kritisch beantwortet. Beim Pre- und beim Posttest stimmt die Mehrheit dieser Aussage zu, allerdings steigt diese im Wesentlichen nur bei der internen

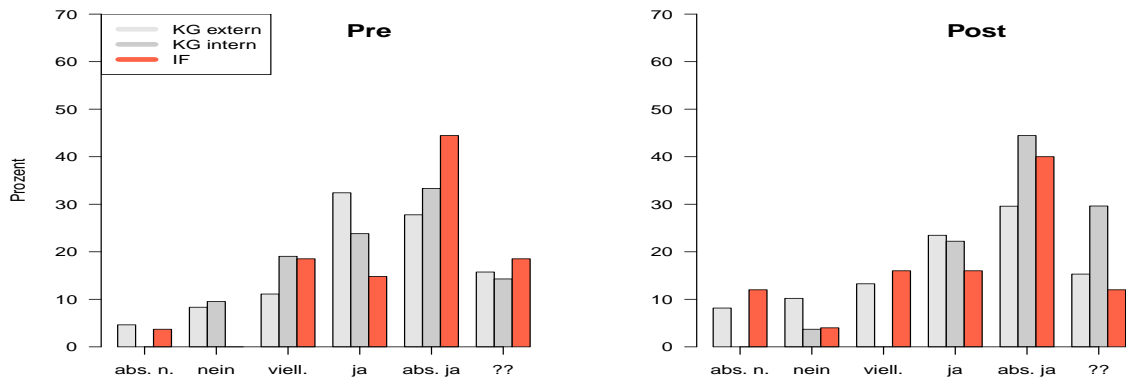


Abbildung 8.71: Mädchen sind genauso erfolgreich wie Jungen in Informatik.

Kontrollgruppe an. Die Veränderungen dürften auch hier in erster Linie durch die zuvor Unentschlossenen entstanden sein.

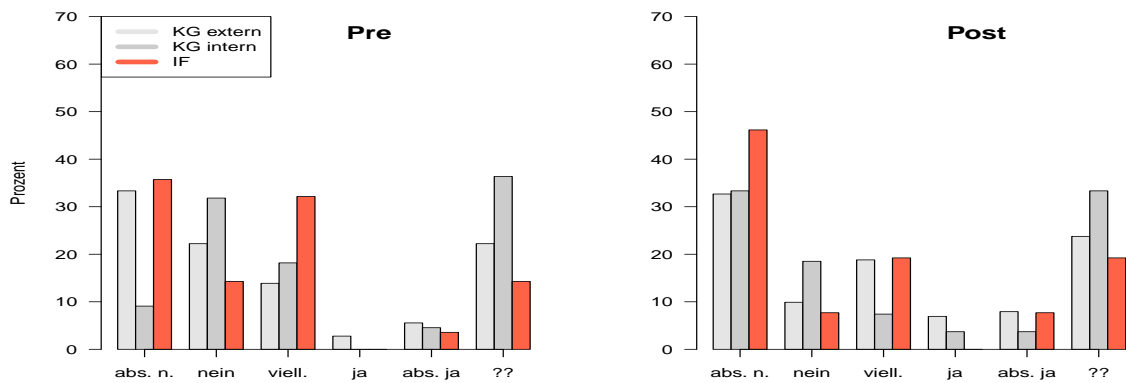


Abbildung 8.72: Mädchen sind besser in Informatik als Jungen.

In der nächsten Frage, ‚Mädchen sind besser in Informatik als Jungen‘, zeigt sich jedoch, dass ein Teil der Schülerinnen und Schüler der Meinung ist, dass Mädchen sogar besser in Informatik als Jungen sind Abbildung [8.72]. Während beim Pretest in keiner der drei Gruppen die Zustimmung über 10% liegt, steigt diese in allen Gruppen auffällig. Bei der Informatikgruppe nimmt der Anteil von unter 4% um über 15 Prozentpunkte zu.

### Sichtweise von Mädchen und Jungen

Natürlich interessiert gerade in der Genderfrage, ob sich die Antworten der Mädchen von denen der Jungen unterscheiden. Unsere Absicht ist es unter anderem, die Mädchen für Informatik zu sensibilisieren und zu interessieren und dabei Ängste und Vorurteile abzubauen. Die

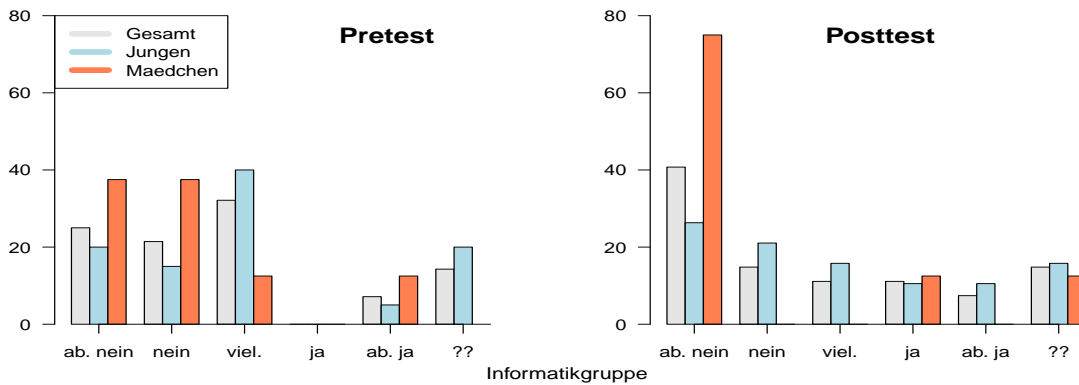


Abbildung 8.73: In Informatik haben Jungen mehr Erfolg als Mädchen.

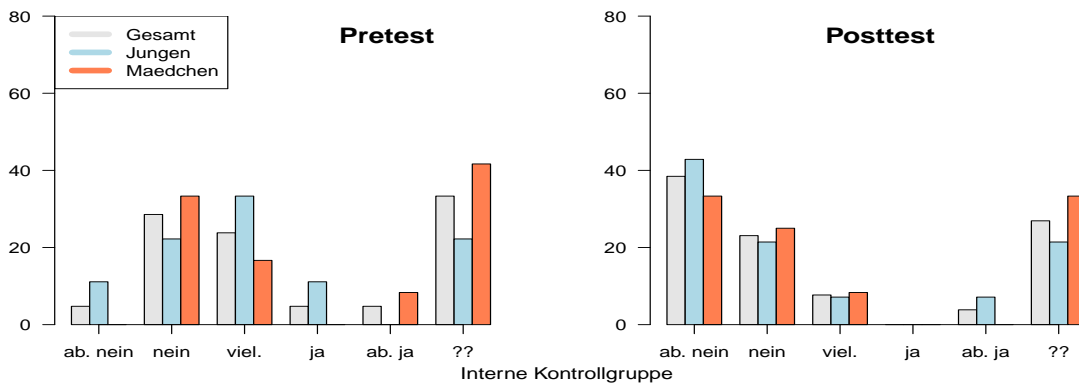


Abbildung 8.74: In Informatik haben Jungen mehr Erfolg als Mädchen.

Antworten zu den Fragen zeigt, dass dies allerdings für die Gesamtheit aller Schülerinnen und Schüler nicht so zutrifft.

Die Grafiken entsprechend 8.73 und 8.74 zeigen, dass die Sichtweise von Mädchen und Jungen in den verschiedenen Gruppen unterschiedlich ist. Die Mädchen und Jungen in der internen Kontrollgruppe haben ähnliche Meinungen. Sie sind mehrheitlich überzeugt, dass Jungen keinen grösseren Erfolg in Informatik haben. Dies entspricht dem Anspruch der FSG, keine Unterschiede zwischen Jungen und Mädchen in der schulischen Wirklichkeit zuzulassen und zu praktizieren. Allerdings gibt es gerade unter den Mädchen im Pretest einen Anteil von fast 40%, die nicht wissen, ob diese Aussage falsch oder richtig ist. Dieser Anteil sinkt zum Posttest nur wenig ab. Es ist zu vermuten, dass dieser doch grosse Anteil in dieser Frage verunsicherter Schülerinnen in den späteren Jahren die üblichen Rollenzuweisungen übernehmen werden. Dieses gilt auch dann, wenn man berücksichtigt, dass sich in allen Gruppen die Position verstärkt, dass Mädchen und Jungen bezüglich der Informatik gleichwertig sind. Hier werden sicher ausserschulische Erfahrungen und altersgemässe Entwicklungen ihre Auswir-

kung gehabt haben<sup>14</sup>. In der Informatikgruppe sind die Ergebnisse ausgeprägter. Bis auf Ausnahmen haben die Mädchen ein starkes Selbstbewusstsein, dass sich zum Posttest sogar noch verstärkt. Allerdings ist erkennbar, dass die Jungen vom Pretest zum Posttest teilweise eine Meinung entwickeln, dass sie Informatik besser könnten als Mädchen. Dies kann dadurch begründet sein, dass sie im konkreten Unterricht erfahren haben, dass es auch Mädchen mit nicht so guten Leistungen gibt und sie diese Erfahrungen verallgemeinern. Auch nicht so gute Leistungen in diesem Fach führen bei den Mädchen dagegen nicht zur Meinung, dass es an ihrem Geschlecht liegt. Der Vergleich macht deutlich, dass ein Informatikunterricht unabhängig von der eigenen Leistungsfähigkeit bei den Mädchen zu einem geschlechtsspezifischen Selbstbewusstsein führt, dagegen fehlender Informatikunterricht bei einem grossen Teil der Mädchen zu einer Unsicherheit bezüglich des Faches Informatik führt. Es sei hier aber kritisch angemerkt, dass diese Aussagen mit etwas Vorsicht zu betrachten sind und nicht überbewertet dürfen, da in der Informatikgruppe nur 8 Mädchen vertreten sind.

### Einfluss des Lehrers

Das Fach Informatik an der Fritz-Steinhoff-Gesamtschule bzw. das Fach Informatik schwerpunktmässig als informationstechnologische Grundbildung an der Realschule Halden sind Fächer im Rahmen des Wahlpflichtangebotes. Neben sachlichen Erwägungen seitens der Schüler und deren Eltern spielen viele emotionale Gründe eine Rolle, dieses Fach zu wählen. Gerade an der FSG, an der dieses bisher einmalige Projekt stattfindet, ist sicher ein Einfluss der Lehrer vorhanden, dieses Fach zu wählen.

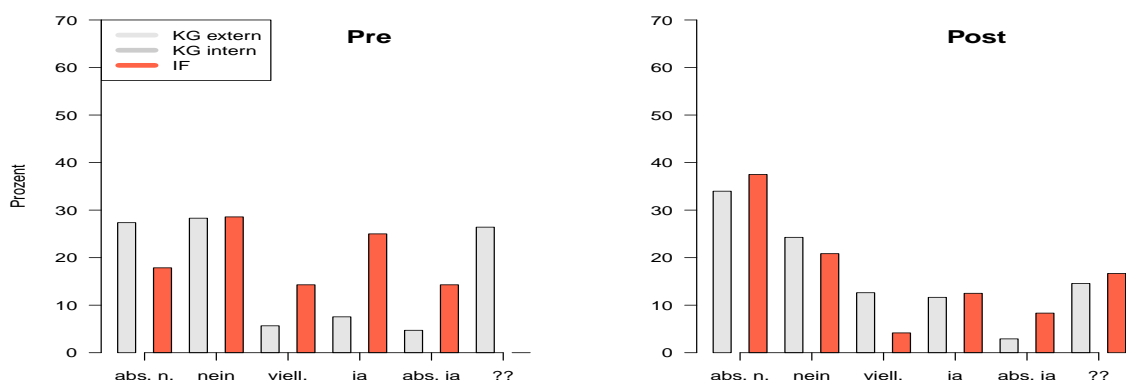


Abbildung 8.75: Mein Lehrer hat mir Informatik als Fach empfohlen.

Dies zeigt sich an den Antworten entsprechend Abbildung [8.75], dass verständlicherweise die Schüler an der FSG in einem höheren Maße eine Diskussion mit ihren Lehrern über dieses neue Angebot geführt haben.

Der Einfluss auf die konkrete Wahl war dann offensichtlich deutlich geringer. Es zeigt, dass Lehrer mit ihren Empfehlungen nur einen beschränkten Einfluss auf die konkrete Wahl des

<sup>14</sup>siehe hierzu auch *Taub et al.* [TBAA09]

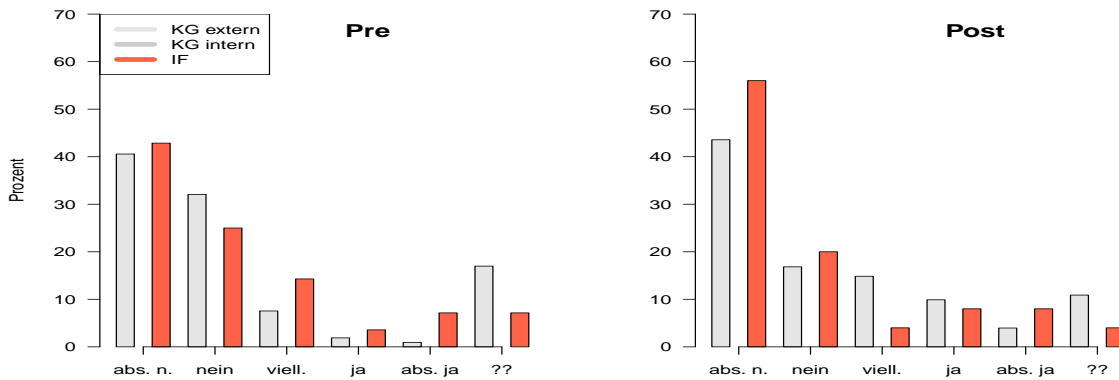


Abbildung 8.76: Ich habe Informatik wegen des Lehrers gewählt.

Schülers haben.

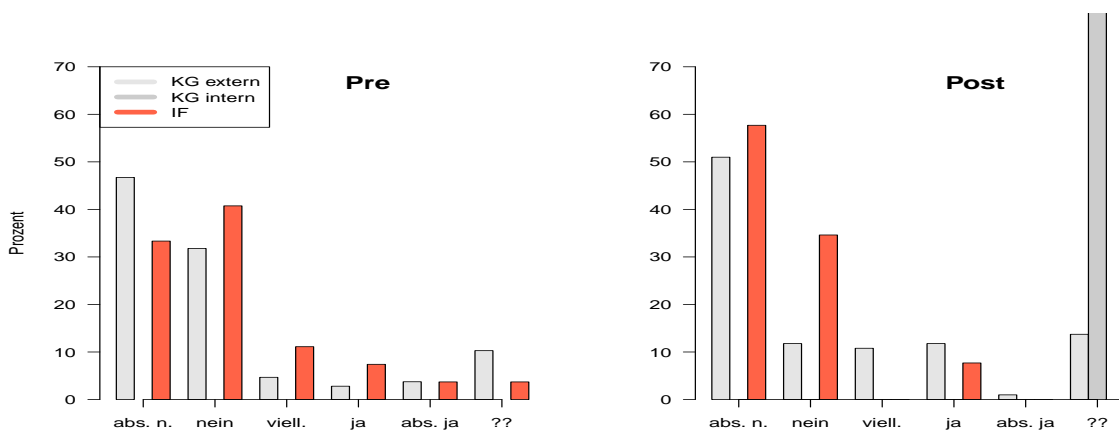


Abbildung 8.77: Ich habe Informatik gewählt, weil meine Eltern das wollten.

Der Einfluss der Eltern ist ebenfalls nicht sehr gross. Es ist nachvollziehbar, dass die Schülerinnen und Schüler an der FSG in dem besonderen Fall des Wahlpflichtkurses Informatik/Physik in etwas stärkeren Maß als üblich in die Diskussion und Entscheidung um das Wahlpflichtfach einbezogen waren.

Dieses war schon bei der Vorstellung des Angebotes im März 2010 zu spüren. Die dort zahlreich anwesenden Eltern und Schüler haben sich dort umfassend über das Fach informiert. Das es anschliessend mit den Kindern zu Hause intensive Diskussionen und teilweise Empfehlungen für oder gegen das Fach Informatik gab, ist naheliegend.

Die meisten Schülerinnen und Schülern in der FSG und in der RS Halden haben schliesslich die Wahl aus eigenen Erwägungen getroffen. Dies ist zweifelsohne sehr erfreulich.

## 8.9.2 Zusammenfassung

Der Computer ist ein täglicher Begleiter heutiger Jugendlicher. Überwiegend wird er zu Freizeitaktivitäten wie Spielen verwendet. Das ist verständlich, denn Spielen ist ein sehr wichtiger Teil des kindlichen Daseins und für die Entwicklung des Kindes und Jugendlichen notwendig. „Je länger die Entwicklungszeit bis zum Erwachsenenalter dauert, desto bedeutsamer und vielfältiger wird das Spiel. Beim Menschen tritt Spielverhalten in allen Kulturen auf und nimmt eine bedeutende Rolle mindestens bis zur Pubertät ein.“ [Oer07] stellt der Entwicklungspsychologe *Rolf Oerter* fest. Ein Gegenstand der Alltagswelt wird von Kindern und Jugendlichen ausserhalb des schulischen Umfeldes weitgehend spielerisch erobert. Das Erobern der Funktionsweise von Computern und der dahinter stehenden Technologien ist mangels entsprechenden Spielmaterials und Anleitung seitens der Kinder nur beschränkt möglich. Allein deshalb ist die Auseinandersetzung mit der Technik, Funktionsweise und Struktur des Computers zumindest heute eine Aufgabe von Schule. Die Schülerinnen und Schüler wollen aber mehr als nur *Spielen*. Das gilt für die Schüler aus allen untersuchten Gruppen, jedoch deutlich mehr für die Informatikgruppe. Diese unterstützen auch deutlich mehr und dies noch stärker nach einem Jahr Unterricht in Informatik die Aussage, dass Basiskenntnisse in Programmieren wichtig sind. Diese werden nach der Einschätzung der meisten Schüler aber erst gelernt, wenn man diese auch konkret am Computer ausprobiert.

Vorstellung von der Berufswelt und eigene Ideen für die zukünftige Berufsentscheidung sind nur teilweise vorhanden. Auffällig ist, dass nach einem Jahr Informatik der Anteil der Schülerinnen und Schüler, die noch nicht wissen, welchen Beruf sie später ausüben wollen, zugenommen hat. Gleichzeitig nimmt bei dieser Gruppe der Anteil zu, die sich vorstellen können, einen Beruf im Umfeld der Informationstechnik auszuüben. Natürlich gibt es auch weiterhin einen Anteil, die dies nicht möchten. Es sehen auch einige in den Informatikern Fachidioten. Dieses Bild wird hoffentlich in den folgenden Unterrichtsjahren abgemildert.

In allen Gruppen ist die deutliche Mehrheit der Meinung, dass Mädchen und Jungen gleichermaßen für Informatik geeignet sind. Die eigenen Erfahrung mit den Teilnehmern in ihrem Kurs, der nicht repräsentativ für alle Mädchen und Jungen ist, führt dazu, dass Jungen und Mädchen viel differenzierter die Fähigkeiten von Jungen und Mädchen beurteilen. Das Selbstbewusstsein der (relativ wenigen) Mädchen im Informatikkurs wird gestärkt, auch wenn einige Mädchen persönliche inhaltliche Probleme mit dem Fach haben.

Generell lässt sich feststellen, dass Veränderungen erkennbar sind. Veränderungen, die eine etwas grössere Offenheit gegenüber dem Fach signalisieren. Ebenso werden die in diesem Fach ansatzweise vermittelten Kenntnisse als notwendige Kenntnisse erachtet. Dieses Veränderungen lassen hoffen, dass die Schülerinnen und Schüler für ihre eigenen Berufs- und Lebensentscheidungen den Bereich der IT einbeziehen.

## 8.10 Follow-Up – Untersuchungen

Ziel schulischer Unterweisung ist der Erwerb von Kompetenzen fachlicher sowie überfachlicher Art. Diese werden nur über einen langen Zeitraum erworben. Es wäre vermessen zu erwarten, dass nach dem einmaligen Unterrichten einer Thematik die Inhalte nach einer be-

stimmten Zeit vollständig erinnert oder sogar angewandt werden können. Andererseits sollte natürlich auch nicht Alles vergessen werden, da dies Unterricht generell in Frage stellen würde.

Um zu überprüfen, welche Inhalte und Kompetenzen über einen längeren Zeitraum erhalten geblieben sind, wurde eine *Follow-Up* - Untersuchung durchgeführt. Diese fand in der Form einer *Panel-Studie* [Fri90, S.366] statt. Dabei wurden denselben Schülern im Februar 2012 die Fragebogen noch einmal ausgeteilt. Ebenso wurde ein Teil der Schüler wiederum in kurzen *Think-Aloud*-Interviews befragt. Erwartet wurde, dass wesentliche im Unterricht behandelte Begriffe zumindest teilweise von den Schülern erinnert werden konnten, jedoch die inhaltlichen Eigenschaften nicht mehr unbedingt beherrscht werden.

Es sollen hier nicht die Ergebnisse der gesamten Follow-Up-Befragung behandelt werden, sondern wir beschränken uns auf die Fragen zu den im ersten Jahr im Unterricht intensiver behandelten Themen sowie auf interessante Ergebnissen in der Einstellung der Schülerinnen und Schüler<sup>15</sup>.

## 8.10.1 Ausgewählte Beispiele aus der Follow-Up-Untersuchung

### Grafikformate

Im ersten Jahr des Unterrichts wurde in verschiedenen Einheiten mit Grafiken gearbeitet. Ein Ziel war das Herausarbeiten der Unterschiede zwischen Raster- und Vektorgrafiken. Diese wurden in projektartigen Zusammenhängen beispielsweise bei der Erstellung der eigenen Stundenpläne eingebettet in eine Webseite verwendet. Bei der Einführung von Pixelgrafiken wurde auch mit konkreten Grafikdateiformaten wie dem *jpg*-, *gif*- und dem *xpm*-Format gearbeitet. Die Schülerinnen und Schüler sollten daher sowohl wissen, dass eine Grafik in unterschiedlicher Struktur dargestellt werden kann, sie sollten die prinzipiellen Eigenschaften der Strukturen kennen als auch konkrete Formate zuordnen können. Die Ergebnisse der Befragung zu den Formaten sehen wir in den Abbildungen [8.78] und [8.79].

Es zeigt sich, dass in der Follow-Up-Untersuchung vergleichbare Werte wie in der Post-Untersuchung gemessen wurden. Das gilt für beide Formate und für alle untersuchten Gruppen. Die Werte für die Informatik-Gruppe unterscheiden sich dabei sehr stark von den Werten der Pre-Untersuchung. Die Begrifflichkeit und die Zuordnung dieser beiden Formate zu einer konkreten Grafikstruktur ist durch die relative häufige Verwendung im Unterricht ausreichend verankert worden, sodass die Bedeutung auch bis zur Nachuntersuchung nicht vergessen wurde. Bei der zusammengefassten Kontrollgruppe ist der Unterschied zur Pre-Untersuchung nur bedingt erkennbar. Insbesondere ist beim GIF-Format nur eine Steigerung der Ja-Stimmen von knapp 30% auf 40% zu beobachten, während der Kenntnisstand in der Informatikgruppe von knapp 30% auf über 70% steigt. Beim JPG-Format ist dieser Unterschied nicht ganz so deutlich, aber vorhanden. Ein derartig grosser Unterschied ist durch das unterrichtliche Geschehen zu begründen.

Die Kontrollfragen zur Pixel- und zur Vektorgrafik fallen vom Ergebnis sehr ähnlich aus. Vor allem die Zustimmung zur Kenntnis von Pixel- bzw. Vektorgrafik bei der Informatikgrup-

---

<sup>15</sup>Die grafische Gesamtdokumentation der Befragung *IF-Gruppe im Vergleich zu allen anderen Gruppen* befindet sich im Anhang( 11.9)

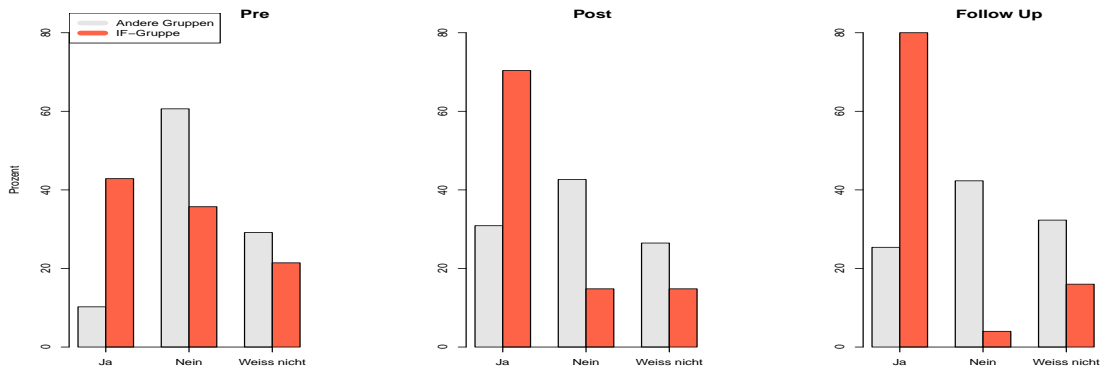


Abbildung 8.78: Ich kenne das JPG-Format.

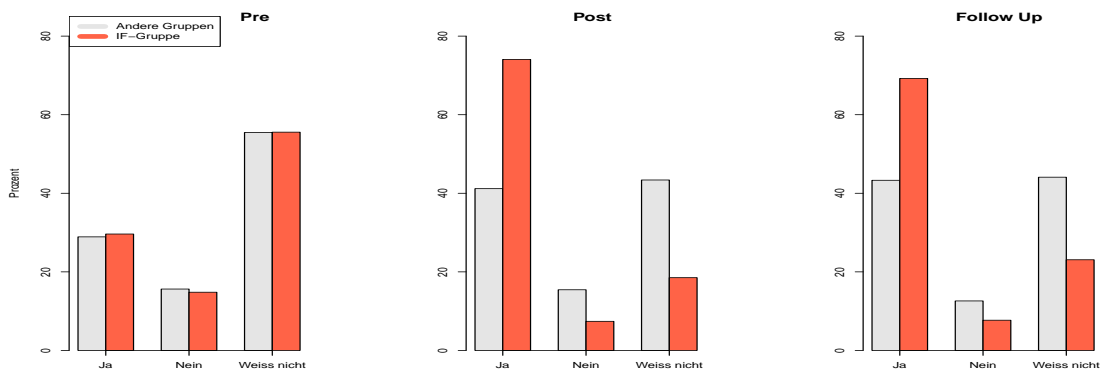


Abbildung 8.79: Ich kenne das GIF-Format.

pe ist noch höher als bei der Frage nach den Formaten. Sofern diese Kenntnisse in der Selbsteinschätzung mit teilweise über 80% Zustimmung auch nur annähernd in der Think-Aloud-Untersuchung bekräftigt werden sollten, kann hier zweifelsohne von einem unterrichtlichen Erfolg gesprochen werden.

## Variablen

Zum Ende des ersten Jahres Informatikunterricht wurde eine erste Einführung des Variablenbegriffes vorgenommen. Dieses Thema ist zweifelsohne deutlich abstrakter und damit schwieriger, da viele Schüler das dafür notwendige abstrakte Denkniveau noch nicht erreicht haben. Entsprechend handelte es sich im Jahrgang 6 um eine erste Einführung, die von konkreten Objekten ausging und noch kein vollständiges Verständnis des Variablenbegriffes als auch der Vorteile des Arbeitens mit Variablen erzeugen konnte. Dieses wird erst in späteren Unterrichtseinheiten erfolgen.



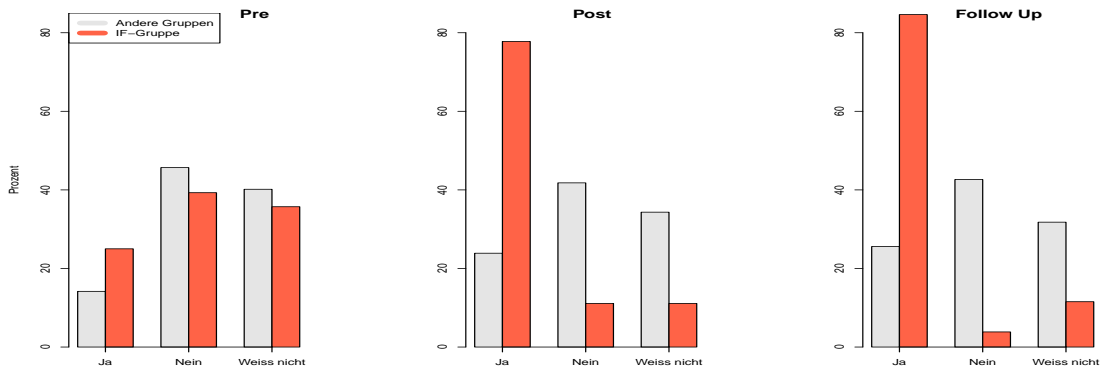


Abbildung 8.80: Ich weiss, was eine Vektorgrafik ist.

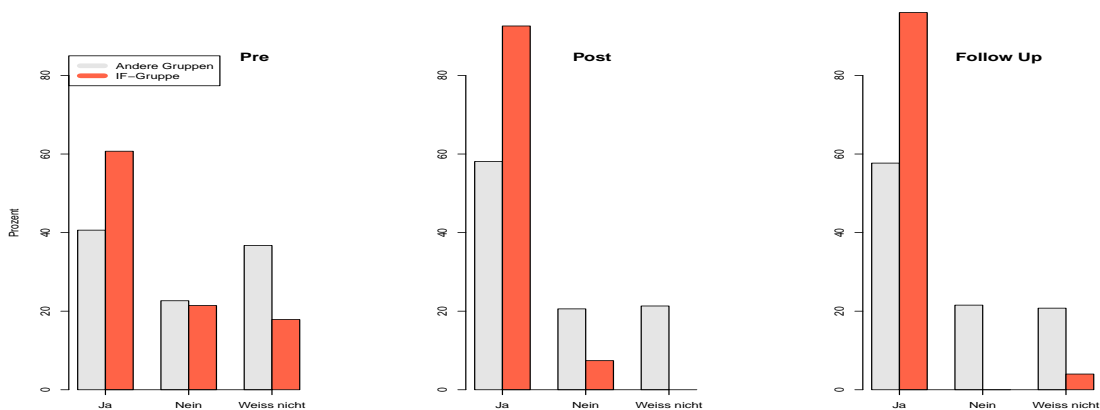


Abbildung 8.81: Ich weiss, was eine Pixelgrafik ist.

Die Abbildung [8.82] zeigt, dass der Begriff Variable den Informatik-Schülern auch bis hin zur Nachuntersuchung in Erinnerung geblieben ist. Die Bedeutung dieses Begriffes dagegen ist schon zur Zeit des Unterrichts entsprechend Abbildung [8.83] nicht allen Schülern klar geworden. Während der Begriff von immerhin über 80% der Schüler gekannt wird, können nur knapp über 55% der Schüler den Bedeutungsinhalt angeben. Diese Kenntnis sinkt in der Nachuntersuchung auf unter 40% und damit fast auf das Niveau der zusammengefassten Kontrollgruppe. Hiermit wird deutlich, dass ein spiralisches Unterrichten zwingend notwendig ist, wenn eine nachhaltige Wirkung unterrichtlicher Inhalte gerade schwieriger Begriffe angestrebt wird. Da die Bedeutung und der Begriff der Variablen in den Kontrollgruppen im Informatikunterricht bzw. ITG-Unterricht praktisch keine Rolle gespielt hat, verändern sich die Kenntnisse in der Gesamtkontrollgruppe über den gesamten Zeitraum praktisch nicht. Dies macht deutlich, dass ein Begriffsverständnis über Variable ausserhalb von Unterricht zumindest in dieser Altersgruppe nicht entwickelt wird.

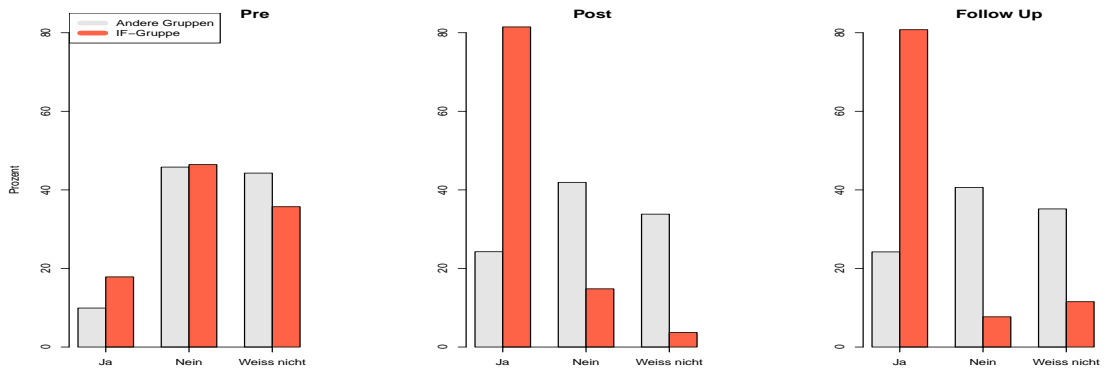


Abbildung 8.82: Ich weiss, was eine Variable ist.

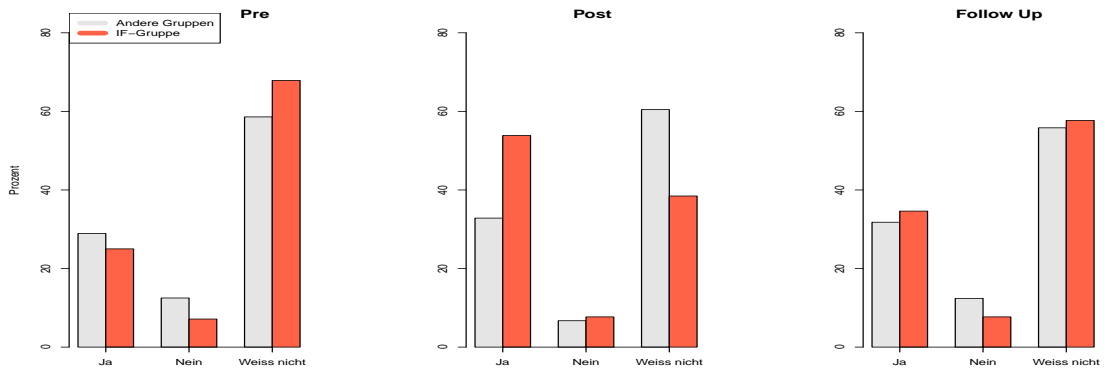


Abbildung 8.83: Ein Wert kann in einer Variablen gespeichert werden.

## Verzeichnisse und Verzeichnisbäume

Computer und Netze gehören zu den Werkzeugen produktiv tätiger Menschen seit dem ausgehenden zwanzigsten Jahrhundert. Die Verwaltung von Dokumenten in Dateien in Verzeichnissen und Verzeichnisbäumen gehört damit bereits zum notwendigen Allgemeinwissen. Dieses Wissen wird nur teilweise im Alltagsleben nebenbei erworben.

Der Begriff *Verzeichnis* bzw. *Ordner* gehört für Schüler inzwischen oft zum Alltagswissen. Nur wenige Schüler können mit diesen Begriffen nichts anfangen. Bei den Schülern der Informatikgruppe sinkt die Quote der Unwissenden von 11% auf 4% (entspricht einem Schüler), bei den Kontrollgruppen sind es immerhin über 20% und im Follow-Up-Test sogar wieder ansteigend über 25%, die mit den Begriffen nichts anfangen können. Dies zeigt, dass selbst bei derartigen scheinbar einfachen Alltagsbegriffen ohne Unterricht nicht mit einem Wissenszuwachs gerechnet werden kann.

Deutlich extremer sind die Unterschiede beim Begriff *Verzeichnisbaum*. Bei den Kontroll-

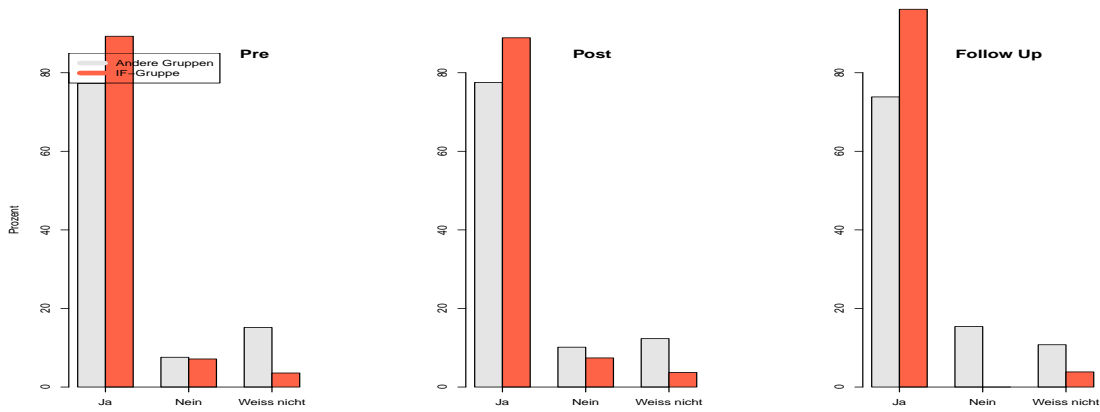


Abbildung 8.84: Ich weiss, was ein Verzeichnis bzw. ein Ordner ist.

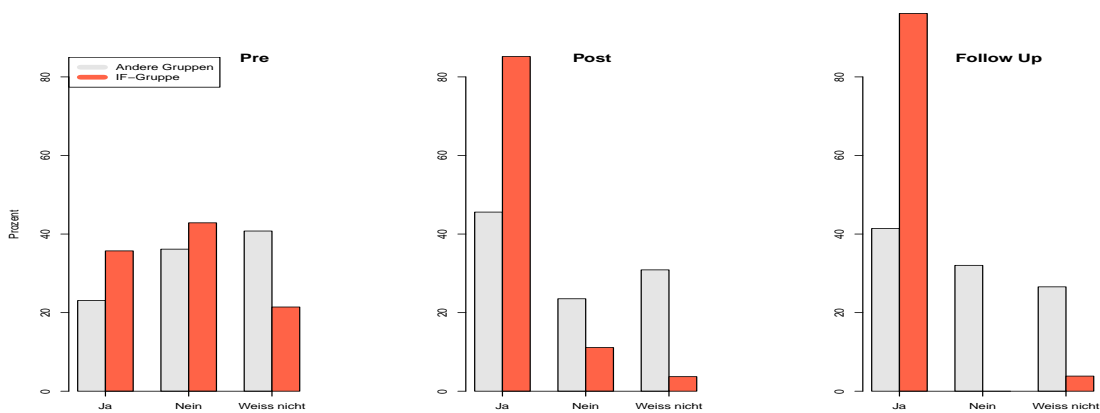


Abbildung 8.85: Ich weiss, was ein Verzeichnisbaum ist.

gruppen erreichen wir maximal im Posttest einen Kenntnisstand von knapp 45%, der sogar zum Follow-Up-Test auf fast 40% sinkt, während bei den Informatikschülern der Begriff als solcher praktisch dieselben Werte erreicht wie die beim Begriff ‚Verzeichnis‘.

Es ist zu vermuten, dass hinter den nicht verfestigten Begriffen auch ein unsicheres und unsystematisches Umgehen mit Verzeichnissen und überlegtem Ablegen von Dokumenten in Verzeichnisstrukturen einhergeht.

## Programmieren

Die Schüler haben im ersten Jahr nur rudimentäre Ansätze von Programmiersprachen und des Programmierens kennen gelernt. Allerdings wurden an den verschiedensten Stellen Sequenzen von Anweisungen erstellt und diese direkt in die Konsole eingegeben oder mit einem Editor erstellt und dann ausgeführt. In diesem Zusammenhang wurde gelegentlich, wenn auch nicht systematisch das Wort ‚Programmieren‘ verwendet. Die Schüler im Informatikkurs kön-

nen sich daher *Programmieren* im gewissen Rahmen vorstellen. In den Vergleichskursen ist eine derartige Einführung nicht erfolgt. Es existiert allerdings eine sicher diffuse Vorstellung aus der Alltagswelt über das ‚Programmieren‘ als Tätigkeit in der Informatik.

Bei den Schülerinnen und Schülern in der Kontrollgruppe bleibt die Zustimmung bei der Frage nach der Wichtigkeit des Programmierens mit ‚stimmt‘ und ‚stimmt völlig‘ entsprechend Abbildung [8.80] zu allen Zeitpunkten mit 57% bis 60% praktisch unverändert, während sie in der Informatikgruppe von 66% über 74% auf 84% ansteigt. Es ist auffällig, dass in der Post-Befragung die Zustimmung in der höchsten Kategorie ‚stimmt völlig‘ überwiegt, während in der Pre- und in der Follow-Up-Untersuchung gegenüber die ‚stimmt‘-Kategorie überwiegt.

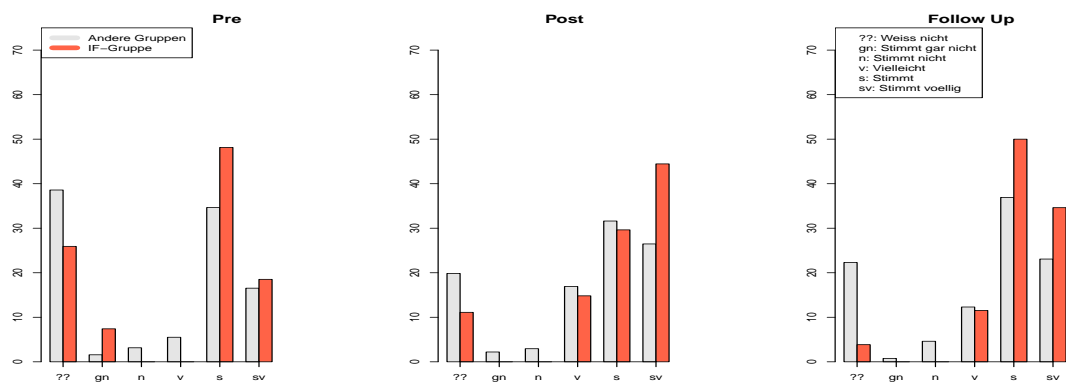


Abbildung 8.86: Programmierung ist in Informatik sehr wichtig.

Diese Tendenz wird auch in der Kontrollfrage bestätigt, in der nach der Rolle der Programmiersprache gefragt wird. Abbildung [8.87] zeigt, dass die Grössenordnung in der Zustimmung der summierten oberen Kategorien von 47% auf 60% steigt, jedoch in der Follow-Up-Befragung eine deutliche Verschiebung von ‚stimmt völlig‘ nach ‚stimmt‘ stattfindet.

Bei der Frage nach der Schwierigkeit des Programmierens zeigt sich, dass die Position der Schüler der Kontrollgruppen in allen drei Befragungen bei allen Fragen zur Programmierung praktisch unverändert ist. Verständlicherweise ist daher die Kategorie mit der grössten Zustimmung zumeist ‚Weiss nicht‘. Die Informatikschüler sind nach einem halben Jahr ohne Unterricht in Informatik nicht mehr sicher, ob Programmieren leicht ist. Fast 54% antworten mit ‚Vielleicht‘, während in der Post-Befragung nur 38% mit ‚vielleicht‘ geantwortet hatten und doch zumindest 30% statt schlussendlich 8% mit ‚stimmt‘ oder ‚stimmt völlig‘ votiert hatten.

Diese stärkere Verunsicherung wird auch bei in der leicht geänderten Fragestellung ‚Ein Computerprogramm ist nicht einfach zu erstellen‘ deutlich. Während in der Post-Untersuchung 60% gegenüber 39% in der Pre-Untersuchung dieser Aussage zustimmen, sind es in der Follow-Up – Untersuchung entsprechend Abbildung [8.89] nur noch 31%.

Auch das eigene Selbstbewusstsein in Sachen ‚Programmierung‘ leidet mit dem zeitlichen Abstand. Die von 41% auf 36% leicht gesunkene Überzeugung zwischen Pre- und Posttest,

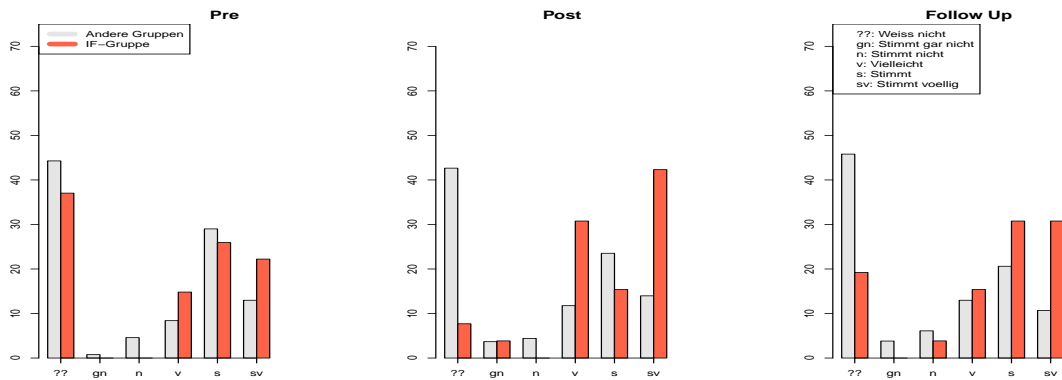


Abbildung 8.87: In der Informatik sind Programmiersprachen wichtig.

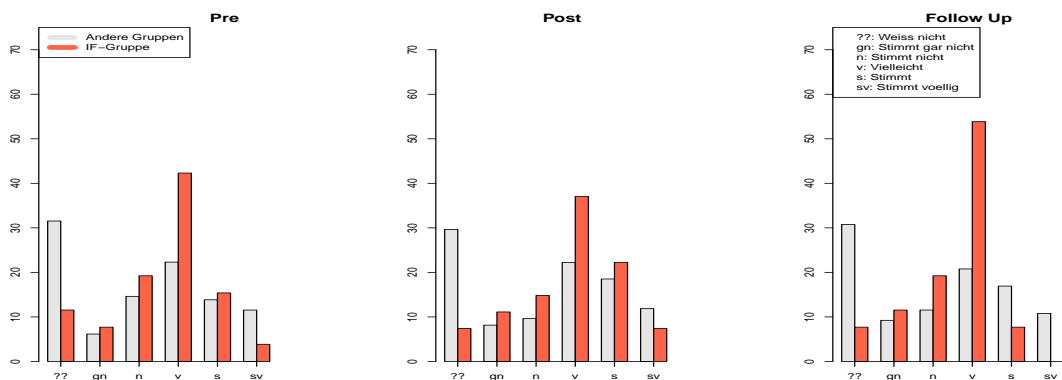


Abbildung 8.88: Programmieren ist sehr leicht.

gut programmieren zu können, sinkt zur Follow-Up – Untersuchung auf 24% ab. Es ist zu vermuten, dass ein kontinuierlicher Unterricht in Informatik die in diesen Fragen dargestellten Verunsicherungen aufheben kann.

### Basiskonntnisse im Programmieren

Aus der Bedeutung der Programmiersprache für die Informatik folgt, dass im Rahmen schulischer Bildung in Informatik Grundkenntnisse einer oder mehrerer Programmiersprachen unterrichtet werden. Abbildung [8.91] zeigt, dass diese Einschätzung schon zu Beginn der Untersuchung von den Informatikschülern mit einer Zustimmung von 39,8% geteilt wird, obwohl sie zu Beginn des Jahrganges 6 noch keine Programmiersprache kennen. Dementsprechend beantworten auch 35,7% der Schüler diese Frage mit ‚Weiss nicht‘. Im Posttest unterstützen die Aussage nach der Bedeutung von Basiskonntnissen 58% der Schülerinnen und Schüler und nur 20% enthalten sich bei dieser Frage. Im Follow-Up-Test sinkt diese Zustimmung

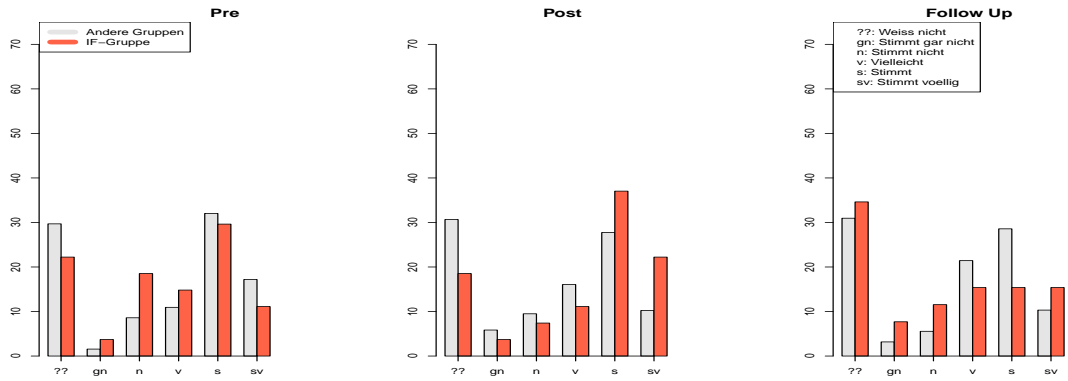


Abbildung 8.89: Ein Computerprogramm ist nicht einfach zu erstellen.

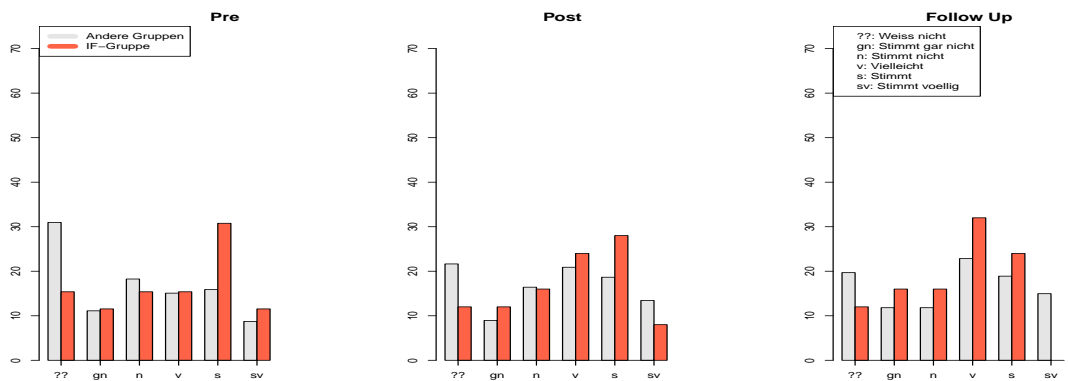


Abbildung 8.90: Ich kann sehr gut programmieren.

dann überraschend stark auf nur noch 34% ab und liegt damit sogar unter der Zustimmung der Kontrollgruppen mit 41,2%. Die Zustimmung bei diesen Gruppen lag zuvor bei 20,1% bzw. 26,8%. Allerdings war die Anzahl der Unentschlossenen in diesen Gruppen in allen Befragungen trotz tendenziellem Absinkens hoch und lag auch zum Schluss noch bei 37,7%.

Der zeitliche Abstand mit der Beschäftigung mit einer Programmiersprache im Unterricht lässt die Bedeutung von Programmiersprachen verblassen. Wenn die Bedeutung eines Inhaltes aus Sicht der Schülerinnen und Schüler sinkt, dann ist zu befürchten, dass auch das Interesse einer Berufstätigkeit in diesem Berufsfeld nachlässt. Dieses kann nicht im Sinne der Gesellschaft und der objektiven Interessen der Schüler sein. Da die Beschäftigung im Informatik-Unterricht mit Programmiersprache eine deutliche Steigerung der Zustimmung beim Posttest bei dieser Frage ergeben hat, ist zu vermuten, dass dieser Zustimmungsverlust durch einen fortlaufenden Informatikunterricht vermieden werden kann.

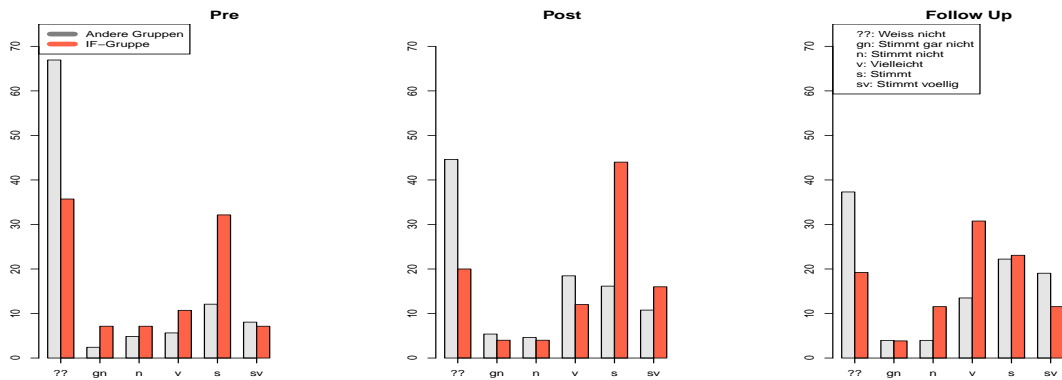


Abbildung 8.91: Es ist wichtig, Basiskenntnisse im Programmieren zu haben.

## Anwendungsprogramme

Das Zutrauen zu den eigenen Programmierfähigkeiten hat bei den Informatikschülerinnen und -schülern gelitten. Dagegen ist ihr Vertrauen in ihre Fähigkeiten bei den klassischen Anwendungsprogrammen Textverarbeitung und Tabellenkalkulation entsprechend den Abbildungen [8.92] und [8.93] überraschenderweise gestiegen.

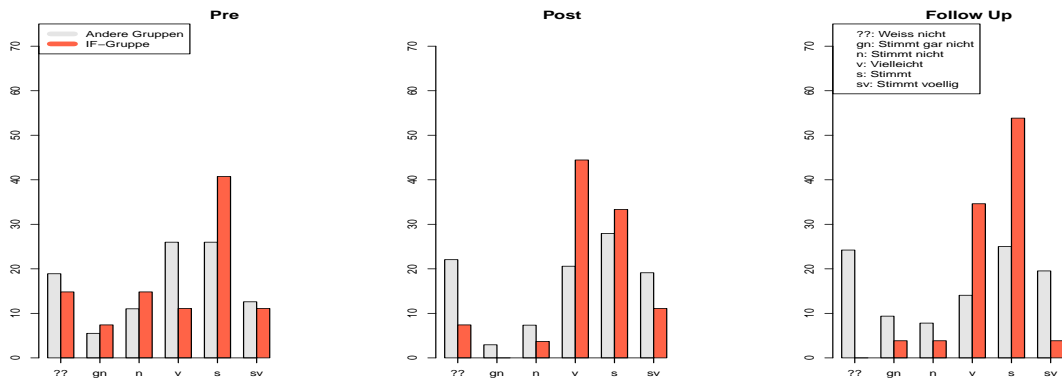


Abbildung 8.92: Ich kenne mich sehr gut mit Textverarbeitung aus.

Vor allem der Anstieg beim Zutrauen in die Arbeit mit Tabellenkalkulationsprogrammen von ursprünglich 32% über 60% auf 84% in der Abschlussbefragung ist auffällig. Dies ist umso mehr auffällig, weil Textverarbeitung und Tabellenkalkulation im Informatikkurs nicht behandelt worden sind. Die Schüler in den Vergleichskursen haben nur in der Arbeit mit Tabellenkalkulationsprogrammen einen nennenswerten Vertrauenszuwachs von 28% auf endgültig 40% erfahren. Dabei sind in der Hauptschule Vorhalle und in der Realschule Halden genau diese beiden Programmklassen nicht unwesentlich behandelt worden.

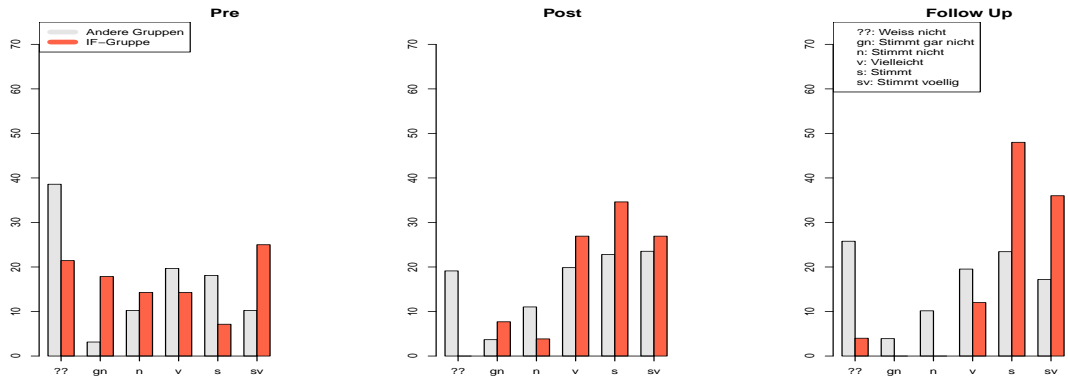


Abbildung 8.93: Ich kenne mich sehr gut mit Tabellenkalkulation aus.

Generelles Ziel von Unterricht ist, dass Schülerinnen und Schüler lernen, sich etwas zuzutrauen. Natürlich sollen sie ihr Können auch realistisch einschätzen können. Selbstvertrauen in die eigene Leistung ist eine wesentliche Voraussetzung, um in der komplexen Welt seinen eigenen Weg zu finden. Nach den Bildungsstandards Informatik ist die bewusste Nutzung von Informatiksystemen ein inhaltsbezogenes Ziel [Arb08, S.13]. Die Schülerinnen und Schüler der Informatikgruppe haben nach ihrer Selbsteinschätzung ein Vertrauen, die ihnen bekannten Systeme Textverarbeitung und Tabellenkalkulation sachgerecht einsetzen zu können. Eine spezielle Unterweisung ist nach dieser Einschätzung nicht nötig, sofern diese Selbsteinschätzung auch mit der angenommenen Nutzungskompetenz einhergeht.

## Arbeits- und Freizeitwelt

Die oben dargestellte Veränderung der Einschätzung der Bedeutung der Basiskenntnisse einer Programmiersprache schlägt sich auch in den Ansprüchen für Freizeit und Arbeitswelt nieder.

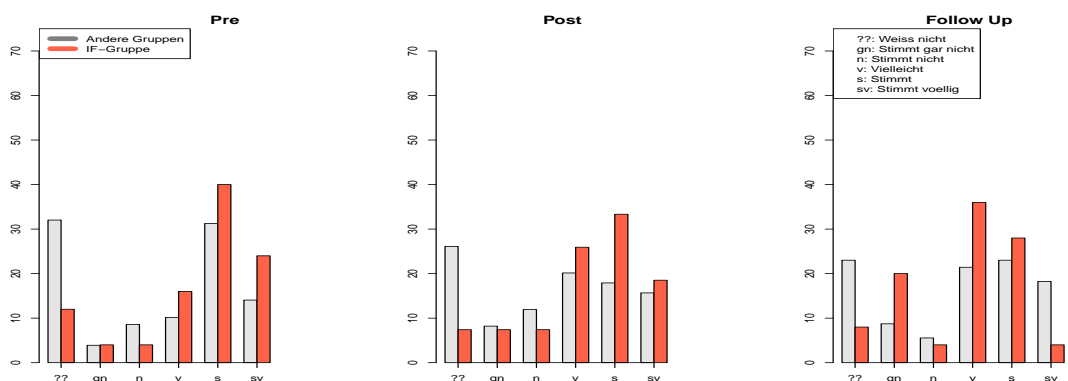


Abbildung 8.94: Die Arbeit am Computer beim Programmieren macht Spass.



Der ‚Spasfaktor‘ liegt bei der Informatik-Schülergruppe beim Pre-Test laut Abbildung [8.94] bei 64% und sinkt zum Post-Test auf 51,8% ab und bricht dann zum Follow-Up-Test auf die Hälfte des Ursprungwertes 32% ein. Bei den Kontrollgruppen sehen wir nur ein Absinken von 45% auf schliesslich 41%.

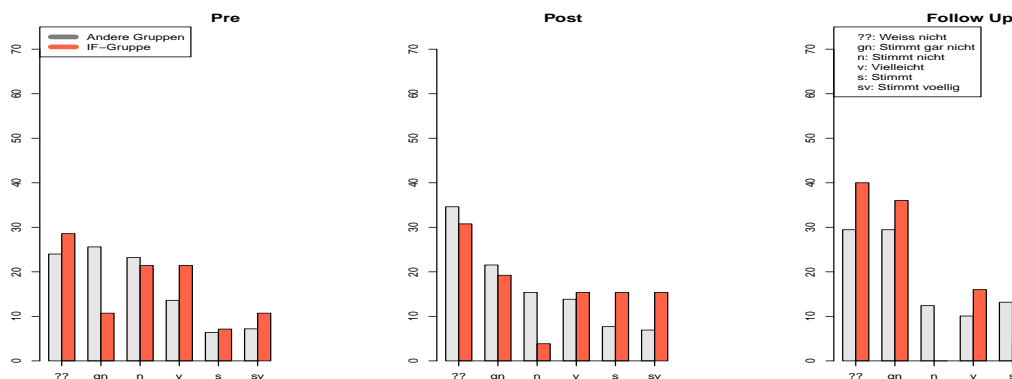


Abbildung 8.95: Ich möchte später irgend etwas mit Informatik oder Computern machen.

Es verwundert nicht, dass darauf aufbauend die Berufswünsche für die Zukunft nur noch eingeschränkt im Umfeld der Informatik gesehen werden. Das Arbeitsfeld ‚Informatik‘ hatte im Pretest eine Präferenz von 17,8% und steigerte sich zum Posttest auf 30,8% und bricht dann auf 8% ein. Bei der Kontrollgruppe liegen diese Werte bei 13,6%, 14,5% und dann 18,6%.

Natürlich müssen die Werte der Kontrollgruppe mit grosser Vorsicht betrachtet werden, da diese u.a. relativ geringe Vorstellungen über das Programmieren besitzen. Bedenklich dagegen sind die Aussagen der Schülerinnen und Schüler aus der Informatikgruppe. Zweifelsohne wäre es hier interessant gewesen, wenn eine Vergleichsgruppe auch einen fortlaufenden Informatikunterricht gehabt hätte.

### 8.10.2 Zusammenfassung der Aussagen zum Follow-Up – Test

Begrifflichkeiten und Faktenwissen sind im ersten Unterrichtsjahr bis hin zum Follow-Up-Test von den Schülerinnen und Schülern erarbeitet worden. Bei den abstrakteren Themen wie der Einführung des Variablenbegriffes sind noch erhebliche Lücken im Wissen erkennbar, die auch mit der zeitlichen Differenz wieder grösser werden. Deutlich gerade im Vergleich zu den Kontrollgruppen wird, dass Begriffe und Fakten aus dem informatischen Umfeld nicht durch das Alltagswissen hinreichend erworben werden.

Die Bedeutung von Programmierung und Programmiersprachen wird von den Schülerinnen und Schülern der Informatikgruppe in allen Befragungen als wichtig angesehen, allerdings werden die Schwierigkeiten zur Erstellung eines Computerprogrammes mit der zeitlichen Differenz zum eigenen Unterricht in Informatik unsicherer eingeschätzt. Mit der zeitlichen Differenz leidet auch das eigene Selbstbewusstsein in die eigenen Programmierfähigkeiten.

Vergleichbar verhält es sich mit der Zustimmung zur Bedeutung von Basiskenntnissen von Programmiersprachen sowie der Spass bei der Arbeit am Computer. Parallel bricht der Wunsch nach einem zukünftigen Beruf im Umfeld der Informatik sehr stark ein. Das Selbstvertrauen in die Nutzung von Anwendungsprogrammen wird davon allerdings nicht tangiert.

Die beschriebenen negativen Effekte dürften ihre Ursache darin haben, dass im halben Jahr vor der Follow-Up – Untersuchung kein Informatikunterricht stattgefunden hat. Daraus kann gefolgert werden, dass ein Informatikunterricht fortlaufend erteilt werden muss, sofern ein Interesse an einer Beseitigung dieser negativen Tendenzen besteht.

## 8.11 Auswertung der *Think-Aloud*-Interviews

Am Ende des Schuljahres 2010/2011 wurde ein Teil der Schülerinnen und Schüler in kurzen Interviews zu zentralen Inhalten des vergangenen Jahres im Unterricht befragt. Da dies parallel zum Unterricht in den drei Schulen passieren musste, standen nicht überall eine gleiche Anzahl von Schülern für diese Interviews zur Verfügung. Die meisten Interviews wurden mit 19 in der Informatik-Schülergruppe und 13 in der Kontrollgruppe ohne jeglichen Informatikunterricht durchgeführt. An der Hauptschule Vorhalle wurden in den insgesamt drei Gruppen 16 Interviews und an der Realschule Halden nur 6 Interviews eingeholt. Auch die Follow-Up-Untersuchung wurden parallel mit Think-Aloud - Interviews begleitet.

Mit diesen Interviews von ca. 5 Minuten Dauer sollte überprüft werden, ob die Selbsteinschätzungen seitens der Schülerinnen und Schüler bezüglich ihres Wissens korrekt sind. Sie stellen damit ein Indiz dar, wie weit die Aussagen aus den Fragebögen den tatsächlichen Unterrichtsverlauf und -erfolg widerspiegeln. Es zeigte sich bei der Auswertung, dass die Schülerinnen und Schüler ihr Wissen höher einschätzen als es in den Interviews präsentiert wurde. Dieses ist sicher so zu erwarten gewesen. Damit die Aussagen aus den Think-Aloud-Interviews mit der Selbsteinschätzung verglichen werden können, wurden die Ergebnisse für *teilweise bzw. in Ansätzen verstanden* auf die beiden Gruppen *verstanden* und *nicht verstanden* verteilt, da in der Selbsteinschätzung zu einer inhaltlichen Aussage neben *ich weiss nicht* die Schüler nur *stimmt* oder *stimmt nicht* angeben konnten. Dies wird dadurch gerechtfertigt, weil in längeren Interviews mancher Schüler doch mehr Verständnis und manch anderer mehr Unverständnis gezeigt hätte.

### 8.11.1 Interviews nach dem Posttest

#### Der Variablenbegriff

Es zeigt sich in den Interviews, dass der Variablenbegriff nach der ersten Einführung zum Ende des Schuljahres nur bei einem Teil der Schüler der Informatikgruppe verstanden wurde. Ein Vorverständnis dieses Begriffes ohne Unterricht in den anderen Gruppen ist nicht vorhanden. Dies Ergebnis bestätigt, dass ohne Unterricht das Konzept der Variablen nicht zu verstehen ist. In der Tabelle [8.3] sind die Ergebnisse der Kontrollgruppen nach Schulen getrennt gegenüber der Informatikgruppe aufgelistet, Tabelle [8.4] sind die Ergebnisse aller Kontrollgruppen

Gruppe	verstanden	nicht verstanden	nicht gefragt etc.
HSV SE	34,5	6,9	58,6
HSV TA	0,0	81,3	18,8
RSH SE	38,8	10,2	51,0
RSH TA	0,0	83,3	16,7
6.2 SE	18,5	0,0	81,5
6.2 TA	0,0	100,0	0,0
IF SE	53,8	7,7	38,5
IF TA	34,2	60,5	5,3

Tabelle 8.3: Ein Wert kann in einer Variablen gespeichert werden. (I)

Gruppe	verstanden	nicht verstanden	nicht gefragt etc.
KG SE	32,8	6,7	60,4
KG TA	0,0	88,6	11,4
IF SE	53,8	7,7	38,5
IF TA	34,2	60,5	5,3

Tabelle 8.4: Ein Wert kann in einer Variablen gespeichert werden. (II)

zusammengefasst<sup>16</sup>. Immerhin hat praktisch ein Drittel der Informatik-Gruppe diesen Begriff ausreichend verstanden. Das sind überraschend viele Schüler. Beispielhaft antworteten die Schüler mit: „Eine Variable ist ein Behälter, wo zum Beispiel Bilder drin sind.“ oder mit „Eine Variable ist etwas Unfestes, kann man immer wieder aufmachen, vom externen Speicher auf den Arbeitsspeicher kopieren; wie ein Etui; Bleistifte reintun, raustun, kopieren.“

Bei der Selbsteinschätzung waren es knapp über 50%, die diesen Begriff inhaltlich korrekt beantworteten. Es ist nicht ganz überraschend, dass mehr Schüler im Interviews zugeben, diesen Begriff nicht zu kennen als dies in der Selbsteinschätzung deutlich wird. Dadurch sind die Werte für die Zustimmung und vor allem für die Ablehnung der inhaltlichen Aussage überhöht.

Es interessiert hier an dieser Stelle in erster Linie die Stimmigkeit der Aussagen der Schüler in ihrer Selbsteinschätzung. Daher werden in den folgenden ausgewählten Ergebnissen die Kontrollgruppen zusammengefasst dargestellt.

## Grafikformate

Die Grafikformate wurden in verschiedenen Unterrichtssequenzen benötigt und verwendet. Im Vergleich zum Variablenbegriff ist zumindest der Aufbau einer Pixelgrafik relativ einfach zu verstehen. Viele Schülerinnen und Schüler kennen teilweise die Begriffe, aber fragt man genauer nach, so haben sie doch wenig Informationen über die tatsächlichen Hintergründe. Nach Tabelle [8.5] glauben fast 60% der Schüler der Kontrollgruppen, dass sie wüssten, was eine Pixelgrafik ist, tatsächlich sind es aber nur knapp über 10%. Fast alle Schüler der Infor-

<sup>16</sup> ‚SE‘ steht für Selbsteinschätzung, ‚TA‘ für Think Aloud

Gruppe	verstanden	nicht verstanden	nicht gefragt etc.	Frage nach
KG SE	58,1	20,6	21,3	Pixelgrafik
KG TA	11,4	22,9	65,7	
IF SE	92,6	7,4	0,0	
IF TA	50,0	28,9	21,1	
KG SE	23,9	41,8	34,3	Vektorgrafik
KG TA	0,0	37,1	62,9	
IF SE	77,8	11,1	11,1	
IF TA	13,2	81,6	5,3	

Tabelle 8.5: Ich weiss, was eine Pixel- bzw. eine Vektorgrafik ist.

matikgruppe sind der Meinung, sie kennen die auch häufig im Unterricht besprochenen Pixelgrafiken, tatsächlich sind es aber nur genau die Hälfte der Schüler. Dagegen hat auch kein Schüler der Informatikgruppe den Begriff der Vektorgrafik verstanden, sie kennen ihn allerhöchstens. Dies ist nachvollziehbar, da die Struktur der Vektorgrafik schwerer zu beschreiben ist. Allerdings haben die Schüler mit Hilfe von *Tcl/Tk* Vektorgrafiken erstellt. Dies ist ihnen offensichtlich nicht bewusst geworden. Daher muss im zukünftigen Unterricht das Bewusstsein dafür geschärft werden. Für das Verständnis kommt erschwerend hinzu, dass die Schüler den Begriff ‚Vektor‘ unabhängig von der Vektorgrafik nicht kennen. Es wäre überlegenswert, aus didaktischen Gründen noch eine zweite leichter eingängige Vokabel zu benutzen. Ein Viertel der Schülerinnen und Schüler der Kontrollgruppen glaubt diesen Begriff zu kennen, es hat ihn aber keiner inhaltlich verstanden.

Folgende Beispiele einzelner Aussagen zeigen das Verständnis bzw. das Nicht-Verständnis einzelner Schüler: „Ich meine, das wäre, wie hell der Computer sein muss, also der Bildschirm“ sagt beispielsweise ein Realschüler aus Halden zur Frage, was eine Grafik ist. Ein Schüler der Informatikgruppe hat den Begriff nur sehr wenig verstanden, indem er antwortet: „Eine Grafik ist, wenn eine Datei nicht vergrößert werden kann, weil es zu viele Pixel gibt.“ Ein anderer antwortet dagegen bei der Frage nach Pixelgrafiken: „Pixel sind so Vierecke; ein Bild besteht aus Pixeln, aber aus sehr vielen kleinen Vierecken und das ist dann eben eine Pixelgrafik, man kann damit Bilder erzeugen.“

Die Selbsteinschätzung des Verständnisses der beiden Gruppen ist für beide Formate erheblich höher als das tatsächliche Verständnis. Die Differenz bezüglich des Verständnisses ist allerdings sowohl bei der Selbsteinschätzung als auch bei dem tatsächlichen Verständnis mit ca. 40% bis 50% praktisch gleich. Der Unterschied zwischen den beiden Gruppen macht deutlich, dass ohne Unterricht die Eigenschaften derartiger Grafiken aus den Erfahrungen im Alltagsleben kaum zu erfassen sind.

## Betriebssystem und Anwendungssoftware

Auch bei den beiden Softwareklassen *Betriebssysteme* und *Anwendungssoftware* ergibt sich ein ähnliches Bild. Die Selbsteinschätzung liegt in beiden Gruppen höher. Das tatsächliche Verständnis ist niedriger. Bei der Informatikgruppe sinkt der Wert bezüglich der Kenntnis von

Gruppe	verstanden	nicht verstanden	nicht gefragt etc.	Frage nach
KG SE	68,1	17,0	14,8	Betriebssystem
KG TA	28,6	45,7	38,6	
IF SE	85,2	7,4	7,4	
IF TA	60,5	28,9	10,4	
KG SE	51,5	30,6	17,9	Anwendersoftware
KG TA	17,1	45,7	37,1	
IF SE	60,0	20,0	20,0	
IF TA	42,1	31,6	26,3	

Tabelle 8.6: Ich weiss, was ein Betriebssystem bzw. Anwendersoftware ist.

Gruppe	verstanden	nicht verstanden	nicht gefragt etc.	Frage nach
KG SE	77,5	10,1	12,3	Verzeichnis
KG TA	17,1	65,7	17,1	
IF SE	88,9	7,4	3,7	
IF TA	52,6	36,8	5,3	
KG SE	45,6	23,5	30,9	Verzeichnisbaum
KG TA	11,4	34,3	54,3	
IF SE	85,2	11,1	3,7	
IF TA	34,2	23,7	31,6	

Tabelle 8.7: Ich weiss, was ein Verzeichnis/Ordner bzw. ein Verzeichnisbaum ist.

Betriebssystemen allerdings nur von 85% auf 60%, während der Wert bei der Kontrollgruppe mehr als halbiert wird. Ähnlich sehen die Verhältnisse bei der Anwendersoftware aus. Etwa die Hälfte der Schülerinnen und Schüler in der Informatikgruppe verfügen noch über kein ausgeprägtes Verständnis dieser Softwarearten.

Missverständnis und Verständnis seitens der Schülerinnen und Schüler veranschaulichen folgende beispielhafte Aussagen: „Es gibt viele Betriebssysteme, zum Beispiel Google und Wikipedia“ antwortet ein Realschüler aus Halden, ein Vorhaller Hauptschüler meint, „ein System, was den Computer betreibt“. Ein Schüler der Informatikgruppe vermutet, dass „ein Betriebssystem ist zum Beispiel ein Arbeitsspeicher.“ Ein anderer meint: „Der Computer hat ein Betriebssystem, dass das am Laufen erhält.“ Ein anderer formuliert sehr passend: „Wenn der PC kein Betriebssystem hätte, würde er nicht funktionieren; dann wüsste er keine Befehle.“ Ein Schüler der Kontrollklasse 6.2 meint dagegen: „Facebook oder SchülerVZ, also mehrere Seiten; auch diese Schreibseiten.“ Zur Frage nach der Anwendungssoftware antwortet ein Schüler der Informatikgruppe: „Das bedeutet dann, dass der PC dir sagt: ‚Jetzt kannst Du was auf meinem Programm Linux jetzt was schreiben, indem Du ein anderes Programm auf mir aufrufst.‘“

## Verzeichnisse und Verzeichnisstrukturen

Noch deutlicher fallen die Ergebnisse entsprechend Tabelle [8.7] bezüglich der Begriffe im Umfeld von *Verzeichnisstrukturen* aus. Nach der Selbsteinschätzung der Schülerinnen und Schüler wissen mehr als drei Viertel aller Schüler, was sich hinter den Begriffen *Verzeichnis* bzw. *Ordner* verbirgt. Tatsächlich haben ein wirkliches Verständnis davon immerhin etwas mehr als die Hälfte der Informatikschüler, aber weniger als ein Fünftel der anderen Schüler. Höhere Begrifflichkeiten sind bei diesen Schülern nur mit knapp 10% verankert, während bei den Informatikschülerinnen und -Schüler immerhin ein Drittel diese Begrifflichkeiten verstanden hat.

Zum Verhältnis von Datei zu Verzeichnis meint ein Vorhaller Hauptschüler: „... Datei ist ja halt irgendetwas, das wir machen, sozusagen. Sagen wir jetzt zum Beispiel, wird etwas gemacht: Das ist ja dann halt die Datei und die Datei speichern wir halt dem Ordner.“ Ein Informatikschüler der FSG meint: „Ein Verzeichnis ist ein Ordner, wo die Dateien abgespeichert und verwaltet werden können.“ Ein Schüler aus der Kontrollklasse 6.2 sagt: „Anhand des Beispiels ist Informatik der Ordner und die Unterordner sind Dateien.“ Zu Verzeichnisstrukturen formuliert ein Schüler aus der WP-Informatikgruppe: „Ein Ordnerbaum ist eine grafische Darstellung, was in Ordnern oder anderen Dateien drin ist.“

Es bleibt festzustellen, dass selbst bei diesen grundsätzlichen Strukturen offensichtlich ein umfassendes Verständnis ohne Informatikunterricht nicht im ausreichenden Umfang zu erwarten ist.

## Zusammenfassung der Auswertung der Interviews nach dem Posttest

Generell lässt sich aussagen, dass die Ergebnisse der Interviews die Selbsteinschätzung der Schülerinnen und Schüler bestätigen. Natürlich überschätzen sie ihre Fähigkeiten teilweise nicht unerheblich und sie geben auch eher eine falsche Antwort, bevor sie eine Frage nicht oder mit ‚Weiss nicht‘ beantworten. Der Trend in den Interviews ist allerdings derselbe wie in den Selbsteinschätzungen. Dabei liegen die Ergebnisse der Schüleraussagen der Informatikgruppe durchweg relativ wie absolut noch näher an der Selbsteinschätzung gegenüber den Schülern der Nicht-Informatikgruppen. Erhebliche Wissensunterschiede sind demnach in allen untersuchten Bereichen anzutreffen, nicht nur bei den Themen, die der Sache nach tatsächlich in einem Informatikunterricht unterrichtet werden. Dabei erstaunt, dass selbst Begriffe aus der eigentlichen heutigen kindlichen Alltagswelt wie beispielsweise *Verzeichnis* bzw. *Ordner* von dem überwiegenden Teil der Nicht-Informatik-Schüler nicht vollständig erfasst sind.

### 8.11.2 Interviews nach dem Follow-Up-Text

Im Februar 2012 wurde ein Teil der Schülerinnen und Schüler erneut zu zentralen Begriffen aus dem Unterricht des vorhergehenden Schuljahres befragt. Wir wollten wissen, ob und welche Kenntnisse aus dem Unterricht in den verschiedenen Gruppen noch vorhanden waren. Ebenso war von Interesse, ob eventuell durch den persönlichen Entwicklungsprozess der einzelnen Schülerinnen und Schüler aufgrund ihres Umfeldes eine Veränderung des Wissensstandes eingetreten ist. In der 2. Befragung wurden in der Informatikgruppe 18 Schülerinnen

Fragen	IF		FSG 6.2		HSV/RSH	
	6/11	2/12	6/11	2/12	6/11	2/12
Was ist eine Datei	67	50	38	55	39	53
Was ist ein Verzeichnis	47	50	8	29	13	27
Was ist ein Ordner	75	75	70	75	53	71
Was ist eine Verzeichnisstruktur	55	50	0	10	10	20
Was ist ein Programm	61	58	38	40	52	40
Was ist eine Variable	40	18	0	18	0	0
Was ist eine Anweisung	82	58	31	9	17	46
Was ist eine Betriebssystem	53	50	0	10	24	29
Was ist ein Pixel	67	75	33	50	—	38
Was ist eine Pixelgrafik	47	50	22	0	0	12
Was ist eine Vektorgrafik	39	0	0	0	0	0
Wie zentriert man ‚automatisch‘	75	80	18	80	33	67
Wie zentriert man ‚manuell‘	75	80	100	100	64	82

Tabelle 8.8: Positiv bewertete Aussagen (in Prozent) bei den Think-Aloud – Befragungen Ende des Schuljahres 2010/2011 sowie ein halbes Jahr später.

und Schüler befragt, in der Kontrollgruppe der Klasse 7.2 elf Schülerinnen und Schüler, an der Hauptschule Vorhalle 13 Schülerinnen und Schüler sowie an der Realschule Halden vier Schüler. Da die Anzahl der Realschüler sehr klein war, wurden zur Auswertung die externen Gruppen zusammengefasst. In der Tabelle 8.8 sind die korrekten Aussagen in Prozent der drei verschiedenen Schülergruppen dargestellt.. Neben diesen korrekten Aussagen gab es teilweise korrekte Aussagen, die hier aber nicht gezählt wurden. Nicht alle Fragen wurden allen Schülern gestellt.

Es zeigt sich, dass im Teilbereich *Dateien und Dateisysteme* alle Schüler einen Lernzuwachs erfahren haben, der auch über das Halbjahr erhalten bzw. sogar gesteigert wurde. Dies zeigt, dass in diesem Alter die Beschäftigung mit informationstechnischen Systemen relativ unabhängig vom Bildungsprozess einen Wissenszuwachs generiert. Die Informatikschüler zeigen eine etwas mehr generalisierende Kenntnis über Dateisysteme.

Im Teilbereich *Grafik* zeigt sich, dass die Kenntnisse der Schüler ohne entsprechenden Unterricht sehr gering sind. Aber auch die Schüler der Informatikgruppe vergessen die Kenntnisse über den relativ schwierigen Zusammenhang bei *Vektorgrafiken*.

Im Teilbereich *Begriffe* zeigen die Informatikschüler auch nach einem halben Jahr, dass sie mehr wissen als die Schüler in den Kontrollgruppen, aber auch sie haben einiges wieder verlernt. Dies gilt vor allem bei den schwierigen Begriffen wie beispielsweise beim Verständnis des Begriffes *Variable*.

Im Teilbereich *Texte* zeigen fast alle Schüler, dass sie grundlegende Eigenschaften derartiger Systeme kennen. Selbst die Schüler der internen Kontrollgruppe ohne jeglichen Unterricht in Informatik oder ITG zeigen einen teilweise erheblichen Zuwachs. Das zeigt den zweifelhaften allgemeinbildenden Charakter derartiger Ziele.

Zusammenfassend lässt sich formulieren, dass die Schülerinnen und Schüler im entspre-

Fragen	6/11	2/12	7/12
Was ist eine Datei	67	50	75
Was ist ein Verzeichnis	47	50	43
Was ist ein Ordner	75	75	72
Was ist eine Variable	40	18	35
Was ist eine Anweisung	82	58	73
Was ist ein Pixel	67	75	41
Was ist eine Pixelgrafik	47	50	63
Was ist eine Vektorgrafik	39	0	0
Wie zentriert man ‚automatisch‘	75	80	58
Wie zentriert man ‚manuell‘	75	80	73

Tabelle 8.9: Positiv bewertete Aussagen (in Prozent) bei den Think-Aloud – Befragungen Ende des Schuljahres 2010/2011, nach dem 1. Halbjahr und am Ende des Schuljahres 2011/2012 der befragten Informatikschüler.

chenden Unterricht einen Wissenszuwachs generiert haben, der auch nach einem halben Jahr ohne Unterricht in Informatik messbar ist. Es zeigt sich aber auch, dass die schwierigeren und abstrakteren Zusammenhänge wieder etwas in Vergessenheit geraten. Dies verdeutlicht die Notwendigkeit eines frühzeitigen und permanenten Unterrichtes in Informatik, da sich diese Inhalte nicht durch ein einmaliges Lernen in Informatik dauerhaft erwerben lassen. Entsprechend der Idee der *Curriculumspirale* nach Bruner müssen diese Gegenstände erneut aufgegriffen und vermittelt werden.

### 8.11.3 Interviews am Ende des Schuljahres 2011/2012

Die Wahlpflichtgruppe Informatik/Physik hat aufgrund der Umbausituation in der Fritz-Steinhoff-Schule im 6. Schuljahr ausschliesslich Informatik-Unterricht erhalten. Im Ausgleich dazu wurde im 7. Schuljahr weitgehend Unterricht in Physik erteilt. Erst am Ende des 2. Halbjahres in den letzten 6 Wochen wurde der Informatik-Unterricht wieder aufgenommen. Dazu wurden die zentralen Begriffe anhand von erweiterten Übungen wiederholt und ausgebaut. Ziel anschliessend war in erster Linie der weitere Aufbau eines *Variablenbegriffes*. Dieser Anspruch konnte nicht ganz realisiert werden, da der Unterrichtende aufgrund eines Sportunfalles die letzten 2,5 Wochen nicht unterrichten konnte. Trotzdem wurden 22 Schülerinnen und Schüler in der letzten Schulwoche noch ein weiteres Mal in kurzen Interviews nach zentralen Begriffen gefragt. Dadurch konnten wir feststellen, ob nach einer relativ kurzen Zeit des Wiedereinstiegs in den Unterricht in Informatik teilweise vergessenes Wissen reorganisiert werden konnte.

Tabelle 8.9 zeigt, dass eine derartige Reorganisation tatsächlich eingetreten ist. So steigt die Zahl der korrekten Antworten zum Begriff *Variable* von ursprünglich 40% nach dem Tief von 18% wieder auf 35% an. Auch die Anzahl der korrekten Antworten zum Begriff *Anweisungen* konnte wieder fast den Wert von vor einem Jahr erreichen. Verständlicherweise hat sich beispielsweise der Wert bezüglich der *Vektorgrafiken* nicht verändert, da diese in diesem Schuljahr überhaupt nicht angesprochen wurden.



Ja/Nein/Weiss nicht - Fragen	IF		FSG 6.2		RSH		HSV	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post
Ich kenne mehrere Betriebssysteme.	57	81	27	70	37	74	25	53
Ich kenne das JPG-Format.	43	70	9	19	10	41	11	28
Ich weiss, was Anwendersoftware ist.	36	60	30	56	33	66	26	37
Ein Wert kann in einer Variablen gespeichert werden.	25	54	18	19	27	39	35	35
Ein GIF-Bild ist eine Pixelgrafik.	30	74	14	37	25	45	38	40
Ich weiss, was eine Vektorgrafik ist.	25	78	13	31	12	28	17	17
Ich weiss, was eine Variable ist.	18	81	4	15	12	34	11	20
Ich weiss, was eine Pixelgrafik ist.	61	93	39	56	40	73	42	48
Ich weiss, was ein Verzeichnisbaum ist.	36	85	27	48	33	63	13	30

Tabelle 8.10: Positive bewertete Aussagen (in Prozent) bezüglich der Selbsteinschätzung zwischen Pre- und Posttest.

## 8.12 Quantitative Untersuchungen

### 8.12.1 Fragebogen

Auch wenn die qualitativen Ergebnisse kaum einen Zweifel lassen, dass die Schülerinnen und Schüler der Informatikgruppe gegenüber den Schülern in den verschiedenen Kontrollgruppen einen unterschiedlichen Lernprozess erfahren haben, soll hier an ausgesuchten Fragestellungen gezeigt werden, ob und welcher signifikante Unterschied zwischen Pre- und Posttest nachweisbar ist [PV12].

#### Auswertung des Fragebogens ‚Wissen‘

Zuerst betrachten wir die Ergebnisse innerhalb der einzelnen Gruppen. In der Tabelle 8.10 sind die zustimmenden bzw. korrekten Aussagen der Schülerinnen und Schüler im Pre- und Posttest in Prozent wiedergegeben. Die zugehörigen Signifikanzwerte nach dem  $\chi^2$ -Test sind in Tabelle 8.11 aufgelistet. Aufgeführt sind bis eine Ausnahme die Items, in denen sich ein signifikanter Unterschied bei der Informatik-Gruppe ergeben hat<sup>17</sup>.

Es ist ablesbar, dass die qualitativen Veränderungen in der Informatikgruppe durch diese quantitativen Zahlen bestätigt werden. Die schulinterne Kontrollgruppe zeigt im Vergleich dazu bis auf eine Ausnahme keine signifikante Veränderung. Bei den externen Kontrollgruppen zeigen sich bei der Realschule Halden signifikante Veränderungen, während bei den Schülerinnen und Schülern der Hauptschule Vorhalle kaum Abweichungen signifikanter Art nachweisbar sind. Unterschiede im Curriculum (siehe Kapitel 8.3.3 und 8.3.2) zwischen den externen

<sup>17</sup>Zu berücksichtigen ist, dass sich die Signifikanzwerte aufgrund der gesamten Stichprobe und nicht nur alleine aus der Zustimmung bzw. Korrektheit der Aussage ergeben. Damit erklärt sich der niedrige p-Wert bei der Hauptschule Vorhalle bei praktisch gleich gebliebenen zustimmenden Werten zur Aussage ‚Ein Wert kann in einer Variable gespeichert werden.‘

Ja/Nein/Weiss nicht - Fragen	If	6.2	RSH	HSV
Ich kenne mehrere Betriebssysteme.	0.005	0,008	0,0009	0,002
Ich kenne das JPG-Format.	0.10	0,35	0,001	0,08
Ich weiss, was Anwendersoftware ist.	0,004	0,12	0,004	0,48
Ein Wert kann in einer Variablen gespeichert werden.	0,08	0,28	0,14	0,05
Ein GIF-Bild ist eine Pixelgrafik.	0.004	0,20	0,11	0,65
Ich weiss, was eine Vektorgrafik ist.	0.0005	0,29	0,09	0,82
Ich weiss, was eine Variable ist.	$1.1 * 10^{-5}$	0,43	0,02	0,36
Ich weiss, was eine Pixelgrafik ist.	0,01	0,38	0,005	0,54
Ich weiss, was ein Verzeichnisbaum ist.	0,0009	0,11	0,009	0,03

Tabelle 8.11: Signifikanz bezüglich der Selbsteinschätzung zwischen Pre- und Posttest ( $p$ -Werte entsprechend dem  $\chi^2$ -Test) innerhalb der Gruppen.

Ja/Nein/Weiss nicht - Fragen	Pretest		Posttest	
	IF-KG <sub>int</sub>	IF-KG <sub>ext</sub>	IF-KG <sub>int</sub>	IF-KG <sub>ext</sub>
Ich weiss, was eine Variable ist.	0.33	0.54	$0.99 * 10^{-6}$	$0.6 * 10^{-7}$
Ein Wert kann in einer Variablen gespeichert werden.	0.84	0.47	0.005	0.28
Ich weiss, was eine Pixelgrafik ist.	0.26	0.11	0.01	0.003
Ich weiss, was eine Vektorgrafik ist.	0.29	0.40	0.001	$0.25 * 10^{-7}$
Ich weiss, was ein Verzeichnis bzw. ein Ordner ist.	0.21	0.21	1	0.25
Ich weiss, was ein Verzeichnisbaum ist.	0.19	0.15	0.01	0.0007

Tabelle 8.12: Signifikanz bezüglich der Selbsteinschätzung der Informatik-Gruppe relativ zu den verschiedenen Kontrollgruppen ( $p$ -Werte entsprechend dem  $\chi^2$ -Test).

Schulen und die unterschiedliche Schülerschaft ermöglichen an der Realschule Halden einen Unterricht, der bezüglich der gestellten Fragen eine höhere Wirkung ergibt.

In Tabelle [8.12] sind die Signifikanz-Werte der Selbsteinschätzung aus dem Pretest den Werten aus dem Posttest zwischen der Informatikgruppe und den Kontrollgruppen gegenüber gestellt. Bei einigen Fragen haben sich die Ergebnisse vom Pre- zum Posttest erheblich verschoben. Bei vielen Fragen ergeben sich ausgehend von einem Signifikanz-Niveau von 5 % deutliche Unterschiede. Dies gilt nicht für den Begriff des *Verzeichnisses*, der sich vielen Schülern aus der üblichen Beschäftigung mit dem Computer in ihrer Alltagswelt nach ihrer Einschätzung erschliesst, Darauf aufbauende Strukturen wie *Verzeichnisbaum* sind nur den Informatik-unterrichteten Schülern geläufig, entsprechend erhalten wir hier einen signifikanten Unterschied.

Auch bei den grafischen Strukturen ist nur bei den Informatik-Schülern ein tiefgehendes Verständnis erkennbar. Allerdings wird deutlich, dass der Begriff Vektorgrafik bei allen Schülergruppen noch erheblich aufgearbeitet werden muss. Der Variablenbegriff ist praktisch ausschliesslich den Informatik-Schülern geläufig, wobei etwa die Hälfte der Schüler den Be-

Fragen	If	6.2	RSH	HSV
Ein Programmtext sollte sehr gut lesbar sein.	0,003	0,54	0,006	0,01
Programmierung ist in Informatik sehr wichtig.	0,05	0,36	0,009	0,48
Ich kenne mich sehr gut mit Tabellenkalkulation aus.	0,08	0,84	$10^{-5}$	0,57
Wenn ein Programm das tut, was es soll, ist es egal, wie es konstruiert wurde.	0,08	0,79	0,87	0,43
Es ist bei jedem Programm sehr wichtig, die aktuellste Softwareversion zu haben.	0,75	0,28	0,35	0,06
Ich kann sehr gut programmieren.	0,87	0,99	0,07	0,20
Ich kenne mich sehr gut mit Textverarbeitung aus.	0,02	0,39	0,58	0,19
In der Informatik sind Programmiersprachen sehr wichtig.	0,05	0,69	0,17	0,98

Tabelle 8.13: Signifikanz zu Aussagen über Informatik zwischen Pre- und Posttest ( $p$ -Werte entsprechend dem  $\chi^2$ -Test).

griff zwar kennen, ihn aber noch nicht in seiner Tragweite erfasst haben. Entsprechend gibt es beim Verständnis des Variablenbegriffes auch keinen signifikanten Unterschied zwischen den Gruppen.

### Auswertung des Fragebogens zu Aussagen über die Informatik

Viele Fragen in diesem Abschnitt des Fragebogens konnten mit den bisherigen Themen des konkreten Unterrichts nicht beantwortet werden. Sie werden in zukünftigen wiederholten Befragungen eine grössere Rolle spielen. Daher betrachten wir hier nur die Befragungsergebnisse, die bezüglich der Signifikanz vor allem bei der Informatikgruppe auffällig sind. Es gilt natürlich auch hier wie beim Fragebogen zu den Wissensthemen, dass die rechnerischen Ergebnisse zur Signifikanz sich auf die Verteilung in der gesamten Stichprobe beziehen und daher nur Aussagen über die gesamte Stichprobe und nicht unbedingt über einzelne interessante Ausprägungen ermöglichen. Um diese negativen Nebeneffekte zu minimieren, wurden die jeweils beiden unterstützenden und ablehnenden Antwortstufen zusammengefasst, sodass in den Stichproben nur noch die Stufen *Weiss nicht*, *Ja*, *Vielleicht*, *Nein* vorhanden sind. Die Signifikanzwerte nach dem dem  $\chi^2$ -Test zwischen den Ergebnissen der Stichproben zwischen Pre- und Posttest sind für die unterschiedlichen Gruppen in Tabelle 8.13 zusammengefasst.

Es werden wesentliche Aussagen aus den qualitativen Betrachtungen unterstützt. Wir erkennen, dass sich bei der Informatikgruppe zwischen Pre- und Posttest signifikante Unterschiede zeigen, während bei der internen Kontrollgruppe und der Hauptschule in Vorhalle keine wesentlichen Veränderungen registrierbar sind. Bei den Gruppen aus der Realschule in Halden sind ebenfalls teilweise parallel zu den Ergebnissen der Informatikgruppe Veränderungen signifikant. Auffällig ist, dass einerseits die Aussage *Ich kann gut programmieren* von den Informatikschülern aus der FSG in den beiden Tests nicht sehr unterschiedlich beantwortet wird, die Realschüler, die praktisch keine Vorkenntnisse auf diesem Gebiet erworben haben,

Fragen	If	6.2	RSH	HSV
Die Arbeit am Computer beim Programmieren ist schwierig und anstrengend.	0,04	0,59	0,73	0,34
Informatik kann man nur lernen, wenn man jede Stunde aufmerksam ist.	0,01	0,52	0,41	0,04
Ein Informatiker sollte ein guter Team-Worker sein.	0,56	0,10	0,004	0,19
Ich möchte mehr können als nur am Computer zu spielen.	0,24	0,98	0,02	0,23
Die Arbeit am Computer beim Programmieren macht Spass.	0,63	0,59	0,002	0,51

Tabelle 8.14: Signifikanz zu Einstellungen zur Informatik zwischen Pre- und Posttest ( $p$ -Werte entsprechend dem  $\chi^2$ -Test).

hier ein deutlich stärkeres Selbstbewusstsein entwickelt haben. In beiden Gruppen wird die Bedeutung der Programmierung im Posttest hervorgehoben. Bei dem Selbstvertrauen in die eigenen Fähigkeiten bezüglich der Textverarbeitung sehen wir bei den Informatik-Schülern einen signifikanten Unterschied, obwohl diese nun im Unterricht nicht mit einem Textverarbeitungssystem gearbeitet haben. Realschüler sowie die Informatikschüler der FSG entwickeln zudem ein gestärktes Bewusstsein bezüglich der Tabellenkalkulation, während die interne Kontrollgruppe und die Hauptschüler hier keine Veränderung deutlich werden lassen.

Zusammengefasst lässt sich sagen, dass der Unterricht der Informatik-Schüler in der FSG und der Realschüler in Halden in verschiedener Ausprägung teilweise bei unterschiedlichen Aussagen zu signifikanten Veränderungen geführt hat, während die interne Kontrollgruppe und die Hauptschüler keine wesentlichen Änderungen zeigen.

### **Auswertung des Fragebogens zu Einstellungen, Meinungen und Verhalten**

Im letzten Teil des Fragebogens wurden die Schülerinnen und Schüler zu Einstellungen, Meinungen und Verhalten befragt. Wie bei den Fragen zu Aussagen zur Informatik gilt auch hier entsprechend Tabelle 8.14, dass es signifikante Änderungen bis auf eine Ausnahme nur bei der Informatikgruppe in der FSG und den Gruppen in der Realschule Halden gab. Die Schülerinnen und Schüler der Informatikgruppe haben erkannt, dass das zielgerichtete Arbeiten am Computer nicht immer einfach ist. Ebenso empfinden sie nach einem Jahr deutlich mehr als zuvor, dass eine hohe Aufmerksamkeit im Unterricht notwendig ist. Dies empfinden auch die Schüler der Hauptschule so, nur die Schüler aus der Realschule Halden sehen dies anders. Dagegen formulieren diese vermehrt, dass der Computer nicht nur zum Spielen benutzt werden soll. Dieser Meinung waren die Informatik-Schüler der FSG schon zu Beginn der Untersuchung. Zudem erklären die Realschüler am Ende der Untersuchung, dass sie weniger Spass als zu Beginn der Untersuchung am Unterricht mit dem Computer haben.

Die Realschüler scheinen mit dem Unterricht in der bisherigen Form nicht ganz einverstanden zu sein. Da sie gegenüber den Hauptschülern deutlich mehr Lernzuwachs auch in den eher informatiknahen Fragen erworben haben, kann daraus abgeleitet werden, dass sie mit

Ja/Nein/Weiss nicht - Fragen	IF-KG
Ein Wert kann in einer Variablen gespeichert werden.	$8,7 * 10^{-10}$
Ich weiss, was eine Pixelgrafik ist.	$4,01 * 10^{-11}$
Ich weiss, was eine Vektorgrafik ist.	$8,9 * 10^{-18}$
Ich weiss, was ein Betriebssystem ist.	$2,1 * 10^{-7}$
Ich weiss, was Anwendersoftware ist.	0.001
Ich weiss, was ein Verzeichnis bzw. ein Ordner ist.	$9,5 * 10^{-8}$
Ich weiss, was ein Verzeichnisbaum ist	$8,2 * 10^{-5}$

Tabelle 8.15: Signifikanz bezüglich der Think-Aloud-Interviews der Informatik-Gruppe relativ zu der zusammengefassten Kontrollgruppe ( $p$ -Werte entsprechend dem  $\chi^2$ -Test).

dem Schwerpunkt des Unterrichtes auf dem Gebiet der ITG unzufrieden sind und gegebenenfalls mit einem Unterricht, wie ihn die Schülerinnen und Schüler der Informatikgruppe in der FSG erfahren haben, besser angesprochen werden können.

### 8.12.2 Think-Aloud

Wie wir weiter oben bei der qualitativen Auswertung bereits gesehen haben, besteht zahlenmässig zwischen der Selbstauskunft und den überprüfenden Think-Aloud-Interviews teilweise ein erheblicher Unterschied, auch wenn zumeist der Trend in den jeweiligen Gruppen Bestand hatte. Daher ist es sinnvoll, auch die Think-Aloud-Interviews quantitativ auszuwerten. Da Interviews für den Pretest nicht erhoben wurden, ist dies nur für die Posttests möglich. Wie bei den qualitativen Auswertungen, wurden für diese Auswertung alle Kontrollgruppen zusammengefasst. Diese Werte wurden ebenfalls mit dem  $\chi^2$ -Test auf Signifikanz getestet. Angegeben in der Tabelle [8.15] sind die  $p$ -Werte zwischen der unterrichteten Informatik-Gruppe gegenüber der zusammengefassten Kontrollgruppe.

Wir können erkennen, dass alle Versuchsreihen zwischen der Informatikgruppe und der Kontrollgruppe signifikant unterschiedlich sind. Die qualitative Aussage, dass die Schüler im Informatikunterricht tatsächlich mehr wissen als die Schüler aus den Kontrollgruppen, wird damit bestätigt. Selbst bei den Fragen, die nach der Selbsteinschätzung keinen signifikanten Unterschied zeigen, sind hier signifikant unterschiedlich.

### 8.12.3 Gender-Problematik

In der qualitativen Auswertung wurde bereits auf die Genderproblematik hingewiesen und die unterschiedlichen Positionen vor allem der Mädchen in den Gruppen besprochen. Die Anzahl von 8 Mädchen im Informatik-Kurs ist für eine quantitative statistische Untersuchung kritisch. Deswegen versagt hier der  $\chi^2$ -Test. Als Alternative bietet sich der Rangsummentest nach *Wilcoxon* an, der auch noch für kleine Anzahlen ein sinnvolles Ergebnis produzieren kann.

Der  $p$ -Wert in der Tabelle [8.16] zeigt an, dass alle Gruppen ihre Meinung signifikant geän-

Gruppe	<i>n</i>	Pretest	Posttest	<i>p</i> -Wert
IF (weiblich)	8	2.38 (1.77)	2.11 (1.76)	0.019
KG <sub>int</sub> (weiblich)	13	3.08 (1.16)	2.17 (1.03)	0.014
IF (männlich)	19	3.00 (1.38)	3.14 (1.74)	0.003
KG <sub>int</sub> (männlich)	14	3.11 (1.27)	2.21 (1.48)	0.005

Tabelle 8.16: In Informatik haben Jungen mehr Erfolg als Mädchen.

Mittelwert(Standardabweichung) in einer sechsstufigen Skala und *p*-Wert für den Wilcoxon-Rangsummentest.

dert haben. Damit werden die im Kapitel [8.9.1] über die Genderfrage dargelegten Aussagen bestätigt.

## 8.13 Informatik versus ITG

Informatische Bildung wird in der Sekundarstufe I nur zu einem kleinen Teil in einem originären Informatikunterricht erteilt [Wee07, Sta10]. In vielen Bundesländern wird versucht, diese Ziele in Form der ITG zu erreichen [BHW85, NRW85, Bik90]. Diese Konzepte gehen durchweg von der Anwendungs- und Benutzerorientierung aus und sehen die Bedeutung der Informatik im Rahmen der Allgemeinbildung als Lieferant für Werkzeuge, deren sinnvolle Verwendung von den Schülern angeeignet werden sollen.

Zweifelsohne existieren heute viele Werkzeuge als informatische Produkte (digitale Artefakte [SD08]), die die Arbeit erleichtern und einen sinnvollen Beitrag im Leben des Einzelnen spielen können und die dementsprechend bei Bedarf beherrscht werden wollen. Zu einer Beherrschung gehören auf der einen Seite das Verständnis der Aufgabe eines konkreten Werkzeuges, die Handhabung desselben und auch das Selbstvertrauen des Einzelnen, im konkreten Fall sich in der Lage zu fühlen, bei einer gestellten Aufgabe ein solches Werkzeug auszusuchen, sich anzueignen bzw. die notwendigen Hilfen zur Aneignung zu finden.

Inzwischen geradezu ‚klassische‘ Produkte sind Textverarbeitung und Tabellenkalkulation. In der Befragung (8.8) haben wir die Selbsteinschätzung der Schülerinnen und Schüler zu diesen Produktklassen erfasst (8.8) und in der Follow-Up – Untersuchung (8.10) noch einmal hinterfragt. Dabei fiel auf, dass das Selbstvertrauen zur Beherrschung dieser Werkzeuge der Schülerinnen und Schüler der Informatikgruppe zu dieser Untersuchung sehr stark gestiegen war. Dabei wurde im Informatikunterricht im Gegensatz zum Unterricht in den externen Kontrollgruppen nicht mit diesen Produkten gearbeitet.

In den Think-Aloud – Interviews sind die Schüler daher mit Fragen aus dem ITG-Curriculum befragt worden, um zu überprüfen, wie die dort angegebenen Ziele im ITG-Unterricht in den externen Kontrollgruppen im Vergleich zu der internen Kontrollgruppe ohne jeglichen Informatikunterricht und der Informatikgruppe ohne spezifischen Unterricht in der Nutzung von Werkzeugen erreicht werden. Entsprechend dem *Curriculum der UNESCO* [BDF<sup>+</sup>00] wurden die Schüler anhand des Materials [8.96] zu den Themen ‚Dateien und Dateisysteme‘ [BDF<sup>+</sup>00, S.59] und zum Thema ‚Textverarbeitung‘ [BDF<sup>+</sup>00, S.61] befragt.

Die Trichter

Zwei Trichter wandeln durch die Nacht.  
 Durch ihres Rumpfs verengten Schacht  
 fließt weißes Mondlicht  
 still und heiter  
 auf ihren  
 Waldweg  
 u. s.  
 w.



Die Trichter

Zwei Trichter wandeln durch die Nacht.  
 Durch ihres Rumpfs verengten Schacht  
 fließt weißes Mondlicht  
 still und heiter  
 auf ihren  
 Waldweg  
 u. s.  
 w.

Abbildung 8.96: Material zu den ITG-Fragen.

Verzeichnisbaum	n	unbekannt	Verständnis		
			kein	partiell	voll
IF	13	38%	15%	23%	23%
KG <sub>int</sub> + KG <sub>ext</sub>	12	58%	33%	8%	0%
KG <sub>ext</sub>	4	75%	0%	25%	0%

Tabelle 8.17: Auswertung der Interviews zum Verständnis der Struktur eines Verzeichnisbaumes

## Verzeichnisstrukturen

Die Tabelle [8.17] zeigt, dass die Schülerinnen und Schüler der Informatikgruppe ein tieferes Verständnis bezüglich der hierarchischen Strukturen in einem Verzeichnisbaum erworben haben. Dies verdeutlicht noch einmal die Notwendigkeit, diese Thematik im konkreten Unterricht intensiv zu behandeln, da ein Verständnis derartiger informatiktypischer Strukturen nicht durch das Alltagserfahren erfolgt.

## Textverarbeitung

Die Auswertung der Interviews zum Thema Verzeichnisstrukturen mag noch nicht sehr überraschend sein, war doch an verschiedenen Stellen im Unterricht dieses Thema konkret im Unterricht behandelt wurden und auch teilweise in den Testen (Kursarbeiten) erfragt worden. Das in den Interviews deutlich dokumentierte Unverständnis der meisten Schüler in den Kontrollgruppen fordert demnach eine intensive Behandlung dieser Thematik im Unterricht aller Schüler.

Überraschend dagegen ist die Auswertung der Interviews zum Thema ‚Zentrierung‘ entsprechend Tabelle [8.18] beispielsweise in Textverarbeitungssystemen. Im Unterricht der Informatikgruppe ist zu keiner Zeit mit einem Textverarbeitungssystem gearbeitet worden. Texte wurden ausschliesslich mit verschiedenen Editoren erstellt. In den externen Kontrollgruppen

<i>Zentrierung</i>	<i>n</i>	keine Idee	Leerzeichen	Ausrichten	Beides
IF	14	0%	21%	43%	36%
KG <sub>int</sub>	12	34%	25%	41%	0%
KG <sub>ext</sub>	19	21%	42%	31%	4%

Tabelle 8.18: Auswertung der Interviews zum Verständnis des Zentrierens von Textdokumenten

war neben der intensiven Beschäftigung mit Tabellenkalkulationssystemen die Arbeit an und mit Textverarbeitungssystemen ein Schwerpunkt. Daher musste erwartet werden, dass einem Grossteil dieser Schülerinnen und Schüler diese grundlegenden Arbeitsweisen bekannt sind, ist doch das ‚Zentrieren‘ in praktisch allen Textverarbeitungssystemen als Button auf einer der Hauptleisten auf der Programmoberfläche realisiert. Dass nun gerade aus diesen Gruppen ein nicht unbeträchtlicher Teil von über 20% nicht einmal eine Idee hat, wie denn dieses Problem zu lösen ist, ist schon auffällig. Dagegen fällt positiv auf, dass alle(!) befragten Schüler der Informatikgruppe zumindest eine Lösungsidee hatten und zu mehr als einem Drittel beide Alternativen angeben konnten. Praktisch 80% sahen in der Zentrierung ein Konzept. Da das Konzept der Zentrierung auf der Quelltextebene in *HTML* behandelt worden ist, hat hier seitens der Schüler offensichtlich ein Transfer stattgefunden, der zweifelsohne informatisches Denken voraussetzt und repräsentiert.

### **Das Loch in der Wand – Experiment**

Es wäre sicher vermessen zu behaupten, die Beantwortung dieser zwei Fragestellungen aus dem *UNESCO-Curriculum* [BDF<sup>+</sup>00, S.59/61] für eine informationstechnologische Grundbildung beweise generell die Unnötigkeit einer derartigen Unterweisung in der Schule. Aber natürlich ist umgekehrt die Frage zulässig, ob denn zumindest ein Informatikunterricht bis zum Jahrgang 10 in der in dieser Arbeit beschriebenen Art eine ITG überflüssig macht. Immerhin haben die Informatikschüler die Anforderungen aus dem UNESCO-Curriculum auf einem höheren Niveau erfüllt als die Schüler, die sich im Unterricht schwerpunktmässig mit Anwendungen gerade aus dem Office-Bereich beschäftigen.

Da Unterrichtszeit nicht beliebig zur Verfügung steht, bedarf dies einer Klärung im Interesse zukünftiger Schüler und erfordert daher zur Verifikation oder Falsifikation eine empirische Untersuchung nicht nur einer oder einiger weniger Gruppen. Das erfordert eine Einführung eines Informatikunterrichtes zumindest in ausreichend vielen Schulen.

Es darf dabei erwartet werden, dass die Ergebnisse eines solchen Versuches nicht unseren Erkenntnissen widersprechen. Diese Vermutung wird erhärtet durch das ‚Loch in der Wand – Experiment‘<sup>18</sup> [MK99], [Lös02, S.154-156]. Dieses Experiment fand in Indien statt und bestand darin, in der Wand zwischen einem Slum-Viertel und einer IT-Firma einen Monitor samt Zugriff zu einer Tastatur und Maus zu deponieren und zu beobachten, wie die Kinder und Jugendlichen, die ansonsten in wahrhaft katastrophalen Verhältnissen leben und spielen mussten und praktisch ohne schulische Bildung aufwachsen, auf dieses ‚Angebot‘ reagieren

<sup>18</sup><http://www.hole-in-the-wall.com>, letzter Zugriff: 27.9.2012



würden. Ohne jede Anleitung und ohne jede bisherige Verbindung zur Computer- und Online-Welt eroberten sich die Kinder nach kurzem Zögern die ihnen gebotenen Möglichkeiten. Ohne Anleitung und Hilfen waren diese Kinder in der Lage, mit und über diese Computer zu kommunizieren und daraus Nutzen zu ziehen. Die Ergebnisse aus diesem ersten *Loch in der Wand*-Versuch konnten durch Wiederholungen in anderen Ländern bestätigt werden. Sofern sich diese Beobachtungen auch im schulischem Umfeld in ähnlicher Weise zeigen sollten, sind danach die wesentlichen Lernziele der heutigen ITG nach den Kriterien von Heymann [Hey96, S.62] kein schulischer Lerngegenstand.



# 9 Fazit und weitere Untersuchungen

## 9.1 Fazit Informatikunterricht nach 1 Jahr

### 9.1.1 Unterrichtsfach Informatik

Nach dem ersten Jahr Informatikunterricht im 6. Jahrgang kann eine Gesamteinschätzung sicher nur vorläufigen Charakter haben. Ob die angestrebten Ziele für die Sekundarstufe I insgesamt erreicht werden, lässt sich abschliessend frühestens nach Abschluss der Jahrgangsstufe 10 klären. Im Rahmen der Evaluation können wir allerdings erste Tendenzen beschreiben.

Wir sind von folgenden Fragen ausgegangen (siehe Kapitel 7.1.3):

#### Hauptfragen

- Können wir bereits in den Anfangsjahren der Sekundarstufe I Informatik als vollwertiges Fach unterrichten?
- Können wir Informatik sogar als Hauptfach unterrichten?

Darüberhinaus wollten wir klären:

#### Spezielle Fragen

- Erkennen alle Beteiligten Informatik als ein vollwertiges Fach an?  
(Akzeptanz)
- Unterscheiden sich die Noten in Informatik von denen in Mathematik & Naturwissenschaften?  
(Noten)
- Unterscheiden sich die Inhalte in Informatik von denen in einem ITG-Kurs?  
(Inhalte)

Die Beantwortung der speziellen Fragen hilft, eine Antwort für die Hauptfragen zu formulieren. Daher beginnen wir die vorläufige Antwort mit den speziellen Fragen:

- **Akzeptanz:**  
**Erkennen alle Beteiligten Informatik als ein vollwertiges Fach an?**

Im Kapitel 7.1.4 wurde beschrieben, dass viele Eltern bereits auf der Vorstellung des Wahlpflichtfaches Informatik/Physik die Wichtigkeit eines solchen Angebotes von sich aus betont haben. Allein die grosse Anzahl der anwesenden Eltern an diesem Abend verdeutlichte das Interesse seitens der Elternschaft.

Für die Schüler ist die Vollwertigkeit aus verschiedener Sicht gegeben: Das Wahlpflichtfach ist Hauptfach und damit nur unwesentlich mit entscheidend für den Abschluss am Ende der Sekundarstufe I. Im Kapitel 8.1.2 zeigte sich bei der Analyse der Fragen zum Lieblingsfach, dass Informatik eine gleichwertige und damit vollwertige Stellung zwischen den mathematisch-naturwissenschaftlichen Fächern eingenommen hat.

Im Fragebogen wurde gefragt, ob die Schüler der Meinung sind, ob es nach ihrer Meinung wichtig ist, über *Basiskenntnisse im Programmieren* zu verfügen (siehe Kapitel 8.9.1). Bereits im Pre-, aber noch mehr im Posttest wird diese Meinung von etwa 50% der Schüler geteilt und weitere 30% halten das nicht für ausgeschlossen. Die Programmierung steht hier stellvertretend für die Informatik, sodass aus sachlicher Sicht die Vollwertigkeit der Informatik seitens der Schüler erkennbar ist.

Die Schulleitung und das Kollegium der Fritz-Steinhoff-Schule hat es ermöglicht, dass dieser Kurs überhaupt zustande gekommen ist. Diese Schulöffentlichkeit als Teil der Gesamtöffentlichkeit hat damit die Bedeutung unseres Ansatzes zumindest wohlwollend zur Kenntnis genommen.

**Die Frage nach der Akzeptanz lässt sich damit mit Ja beantworten.**

- **Noten:**

**Unterscheiden sich die Noten in Informatik von denen in Mathematik & Naturwissenschaften?**

Im Kapitel 8.1.1 wurde beschrieben, dass die Zeugnisnoten in Mathematik fast einer idealen *Gauß'schen Normalverteilung* entsprechen und sowohl Informatik, aber auch die Naturwissenschaften davon etwas abweichen. In beiden Fächern gibt es eine Verteilung, die je zwei lokale Maxima enthält. Die Noten verteilen sich fast über das gesamte Spektrum der Notenskala, sodass deutlich wird, dass die Schülerinnen und Schüler als Gruppe weder unter- oder überfordert sind.

Auch die Noten in den Kursarbeiten (siehe Kapitel 8.1) entsprechen diesem Bild. Die ersten Arbeiten zeigten die Gefahr einer ‚Spaltung‘ zwischen besseren und schlechteren Schülern. Durch eine verbesserte Übungsstruktur konnte der Graben zwischen diesen beiden Gruppen überwunden werden und eine ‚normale Verteilung‘ in den letzten Kursarbeiten erreicht werden. Es zeigte sich damit, dass in Informatik in den unteren Jahrgängen Arbeiten geschrieben werden können, mit denen die Leistungsfähigkeit der Schülerinnen und Schüler gut abgebildet wird.

**Die Noten im Schulfach Informatik unterscheiden sich im positiven Sinn nicht qualitativ von denen in den affinen Fächern Mathematik und Naturwissenschaften.**

- **Inhalte:**

**Unterscheiden sich die Inhalte in Informatik von denen in einem ITG-Kurs?**

Im Kapitel 7.5 *Unterrichtsverlauf Informatik Jahrgangsstufe 6* ist das Curriculum des Informatikkurses ausführlich vorgestellt worden. Es unterscheidet sich wesentlich von dem Curriculum eines *ITG-Kurses*, das im Kern den Werkzeug- und teilweise zusätzlich den *Mediencharakter* informatischer Produkte betrachtet [BDF<sup>+</sup>00]. In der Evaluation zeigte sich, dass einige aus dem Kanon der UNESCO entnommenen Fragenkomplexe besser von den Informatikschülern als von den ITG-Schülern in Halden und Vorhalle gelöst wurden (siehe Kapitel 8.13). Daher stellt sich die Frage nach der generellen Rechtfertigung eines *ITG-Unterrichtes*, während im Gegenzug dazu ein Informatik-Unterricht zu fordern ist.

**Die Untersuchung legt die Feststellung nahe, dass ein *ITG-Unterricht* in der Sekundarstufe I durch einen Informatikunterricht ersetzt werden kann, aber nicht umgekehrt.**

Die Antwort zu den speziellen Fragen leiten uns zur Beantwortung der Hauptfragen:

- **Pflichtfach:**

**Können wir bereits in den Anfangsjahren der Sekundarstufe I Informatik als vollwertiges (Pflicht-)Fach unterrichten?**

Im Kapitel 8 Evaluation wurde gezeigt, dass in vielfältiger Weise Unterschiede im Wissen, Einschätzen und auch Meinungen und Einstellungen in und über die Informatik zwischen den Schülern der Informatikgruppe und den verschiedenen Kontrollgruppen existieren.

Im Gegensatz dazu zeigte sich, dass viele der im heutigen Leben unverzichtbaren informatischen auftretendem Begriffe, Inhalte und Kompetenzen nicht von selbst im auserschulischen Bereich und wenn doch, nicht automatisch korrekt bei den Schülern aufgenommen und verstanden werden. Die Vermittlung ist somit nach dem Allgemeinbildungsbegriff nach Heymann Aufgabe der Schule [Hey96].

Nach weiteren 5 Jahren Unterricht bis zum Ende der Sekundarstufe I im Jahrgang 10 werden die bisher ermittelnden Unterschiede wahrscheinlich noch gravierender werden. Ansätze eines *informatischen Denkens* konnten im Informatikkurs daher gelegt und nachgewiesen werden.

Darüber hinaus lässt sich das Schulfach Informatik vergleichbar anderen Fächern in dieser Altersstufe für die Schüler organisieren und durchführen. Schriftliche Leistungsmessungen in Form von Kursarbeiten sind altersgemäß durchführbar und die Notenverteilung entspricht den anderer Fächer.

Eine wichtig Aufgabe von Schule ist die Vorbereitung auf die Berufswelt. Diese ist für die Schüler in der Jahrgangsstufe 6 noch in relativ weiter Zukunft. Es zeigte sich, dass

die Beschäftigung mit informatischen Themen die Offenheit für Berufe im IT-Bereich vergrößert. Das ist aus Sicht der Gesellschaft unter dem Eindruck des wahrscheinlich wachsenden Fachkräftemangels und aus Sicht der Schüler bezüglich der Erlangung eines sicheren und interessanten Arbeitsplatzes beiderseits positiv zu werten.

In dem untersuchten Kurs waren relative wenige Mädchen vertreten. Ihr Selbstbewusstsein bezüglich Informatik wurde gestärkt, selbst wenn die eigenen Leistungen im Einzelfall verbesserungswürdig waren.

Besonders zu berücksichtigen ist, dass dieser Kurs weitgehend der ‚normalen‘ Schülerschaft der FSG entspricht. Die FSG hat zudem einen relativ kleinen Anteil an typischen Gymnasialschülern. Die Untersuchung unterstrich damit die Möglichkeit, anspruchsvollen Informatik-Unterricht auch für Schüler zu konzipieren und durchzuführen, die nicht unbedingt eine besondere Begabung oder Interesse für informatische Themen mitbringen. Wie in jedem anderen Unterrichtsfach auch ist bei der Planung entsprechend zu berücksichtigen, dass gerade für eine derartige Schülerschaft ausreichend Zeit für die Themen und vor allem auch Übungszeit mit geeignetem Übungsmaterial angeboten wird.

Dieser Kurs war kein Pflichtfach für alle Schüler. Da allerdings die Zusammensetzung sich nicht unwesentlich von einer normalen Schülergruppe im anderen Unterricht der Schule unterschieden hat, ist ein Übertragen der Ergebnisse auf einen Pflichtfachunterricht gerechtfertigt.

**Fazit:**

**Informatik hat sich im Untersuchungszeitraum als gleichwertiges (Pflicht-)Unterrichtsfach für den 6. Jahrgang herausgestellt.**

• **Hauptfach:**

**Können wir Informatik sogar als Hauptfach unterrichten?**

Der Kurs an der Fritz-Steinhoff-Gesamtschule Hagen was als Wahlpflicht-Kurs organisiert und durchgeführt. Er hat damit aus Sicht der Schülerinnen und Schüler Hauptfachcharakter. Entsprechend diesem Charakter nahmen und nehmen die Schülerinnen und Schüler diese Kurs ernst, sodass bis auf Ausnahmen die bisherigen Abschlussnoten ihre angestrebten und gerechtfertigten Abschlüsse am Ende der Sekundarstufe I nicht gefährden. Die inhaltlichen Anforderungen sind anderen Wahlpflichtfächern im Umfang und Schwierigkeit nicht nachgeordnet.

**Fazit:**

**Informatik kann als Hauptfach vergleichbar den Naturwissenschaften im Wahlpflichtbereich unterrichtet werden.**

## 9.1.2 Beschränkung des Curriculums

Es ist nachvollziehbar, dass die Schüler zu Beginn eines solchen Unterrichtes aufgrund keiner oder geringer Vorkenntnisse nur kleinere Kontexte bearbeiten konnten. Schon im Laufe

eines Jahres wurden diese umfangreicher und es ist zu erwarten, dass in den nächsten Jahren grössere und interessantere Kontexte aufbereitet werden können. Als Beispiele seien hier Kontexte aus den Bereichen Musik und Grafik/Animation/Video genannt, die dann auch aufgrund physikalischer Grundkenntnisse, die im zweiten Teil dieses Unterrichtsprojektes, der Physik, vorbereitet werden, adäquat behandelt werden können.

Nach der ursprünglichen Planung war erhofft worden, dass zumindest einer dieser Bereiche schon in diesem ersten Schuljahr Informatikunterricht Thema hätte sein sollen. Das dies nicht erreicht wurde, ist erklärbar:

- Die Schülerinnen und Schüler befanden sich im 6. Schuljahr. Die Erfahrungen mit Informatikunterricht entstammen weitgehend der Altersgruppe des 9. und 10. Jahrganges. Trotz aller sorgfältigen Transferüberlegungen ist das mögliche Tempo im 6. Jahrgang überschätzt worden. Da die Fähigkeiten der Schüler der Gradmesser für das Vorgehen im Unterricht sind, musste das Tempo entsprechend angepasst, in diesem Fall reduziert werden.
- Diese Gruppe war eine Wahlpflichtgruppe. Und doch — das war ja auch unser Wunsch — entsprach diese Gruppe wesentlich mehr in ihrer Zusammensetzung einer ‚normalen‘ Klasse als beispielsweise eine der früheren Wahlpflicht II – Gruppen in den Jahrgängen 9 und 10. Diese waren offensichtlich altersbedingt deutlich interessenorientierter und zielgerichteter. Zudem ist ein Unterricht, der sich an *alle Schüler* und nicht an spezifische Teilgruppen von Schülern richtet, von Natur aus langsamer im Fortgang.
- Es war unser Anspruch, einen ‚normalen‘ Unterricht zu untersuchen. Die WP-Gruppe ist annähernd so gross wie eine normale Klasse an der Fritz-Steinhoff-Gesamtschule.  
Diese Gruppe hatte dazu zumeist einen Raum zur Verfügung, in dem 30 Computer für die Schülerinnen und Schüler vorhanden waren. Die Geduld, Ausdauer und Experimentierfähigkeiten waren bei vielen Schülern in diesem Alter noch nicht sehr ausgeprägt. Sie verlangten daher nach sehr viel Hilfe bei der konkreten Rechnerarbeit. Diese ist bei 28 Schülern nur schwer realisierbar. Die gegenseitige Hilfe muss im Laufe der Zeit gelernt werden und ist nicht immer zielführend. Gelegentlich wurden daher Zweiergruppen gebildet, sodass nur 14 Schülergruppen betreut werden mussten. Auch dieses kann schon für einzelne Gruppen zu Leerlaufzeiten führen.
- Der Unterricht ist nicht für eine sogenannte ‚Elitegruppe‘ entwickelt worden. Daher wird mehr mehr Übungszeit für den einzelnen Schüler benötigt. Diese Zeit wurde den Schülern gewährt und dies war erfolgreich.

## 9.2 Weitere Untersuchungen

Die Untersuchung beschränkte sich auf das erste Unterrichtsjahr dieser Gruppe. Viele Fragen konnten nur vorläufig und teilweise unvollständig beantwortet werden. Weitere Untersuchungen sind daher in dieser Gruppe, aber auch in anderen Gruppen sinnvoll.

- In den nächsten Jahren soll versucht werden, diese Gruppe durch weitere Befragungen und eventuell ergänzende Interviews wissenschaftlich bis zum Ende der Sekundarstufe I zu begleiten. Die Fragen in den Fragebögen der dargelegten Untersuchung sind so konzipiert, dass sie den weiterführenden Lernprozess bis Jahrgangsstufe 10 repräsentieren können. Sollte diese Untersuchung entsprechend stattfinden, läge damit erstmalig eine Langzeituntersuchung eines Unterrichts in Informatik als Hauptfach in der Sekundarstufe I vor.
- Ein Teil der Schülerinnen und Schüler an der FSG wird am Ende der Sekundarstufe I in die Sekundarstufe II an der FSG überwechseln. Noch ist völlig offen, wie viele Schüler das sein werden. Sofern diese Zahl gross genug sein sollte, könnte diese Untersuchung in abgewandelter Form fortgesetzt werden.
- Mit den Ergebnissen können darauf aufbauend die *Bildungsstandards Informatik* [Arb08] überprüft und evaluiert werden. Damit würden diese Bildungsstandards den Charakter einer Vision und eines wünschenswerten Anspruches ablegen und vergleichbar den Bildungsstandards anderer Fächer werden. Zusätzlich könnten derartige Untersuchungen in eine zukünftige Bildungsstandards für die Sekundarstufe II einfließen.
- Andere Schulen können aufgrund eines derartigen Unterrichtsprojektes Interesse entwickeln, auch an ihrer Schule ein entsprechendes Angebot zu schaffen. Es wäre positiv, die Ergebnisse dieser Gruppen mit der untersuchten Gruppe zu vergleichen.

Es bleibt zu hoffen, dass sich mit diesen und weiteren kleinen Schritten das Schulfach Informatik aus seinem Schattendasein befreien kann.



# 10 Schlussbemerkung

Im Kapitel 3 Bildungswissenschaft ist beschrieben worden, wie sich Veränderungen in der Bildungslandschaft oft nur langsam durchsetzen. Es gibt dagegen aber Zeiten, in denen sich Schule merkbar und relativ schnell ändert. Diese haben natürlich dann zumeist mit gesellschaftlichen Umbrüchen zu tun. Eine der Triebfedern sind technologische Entwicklungen, die auf die Gesellschaft und auf die Bildung wirken. Wir können dies beispielsweise an der Geschichte der naturwissenschaftlichen Schulfächer beobachten, die in Deutschland Anfang des 20. Jahrhunderts endgültig den schulischen Durchbruch geschafft haben [MNU05, MNU05, Hir87], nachvollziehen. Auch die Diskussion und die Entwicklung in den 60er und 70er Jahren in der Bundesrepublik Deutschland, die inhaltlich mit dem Bildungsgesamtplan [Bun73] endete, ist ein beredtes Zeugnis für eine solche Entwicklung.

*Saul Benjamin Robinsohn* hat in *Bildungsreform als Revision des Curriculum* [Rob67, S.31-43] beschrieben, wie in fast allen industrialisierten Staaten erhebliche Anstrengungen unternommen wurden, den Bildungsbereich zu reformieren. Unter anderem fand in den Vereinigten Staaten unter Leitung von *Jerome S. Bruner* eine zehntägige Tagung mit 35 führenden Pädagogen, Psychologen und Forschern aus Hochschule und der Wirtschaft statt, um die Bildung unter anderem in den naturwissenschaftlichen Fächern in den Primar- und Sekundarschulen zu verändern [Bru60, S.VII]. Das Werk *The Process of Education* [Bru60] ist der von Bruner erstellte Report dieser Konferenz. Im 'Preface' wird von Bruner dargelegt:

*“Major efforts in curriculum design had been launched by leading physicists, mathematicians, biologists, and chemists, and similar projects were in prospect in other fields of scientific endeavor. Something new was stirring in the land.” [Bru60, S.VII]*

Zum ersten Mal waren bei einer derartigen Konferenz Psychologen beteiligt [Bru60, S.IX]. Dies zeigt ein völlig neues Verständnis von Schule und Unterricht. Es verwundert daher nicht, dass als wichtigste und nachhaltigste Erkenntnis dieser Tagung die Notwendigkeit eines *Spiralcurriculums* ist [Bru60, S.52], [Bru70, S.61].

In der Einleitung lesen wir (hier in der deutschen Ausgabe von 1970):

„Was sollen wir lehren und mit welchem Ziel?

Die neue geistige Einstellung spiegelt vielleicht auch die in unserer Zeit stattfindende tiefgreifende Umwälzung in den Wissenschaften wieder. Dieser Trend wird durch eine höchstwahrscheinlich langfristige Krise der Sicherheit der Nation akzentuiert, eine Krise, für deren Bewältigung es auf eine gut ausgebildete Bürgerschaft ankommt.

Einer der Punkte, an denen sich diese neu auflebende Interesse an Erziehungsfragen gezeigt hat, ist die Curriculumplanung für Primar- und Sekundarschulen. ... In noch nicht dagewesenen Maße haben sich an der Entwicklung der Lehrpläne Geistes- und Naturwissenschaftler von Universitäten beteiligt, Männer, die sich durch Arbeiten an der vordersten Linie ihrer Disziplinen ausgezeichnet hatten. Sie haben Lehrpläne für Primar-

und Sekundarschulen ausgearbeitet, welche nicht nur die jüngsten Forschungsergebnisse berücksichtigen, sondern auch kühn vorseilenden Vorstellungen von der schulischen Praxis eine konkrete Gestalt verleihen.“ [Bru70, S.16]

Die curriculare Entwicklung von Schule darf nicht stehen bleiben, da gesellschaftliche und technische Entwicklungen weiter voranschreiten. Daher benötigen wir auch mehr als fünfzig Jahre später das Engagement vieler Menschen bei der Weiterentwicklung von Schule und deren Inhalte. Waren vor etwa fünfzig Jahren der immer noch geringe Stellenwert der Naturwissenschaften, aber auch der Sozialwissenschaften, die viele Mitbürger mit Sorge betrachtet haben, so ist es heute die Informatik, deren Integration in die Schulen immer noch nicht ausreichend realisiert wurde.



# 11 Anhang

## 11.1 Eltern - Informationsbroschüre: WP-Wahl 3/2010

Die Schülerinnen und Schüler unseres Jahrganges können neben den üblichen Angeboten im Bereich Naturwissenschaften das Fach

### Naturwissenschaften Physik und Informatik

wählen.

Dieses Angebot ist Teil eines Forschungsvorhaben der TU Dortmund in Zusammenarbeit mit der Fritz-Steinhoff-Gesamtschule Hagen.



#### Warum Physik und Informatik?

Die **Physik** ist die „Mutter“ aller klassischen Naturwissenschaften. Im Niederländischen heisst sie daher auch „Naturkunde“. Alle wesentlichen Naturphänomene haben (fast) immer auch einen physikalischen Anteil.

Das gilt auch für das Fach **Informatik**. Auch wenn es dieses Fach kaum mehr als 40 Jahre gibt, so steckt im Mittelpunkt der Informatik ein naturwissenschaftlicher Kern: *Information* ist ein nicht unwesentlicher



Teil auch der Natur, deren Bedeutung aber erst so recht durch die technische Anwendung in Computern und heute zusätzlich in Netzen deutlich wurde. Informatik ist damit auch, aber nicht nur eine Naturwissenschaft.

*Information* benötigt zudem als Träger immer eine physikalische Basis in Form z.B. elektrischer Impulse in Kabeln oder Magnetisierungen auf Festplatten.

In diesem **WP-Fach Naturwissenschaften -- Physik und Informatik** beschäftigen wir uns daher immer wieder um physikalische und informatische Phänomene und deren Wechselwirkungen.

#### Was wird untersucht?

##### Physikalische Grundlagen



So besteht beispielsweise ein Computer aus elektronischen Schaltungen. Will man ihn verstehen, so sind Kenntnisse aus der Elektrik und Elektronik vonnöten. So werden unter Anderem auch digitale Schaltungen thematisiert. Diese basieren auf logischen Prinzipien. Dies hilft sicher auch in der Mathematik.

Mit dem Computer werden u.a. Musik und Bilder gespeichert, bearbeitet und übertragen. Will man dies richtig verstehen und auch praktizieren, sind entsprechende Kenntnisse aus der Physik -- hier Optik und Akustik -- sinnvoll. Dabei werden auch Streifzüge in die Biologie unternommen, denn die Aufbereitung dieser Informationen ist abhängig von der Wahrnehmungsmöglichkeiten des Menschen.

### Programmierung

Die Bearbeitung von derartigen Dokumenten wird von Programmen ermöglicht. Diese müssen erstellt -- *programmiert* -- werden. Dementsprechend ist die **Programmierung** ein wesentlicher Anteil heutiger technischer Gegenwart und Arbeit. Als vor einigen Jahrzehnten die kleinen Computer überall auftauchten, war es notwendig, selber viele Anwendungen herzustellen, sprich zu programmieren. Programmieren und Informatik wurde gleichgesetzt. Heute glauben viele, da Programmieren zur Herstellung von Programmen vom 'Normaluser' nicht mehr notwendig ist, ist das Wissen über Programmieren und dessen Konzepte nicht mehr notwendig und sinnvoll. Aber erst der, der die Prinzipien der Funktions- und Arbeitsweise der Computer an Hand des Programmierens kennengelernt hat, kann ihn zielgerichtet einsetzen.

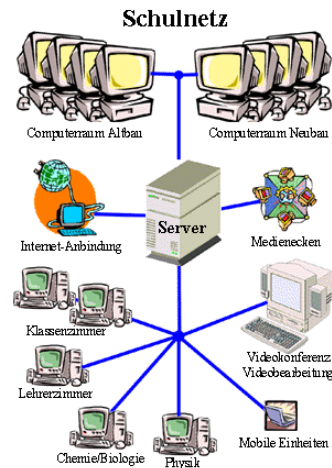
### Die Programmiersprache Tcl/Tk

Wir werden in diesem Kurs mit der **Programmiersprache Tcl/Tk** arbeiten, die leicht zu erlernen ist und mit der (fast) alles gemacht werden kann, was überhaupt mit einer Programmiersprache gemacht werden kann. Die Schülerinnen und Schüler erhalten daher auch Kenntnisse und Fertigkeiten über ein Werkzeug, das sie weit über die Schulzeit hinaus benutzen können.

### Netze

Wir bewegen uns in **Netzen** und transportieren und hinterlegen dort Informationen. Ein verständnisvoller Umgang damit benötigt Kenntnisse über Aufbau und Wirkungsweise derartiger Netze. Es ist doch immer wieder faszinierend zu erkennen, wie über eine Leitung praktisch beliebig viele Personen scheinbar parallel miteinander in Kontakt treten können. Das Verständnis über die Funktionsweise und auch die Grenzen dieser Netze soll zu einem verstärkten bewussten Verhalten führen.

Diese physikalischen und informatischen Systeme haben vor allem in den letzten fünfzig Jahren erhebliche Veränderungen in der Gesellschaft für alle Teilnehmer in Beruf-, Privat- und Freizeitleben verursacht. Auch in den



nächsten Jahren werden erhebliche Veränderungen erwartet.

### **Beruf und Freizeit**

Natürlich versteht sich von selbst, dass Schülerinnen und Schüler eines derartigen WP-Kurses erheblich besser gerüstet für Beruf und Freizeit sind als Schüler, die diesen Kurs nicht besucht haben werden.

### **Fachkräftemangel**

In Anbetracht des immer grösser werden Fachkräftemangels im technisch-informatischen Feld ist es zudem von grossem Vorteil, sich mit diesem Bereich schon in der Schule intensiv zu beschäftigen.

## **Wer sollte das Fach Naturwissenschaften -- Physik und Informatik -- wählen?**

Voraussetzungen für dieses Fach sind:

- Interesse an naturwissenschaftlichen, technischen und informatischen Fragen
- Freude am Planen, Experimentieren, Auswerten und Testen
- Bereitschaft zu kooperativer und konzentrierter Arbeit



Es versteht sich von selbst, dass gerade Mädchen dieses Fach wählen können und sollten. Wenn wir wollen, dass aus den sogenannten *typischen Männerberufen* Berufe für Mann und Frau werden sollen, dann müssen Mädchen schon frühzeitig an diese Fächer herangeführt werden.

Und die Erfahrung zeigt: Mädchen stellen sich im Umfeld der Physik und Informatik nicht im Geringsten dümmer als Jungen an.

Ein eigener Computer zu Hause ist nicht zwingend, da wir in der Schule über ein sehr grosses Schulnetz bis in viele Klassen hinein haben. Schülerinnen und Schüler können an den verschiedensten Stellen der Schule ihre Aufgaben auch für dieses Fach erledigen. Ein eigener Computer zu Hause schadet sicher nicht, solange er für schulische Zwecke genutzt wird. Wir werden nur Software benutzen, die für Eltern und Schüler keine Kosten verursachen.

*(Bildquellen:*

*Netzgrafik: <http://www.ffb.shuttle.de/pluto/ovtgnetz/images/Netz-m.GIF>*

*Titelbild von "Tcl/Tk für Dummies":*

*[http://www.amazon.de/Tcl-Tk-für-Dummies-CD-ROM/dp/3826628721/ref=sr\\_1\\_10?ie=UTF8&s=books&qid=1266955699&sr=8-10](http://www.amazon.de/Tcl-Tk-für-Dummies-CD-ROM/dp/3826628721/ref=sr_1_10?ie=UTF8&s=books&qid=1266955699&sr=8-10)*

*alle anderen Grafiken entstammen der freien OpenClipart Bibliothek)*



# 11.2 Kursmappe

## Kursmappe \* WP If/Ph 6

### 1. Halbjahr Schuljahr 2010/2011

Datum	E/D	Thema	Aufgabe
31.8.2010	E	<ul style="list-style-type: none"> <li>• Pretest</li> <li>• Hefte etc.</li> </ul>	
3.9.2010	E	<ul style="list-style-type: none"> <li>• Namen/Vorstellungen</li> <li>• Wish</li> <li>• Übung in der Wish</li> </ul>	
8.9.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Computer sind saublöd</li> <li>• Wiederholung Wish</li> <li>• Übung in der Wish: puts</li> <li>• Übung: button anlegen</li> <li>• Button mit neuer Hintergrundfarbe</li> <li>• 2. Button anlegen</li> <li>• für die Mappe: Zettel für Kommandos</li> <li>• Button für Programmstarts</li> </ul>	
10.9.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Heftkontrolle</li> <li>• Wdhlg.: Übung: button anlegen</li> <li>• Button mit Programmstart: Opera</li> </ul>	
21.9.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Heftkontrolle</li> <li>• Wdhlg.: Übung: button anlegen mit Programmstart</li> </ul>	
24.9.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Heftkontrolle</li> <li>• Aufschreiben: WZ legen Objekte an. Diese haben Namen und Eigenschaften ...</li> <li>• Als Text erstellen: 2 Buttons mit Programmstart: Opera, nedit</li> </ul>	
1.10.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Heftkontrolle</li> <li>• für die Mappe: Zettel für Kommandos</li> <li>• Werkzeug Label: Text: Elise</li> <li>• Werkzeuge nebeneinander: Label und Button</li> </ul>	
6.10.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Heftkontrolle</li> <li>• Zettel button</li> <li>• für die Mappe: Zettel für Kommandos</li> <li>• Button, Label mit neuer Hintergrundfarbe</li> <li>• Stundenplan mit Labels</li> </ul>	
8.10.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel, Heftkontrolle</li> <li>• Zettel label</li> <li>• Editor</li> <li>• Stundenplan mit Labels</li> <li>• Werkzeuge nebeneinander: Label und Button</li> </ul>	
3.11.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel</li> <li>• Verzeichnisse(-Bäume)</li> </ul>	
5.11.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel, Heftkontrolle</li> <li>• Infzettel für Verzeichnisse vorlesen</li> <li>• Übung durchführen</li> </ul>	
10.11.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel, Heftkontrolle</li> <li>• Übung 1a durchführen</li> </ul>	Aufg. 2
12.11.2010	E	<ul style="list-style-type: none"> <li>• Namenszettel, Heftkontrolle</li> <li>• HA besprechen</li> </ul>	
		<ul style="list-style-type: none"> <li>• Namenszettel, Heftkontrolle</li> </ul>	



19.11.2010	E	<ul style="list-style-type: none"> <li>• Wdhlg. Verzeichnisse, Bäume</li> <li>• Bäume in Anwendungsprogrammen</li> <li>• cp</li> <li>• Stundenplan mit Labels</li> <li>• Werkzeug nebeneinander: mehrere Tage</li> </ul>	
24.11.2010	E	<ul style="list-style-type: none"> <li>• Test: auf Zetteln, kein Kuli, Bleistifte!!!</li> <li>• Wdhlg. cp</li> <li>• Stundenplan mit Labels und mehreren Tagen</li> </ul>	Wdhlg.
26.11.2010	E	Test	
3.12.2010	E	<ul style="list-style-type: none"> <li>• Rückgabe Test</li> <li>• Besprechung</li> </ul>	Korrektur
8.12.2010	E	<ul style="list-style-type: none"> <li>• Installation TCL/TK unter Win</li> <li>• Teilweise Demo</li> </ul>	Korrektur
10.12.2010	E	Weiter: Stundenplan	Korrektur
15.12.2010	E	<ul style="list-style-type: none"> <li>• Einsammeln Test</li> <li>• Fertigstellen Stundenplan</li> <li>• Bild mit import</li> </ul>	
22.12.2010	E	<ul style="list-style-type: none"> <li>• Vorstellen Puppy und Verteilen</li> <li>• Fertigstellen Stundenplan</li> <li>• Bild mit import</li> </ul>	
12.1.2011	E	<ul style="list-style-type: none"> <li>• &lt;i&gt;Nachfragen zu Puppy</li> <li>• Stundenplan als Bild mit import</li> <li>• Webseite mit Bild</li> </ul>	
14.1.2011	E	<ul style="list-style-type: none"> <li>• &lt;i&gt;Ergänzungen zu Puppy</li> <li>• Wdhlg. Stundenplan als Bild mit import</li> <li>• Weiter: Webseite mit Bild</li> <li>• Ergänzungen</li> </ul>	
19.1.2011	E	<ul style="list-style-type: none"> <li>• &lt;i&gt;evtl. Kopieren Bild und Webseite Stundenplan</li> <li>• Weiter und Umbauen: Webseite mit Bild</li> <li>• Ergänzungen: Überschrift 2. Ordnung, Lehrerverz.</li> </ul>	
21.1.2011	E	<ul style="list-style-type: none"> <li>• evtl. Kopieren Bild und Webseite Stundenplan</li> <li>• Skizze erstellen</li> <li>• Weiter und Umbauen: Webseite mit Bildern</li> </ul>	
26.1.2011	E	<ul style="list-style-type: none"> <li>• Testtermin</li> <li>• Wdhlg. Webseite mit Bildern</li> <li>• Bilder kopieren</li> <li>• Umbauen: Webseite mit Bildern</li> </ul>	
28.1.2011	E	<ul style="list-style-type: none"> <li>• Testtermin</li> <li>• Wdhlg. Webseiten-Tags</li> <li>• Was ist zu tun: Bilder nach unten</li> <li>• Bilder drehen</li> <li>• Umbauen: Webseite mit Bildern oben und unten</li> <li>• zusätzlicher Text</li> </ul>	
2.2.2011	E	Test	
4.2.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe und Besprechung Test</li> <li>• Notenbesprechung</li> <li>• Abschrift A1 erstellen</li> </ul>	fertigstellen
9.2.2011	E	<ul style="list-style-type: none"> <li>• Rest-Rückgabe, Verteilen Ausdrucke Test</li> <li>• Zeichnen in TCL, Canvas, Line</li> <li>• Aufschreiben</li> </ul>	Abschrift A2 fertigstellen

## Kursmappe \* WP If/Ph 6

### 2. Halbjahr Schuljahr 2010/2011

Datum	E/D	Thema	Aufgabe
11.2.2011	E	<ul style="list-style-type: none"> <li>• Rest-Rückgabe</li> <li>• Wdhlg.: Zeichnen in TCL, Canvas, Line</li> <li>• Aufschreiben Koordinatensystem</li> <li>• Zeichnet zwei beliebige Linien</li> </ul>	Abschrift A3 fertigstellen
16.2.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Tests an mich</li> <li>• HA kontrollieren</li> <li>• Verteilen Zettel/Vorlesen</li> <li>• A1 auf Papier erledigen</li> <li>• Arbeit am Rechner</li> <li>• Weitere Aufgaben</li> <li>• Zeichnet ein Quadrat</li> <li>• Zeichnet ein Rechteck</li> <li>• Zeichnet ein Rechteck der Breite x und Höhe y</li> </ul>	A2 fertigstellen
18.2.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Tests an mich</li> <li>• HA kontrollieren</li> <li>• Arbeit am Rechner</li> <li>• Weitere Aufgaben</li> </ul>	A1, A2 vom Übungsblatt 3
23.2.2011	E	Test TU Dortmund	
25.2.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Tests an mich</li> <li>• HA kontrollieren: Teste, A1, A2</li> <li>• Arbeit am Rechner</li> </ul>	A4
2.3.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Tests an mich</li> <li>• alte HA kontrollieren: Teste, A1, A2</li> <li>• Besprechen HA: A4</li> <li>• Arbeit am Rechner</li> <li>• Aufgabe A5</li> </ul>	Fertigstellen: A5
4.3.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Tests an mich</li> <li>• alte HA kontrollieren: Teste, A1, A2 ...</li> <li>• Aufgaben Tagungswoche: Hefte, Ax</li> <li>• Besprechen HA: A5</li> <li>• Arbeit am Rechner</li> <li>• Weitere Aufgaben</li> </ul>	Hefte, Ü-Blatt 3a
16.3.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Tests an mich</li> <li>• HA Hefte</li> <li>• Besprechen HA: Ü-Blatt 3a,b</li> <li>• Arbeit am Rechner</li> <li>• Weitere Aufgaben</li> </ul>	Ü3,6
18.3.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Hefte</li> <li>• HA Ü3,A6</li> <li>• Programm</li> <li>• Arbeit am Rechner</li> </ul>	
25.3.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Hefte II</li> <li>• Zeichnet ein Rechteck</li> <li>• Programm</li> <li>• Arbeit am Rechner</li> <li>• Zeichnet ein Quadrat</li> </ul>	Ü3,6
1.4.2011	E	<ul style="list-style-type: none"> <li>• Rückgabe der Hefte III</li> <li>• was war in der Vertretung</li> <li>• Zeichnet ein Rechteck</li> <li>• Zeichnet ein Quadrat</li> <li>• Programm</li> <li>• Arbeit am Rechner</li> <li>• Das Haus vom Nikolaus</li> </ul>	Ü3,6

6.4.2011	E	<ul style="list-style-type: none"> <li>• Hampelmann</li> <li>• Arbeit am Rechner</li> </ul>	Wdhlg.
8.4.2011	E	Test 3	Wdhlg.
13.4.2011	E	<ul style="list-style-type: none"> <li>• Vektorgrafik Definition</li> <li>• Def Pixelgrafik</li> <li>• Demo mit XPM-Format</li> <li>• Spielen am Computer mit XPM-Format</li> </ul>	Wdhlg.
4.5.2011	E	<ul style="list-style-type: none"> <li>• Ankündigung: Praktikanten Uni</li> <li>• Ausfall Test</li> <li>• Rückgabe Test</li> <li>• Besprechung Test</li> </ul>	Berichtigung
6.5.2011	E	<ul style="list-style-type: none"> <li>• Begrüssung Praktikanten Uni</li> <li>• Einsammeln Test</li> <li>• Vorstellung Aufgabe</li> <li>• Durchführen der Aufgabe am Computer</li> <li>• Besprechen</li> </ul>	
13.5.2011	E	<ul style="list-style-type: none"> <li>• Einsammeln Test</li> <li>• Weiter: Aufgabe 1(A)</li> <li>• Besprechen</li> <li>• Durchführen der Aufgaben 2 am Computer</li> <li>• Besprechen</li> </ul>	
18.5.2011	E	<ul style="list-style-type: none"> <li>• Einsammeln Test</li> <li>• Besprechen: Aufgabe 1(A)</li> <li>• Durchführen der Aufgaben 2(a) am Computer</li> </ul>	
20.5.2011	E	<ul style="list-style-type: none"> <li>• Einsammeln Test</li> <li>• Vorstellen AB 3: 2 xpm-Dateien</li> <li>• Bearbeiten</li> <li>• Besprechen</li> </ul>	
3.6.2011	E	<ul style="list-style-type: none"> <li>• Test am 10.6.</li> <li>• Kurze Wiederholung xpm-Dateien</li> <li>• Vorstellen AB 3a: 2 xpm-Dateien</li> <li>• Durchführen der Aufgaben 3a am Computer</li> <li>• Besprechen</li> </ul>	
8.6.2011	E	<ul style="list-style-type: none"> <li>• Test am 10.6.:Bunt-/Bleistift, Geodreieck, Fineliner, alte Teste</li> <li>• Vorstellen AB 4: A1 und A2 auf Papier</li> <li>• Durchführen der Aufgaben am Computer</li> <li>• Besprechen</li> </ul>	
10.6.2011	E	Test 4	
15.6.2011	E	<ul style="list-style-type: none"> <li>• Snickers für Rückgabe von Testen</li> <li>• Rückgabe und Besprechung Test</li> <li>• Korrektur</li> </ul>	.. fertigstellen
22.6.2011	E	<ul style="list-style-type: none"> <li>• Snickers für Rückgabe von Testen</li> <li>• Rückgabe Test II, HA Korrekt./Abschrift</li> <li>• Einsammeln Hefte nächsten Freitag</li> <li>• Variablenarbeitsblatt</li> </ul>	.. fertigstellen
29.6.2011	E	<ul style="list-style-type: none"> <li>• Erinnerung: Snickers für Rückgabe von Testen</li> <li>• Ausgabe Materialien für Quiz: Robert, Eda, Rabia</li> <li>• Einsammeln Hefte Freitag</li> <li>• Variableneinführung</li> </ul>	.. fertigstellen
1.7.2011	E	<ul style="list-style-type: none"> <li>• Hefte am Ende der Stunde</li> <li>• Erinnerung: Snickers für Rückgabe von Testen</li> <li>• Quiz Robert und Co. 3. Stunde</li> <li>• Mitbringen: Papier DIN A4, CF-Schachteln etc., OHP-Schreiber</li> <li>• Variableneinführung</li> </ul>	.. fertigstellen

## 11.3 Materialien

TCL/TK - Grafikwerkzeug:

*button*

---

Syntax:

***button .name -text "Knopfaufschrift" -command Kommando***

Beispiel:

***button .opera\_starten -text "Opera starten" -command opera***

---

### Beschreibung:

Mit dem Grafikwerkzeug *button* wird im grafischen Fenster ein Knopf erzeugt. Der Knopf hat einen *Namen*, der mit einem Punkt beginnt, und kann mehrere Eigenschaften haben.

Die verschiedenen Eigenschaften beginnen mit einem Bindestrich gefolgt von einer Bezeichnung und einem Wert. Im Beispiel oben wird ein Knopf mit dem Namen *.opera\_starten* erzeugt. Auf dem Knopf steht der Text *Opera starten*. Beim Drücken auf diesen Knopf wird das Programm *Opera gestartet*.

Damit der Knopf sichtbar wird, muss er mit `pack .opera_starten` auf die Grafikoberfläche gepackt werden.

---

### Eigenschaften:

- |                                |   |
|--------------------------------|---|
| <i>-text "Knopfaufschrift"</i> | Auf dem Knopf steht der Text <i>Knopfaufschrift</i> .           |
| <i>-background "Farbe"</i>     | Der Knopf erhält die angegebene Farbe als Hintergrund.          |
| <i>-command Kommando</i>       | Beim Drücken auf den Knopf wird das <i>Kommando</i> ausgeführt. |

**Syntax:**

*label* .name -text "Aufschrift"

**Beispiel:**

*label* .opera\_info -text "Opera Browser"

---

**Beschreibung:**

Mit dem Grafikwerkzeug *label* wird im grafischen Fenster ein „Aufkleber“ erzeugt. Der Aufkleber hat einen *Namen*, der mit einem Punkt beginnt, und kann mehrere Eigenschaften haben.

Die verschiedenen Eigenschaften beginnen mit einem Bindestrich gefolgt von einer Bezeichnung und einem Wert. Im Beispiel oben wird ein Aufkleber mit dem Namen *.opera\_info* erzeugt. Auf dem Aufkleber steht der Text *Opera Browser*.

Damit der Aufkleber sichtbar wird, muss er mit `pack .opera_info` auf die Grafikoberfläche gepackt werden.

---

**Eigenschaften:**

-text "Aufschrift" Auf dem Aufkleber steht der Text *Aufschrift*.

-background "Farbe" Der Aufkleber erhält die angegebene Farbe als Hintergrund.

-width "10" Die Breite des Aufklebers beträgt 10 Einheiten.

```
puts "Wir basteln uns unseren Stundenplan"
```

```
puts "Version 3"
```

```
destroy .montag  
destroy .dienstag  
destroy .mittwoch  
destroy .donnerstag  
destroy .freitag
```

```
frame .montag  
pack .montag -side left  
label .montag.tag -text "Montag" -width "15" -height "2"  
pack .montag.tag  
label .montag.stunde1 -text "DE" -background "red" -width "15"  
pack .montag.stunde1  
label .montag.stunde2 -text "MA" -background "orange" -width "15"  
pack .montag.stunde2  
label .montag.stunde3 -text "MA" -background "orange" -width "15"  
pack .montag.stunde3  
label .montag.stunde4 -text "EN" -background "pink" -width "15"  
pack .montag.stunde4  
label .montag.stunde5 -text "AS" -background "purple" -width "15"  
pack .montag.stunde5  
label .montag.stunde6 -text "MU" -background "burlywood" -width "15"  
pack .montag.stunde6
```

```
frame .dienstag  
pack .dienstag -side left  
label .dienstag.tag -text "Dienstag" -width "15" -height "2"  
pack .dienstag.tag  
label .dienstag.stunde1 -text "GL" -background "green" -width "15"  
pack .dienstag.stunde1  
label .dienstag.stunde2 -text "SP" -background "yellow" -width "15"  
pack .dienstag.stunde2  
label .dienstag.stunde3 -text "DE" -background "red" -width "15"  
pack .dienstag.stunde3  
label .dienstag.stunde4 -text "KU" -background "white" -width "15"  
pack .dienstag.stunde4  
label .dienstag.stunde5 -text "M-F" -background "orange" -width "15"  
pack .dienstag.stunde5  
label .dienstag.stunde6 -width "15"  
pack .dienstag.stunde6
```

```

frame .mittwoch
pack .mittwoch -side left
label .mittwoch.tag -text "Mittwoch" -width "15" -height "2"
pack .mittwoch.tag
label .mittwoch.stunde1 -text "EN" -background "Pink" -width "15"
pack .mittwoch.stunde1
label .mittwoch.stunde2 -text "MA" -background "Orange" -width "15"
pack .mittwoch.stunde2
label .mittwoch.stunde3 -text "SP" -background "yellow" -width "15"
pack .mittwoch.stunde3
label .mittwoch.stunde4 -text "WP" -background "blue" -width "15"
pack .mittwoch.stunde4
label .mittwoch.stunde5 -text "OS" -background "purple" -width "15"
pack .mittwoch.stunde5
label .mittwoch.stunde6 -text "KU" -background "white" -width "15"
pack .mittwoch.stunde6

frame .donnerstag
pack .donnerstag -side left
label .donnerstag.tag -text "Donnerstag" -width "15" -height "2"
pack .donnerstag.tag
label .donnerstag.stunde1 -text "MA" -background "orange" -width "15"
pack .donnerstag.stunde1
label .donnerstag.stunde2 -text "GL" -background "green" -width "15"
pack .donnerstag.stunde2
label .donnerstag.stunde3 -text "DE" -background "red" -width "15"
pack .donnerstag.stunde3
label .donnerstag.stunde4 -text "NW" -background "cyan" -width "15"
pack .donnerstag.stunde4
label .donnerstag.stunde5 -text "ZK" -background "grey" -width "15"
pack .donnerstag.stunde5
label .donnerstag.stunde6 -text "REL" -background "brown" -width "15"
pack .donnerstag.stunde6

frame .freitag
pack .freitag -side left
label .freitag.tag -text "Freitag" -width "15" -height "2"
pack .freitag.tag
label .freitag.stunde1 -text "DE" -background "red" -width "15"
pack .freitag.stunde1
label .freitag.stunde2 -text "WP" -background "blue" -width "15"
pack .freitag.stunde2
label .freitag.stunde3 -text "EN" -background "pink" -width "15"
pack .freitag.stunde3
label .freitag.stunde4 -text "SP" -background "yellow" -width "15"
pack .freitag.stunde4

```

```
label .freitag.stunde5 -text "AS" -background "purple" -width "15"  
pack .freitag.stunde5  
label .freitag.stunde6 -width "15"  
pack .freitag.stunde6
```



Dateien werden in Verzeichnissen verwaltet. Um im Verzeichnisbaum zu navigieren, gibt es eine Reihe an Betriebssystemwerkzeugen.

## Werkzeuge/Kommandos

---

### Listen:

**Syntax:** *ls*

**Beschreibung:** Der Inhalt des Verzeichnisses wird aufgelistet.

---

### Ausgabe des aktuellen Verzeichnisses:

**Syntax:** *pwd*

(print working directory)

---

### Wechseln des Verzeichnisses (change directory)

**Syntax:** *cd Verzeichnis*

**Beschreibung:** Es wird in das angegebene Verzeichnis gewechselt.

**Beispiel:** *cd psa*

---

### Erzeugen eines Verzeichnisses (make directory)

**Syntax:** *mkdir Verzeichnis*

**Beschreibung:** Es wird das angegebene Verzeichnis erzeugt.

**Beispiel:** *mkdir emil*

---

### Löschen eines Verzeichnisses (remove directory)

**Syntax:** *rmdir Verzeichnis*

**Beschreibung:** Es wird das angegebene Verzeichnis gelöscht.

**Beispiel:** *rmdir dokumente*

---

---

## Übung:

Auf dem Fileserver F ist ein Verzeichnis *ueben* angelegt worden.

Wechsel in dieses Verzeichnis und lege dort ein Verzeichnis mit Deinem Vornamen an. Schreibe in dieses Verzeichnis mit dem Programm *nedit* einen kleinen Text in die Datei *toller\_text*.

## Lösung

```
cd F
cd uebung
mkdir fritzchen    (Bitte den eigenen Namen verwenden!)
cd fritzchen
nedit toller_text
```

## Syntax:

**canvas .name -width breite -height höhe -background farbe**

## Beispiel:

**canvas .leinwand -width 350 -height 500 -background yellow**

---

## Beschreibung:

Das Grafikwerkzeug erzeugt mit dem Aufruf *canvas* im grafischen Fenster eine (Leinwand-) Fläche für Zeichenobjekte. Die Leinwandfläche hat einen *Namen*, der mit einem Punkt beginnt, und kann mehrere Eigenschaften haben.

Die verschiedenen Eigenschaften beginnen mit einem Bindestrich gefolgt von einer Bezeichnung und einem Wert. Die wichtigsten Eigenschaften sind *Breite, Höhe und Hintergrundfarbe*.

Im Beispiel oben wird eine Leinwandfläche mit dem Namen *.leinwand* erzeugt. Diese Leinwandfläche ist 350 Pixel breit und 500 Pixel hoch. Die Hintergrundfarbe ist gelb.

Damit die Leinwandfläche sichtbar wird, muss sie mit *pack .leinwand* auf die Grafikoberfläche gepackt werden.

---

## Eigenschaften:

*-width Pixelanzahl* Die Leinwand ist *Pixelanzahl* Pixel breit.

*-height Pixelanzahl* Die Leinwand ist *Pixelanzahl* Pixel hoch.

*-background Farbe* Die Leinwand erhält die angegebene Farbe als Hintergrund.

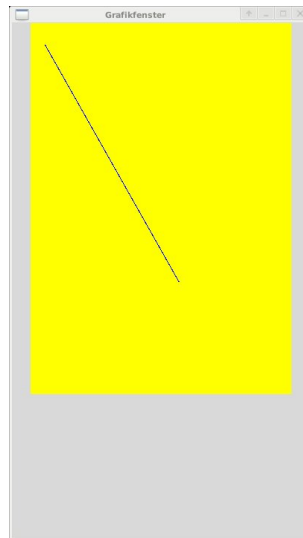
---

## Zeichnen auf der Leinwand:

Auf einer derartigen Leinwand können wir Zeichenobjekte platzieren.

## Beispiel:

`.leinwand create line 20 30 200 350 -fill blue`



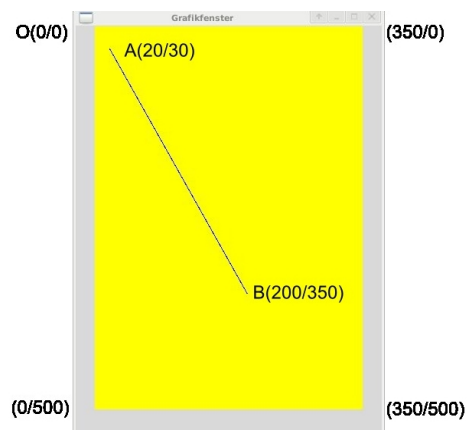
## Beschreibung:

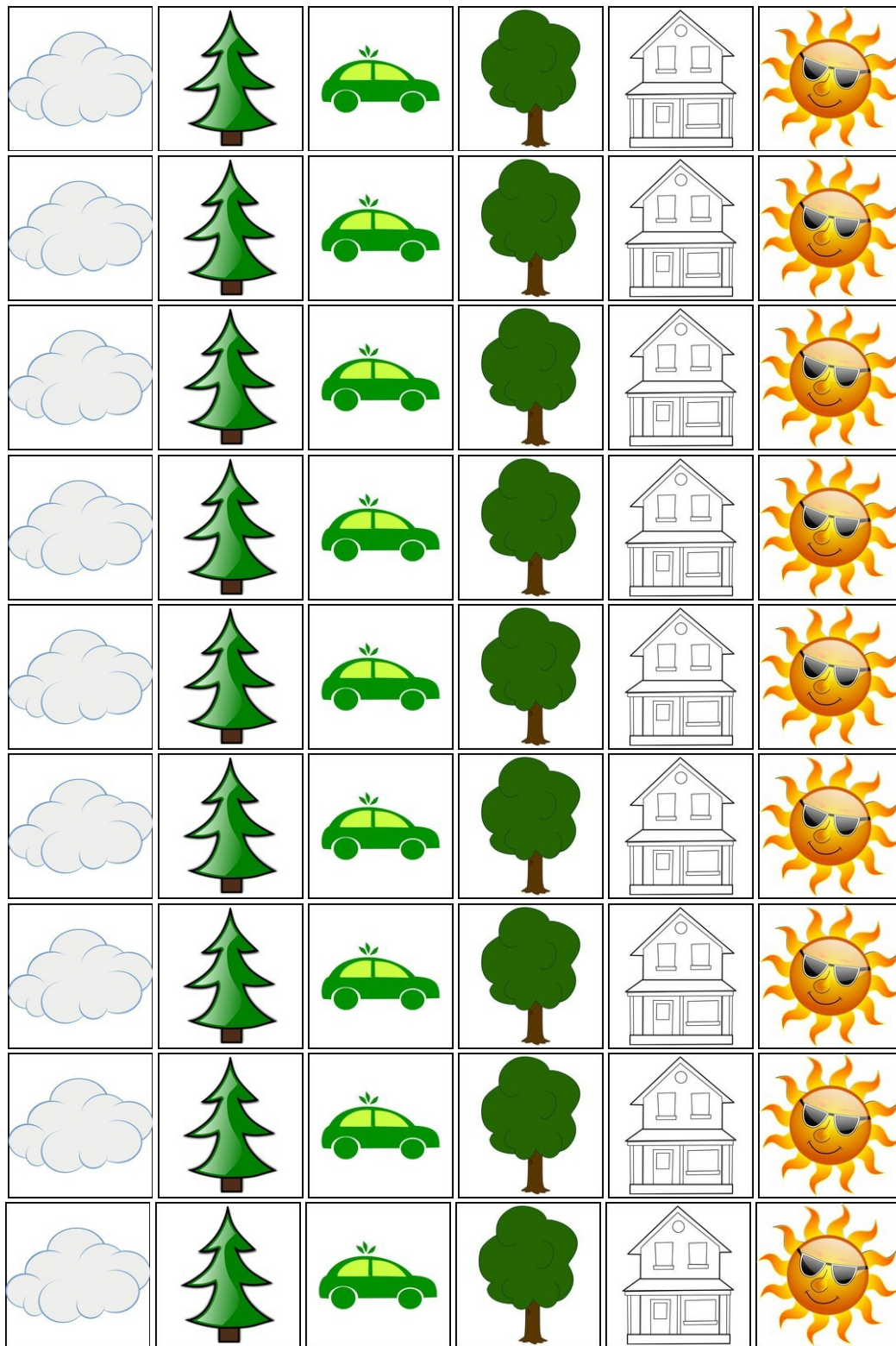
Eine Strecke vom Punkt  $A(20/30)$  bis zum Punkt  $B(200/350)$  wird in der Farbe *blau* gezeichnet.

Achtung:

Der *Ursprungspunkt*  $O(0/0)$  ist – anders als in der Mathematik üblich – in der linken oberen Ecke der Leinwand.

Beschriftet sieht die Leinwandfläche dann folgendermassen aus:





## 11.4 Übungen

### Übung 1:

### Verzeichnisse

psa/10.11.2010

Dateien werden in Verzeichnissen verwaltet.

Wir können in TCL/TK sehr einfach in Verzeichnissen navigieren und direkt dort mit Programmen wie beispielsweise `nedit` Dateien erzeugen und speichern.

#### Beispiel:

Es werden folgende Kommandos in der Konsole ausgeführt:

```
mkdir daten
mkdir texte
mkdir tabellen

cd daten
nedit verkauf10.dat
nedit verkauf11.dat
nedit verkauf12.dat
cd ..

cd texte
nedit protokol11.txt
nedit protokol12.txt
cd ..

cd tabellen
nedit geld5.tab
nedit geld6.tab
```

Wir erhalten (mit dem Werkzeug `tree`) eine Verzeichnisstruktur, wie sie hier abgebildet ist (Die Wurzel ist der Punkt):

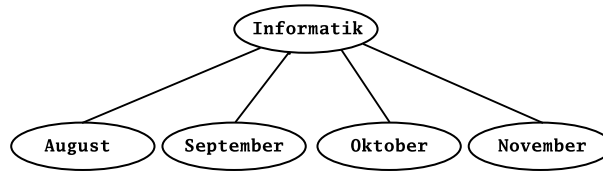
```
.
|-- daten
|   |-- verkauf10.dat
|   |-- verkauf11.dat
|   `-- verkauf12.dat
|-- tabellen
|   |-- geld5.tab
|   `-- geld6.tab
`-- texte
    |-- protokol11.txt
    `-- protokol12.txt
```

#### Aufgabe 1

- Zeichne diesen Baum um, sodass – wie im Unterricht besprochen – die Wurzel ganz oben ist und die Dateien ganz unten sind.
- Der Teilbaum `tabellen` wird nicht mehr benötigt. Du befindest dich im Teilbaum `texte`. Du möchtest nun alle Dateien im Teilbaum `tabellen` und den Baum `tabellen` löschen. Welche Kommandos musst du eingeben?

## Aufgabe 2

Du möchtest Dir eine Verzeichnisstruktur anlegen, die dem Verlauf unseres Unterrichts entspricht. Das Wurzelverzeichnis heisst `Informatik`.



- Welche Kommandos müssen eingegeben werden?
- In jedem dieser Verzeichnisse soll ein Unterverzeichnis für Protokolle und Übungen erstellt werden.

## Aufgabe 3

Folgende Kommandos werden an der Konsole eingegeben. Stelle den entstehenden Verzeichnisbaum zeichnerisch dar!

```
mkdir hans
mkdir adele

cd hans
mkdir deutsch
mkdir mathe
cd ..

cd adele
mkdir deutsch
mkdir mathe
cd mathe
nedit freitag.txt
cd ..
cd ..

cd hans
cd deutsch
nedit montag.txt
```

## Aufgabe 4

Du möchtest das Chaos auf der Festplatte auf Deinem Computer aufräumen. Dir erscheint sinnvoll, Dateien für die Schule und Privates zu trennen. In der Schule hast Du öfters in

Mathematik, Deutsch und Informatik Dateien erstellt. Im privaten Bereich hast Du für Deine Hobbys Theater und Schwimmen gelegentlich Dateien produziert. Einige weitere Dateien existieren auch noch auf der Festplatte, die keinem dieser Gebiete zugeordnet werden können.

Wie könnte ein sinnvoller Dateibaum aussehen?

- a) Zeichne einen geeigneten Vorschlag für einen Dateibaum.
- b) Wie lauten die Kommandos, die Du entsprechend in unserer Konsole eingeben müsstest.



# Aufgabe:

# Stundenplan als Webseite

psa/14.1.2011

## Bilder aus Fensterinhalten erstellen

Wir können mit Tcl/Tk Fensterinhalte gestalten. Einen solchen Fensterinhalt wollen wir nun als Bild erstellen und in unserem Verzeichnis sichern.

Starte daher zuerst *Wish* und wechsele dann in dein Verzeichnis:

```
cd F
cd erwin                Deinen Verzeichnisnamen verwenden!
```

In diesem Verzeichnis ist als Beispiel der Stundenplan `plan3.tcl` hinterlegt worden. Wir können diesen Plan auf zwei Arten ausführen:

```
source "plan3.tcl"      Dann wird dieser Plan im Grafikfenster angezeigt.
oder
wish plan3.tcl &       Dann wird ein neues Fenster "plan3.tcl" aufgemacht.
```

Diese Fenster können wir dem Werkzeug `import` in ein Bild „verwandeln“. (Der Name dieses Werkzeuges ist etwas merkwürdig.)

```
import -window plan3.tcl -pause 10 plan_erwin.jpg
```

Dann wird nach einer kurzen Pause von 10 Sekunden ein Bild von diesem Fenster mit der Bezeichnung `plan_erwin.jpg` im aktuellen Verzeichnis abgelegt. Dieses Bild können wir mit den verschiedensten Bildbetrachtern oder auch einem Browser ansehen, also beispielsweise:

```
xv plan_erwin.jpg      oder:
display plan_erwin.jpg oder auch:
opera plan_erwin.jpg
```

## Ein Bild in einer Webseite

Wir erstellen nun eine sehr kleine Webseite mit dem Bild unseres Planes als Inhalt. Starte unseren Editor mit `nedit plan_erwin.html` und schreibe folgenden Text:

```
<html>
<head><title>Stundenplan von Erwin</title></head>

<body>
  <center>
    <h1>Stundenplan von Erwin</h1>
    
  </center>
</body>

</html>
```

Dieses Seite können wir anschliessend mit einem Browser ansehen, also beispielsweise:

```
opera plan_erwin.html
```

## Bilder aus Fensterinhalten

In den letzten beiden Stunden haben wir den Stundenplan als Bild importiert. Wer dieses nicht geschafft hat, kann aus dem *Materialverzeichnis* eine Vorlage kopieren.

Vorgehen:

Starte daher zuerst *Wish* und wechsele dann in dein Verzeichnis:

```
cd F
cd erwin                               Deinen Verzeichnisnamen verwenden!
cp ../material/plan_vorlage.jpg plan_erwin.jpg    Auch hier deinen Namen verwenden!
```

Dieses Bild können wir – wie schon beschrieben – mit den verschiedensten Bildbetrachtern oder auch einem Browser ansehen, also beispielsweise:

```
xv plan_erwin.jpg           oder:
display plan_erwin.jpg      oder auch:
opera plan_erwin.jpg
```

## Erweiterungen unserer Webseite

Anschliessend haben wir eine Webseite mit dem Bild erstellt. Wer dies nicht geschafft hat kann – wie oben dargestellt – eine Vorlage kopieren.

```
cd F
cd erwin                               Deinen Verzeichnisnamen verwenden!
cp ../material/plan_vorlage.html plan_erwin.html    Auch hier deinen Namen verwenden!
```

Verändere diese Seite wie folgt:

```
<html>
<head><title>Stundenplan von Erwin</title></head>

<body>
  <center>
    <h1>Stundenplan von Erwin</h1>
    <h2>Dieser Plan gilt im 1. Halbjahr</h2>
    
  </center>

  <p>
    Lehrerverzeichnis:<br>
    psa: WP<br>
    gem: Ma, GL <br>
    klb: EN<br>
    ...:<br>
  </p>
</body>
</html>
```

Dieses Seite können wir wieder mit einem Browser ansehen, also beispielsweise:

```
opera plan_erwin.html
```

## Bilder

Wir wollen neben unserem Stundenplan weitere Bilder in unsere Webseite stellen: Eine selbst erstellte *Mondgesicht-Skizze* und ein Foto.

Das *Mondgesicht* erstellen wir mit dem Programm `xpaint`, einem kleinen und schon recht betagtem Malprogramm. Ähnliche Programme gibt es viele auf den verschiedensten Betriebssystemen.

Vorgehen:

Starte daher zuerst *Wish* und wechsele dann in dein Verzeichnis:

```
cd F
cd erwin                Deinen Verzeichnisnamen verwenden!
xpaint
```

In diesem Programm kannst Du unter `File` eine neue Leinwand – auf Englisch `canvas` – öffnen und dort zeichnen.

Dein Mondgesicht speicherst Du unter `skizze_erwin.jpg`.

Dieses Bild können wir – wie schon beschrieben – mit den verschiedensten Bildbetrachtern oder auch einem Browser ansehen, also beispielsweise: `xv plan_erwin.jpg`

Ein Foto von Dir erstellt Dein Informatiklehrer. Du kannst es mit:

```
cp ../material/foto_erwin.jpg .
```

in Dein Verzeichnis laden. Auch dieses Foto kannst Du beispielsweise mit `xv foto_erwin.jpg` ansehen.

## Erweiterungen unserer Webseite

Wenn Du das letzte Mal die Webseitenvorlage nicht kopiert hast, kannst Du das heute nachholen oder Du arbeitest mit deiner bisherigen Webseite weiter.

```
cd F
cd erwin                Deinen Verzeichnisnamen verwenden!
cp ../material/plan_vorlage.html plan_erwin.html Auch hier deinen Namen verwenden!
```

Starte mit `nedit plan_erwin.html` unseren Editor und verändere diese Seite wie folgt:

```
<html>
<head><title>Stundenplan von Erwin</title></head>

<body>
<h1>Dies ist die Webseite von Erwin</h1>

<hr>

<p>
<h3>Hier siehst Du eine Mondgesicht-Skizze und ein Bild von mir</h3>


</p>

<hr>

<p>
<center>
<h1>Stundenplan von Erwin</h1>
<h2>Dieser Plan gilt im 1. Halbjahr</h2>

</center>
</p>

<hr>
```

```
<p>
  Lehrerverzeichnis:<br>
  psa: WP<br>
  gem: Ma, GL <br>
  klb: EN<br>
  ...:<br>
</p>
</body>
</html>
```

Dieses Seite können wir wieder mit einem Browser ansehen, also beispielsweise:

```
opera plan_erwin.html
```

# Übung 3:

## canvas

psa/17.2.2011

Wir können in TCL/TK sehr einfach Zeichnungen erstellen. Dabei werden auf einer Leinwand, in TCL/TK *canvas* genannt, Objekte wie beispielsweise Strecken, in TCL/TK *line* genannt, gezeichnet.

Der Aufruf:

```
.leinwand create line 20 30 200 350 -fill blue
```

zeichnet eine Strecke vom Punkt  $A(20/30)$  bis zum Punkt  $B(200/350)$  in der Farbe *blau*.

Dabei liegt der *Ursprungspunkt*  $O(0/0)$  ist – anders als in der Mathematik üblich – in der linken oberen Ecke der Leinwand.

Bei allen folgenden Aufgaben gilt, dass eine Leinwand mit:

```
canvas .leinwand -width 350 -height 500 -background yellow
```

```
pack .leinwand
```

erzeugt werden soll.

**Wichtiger Hinweis:** Erstelle in Deinem Heft immer zuerst eine Zeichnung, bevor Du aufschreibst, welche Anweisungen Du in TCL/TK eingeben willst.

## Aufgabe 1

Zeichne auf der Leinwand eine Strecke, die von ganz rechts nach ganz links genau in der Mitte der Leinwand verläuft.

## Aufgabe 2

Zeichne auf der Leinwand eine Strecke, die von ganz oben nach ganz unten genau in der Mitte der Leinwand verläuft.

## Aufgabe 3

Zeichne beide Strecken auf eine Leinwand.

## Aufgabe 4

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 20 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 20 von der unteren Leinwandkante sein.

## Aufgabe 5

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 20 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 20 von der rechten Leinwandkante sein.

## **Aufgabe 6**

Zeichne auf der Leinwand drei waagerechte Strecken:

Die Abstände von der oberen und der unteren Kante und zwischen den Strecken sollen alle gleich sein.

## **Aufgabe 7**

Zeichne auf der Leinwand drei senkrechte Strecken:

Die Abstände von der oberen und der unteren Kante und zwischen den Strecken sollen alle gleich sein.

## **Aufgabe 8**

Zeichne auf der Leinwand ein beliebiges Dreieck.

## **Aufgabe 9**

Zeichne auf der Leinwand ein Dreieck mit einer Seitenlänge von 300 Pixelbreiten und einer Seitenlänge mit 200 Pixelbreiten. Die Länge der dritten Seite ist beliebig.

# Übung 3a: Training

canvas

psa/17.2.2011

---

Erstelle eine Leinwand mit:

```
canvas .leinwand -width 350 -height 500 -background yellow  
pack .leinwand
```

**Wichtiger Hinweis:** Erstelle in Deinem Heft immer zuerst eine Zeichnung, bevor Du aufschreibst, welche Anweisungen Du in TCL/TK eingeben willst.

---

## Aufgabe 1

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 20 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 20 von der unteren Leinwandkante sein.

## Aufgabe 2

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 30 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 30 von der unteren Leinwandkante sein.

## Aufgabe 3

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 40 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 40 von der unteren Leinwandkante sein.

## Aufgabe 4

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 50 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 50 von der unteren Leinwandkante sein.

## Aufgabe 5

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 14 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 14 von der rechten Leinwandkante sein.

## **Aufgabe 6**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 21 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 21 von der rechten Leinwandkante sein.

## **Aufgabe 7**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 28 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 28 von der rechten Leinwandkante sein.

## **Aufgabe 8**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 35 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 35 von der rechten Leinwandkante sein.

## **Aufgabe 9**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 42 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 42 von der rechten Leinwandkante sein.



# Übung 3b: Training

canvas

psa/17.2.2011

---

Erstelle eine Leinwand mit:

```
canvas .leinwand -width 350 -height 500 -background yellow  
pack .leinwand
```

**Wichtiger Hinweis:** Erstelle in Deinem Heft immer zuerst eine Zeichnung, bevor Du aufschreibst, welche Anweisungen Du in TCL/TK eingeben willst.

---

## Aufgabe 1

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 10 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 10 von der unteren Leinwandkante sein.  
rand

## Aufgabe 2

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 20 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 20 von der unteren Leinwandkante sein.  
rand

## Aufgabe 3

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 36 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 36 von der unteren Leinwandkante sein.  
rand

## Aufgabe 4

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 51 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 51 von der unteren Leinwandkante sein.  
rand

## Aufgabe 5

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 14 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 14 von der rechten Leinwandkante sein.

## **Aufgabe 6**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 21 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 21 von der rechten Leinwandkante sein.

## **Aufgabe 7**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 28 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 28 von der rechten Leinwandkante sein.

## **Aufgabe 8**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 35 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 35 von der rechten Leinwandkante sein.

## **Aufgabe 9**

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 42 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 42 von der rechten Leinwandkante sein.

Erstelle eine Leinwand mit:

```
canvas .leinwand -width 30 -height 20 -background yellow
pack .leinwand
```

**Achtung:** Die für diese Aufgabe erstellte Leinwand ist viel kleiner als bei Aufgaben, die wir bisher bearbeitet haben!!

**Wichtiger Hinweis:** In Deinem Heft sollst Du für einen Pixel ein Kästchen wählen.

**Achtung:** Es handelt sich nicht um einen Tippfehler: Gemeint ist tatsächlich *pro Pixel ein Kästchen!*

---

## Aufgabe 1

- Erstelle eine Leinwand der Breite 30 und Höhe 20 Pixel. Die Leinwand soll einen gelben Hintergrund haben.
- Zeichne ein Kreuz bestehend aus zwei Diagonalen auf diese Leinwand.  
Eine Diagonale soll von der Ecke oben rechts zur Ecke unten links verlaufen und die andere von oben links nach unten rechts.  
Die Diagonalen sollen rot sein.

## Aufgabe 2

- Öffne den Editor `nedit kreuz.tcl` und schreibe die Anweisung entsprechend Aufgabe 1 als Text und speicher diesen Text wieder als `kreuz.tcl`.
- Führe diese Datei mit `source kreuz.tcl` in der TCL-Konsole aus.

## Aufgabe 3

- Wir wollen diese Leinwand in einem kleinen extra Grafikfenster sehen. Dieses erreichen wir mit:  

```
wish kreuz.tcl
```

Schliesse anschliessend dieses Fenster wieder.

- Damit wir in der TCL-Konsole weiterarbeiten können, müssen wir die Anweisung leicht verändern:  

```
wish kreuz.tcl &
```

(Dieses `'&'` öffnet das Grafikfenster und ermöglicht uns, parallel in der Konsole weiter zu arbeiten.)

## Aufgabe 4

a) Exportiere dieses Fenster als Bilddatei mit:

```
import -pause 10 kreuz.xpm
```

Zur Erinnerung:

- Wenn wir ein Bild abspeichern wollen, öffnen wir es in einer separaten `wish`, damit die angezeigte Leinwand genauso gross wird wie das Bild.
- Innerhalb der zehn Sekunden Pause von `import` musst du die Maus über das Fenster bewegen, welches aufgenommen soll.
- Wenn sich der Mauszeiger zu einem Kreuz ändert, musst du in das Fenster klicken.

b) Zeige diese Bilddatei mit einem Grafikprogramm an:

```
xv kreuz.xpm    oder  
display kreuz.xpm
```

c) Bei früheren Beispielen haben wir auch mit einem Browser wie z.B. *Opera* mit

```
opera kreuz.xpm
```

das Bild angezeigt.

Probiere dies jetzt auch aus.

Was siehst du?

## Aufgabe 5

a) Öffne diese Bilddatei mit unserem Editor `nedit kreuz.xpm`.

b) Beschreibe in Deinem Heft, was du siehst.

---

Wir arbeiten bei diesen Aufgaben mit folgender Leinwand:

```
canvas .leinwand -width 30 -height 20 -background yellow
pack .leinwand
```

**Achtung:** Die für diese Aufgaben erstellte Leinwand ist viel kleiner als bei Aufgaben, die wir bisher bearbeitet haben!!

---

## Aufgabe 1

Wir haben in der letzten Stunde eine kleine Leinwand erzeugt, auf der ein Kreuz zu sehen war. Anschliessend haben wir aus der *TCL/TK-Grafik* eine Bild im Format *xpm* erstellt.

- a) Zeige Dir diese Leinwand mit dem Bild erneut in einem kleinen extra Grafikfenster an. Dieses erreichen wir mit:

```
wish kreuz.tcl
```

Schliesse anschliessend dieses Fenster wieder.

- b) Zeige die daraus erzeugte Bilddatei im Format *xpm* erneut mit einem Grafikprogramm an:

```
xv kreuz.xpm      oder
display kreuz.xpm
```

- c) Bei früheren Beispielen haben wir auch in einem Browser wie z.B. Opera mit

```
opera kreuz.xpm
```

das Bild angezeigt.

Probiere dies jetzt auch aus.

Was siehst du? Beschreibe dies in Deinem Heft!

- d) Betrachte die ersten Zeilen in der Datei. Kannst Du die Bedeutung dieser Zeilen erkennen? Schreibe Deine Vermutung ins Heft.

## Aufgabe 2

- a) Öffne diese Bilddatei mit unserem Editor: `nedit kreuz.xpm`.  
b) Nun öffne diese Bilddatei mit einem anderem Editor: `xedit kreuz.xpm`.  
c) Eine weitere Möglichkeit ist der Editor: `mcedit kreuz.xpm`  
d) Beschreibe in Deinem Heft, was du mit den verschiedenen Editoren siehst.

(Bei jedem Editor sind die Befehle zum Öffnen und Schliessen etc. anders. Probiere aus und/bzw. frage Deinen Lehrer.)

Wir arbeiten bei diesen Aufgaben mit folgender Leinwand:  
`canvas .leinwand -width 30 -height 20 -background yellow`  
`pack .leinwand`

**Wichtiger Hinweis:** Erstelle in Deinem Heft immer zuerst eine Zeichnung, bevor Du aufschreibst, welche Anweisungen Du in TCL/TK eingeben willst.

In Deinem Heft sollst Du für 10 Pixel zwei Kästchen wählen.

---

## Aufgabe 1

a) Kopiere die Datei `kreuz.tcl` als `kreuz2.tcl`

```
cp kreuz.tcl kreuz2.tcl
```

b) Öffne mit `nedit kreuz2.tcl` im Editor die Datei `kreuz2.tcl`.

c) Nun soll eine grüne Linie waagerecht von ganz links nach ganz rechts genau in der Mitte der Leinwand verlaufen. Schreibe diese Anweisung auf und speicher diesen Text wieder als `kreuz2.tcl`.

d) Führe diese Datei mit `source kreuz2.tcl` in der TCL-Konsole aus.

## Aufgabe 2

Wir wollen nun diese Leinwand auch in einem extra Grafikfenster sehen. Führe also die Anweisung `wish kreuz2.tcl & aus`.

## Aufgabe 3

Exportiere dieses Fenster als Bilddatei wieder mit:

```
import -pause 10 kreuz2.xpm
```

## Aufgabe 4

a) Öffne diese Bilddatei mit einem Editor beispielsweise mit `xedit kreuz2.xpm`.

b) Beschreibe, was du siehst.

## Aufgabe 5

a) Öffne die beiden Bilddatei mit einem Editor:

```
xedit kreuz1.xpm & und
```

```
xedit kreuz2.xpm &
```

(Du kannst stattdessen auch die Bilder im Browser Opera betrachten.)

b) Beschreibe die Unterschiede.

Was bedeuten die Zeichen?

Kannst du erkennen, wie groß das Bild ist?

## Aufgabe 1

- a) Kopiere die Datei *kreuz.xpm* als *kreuz2.xpm*

```
cp kreuz.xpm kreuz2.xpm
```

Falls Du auf Deinem Computer über kein *kreuz.xpm* verfügst, kannst Du diese Datei auch vom Server übertragen mit:

```
cp F/material/kreuz.xpm kreuz2.xpm
```

- b) Öffne mit `xedit kreuz2.xpm` im Editor die Datei *kreuz2.xpm* .  
c) Nun soll eine rote Linie waagrecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen. Veränder den Text der Bilddatei entsprechend und speicher diesen Text wieder als *kreuz2.xpm* .  
d) Betrachte diese Datei mit `xv kreuz2.xpm` in der TCL-Konsole.

## Aufgabe 2

- a) Kopiere die Datei *kreuz.xpm* als *kreuz3.xpm*

```
cp kreuz.xpm kreuz3.xpm
```

- b) Öffne mit `xedit kreuz3.xpm` im Editor die Datei *kreuz3.xpm* .  
c) Nun soll eine **grüne** Linie waagrecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen. Veränder den Text der Bilddatei entsprechend und speicher diesen Text wieder als *kreuz3.xpm* .  
d) Betrachte diese Datei mit `xv kreuz3.xpm` in der TCL-Konsole.

## Aufgabe 3

Beschreibe in Deinem Heft, welche Veränderungen Du in Aufgabe 1 und Aufgabe 2 vorgenommen hast.

## Aufgabe 4

Füge Deinem Bild *kreuz3.xpm* noch eine **grüne** senkrechte Linie hinzu, die genau in der Mitte von oben nach unten verläuft.

## Aufgabe 5

Die Farbe der senkrechten Linie gefällt nicht. Ersetze die Farbe durch **blau**!

## Aufgabe 6

Die beiden Diagonalen sollen jetzt zweifarbig sein. Beide sollen in der unteren Hälfte die Farbe **schwarz** haben.





## Aufgabe 1

- a) Kopiere die Datei *kreuz.xpm* als *kreuz3.xpm*  
`cp kreuz.xpm kreuz3.xpm`

Falls Du auf Deinem Computer über kein *kreuz.xpm* verfügst, kannst Du diese Datei auch vom Server übertragen mit:

```
cp F/material/kreuz.xpm kreuz3.xpm
```

- b) Öffne mit `xedit kreuz3.xpm` im Editor die Datei *kreuz3.xpm* .  
c) Nun soll eine rote Linie waagerecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen. Veränder den Text der Bilddatei entsprechend und speicher diesen Text wieder als *kreuz3.xpm* .  
d) Betrachte diese Datei mit `xv kreuz3.xpm` in der TCL-Konsole.

## Aufgabe 2

- a) Kopiere die Datei *kreuz.xpm* als *kreuz4.xpm*  
`cp kreuz.xpm kreuz4.xpm`

- b) Öffne mit `xedit kreuz4.xpm` im Editor die Datei *kreuz4.xpm* .  
c) Nun soll eine rote Linie senkrecht von ganz oben nach ganz unten genau in der Mitte der Leinwand verlaufen. Veränder den Text der Bilddatei entsprechend und speicher diesen Text wieder als *kreuz4.xpm* .  
d) Betrachte diese Datei mit `xv kreuz4.xpm` in der TCL-Konsole.

## Aufgabe 3

- a) Kopiere die Datei *kreuz.xpm* als *kreuz5.xpm*  
`cp kreuz.xpm kreuz5.xpm`

- b) Öffne mit `xedit kreuz5.xpm` im Editor die Datei *kreuz5.xpm* .  
c) Nun soll eine **grüne** Linie waagerecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen. Veränder den Text der Bilddatei entsprechend und speicher diesen Text wieder als *kreuz5.xpm* .  
d) Betrachte diese Datei mit `xv kreuz5.xpm` in der TCL-Konsole.

## Aufgabe 4

Beschreibe in Deinem Heft, welche Veränderungen Du in Aufgabe 1, Aufgabe 2 und Aufgabe 3 vorgenommen hast.

## Aufgabe 5

Füge Deinem Bild `kreuz5.xpm` noch eine **grüne** senkrechte Linie hinzu, die genau in der Mitte von oben nach unten verläuft.

## Aufgabe 6

Die Farbe der senkrechten Linie gefällt nicht. Ersetze die Farbe durch **blau**!

## Aufgabe 7

Die beiden Diagonalen sollen jetzt zweifarbig sein. Beide sollen in der unteren Hälfte die Farbe **schwarz** haben.

**Aufgabe 1**

- a) Wie lauten die zugehörigen Anweisungen in *Tcl/Tk* für folgende Beschreibung:
- Erstelle eine Leinwand der Breite 20 und Höhe 10 Pixel. Die Leinwand soll einen grünen Hintergrund haben.
  - Zeichne ein Kreuz bestehend aus zwei Diagonalen auf diese Leinwand:  
Eine Diagonale soll von der Ecke oben rechts zur Ecke unten links verlaufen und die andere von oben links nach unten rechts. Die Diagonalen sollen rot sein.

b) Erstelle eine Zeichnung in Deinem Heft. Zeichne Sie farbig. (1 Kästchen pro Pixel)

- c) Diese in *Tcl/Tk* beschriebene *Vektorgrafik* wurde in das Pixel-Format *xpm* übertragen. Der Anfang dieser Datei sieht folgendermassen aus:

```
/* XPM */
static char *aufgabe_a[] = {
/* columns rows colors chars-per-pixel */
"22 12 3 1",
"o c green",
". c red",
"X c blue",
/* pixels */
"XXXXXXXXXXXXXXXXXXXXXXXXX",
"X...oooooooooooo...oX",
```

Schreibe den Inhalt der gesamten Datei ohne Kopf auf Dein Blatt. Färbe Deinen Dateitext.

- d) Der Kopf wird folgendermassen geändert:

```
/* XPM */
static char *aufgabe_a[] = {
/* columns rows colors chars-per-pixel */
"28 12 2 1",
"o c green",
". c red",
/* pixels */
```

Wie können die ersten beiden Zeilen lauten? Schreibe die ersten zwei „Pixel-Zeilen“ in Dein Heft.

- e) Kevin ändert den Kopf folgendermassen ab:

```
/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"28 12 4 1",
". c red",
"o c green",
". c blue",
/* pixels */
```

Das Bildbetrachtungsprogramm `display` verweigert die Anzeige. Ja, richtig! Der Kopf enthält ja jetzt auch zwei Fehler! Was ist falsch??

## Aufgabe 2

Der Inhalt der Datei `kreuz.xpm` sieht folgendermassen aus:

```
/* XPM */
static char *kreuz_klein[] = {
/* columns rows colors chars-per-pixel */
"13 11 3 1",
" c yellow",
" . c red",
" X c gray",
/* pixels */
"XXXXXXXXXXXXXXXX",
"X ..... .X",
"X. .... .X",
"X.. .. .X",
"X.... .....X",
"X.... .....X",
"X.. .. .X",
"X. .... .X",
"X ..... .X",
"X.....X",
"XXXXXXXXXXXXXXXX"
};
```

- a) Übertrage den Inhalt dieser Datei in Dein Heft (1 Kästchen für 1 Pixel) und färbe den entsprechenden Text.
- b)
  - Eine rote Linie soll waagrecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen.
  - Eine grüne Linie soll senkrecht von ganz oben nach ganz unten genau in der Mitte der Leinwand verlaufen.
  - Die beiden Diagonalen sollen jetzt zweifarbig sein. Beide sollen in der unteren Hälfte die Farbe **schwarz** haben.Schreibe den korrekten Inhalt der `xpm`-Datei erneut hin. Achte auf notwendige Veränderungen im Kopf.
- c) Die Ursprungsdatei wurde aus einem `Tcl/Tk`-Script erzeugt. Wie musste es lauten?
- d) Wie lautet das Script in `Tcl/Tk` für das veränderte Bild?

## Aufgabe 3

Überprüfe Deine Aufgabe 1 am Computer, indem Du das Script eintippst und testest. Erzeuge daraus die Pixel-Datei und überprüfe Deine Lösungen, indem Du im Editor die Veränderungen vornimmst und testest.

## Aufgabe 4

Erstelle am Computer Dein `Tcl/Tk`-Script aus Aufgabe 2. Entspricht es den Vorgaben? Tippe im Editor die vorgegebene `xpm`-Datei aus Aufgabe 2 und betrachte sie mit einem Bildbetrachter wie `xv` oder `display`. Gib Deine vorgeschlagenen Veränderungen im Editor ein und teste sie.

## Aufgabe 1

- a) Wie lauten die zugehörigen Anweisungen in Tcl/Tk für folgende Beschreibung:
- Erstelle eine Leinwand der Breite 20 und Höhe 10 Pixel. Die Leinwand soll einen grünen Hintergrund haben.
  - Zeichne ein Kreuz bestehend aus zwei Diagonalen auf diese Leinwand: Eine Diagonale soll von der Ecke oben rechts zur Ecke unten links verlaufen und die andere von oben links nach unten rechts. Die Diagonalen sollen rot sein.

### Lösung

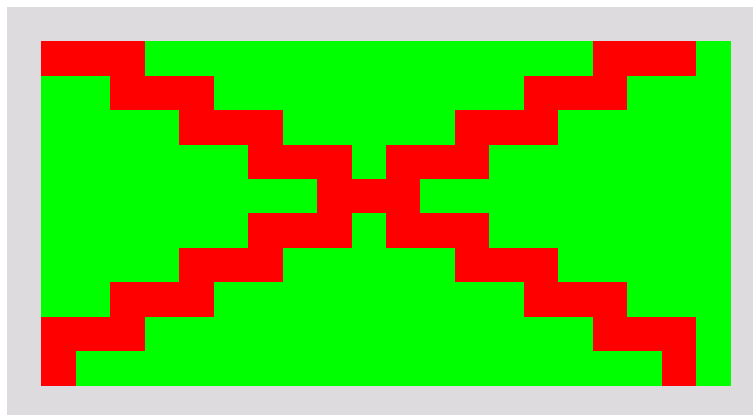
```

canvas .leinwand -width 20 -height 10 -background green
pack .leinwand

.leinwand create line 0 0 20 10 -fill red
.leinwand create line 20 0 0 10 -fill red
    
```

- b) Erstelle eine Zeichnung in Deinem Heft. Zeichne Sie farbig. (1 Kästchen pro Pixel)

### Lösung



- c) Diese in Tcl/Tk beschriebene Vektorgrafik wurde in das Pixel-Format xpm übertragen. Der Anfang dieser Datei sieht folgendermassen aus:

```

/* XPM */
static char *aufgabe_a[] = {
/* columns rows colors chars-per-pixel */
"22 12 3 1",
"o c green",
". c red",
"X c blue",
    
```

```

/* pixels */
"XXXXXXXXXXXXXXXXXXXXXXXXX",
"X...oooooooooooooooo...oX",

```

Schreibe den Inhalt der gesamten Datei ohne Kopf auf Dein Blatt. Färbe Deinen Dateitext.

### Lösung

```

"XXXXXXXXXXXXXXXXXXXXXXXXX",
"X...oooooooooooooooo...oX",
"Xoo...oooooooooooo...oooX",
"Xoooo...oooooo...oooooX",
"Xoooooo...o...oooooooooX",
"Xoooooo...ooooooooooooX",
"Xoooooo...o...oooooooooX",
"Xoooo...oooooo...oooooX",
"Xoo...oooooooooooo...oooX",
"X...oooooooooooooooo...oX",
"X.oooooooooooooooooooo.oX",
"XXXXXXXXXXXXXXXXXXXXXXXXX"
};

```

d) Der Kopf wird folgendermassen geändert:

```

/* XPM */
static char *aufgabe_a[] = {
/* columns rows colors chars-per-pixel */
"28 12 2 1",
"o c green",
". c red",
/* pixels */

```

Wie können die ersten beiden Zeilen lauten? Schreibe die ersten zwei „Pixel-Zeilen“ in Dein Heft.

### Lösung

Die Grösse der Leinwand hat sich vergrössert. Sie ist jetzt mit Rand 28 Pixel breit und nicht mehr 22 Pixel. Ausserdem gibt es nur noch zwei Farben. Daher muss der Rand sinnvollerweise die Farbe des Hintergrundes haben.

Die Zeilen können also folgendermassen aussehen:

```

"oooooooooooooooooooooooooooooooo",
"X...oooooooooooooooo...oX",

```

e) Kevin ändert den Kopf folgendermassen ab:

```

/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"28 12 4 1",
". c red",

```

```
"o c green",  
". c blue",  
/* pixels */
```

*Das Bildbetrachtungsprogramm display verweigert die Anzeige. Ja, richtig! Der Kopf enthält ja jetzt auch zwei Fehler! Was ist falsch??*

## **Lösung**

Es ist im Kopf angegeben, dass es 4 Farben geben soll, es sind dann aber nur drei Farben definiert. Ausserdem sind zwei Farben, hier rot und blau, mit demselben Zeichen codiert. Das kann ein Bildbetrachter nicht mehr korrekt interpretieren.

## Aufgabe 2

Der Inhalt der Datei `kreuz.xpm` sieht folgendermassen aus:

```
/* XPM */
static char *kreuz_klein[] = {
/* columns rows colors chars-per-pixel */
"13 11 3 1",
" c yellow",
". c red",
"X c gray",
/* pixels */
"XXXXXXXXXXXXXXXX",
"X ..... .X",
"X. .... ..X",
"X.. .. ...X",
"X.... .....X",
"X.... .....X",
"X.. .. ...X",
"X. .... ..X",
"X ..... .X",
"X.....X",
"XXXXXXXXXXXXXXXX"
};
```

- a) Übertrage den Inhalt dieser Datei in Dein Heft (1 Kästchen für 1 Pixel) und färbe den entsprechenden Text.

## Lösung

```
"XXXXXXXXXXXXXXXX",
"X ..... .X",
"X. .... ..X",
"X.. .. ...X",
"X.... .....X",
"X.... .....X",
"X.. .. ...X",
"X. .... ..X",
"X ..... .X",
"X.....X",
"XXXXXXXXXXXXXXXX"
};
```

(Anmerkung: Das weisse Kreuz muss gelb sein!! ;-))

- b) • Eine rote Linie soll waagrecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen.



- Eine grüne Linie soll senkrecht von ganz oben nach ganz unten genau in der Mitte der Leinwand verlaufen.
- Die beiden Diagonalen sollen jetzt zweifarbig sein. Beide sollen in der unteren Hälfte die Farbe **schwarz** haben.

Schreibe den korrekten Inhalt der xpm-Datei erneut hin. Achte auf notwendige Veränderungen im Kopf.

## Lösung

```
/* XPM */
static char *kreuz_klein[] = {
/* columns rows colors chars-per-pixel */
"13 11 4 1",
"o c green",
"b c black",
"  c yellow",
". c red",
"X c gray",
/* pixels */
"XXXXXXXXXXXXXXXX",
"X  ...o..  .X",
"X.  ..o.  ..X",
"X..  .o   ...X",
"X.... o.....X",
"X.....o.....X",
"X..bb.obb...X",
"X.bb..o.bb..X",
"Xbb...o..bb.X",
"X.....o.....X",
"XXXXXXXXXXXXXXXX"
};
```

c) Die Ursprungsdatei wurde aus einem Tcl/Tk-Script erzeugt. Wie musste es lauten?

## Lösung

```
canvas .leinwand -width 11 -height 9 -background red
pack .leinwand

.leinwand create line 0 0 11 9 -fill yellow
.leinwand create line 11 0 0 9 -fill yellow
```

d) Wie lautet das Script in Tcl/Tk für das veränderte Bild?

## Lösung

```
canvas .leinwand -width 11 -height 9 -background red
pack .leinwand

.leinwand create line 0 0 6 5 -fill yellow
.leinwand create line 6 5 11 9 -fill black
.leinwand create line 11 0 6 5 -fill yellow
.leinwand create line 6 5 0 9 -fill black

.leinwand create line 0 5 11 5 -fill red
.leinwand create line 6 0 6 9 -fill green
```

Bei diesen Aufgaben ist unser Blatt, welches die Leinwand darstellt, 200 Pixel breit und 130 Pixel hoch.

Das Blatt sollte daher quer vor euch liegen.

Schreibe die Bezeichnungen der Variablen und der Dateien über die Behälter.

---

## Aufgabe 1

- Benenne deine drei Blätter mit `externer Speicher`, `Arbeitsspeicher` und `Leinwand`.
- Trage die Koordinaten der Ecken deiner Leinwand in diese ein.

## Aufgabe 2

- Führe, wie gerade an der Tafel erklärt, folgende Befehle nacheinander aus:

```
set sonne [image create photo -file sonne.gif]
```

```
.leinwand create image 10 10 -image $sonne
```

- Übertrage die Befehle in dein Heft und zeichne eine verkleinerte Abbildung der drei Blätter in dein Heft.

## Aufgabe 3

- Führe folgende Befehle nacheinander aus:

```
set wolke [image create photo -file wolke.gif]
```

```
.leinwand create image 100 50 -image $wolke
```

- Übertrage die Befehle in dein Heft und zeichne eine verkleinerte Abbildung der drei Blätter in dein Heft.

## Aufgabe 4

- Führe folgende Befehle nacheinander aus:

```
set wetter [image create photo -file wolke.gif]
```

```
.leinwand create image 150 40 -image $wetter
```

b) Führe folgenden Befehl nacheinander aus:

```
.leinwand create image 30 50 -image $wetter
```

c) Übertrage die Befehle in dein Heft und zeichne eine verkleinerte Abbildung der drei Blätter in dein Heft.

## Aufgabe 5

a) Führe folgende Befehle nacheinander aus:

```
set baum [image create photo -file baum.gif]
```

```
.leinwand create image 80 80 -image $baum
```

b) Führe folgenden Befehl nacheinander aus:

```
.leinwand create image 20 80 -image $baum
```

c) Übertrage die Befehle in dein Heft und zeichne eine verkleinerte Abbildung der drei Blätter in dein Heft.

Wir arbeiten bei diesen Aufgaben mit folgender Leinwand:  
`canvas .leinwand -width 300 -height 500 -background yellow`  
`pack .leinwand`

---

## Aufgabe 1

- a) Auf dem Fileserver liegt eine Datei mit einer Pixelgrafik im `gif` - Format.  
Kopiere diese Datei `auto.gif` vom Verzeichnis `F` des Servers in Dein Verzeichnis:

```
cp F/material/stadt/auto.gif auto.gif
```

- b) Erstelle mit `nedit auto1.tcl` die Datei `auto1.tcl` .  
c) Übertrage nun folgende Textzeilen in diese Datei:

```
set bild [image create photo -file auto.gif]

canvas .leinwand -width 300 -height 500 -background yellow
pack .leinwand

.leinwand create image 70 220 -image $bild
```

- d) Führe diese Datei mit `wish auto1.tcl` aus.

## Aufgabe 2

- a) Notiere in Dein Heft, welche Anweisungen neu für Dich sind.  
b) Überlege, was diese Anweisungen bewirken. Schreibe Deine Überlegungen in dein Heft unter die zugehörigen Anweisungen.  
c) Was bedeuten die Wörter und Zahlen in den Anweisungen? Schreibe Deine Vermutungen in Dein Heft.

## Aufgabe 3

- a) Kopiere die Datei `auto1.tcl` als `auto2.tcl` :  
`cp auto1.tcl auto2.tcl`  
b) Wir wollen, dass unser Auto auf einer Strasse fährt. Öffne dazu `auto2.tcl` im Editor mit `nedit auto2.tcl` . Füge eine schwarze Linie als Strasse genau unter das Auto hinzu.

**Aufgabe 1**

- a) Kopiere die Datei `auto2.tcl` vom Fileserver F in Dein Verzeichnis:  
`cp F/material/stadt/auto2.tcl auto2.tcl`
- b) Führe die Datei mit `source auto2.tcl` aus.
- c) Führe die Anweisung `destroy .leinwand` aus.
- d) Öffne die Datei `auto2.tcl` mit dem Editor `nedit auto3.tcl`
- e) Verändere die Position des Autos auf der Straße, sodass das Auto am rechten Bildrand steht. Dies kannst Du durch die Veränderung der Koordinaten in der folgenden Zeile erreichen:

```
.leinwand create image 50 222 -image $bild
```

- f) Führe die Datei mit `source auto2.tcl` aus und überprüfe, ob das Auto richtig platziert wurde.  
*(Wenn das Auto noch nicht richtig dargestellt wird oder noch Fehler in der TCL-Datei vorhanden sind, kannst Du Mit der Anweisung `destroy .leinwand` die Leinwand zurücksetzen. Dann kannst du die TCL-Datei mit dem Editor bearbeiten und anschließend mit `source auto2.tcl` erneut ausführen.)*

**Aufgabe 2**

In dieser Aufgabe wollen wir mehrere Objekte auf die Leinwand zeichnen lassen.

- a) Kopiere zunächst die Datei `auto2.tcl` als `auto3.tcl`:  
`cp F/material/stadt/auto2.tcl auto3.tcl`
- b) Öffne die Datei `auto3.tcl` mit dem Editor `nedit auto3.tcl`.
- c) Mit der Anweisung

```
set baum [image create photo -file F/material/stadt/baum.gif]
```

wird in der Variable `baum` das Bild von einem Baum gespeichert. Mit der Anweisung

```
.leinwand create image X Y -image $baum
```

wird der Baum an der Position (X/Y) platziert. Dieser Baum soll unten links in der Ecke platziert werden.

- d) Mit der Anweisung

```
set haus [image create photo -file F/material/stadt/haus.gif]
```

wird in der Variable `haus` das Bild von einem Haus gespeichert. Mit der Anweisung

```
.leinwand create image X Y -image $haus
```

wird das Haus an der Position (X/Y) platziert. Dieses Haus soll direkt an der Straße, am linken Bildrand platziert werden.

e) Mit der Anweisung

```
set sonne [image create photo -file F/material/stadt/sonne.gif]
```

wird in der Variable *sonne* das Bild von einer Sonne gespeichert. Mit der Anweisung

```
.leinwand create image X Y -image $sonne
```

wird die Sonne an der Position (X/Y) platziert. Die Sonne soll oben rechts in der Ecke platziert werden.

f) Führe die Datei mit `source auto3.tcl` aus und überprüfe, ob alle Elemente richtig platziert wurden.

### Aufgabe 3

Die Leinwand aus Aufgabe 2 ist ziemlich leer, daher sollst Du weitere Elemente auf die Leinwand platzieren.

a) Kopiere zunächst die Datei *auto3.tcl* als *auto4.tcl*:

```
cp auto3.tcl auto4.tcl
```

b) Öffne die Datei *auto4.tcl* mit dem Editor `nedit auto4.tcl`.

c) Erstelle im unteren Teil der Leinwand, unterhalb der Straße einen kleinen Wald, der aus mindestens 5 Bäumen besteht.

*(Ganz heisser Tipp:*

*Erstelle zuerst eine Zeichnung in Deinem Heft! )*

d) Platziere ein weiteres Haus in der Nähe der Straße.

e) Führe die Datei mit `source auto4.tcl` aus und überprüfe, ob alle Elemente richtig platziert wurden.

## Aufgabe 1

Auf einer Leinwand der Grösse 400x500 Pixel soll folgendes Bild erzeugt werden.

Als „Bausteine“ liegen auf dem Server im Verzeichnis F/material/stadt/ folgende Dateien:

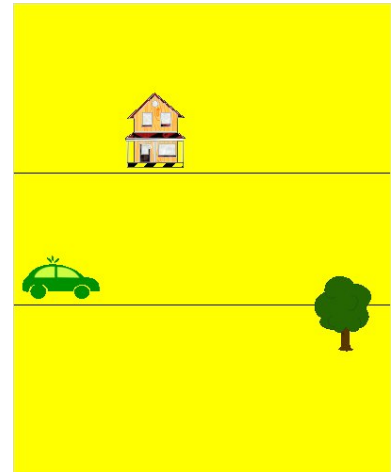
auto.gif, haus.gif, autoflip.gif,  
baum.gif,  
mehrere\_wolken.gif, sonne.gif,  
tanne.gif, wolke.gif, villa.gif

*Zur Erinnerung: Die Datei baum.gif übertragen wir in eine Variable mit dem Name baum mit der Anweisung:*

```
set baum [image create photo -file  
Bilder/baum.gif]
```

*Aus der Variablen auto übertragen wir eine Kopie auf die Leinwand auf beispielhaft Position (200/300) mit*

```
.leinwand create image 200 300 -image  
$auto
```

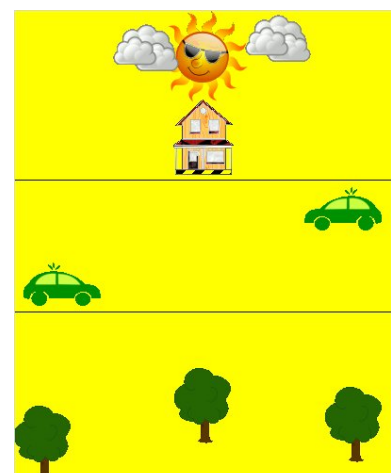


## Aufgabe 2

Erweitere bzw. verändere Dein Bild entsprechend folgender Abbildung rechts.

## Aufgabe 3

Füge Deinem Bild noch weitere Elemente nach eigenen Vorstellungen hinzu. Du solltest noch mindestens vier weitere Elemente einbauen.





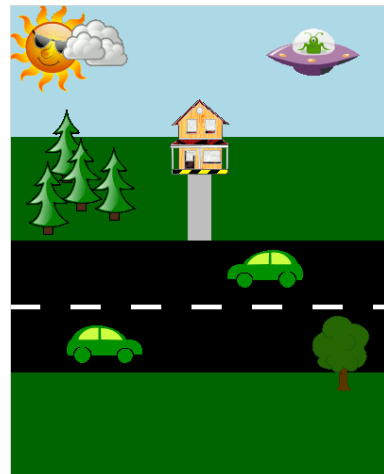
## Zusatzaufgabe

Tcl/Tk hat noch viel mehr Anweisungen und Möglichkeiten, als wir bisher kennengelernt haben. Falls du mit den anderen Aufgaben schon fertig bist, kannst du in dieser Zusatzaufgabe noch neue Sachen entdecken, die dir mehr Gestaltungsmöglichkeiten für deine Bilder geben.

Bei dem nebenstehenden Bild wurde die Linienanweisung von Tcl/Tk um einige zusätzliche Attribute ergänzt. Die Strasse, der Himmel und der Weg vom Haus zur Strasse wurden mit Linien gezeichnet, die jedoch dicker als normal gezeichnet sind.

Ein paar zusätzliche Hinweise:

- Die Farbe des Himmels ist „lightblue“.
- Die Farbe der Leinwand ist „darkgreen“.
- Das Ufo befindet sich in der Datei „ufo.gif“.
- Wenn sich Bilder auf der Leinwand überlappen, müssen die hinteren Bilder zuerst erstellt werden (z.B. bei dem Tannenwäldchen).



Die Anweisung für die Strassenmarkierung lautet wie folgt:

```
.leinwand create line 0 320 400 320 -fill white -width 5 -dash {32 32}
```

Das Attribut **-width 5** kennst du schon von der Erstellung einer Leinwand. Wenn man es bei einer Linie angibt, dann wird diese nicht mehr so fein gezeichnet wie normal, sondern in dem angegebenen Fall 5 Pixel breit.

Das Attribut **-dash {32 32}** sorgt dafür, dass die Linie nicht in einem Zug durchgezeichnet, sondern „gestrichelt“ dargestellt wird. Konkret bedeutet die erste Zahl die Länge der einzelnen Liniensegmente, die zweite Zahl gibt den Abstand zwischen den Liniensegmenten an.

Versuche das Bild in Tcl/Tk nachzubilden.

## 11.5 Teste

<b>Name:</b>	<b>Vorname:</b>

### **1. Test      WP-Kurs NW-If/Ph Jg.6      26.11.2010/psa**

---

#### **Aufgabe 1: *Buttons* und *Labels***

a) Folgende Kommandos werden in der Konsole eingegeben:

```
label .info1 -text "Ganz aktuell:"  
pack .info1  
button .knopf1 -text "Drueck mich" -command opera  
pack .knopf1
```

Zeichne, was Du im Grafikfenster siehst!

Was kann man dort machen und was passiert dann?

b) Du möchtest einen weiteren Knopf haben, der den Editor `nedit` startet. Was musst Du eingeben.

#### **Aufgabe 2: Kommandos**

a) Beschreibe mit Deinen eigenen Worten, was bei der Eingabe der folgenden Kommandos (Werkzeuge) vom Computer gemacht wird.

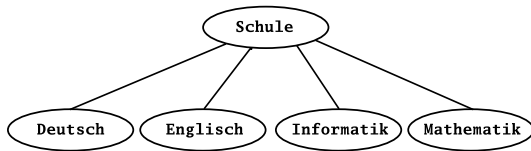
1. `mkdir kalender`
2. `cd kalender`
3. `mkdir januar`
4. `mkdir februar`
5. ...
6. `mkdir dezember`
7. `cd dezember`
8. `nedit wunschzettel`
9. `cd ..`
10. `tree`

b) Du befindest Dich im Verzeichnis, von dem Du zuerst `mkdir kalender` ausgeführt hast.

Du gibst nun `tree` ein. Welche Ausgabe erhältst Du?

c) Stelle diesen Dateibaum wie in der Aufgabe 3) dar!

### Aufgabe 3: Dateibäume



- a) Du möchtest diesen Dateibaum in der Konsole erzeugen. Schreibe die dazu notwendige Befehlsfolge auf.
- b) Wie sieht die Ausgabe aus, wenn Du jetzt das Kommando `tree` eingibst?
- c) Was musst Du eingeben, wenn Du im Verzeichnis `Deutsch` noch Unterverzeichnisse für `Januar` und `Februar` anlegen möchtest.

---

Aufgabe 1 Punkte	
Aufgabe 2 Punkte	
Aufgabe 3 Punkte	
Summe Punkte	

# Lösung 1. Test WP-Kurs If/Ph Jg.6 26.11.2010/psa

## Aufgabe 1: *Buttons* und *Labels*

a) Folgende Kommandos werden in der Konsole eingegeben:

```
label .info1 -text "Ganz aktuell:"  
pack .info1  
button .knopf1 -text "Drueck mich" -command opera  
pack .knopf1
```

*Zeichne, was Du im Grafikfenster siehst!*

*Was kann man dort machen und was passiert dann?*

b) Du möchtest einen weiteren Knopf haben, der den Editor `nedit` startet. Was musst Du eingeben.

## Lösung

zu a)

Wir erhalten beim Ausführen dieser Zeilen folgende Ausgabe im Grafikfenster:



Der Text *Ganz aktuell* steht auf einem *Label*, das sich vom Untergrund nicht abhebt, der Text *Drueck mich* auf einem *Button*. Beim Drücken dieses Knopfes wird das Programm *Opera* gestartet.

zu b)

Mit folgenden Zeilen wird ein weiterer Knopf für das Programm *nedit* eingerichtet.

```
button .knopf2 -text "Nedit starten" -command nedit  
pack .knopf2
```

## Aufgabe 2: Kommandos

a) Beschreibe mit Deinen eigenen Worten, was bei der Eingabe der folgenden Kommandos (Werkzeuge) vom Computer gemacht wird.

1. `mkdir kalender`
2. `cd kalender`
3. `mkdir januar`
4. `mkdir februar`
5. ...
6. `mkdir dezember`
7. `cd dezember`
8. `nedit wunschzettel`
9. `cd ..`

10. `tree`

b) Du befindest Dich im Verzeichnis, von dem Du zuerst `mkdir kalender` ausgeführt hast.

Du gibst nun `tree` ein. Welche Ausgabe erhältst Du?

c) Stelle diesen Dateibaum wie in der Aufgabe 3) dar!

## Lösung

zu a)

1. `mkdir kalender`

Es wird ein Verzeichnis mit dem Namen *kalender* eingerichtet.

2. `cd kalender`

Es wird in das Verzeichnis gewechselt.

3. `mkdir januar`

4. `mkdir februar`

5. ...

6. `mkdir dezember`

In diesem Verzeichnis *kalender* werden für die Monate Januar bis Dezember Unterverzeichnisse eingerichtet.

7. `cd dezember`

Es wird in das Verzeichnis *dezember* gewechselt.

8. `nedit wunschzettel`

Der Editor *nedit* wird gestartet und der Benutzer kann einen Text eingeben und speichern.

9. `cd ..`

Es wird eine Verzeichnisebene nach oben gewechselt.

10. `tree`

Die Verzeichnisstruktur wird ausgegeben.

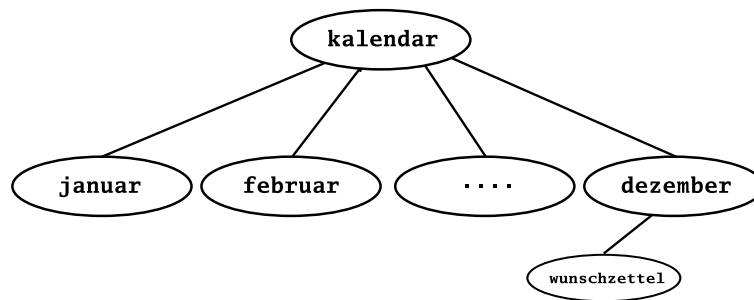
zu b)

Wir erhalten folgende Ausgabe:

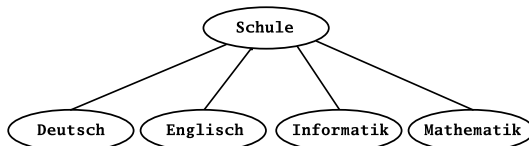
```
.
|-- kalender
    |-- dezember
        |-- wunschzettel
    |-- februar
    |-- januar
```

zu 2c)

Wir erhalten also folgenden Baum:



### Aufgabe 3: Dateibäume



- Du möchtest diesen Dateibaum in der Konsole erzeugen. Schreibe die dazu notwendige Befehlsfolge auf.
- Wie sieht die Ausgabe aus, wenn Du jetzt das Kommando `tree` eingibst?
- Was musst Du eingeben, wenn Du im Verzeichnis `Deutsch` noch Unterverzeichnisse für `Januar` und `Februar` anlegen möchtest.

### Lösung

zu a)

Wir müssen folgende Kommandos eingeben:

```

mkdir Schule
cd Schule
mkdir Deutsch
mkdir Englisch
mkdir Informatik
mkdir Mathematik
  
```

zu b)

Mit dem Werkzeug `tree` erhalten wir dann folgende Ausgabe:

```

.
|-- Schule
  
```

```
|-- Deutsch  
|-- Englisch  
|-- Informatik  
'-- Mathematik
```

zu c)

Wir müssen ergänzen:

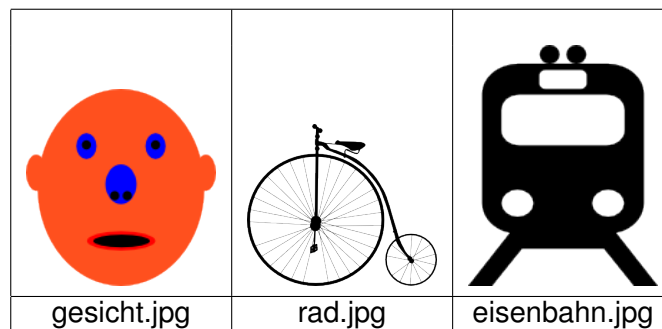
```
cd Deutsch  
mkdir Januar  
mkdir Februar
```

<b>Name:</b>	<b>Vorname:</b>

## 2. Test      WP-Kurs NW-If/Ph Jg.6      2.2.2011/psa

### Aufgabe 1: *Webseite von Emil*

Folgende Bilddateien hat Emil erstellt bzw. erhalten:



Diese verwendet er in seiner neuen Webseite mit dem Text:

```
<html>
  <head><title>Webseite von Emil</title></head>
<body>
  <center>
    <h1>Webseite von Emil Saueressig</h1>
    Hier seht ihr meine aktuelle Webseite.<br>
    Zuerst ein Bild von mir:<br>

    <br>

    <h2>Meine Hobbys:<br>
    
    
    <br>
    Radfahren und Eisenbahn.
  </center>
</body>
</html>
```

Wie sieht diese Webseite im Browser aus?  
Erstelle ein Abbild auf einer DIN-A4-Seite.

(Bildquellen: psa: gesicht.jpg \* OpenClipart-Bibliothek: rad.jpg, eisenbahn.jpg)



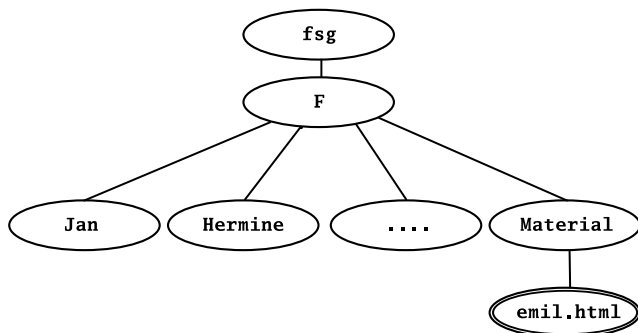
## Aufgabe 2: HTML

Marvin hat im Unterricht nicht gut aufgepasst und anschliessend schlecht abgeschrieben:

```
{html}
<hut><titel>Hausaufgaben</titel></hut>
<buddy>
  <Zentrum>
  <h1>Hausaufgaben</d1>
  Zu Mittwoch haben wir keine Hausaufgaben auf.
</head>
</htm>
```

Schreibe diese HTML-Datei ohne Fehler neu auf.

## Aufgabe 3: Dateibäume



Jan möchte aus dem Verzeichnis `Material` die Datei `emil.html` in sein Verzeichnis als `jan.html` kopieren. Er befindet sich im Verzeichnis `fsg`.

In einem Heft steht eine alte Vorlage. Diese war aber ziemlich hingeschmiert und er kann nur Folgendes entziffern:

(Die mit XXX, YYY, ZZZ dargestellten Zeichen sind nicht mehr lesbar.)

```
cd F
cd XXX
cp ../Material/YYYY ZZZ
```

- Wie muss die korrekte Folge an Anweisungen aussehen?
- Wie sieht der Dateibaum danach aus? Erstelle eine Skizze wie oben dargestellt!
- Wie sieht die Ausgabe aus, wenn Du dann im Verzeichnis `fsg` das Kommando `tree` eingibst?
- Auch in das Verzeichnis von `Hermine` soll diese Datei mit dem Namen `hermine.html` kopiert werden. Wie lautet eine mögliche Anweisungsfolge?

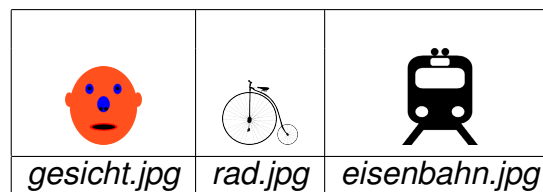
Aufgabe 1 Punkte	
Aufgabe 2 Punkte	
Aufgabe 3 Punkte	
Summe Punkte	

# Lösung zum

## 2. Test WP-Kurs NW-If/Ph Jg.6 2.2.2011/psa

### Aufgabe 1: Webseite von Emil

Folgende Bilddateien hat Emil erstellt bzw. erhalten:



Diese verwendet er in seiner neuen Webseite mit dem Text:

```
<html>
  <head><title>Webseite von Emil</title></head>
  <body>
    <center>
      <h1>Webseite von Emil Saueressig</h1>
      Hier seht ihr meine aktuelle Webseite.<br>
      Zuerst ein Bild von mir:<br>

      <br>

      <h2>Meine Hobbys:<br>
      
      
      <br>
      Radfahren und Eisenbahn.
    </center>
  </body>
</html>
```

Wie sieht diese Webseite im Browser aus? Erstelle ein Abbild auf einer DIN-A4-Seite.

### Lösung

Wir erhalten folgende Darstellung in einem Browser wie beispielsweise *Firefox*, *Safari*, *Internet-Explorer* oder unserem *Opera*:

[Webseite von Emil - Opera](#)

(steht in der blauen Zeile)

## Webseite von Emil Saueressig

Hier seht ihr meine aktuelle Webseite.  
Zuerst ein Bild von mir:



**Meine Hobbys:**



**Radfahren und Eisenbahn.**

### Aufgabe 2: HTML

Marvin hat im Unterricht nicht gut aufgepasst und anschliessend schlecht geschrieben:

```
{html}
<hut><titel>Hausaufgaben</titel><hut>
<buddy>
  <Zentrum>
    <h1>Hausaufgaben</h1>
    Zu Mittwoch haben wir keine Hausaufgaben auf.
  </head>
</htm>
```

Schreibe diese HTML-Datei ohne Fehler neu auf.

### Lösung

Der HTML-Text lautet folgendermassen korrekt:

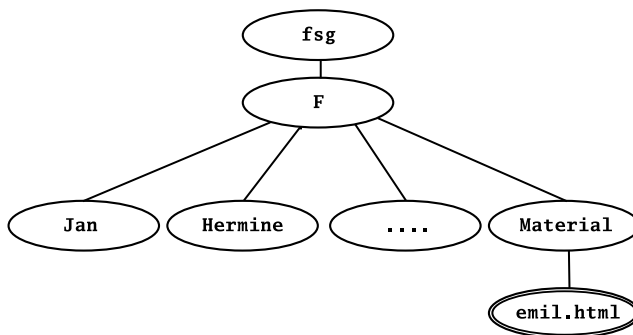
```
<html>
  <head><title>Hausaufgaben</title></head>
  <body>
    <center>
```

```

<h1>Hausaufgaben</h1>
  Zu Mittwoch haben wir keine Hausaufgaben auf.
</center>
</body>
</html>

```

### Aufgabe 3: Dateibäume



Jan möchte aus dem Verzeichnis Material die Datei emil.html in sein Verzeichnis als jan.html kopieren. Er befindet sich im Verzeichnis fsg.  
 In einem Heft steht eine alte Vorlage. Diese war aber ziemlich hingeschmiert und er kann nur Folgendes entziffern:  
 (Die mit XXX, YYY, ZZZ dargestellten Zeichen sind nicht mehr lesbar.)

```

cd F
cd XXX
cp ../Material/YYY ZZZ

```

- Wie muss die korrekte Folge an Anweisungen aussehen?
- Wie sieht der Dateibaum danach aus? Erstelle eine Skizze wie oben dargestellt!
- Wie sieht die Ausgabe aus, wenn Du dann im Verzeichnis fsg das Kommando tree eingibst?
- Auch in das Verzeichnis von Hermine soll diese Datei mit dem Namen hermine.html kopiert werden. Wie lautet eine mögliche Anweisungsfolge?

### Lösung

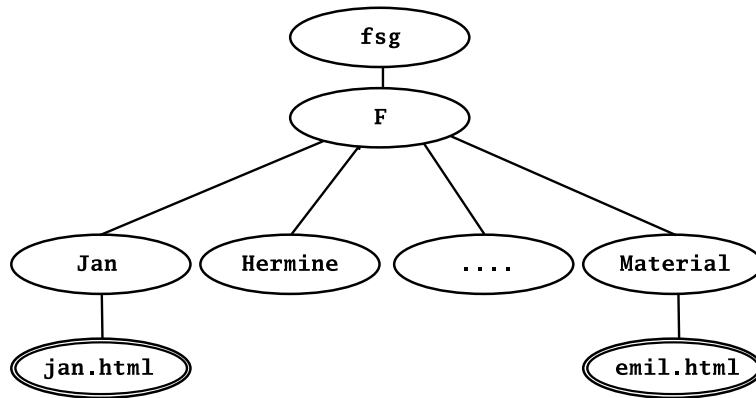
zu a) Die Anweisungsfolge muss lauten:

```

cd F
cd Jan
cp ../Material/emil.html jan.html

```

zu b) Der Baum sieht dann folgendermassen aus:



zu c) Wir erhalten folgende Ausgabe bei der Eingabe des Kommandos `tree` im Verzeichnis

```

fsg.
fsg
|-- F
    |-- Hermine
    |-- Jan
    `-- Material
        `-- emil.html
  
```

zu d) Wir nehmen an, dass wir uns wieder im Verzeichnis `fsg` befinden. Dann kann eine Anweisungsfolge folgendermassen lauten:

```

cd F
cd Hermine
cp ../Material/emil.html jan.html
  
```

<b>Name:</b>	<b>Vorname:</b>

### 3. Test      WP-Kurs NW-If/Ph Jg.6      8.4.2011/psa

---

Bei allen Aufgaben mit einer Leinwand gilt, dass diese Leinwand mit folgenden Anweisungen erzeugt werden soll:

```
canvas .leinwand -width 350 -height 500 -background yellow
pack .leinwand
```

Bevor ihr die Lösung zu einer Aufgabe in *TCL/TK* schreibt, erstellt immer zuerst eine Zeichnung auf Papier!

**Wichtige zusätzliche Hinweise:**

Das Benutzen eines *Kugelschreibers* ist *verboten!*      **Schreibt leserlich!**  
*Skizzen* werden mit *Blei- und Buntstiften* erzeugt.  
 Benutze ein *Lineal* oder ein *Geodreieck* für das *Zeichnen* von Strecken.

Eine Raute auf dem Papier hat eine Breite von 25 Pixeln und eine Höhe von 50 Pixeln.

---

### Aufgabe 1: Ein Bild auf der Leinwand

Folgende Anweisungen werden in der TCL-Konsole eingegeben:

```
canvas .leinwand -width 350 -height 500 -background yellow
pack .leinwand

.leinwand create line 50 100 300 100 -fill blue
.leinwand create line 50 400 300 400 -fill blue
.leinwand create line 50 100 50 400 -fill red
.leinwand create line 300 100 300 400 -fill red

.leinwand create line 50 100 300 400 -fill pink
.leinwand create line 300 100 50 400 -fill pink

.leinwand create line 175 100 175 400 -fill black
.leinwand create line 50 250 300 250 -fill black
```

- a) Erstelle eine Zeichnung! (*Wähle wie üblich 1 cm für 50 Pixel*)
- b) Oberhalb des „Gebildes“ soll noch ein *Dreieck* in der Farbe *orange* aufgesetzt werden.  
 Trage dieses Dreieck in deine Zeichnung ein.  
 Welche Anweisungen müssen dafür hinzugefügt werden?

## Aufgabe 2: Strecken

Auf einer Leinwand soll eine Strecke gezeichnet werden, die von ganz rechts nach ganz links genau in der Mitte der Leinwand verläuft. Zusätzlich soll auf der Leinwand eine zweite Strecke erscheinen, die von ganz oben nach ganz unten ebenfalls genau in der Mitte der Leinwand verläuft.

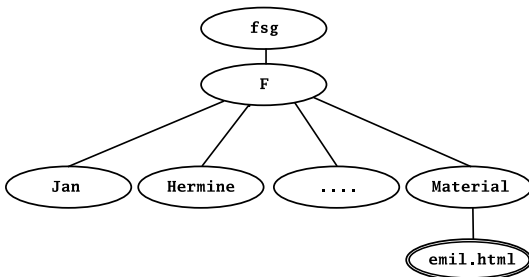
- Erstelle eine Zeichnung!
- Wie lauten die zugehörigen Anweisungen in *TCL/TK*?

## Aufgabe 3: Tisch und Stuhl

Auf einer Leinwand sollen ein Tisch und ein Stuhl – jeweils aus drei Strecken konstruiert – gezeichnet werden.

- Erstelle eine Zeichnung!
- Wie lauten die zugehörigen Anweisungen in *TCL/TK*?

## Aufgabe 4: Dateibäume



Jan möchte aus dem Verzeichnis *Material* die Datei *emil.html* in sein Verzeichnis kopieren. Er befindet sich im Verzeichnis *fsg*.

In einem Heft steht eine alte Vorlage. Diese war aber ziemlich hingeschmiert und er kann nur Folgendes entziffern:

```
cd F
cd XXXXXXXX
cp ../Material/YYYYYYY zzzzzzzzzz
```

- Wie muss die korrekte Folge an Anweisungen aussehen?
- Wie sieht der Dateibaum danach aus? Erstelle eine Skizze!

---

Aufgabe 1 Punkte	
Aufgabe 2 Punkte	
Aufgabe 3 Punkte	
Aufgabe 4 Punkte	
Summe Punkte	

# Lösung zum

## 3. Test

## WP-Kurs NW-If/Ph Jg.6

8.4.2011/psa

---

### Aufgabe 1: Ein Bild auf der Leinwand

Folgende Anweisungen werden in der TCL-Konsole eingegeben:

```
canvas .leinwand -width 350 -height 500 -background yellow
pack .leinwand

.leinwand create line 50 100 300 100 -fill blue
.leinwand create line 50 400 300 400 -fill blue
.leinwand create line 50 100 50 400 -fill red
.leinwand create line 300 100 300 400 -fill red

.leinwand create line 50 100 300 400 -fill pink
.leinwand create line 300 100 50 400 -fill pink

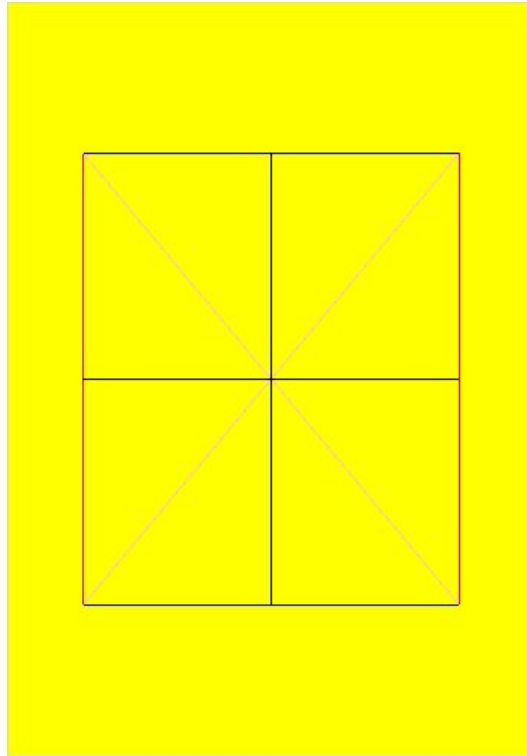
.leinwand create line 175 100 175 400 -fill black
.leinwand create line 50 250 300 250 -fill black
```

- Erstelle eine Zeichnung! (Wähle wie üblich 1 cm für 50 Pixel)
- Oberhalb des „Gebildes“ soll noch ein Dreieck in der Farbe orange aufgesetzt werden.  
Trage dieses Dreieck in deine Zeichnung ein.  
Welche Anweisungen müssen dafür hinzugefügt werden?

### Lösung

zu a) Wir erhalten folgende Darstellung:

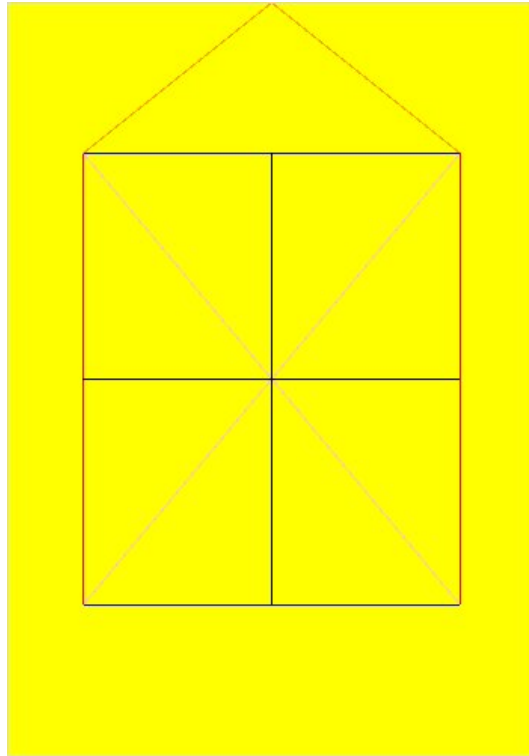




zu b) Wir müssen noch folgende zwei Zeilen hinzufügen, um das Dach zu zeichnen:

```
.leinwand create line 50 100 175 0 -fill orange  
.leinwand create line 175 0 300 100 -fill orange
```

Dann erhalten wir folgende Darstellung:



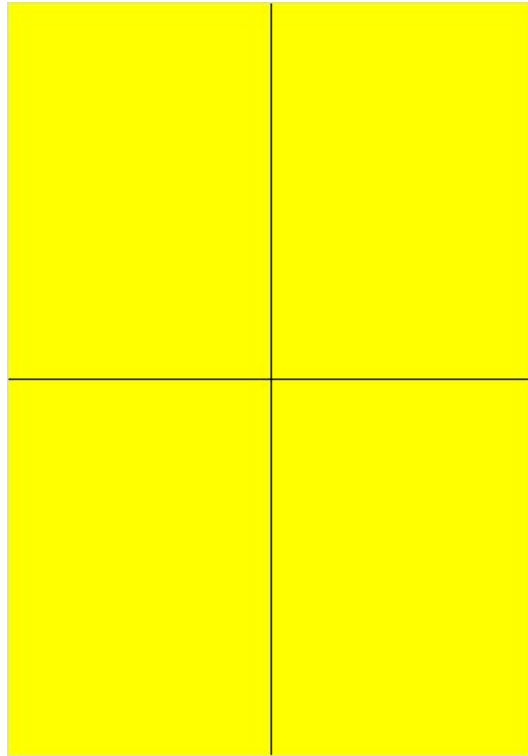
## Aufgabe 2: Strecken

*Auf einer Leinwand soll eine Strecke gezeichnet werden, die von ganz rechts nach ganz links genau in der Mitte der Leinwand verläuft. Zusätzlich soll auf der Leinwand eine zweite Strecke erscheinen, die von ganz oben nach ganz unten ebenfalls genau in der Mitte der Leinwand verläuft.*

- a) *Erstelle eine Zeichnung!*
- b) *Wie lauten die zugehörigen Anweisungen in TCL/TK?*

## Lösung

- a) Wir möchten folgende Zeichnung mit *TCL/TK* erstellen:



b) Die zugehörigen Anweisungen in *TCL/TK* lauten:

```
canvas .leinwand -width 350 -height 500 -background yellow
pack .leinwand

.leinwand create line 175 0 175 500 -fill black
.leinwand create line 0 250 350 250 -fill black
```

### Aufgabe 3: Tisch und Stuhl

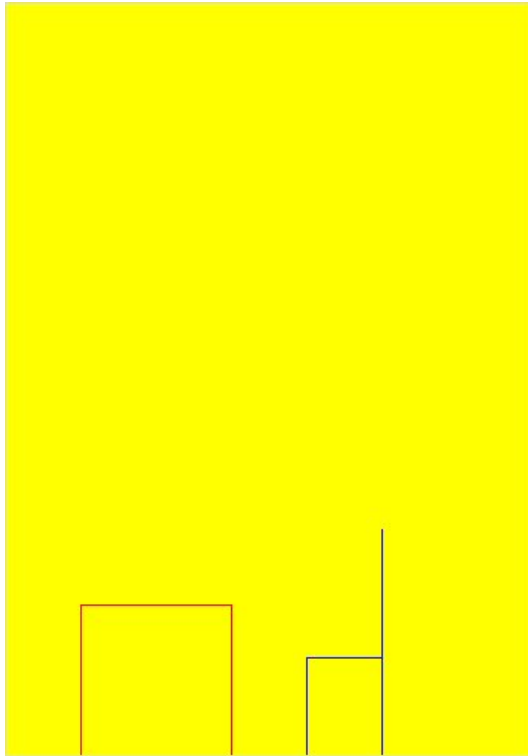
*Auf einer Leinwand sollen ein Tisch und ein Stuhl – jeweils aus drei Strecken konstruiert – gezeichnet werden.*

a) *Erstelle eine Zeichnung!*

b) *Wie lauten die zugehörigen Anweisungen in *TCL/TK*?*

### Lösung

a) Wir können Tisch und Stuhl folgendermassen zeichnen:



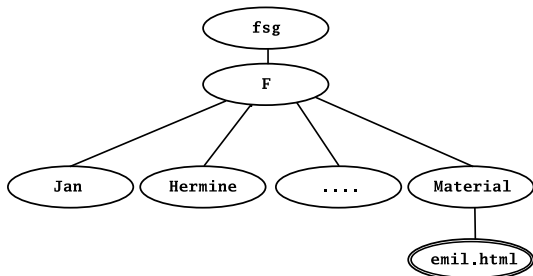
b) Die zugehörigen Anweisungen in *TCL/TK* lauten:

```
canvas .leinwand -width 350 -height 500 -background yellow
pack .leinwand

.leinwand create line 50 500 50 400 -fill red
.leinwand create line 50 400 150 400 -fill red
.leinwand create line 150 500 150 400 -fill red

.leinwand create line 200 500 200 435 -fill blue
.leinwand create line 200 435 250 435 -fill blue
.leinwand create line 250 500 250 350 -fill blue
```

## Aufgabe 4: Dateibäume



Jan möchte aus dem Verzeichnis Material die Datei emil.html in sein Verzeichnis kopieren. Er befindet sich im Verzeichnis fsg .

In einem Heft steht eine alte Vorlage. Diese war aber ziemlich hingeschmiert und er kann nur Folgendes entziffern:

```
cd F
cd XXXXXXXX
cp ../Material/YYYYYYY zzzzzzzzzz
```

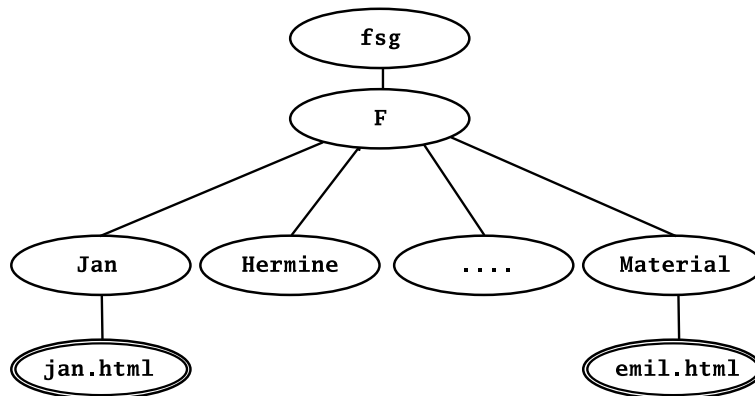
- Wie muss die korrekte Folge an Anweisungen aussehen?
- Wie sieht der Dateibaum danach aus? Erstelle eine Skizze!

## Lösung

zu a) Die Anweisungsfolge muss lauten:

```
cd F
cd Jan
cp ../Material/emil.html jan.html
```

zu b) Der Baum sieht dann folgendermassen aus:



<u>Name:</u>	<u>Vorname:</u>

## 4. Test      WP-Kurs NW-If/Ph Jg.6      10.6.2011/psa

---

### Wichtige Hinweise:

Das Benutzen eines *Kugelschreibers* ist *verboten!*      **Schreibt leserlich!**  
*Skizzen und Zeichnungen* werden mit *Blei- und Buntstiften* erzeugt.  
 Benutze ein *Lineal* oder ein *Geodreieck* für das *Zeichnen* von Strecken.

---

### Aufgabe 1:    Aufbau eines XPM Bildes

Eine Datei *bild.xpm* sieht wie folgt aus:

```

/* XPM */
static char *bild[] = {
/* columns rows colors chars-per-pixel */
"11 11 4 1",
"- c red",
". c black",
"X c blue",
"o c green",
/* pixels */
"-----",
"---X.X---",
"---XX.XX---",
"--XXX.XXX--",
"-XXXoooXXX-",

"...ooo...",
"-XXXoooXXX-",
"---XX.XX---",
"---XX.XX---",
"---X.X---",

"-----",
};

```

- a) Wie groß ist die Leinwand des XPM-Bildes?
- b) Übertrage die Tabelle auf Dein Blatt und schreibe die Farb- und Zeichenzuordnung (auf deutsch) hinein:

Farbe	Zeichen

### Aufgabe 2:    Von Tcl/Tk zu XPM

Folgende Anweisungen werden in der TCL-Konsole eingegeben:

```

canvas .leinwand -width 7 -height 5 -background yellow
pack .leinwand

.leinwand create line 0 3 7 3 -fill blue
.leinwand create line 4 0 4 5 -fill red

```

- a) Wie gross ist die erstellte Leinwand?



- Eine **rote** Linie soll senkrecht von ganz oben nach ganz unten genau in der Mitte des Bildes verlaufen.
- Eine **grüne** soll Linie waagrecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen.
- Eine Diagonale soll mit **schwarzer** Farbe gezeichnet werden.

#### **Aufgabe 4: Tisch und Stuhl**

Auf einer Leinwand sollen ein Tisch und ein Stuhl – jeweils aus drei Strecken konstruiert – gezeichnet werden.

a) Erstelle eine Zeichnung!

b) Wie lauten die zugehörigen Anweisungen in *TCL/TK*?

Aufgabe 1 Punkte	
Aufgabe 2 Punkte	
Aufgabe 3 Punkte	
Aufgabe 4 Punkte	
Summe Punkte	



# Lösung zum

## 4. Test      WP-Kurs NW-If/Ph Jg.6      10.6.2011/psa

---

### Aufgabe 1:    Aufbau eines XPM Bildes

Eine Datei bild.xpm sieht wie folgt aus:

```
/* XPM */
static char *bild[] = {
/* columns rows colors chars-per-pixel */
"11 11 4 1",
"- c red",
". c black",
"X c blue",
"o c green",
/* pixels */
"-----",
"---X.X---",
"---XX.XX---",
"---XXX.XXX---",
"---XXXoooXXX-",

"...ooo...",
"---XXXoooXXX-",
"---XX.XX---",
"---XX.XX---",
"---X.X---",
"-----",
};
```

- a) Wie groß ist die Leinwand des XPM-Bildes?
- b) Übertrage die Tabelle auf Dein Blatt und schreibe die Farb- und Zeichenzuordnung (auf deutsch) hinein.

### Lösung

Die Leinwand hat einen Rand von insgesamt 2 Pixeln und einen Bildbereich von 9 Pixeln in der Höhe und Breite, also insgesamt 11x11 Pixel.

Folgende Farbzuzuordnung gilt:

Farbe	Zeichen
rot	-
schwarz	.
blau	X
grün	o

---

### Aufgabe 2:    Von Tcl/Tk zu XPM

Folgende Anweisungen werden in der TCL-Konsole eingegeben:

```

canvas .leinwand -width 7 -height 5 -background yellow
pack .leinwand

.leinwand create line 0 3 7 3 -fill blue
.leinwand create line 4 0 4 5 -fill red

```

- Wie gross ist die erstellte Leinwand?
- Was sehen wir auf dieser Leinwand? Erstelle eine Zeichnung! (Wähle 1 Kästchen für 1 Pixel)
- Wie üblich, wurde eine XPM-Datei aus diesem Fenster erzeugt. Diese Datei wird mit einem Editor geöffnet. Kopf und Anfang der Datei sehen aus wie folgt:

```

/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"9 7 4 1",
"u c blue",
". c red",
"X c yellow",
"o c gray",
/* pixels */
"oooooooo",
"oXXX.XXXo",

```

- Schreibe den Inhalt der gesamten Datei ohne Kopf auf Dein Blatt. Färbe Deinen Dateitext.
- Wie wird diese Datei dargestellt, wenn der Kopf folgendermassen aussieht, der Rest der Datei aber unverändert bleibt:

```

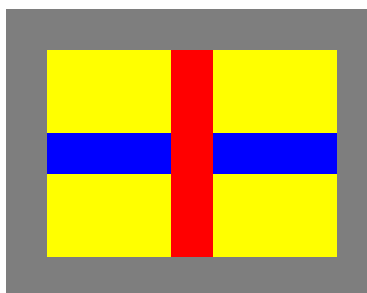
/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"9 7 5 1",
"X c blue",
". c red",
"u c yellow",
"+ c gray",
"o c orange",
/* pixels */

```

Zeichne das entsprechende Bild in Dein Heft!

## Lösung

- Die Leinwand ist 9x7 Pixel gross. Dabei sind insgesamt 2 Pixel Rand.
- Wir sehen folgendes Bild:



- Der vollständige Inhalt der XPM-Datei besteht aus folgenden Zeichen:

```

/* XPM */
static char *auf1[] = {
/* columns rows colors chars-per-pixel */
"9 7 4 1",
"u c blue",
". c red",
"X c yellow",
"o c gray",

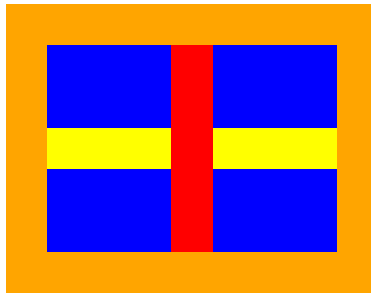
```

```

/* pixels */
"oooooooo",
"oXXX.XXXo",
"oXXX.XXXo",
"ouuu.uuuu",
"oXXX.XXXo",
"oXXX.XXXo",
"oooooooo"
};

```

- d) Die Farben ändern sich: Der Hintergrund wird blau, die Senkrechte bleibt rot, die Waagerechte wird gelb und der Rand orange.  
Damit erhalten wir folgendes verändertes Bild:



### Aufgabe 3: Eine XPM-Datei

Eine XPM-Datei hat folgenden Inhalt:

```

/* XPM */
static char *kreuz[] = {
/* columns rows colors chars-per-pixel */
"13 13 3 1 ",
"o c red",
". c yellow",
"X c gray",
/* pixels */
"XXXXXXXXXXXXX",
"Xo.....oX",
"X.o.....oX",
"X..o.....oX",
"X...o...oX",
"X....o.o...X",
"X.....o....X",
"X.....o.o...X",
"X....o...oX",
"X...o...oX",
"X..o.....oX",
"X.o.....oX",
"Xo.....oX",
"XXXXXXXXXXXXX"
};

```

Schreibe den Text der Bilddatei entsprechend folgender Angaben neu auf Dein Blatt.

- Eine rote Linie soll senkrecht von ganz oben nach ganz unten genau in der Mitte des Bildes verlaufen.
- Eine grüne Linie waagerecht von ganz rechts nach ganz links genau in der Mitte der Leinwand verlaufen.
- Eine Diagonale soll mit schwarzer Farbe gezeichnet werden.

## Lösung

Eine mögliche Lösung sähe folgendermassen aus:

```
/* XPM */
static char *kreuz[] = {
/* columns rows colors chars-per-pixel */
"13 13 5 1 ",
"o c red",
". c yellow",
"X c gray",
"T c green",
"+ c black",
/* pixels */
"XXXXXXXXXXXXXXXX",
"X+...o...oX",
"X.+...o...oX",
"X..+.o..o.X",
"X...+.o.o..X",
"X...+oo...X",
"XTTTT+TTTTX",
"X...oo+...X",
"X...o.o+...X",
"X..o..o..+.X",
"X.o...o...+.X",
"Xo...o...+X",
"XXXXXXXXXXXXXXXX"
};
```

Es gibt aber noch eine zweite Alternative:

```
/* XPM */
static char *kreuz[] = {
/* columns rows colors chars-per-pixel */
"13 13 5 1 ",
"o c red",
". c yellow",
"X c gray",
"T c green",
"+ c black",
/* pixels */
"XXXXXXXXXXXXXXXX",
"Xo...o...+X",
"X.o...o...+.X",
"X..o..o..+.X",
"X...o.o+...X",
"X...oo+...X",
"XTTTT+TTTTX",
"X...+oo...X",
"X...+.o...X",
"X..+.o..o.X",
"X.+...o...oX",
"X+...o...oX",
"XXXXXXXXXXXXXXXX"
};
```

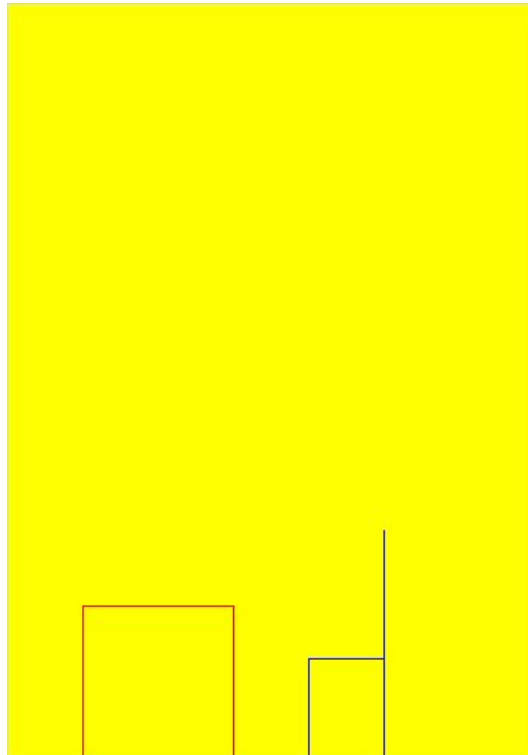
## Aufgabe 4: Tisch und Stuhl

*Auf einer Leinwand sollen ein Tisch und ein Stuhl – jeweils aus drei Strecken konstruiert – gezeichnet werden.*

- a) *Erstelle eine Zeichnung!*
- b) *Wie lauten die zugehörigen Anweisungen in TCL/TK?*

## Lösung

a) Wir können Tisch und Stuhl folgendermassen zeichnen:



b) Die zugehörigen Anweisungen in *TCL/TK* lauten:

```
canvas .leinwand -width 350 -height 500 -background yellow  
pack .leinwand
```

```
.leinwand create line 50 500 50 400 -fill red  
.leinwand create line 50 400 150 400 -fill red  
.leinwand create line 150 500 150 400 -fill red  
  
.leinwand create line 200 500 200 435 -fill blue  
.leinwand create line 200 435 250 435 -fill blue  
.leinwand create line 250 500 250 350 -fill blue
```

## 11.6 Python

### Aufgabe:

### Stundenplan als Webseite

psa/14.1.2011 \* 23.11.2011

---

#### Bilder aus Fensterinhalten erstellen

Wir können mit *Python* Fensterinhalte gestalten. Einen solchen Fensterinhalt wollen wir nun als Bild erstellen und in unserem Verzeichnis sichern.

Starte daher zuerst *bash* und wechsele dann in dein Verzeichnis:

```
cd F
cd erwin                               Deinen Verzeichnisnamen verwenden!
```

In diesem Verzeichnis ist als Beispiel der Stundenplan `plan3.py` hinterlegt worden. Wir können diesen Plan ausführen mit:

```
python2 plan3.py &                    Dann wird ein neues Fenster
                                        mit dem Namen "tk" aufgemacht.
```

Diese Fenster können wir dem Werkzeug `import` in ein Bild „verwandeln“. (Der Name dieses Werkzeuges ist etwas merkwürdig.)

```
import -window tk -pause 10 plan_erwin.jpg
```

Dann wird nach einer kurzen Pause von 10 Sekunden ein Bild von diesem Fenster mit der Bezeichnung `plan_erwin.jpg` im aktuellen Verzeichnis abgelegt. Dieses Bild können wir mit den verschiedensten Bildbetrachtern oder auch einem Browser ansehen, also beispielsweise:

```
xv plan_erwin.jpg                      oder:
display plan_erwin.jpg                 oder auch:
opera plan_erwin.jpg
```

#### Ein Bild in einer Webseite

Wir erstellen nun eine sehr kleine Webseite mit dem Bild unseres Planes als Inhalt. Starte unseren Editor mit `nedit plan_erwin.html` und schreibe folgenden Text:

```
<html>
<head><title>Stundenplan von Erwin</title></head>

<body>
  <center>
    <h1>Stundenplan von Erwin</h1>
    
  </center>
</body>

</html>
```

Dieses Seite können wir anschliessend mit einem Browser ansehen, also beispielsweise:

```
opera plan_erwin.html
```

# Aufgabe:

# Stundenplan als Webseite

psa/14.1.2011 \* 23.11.2011

## Bilder aus Fensterinhalten erstellen

Wir können mit *Python* Fensterinhalte gestalten. Einen solchen Fensterinhalt wollen wir nun als Bild erstellen und in unserem Verzeichnis sichern.

Starte daher zuerst *Python* und wechsele dann in dein Verzeichnis:

```
from os import *
chdir("F")
chdir("erwin")
```

Deinen Verzeichnisnamen verwenden!

In diesem Verzeichnis ist als Beispiel der Stundenplan `plan3.py` hinterlegt worden. Wir können diesen Plan ausführen mit:

```
import plan3          Dann wird ein neues Fenster
                      mit dem Namen "tk" aufgemacht.

bzw:
reload(plan3)        zum erneuten Ausführen
```

Diese Fenster können wir dem Betriebssystemwerkzeug `import` in ein Bild „verwandeln“. (Der Name dieses Werkzeuges ist etwas merkwürdig. Dieses Werkzeug ist nicht zu verwechseln mit der Python-Anweisung *import*!)

```
system("import -window tk -pause 10 plan_erwin.jpg")
```

Dann wird nach einer kurzen Pause von 10 Sekunden ein Bild von diesem Fenster mit der Bezeichnung `plan_erwin.jpg` im aktuellen Verzeichnis abgelegt. Dieses Bild können wir mit den verschiedensten Bildbetrachtern oder auch einem Browser ansehen, also beispielsweise:

```
system("xv plan_erwin.jpg")          oder:
system("display plan_erwin.jpg")     oder auch:
system("opera plan_erwin.jpg")
```

## Ein Bild in einer Webseite

Wir erstellen nun eine sehr kleine Webseite mit dem Bild unseres Planes als Inhalt. Starte unseren Editor mit `system("nedit plan_erwin.html")` und schreibe folgenden Text:

```
<html>
<head><title>Stundenplan von Erwin</title></head>

<body>
<center>
<h1>Stundenplan von Erwin</h1>

</center>
</body>

</html>
```

Dieses Seite können wir anschliessend mit einem Browser ansehen, also beispielsweise:

```
system("opera plan_erwin.html")
```

# Übung 3:

canvas

psa/17.2.2011 \* 21.11.2011

Wir können in Python sehr einfach Zeichnungen erstellen. Dabei werden auf einer Leinwand, in Python *Canvas* genannt, Objekte wie beispielsweise Strecken, in Python *line* genannt, gezeichnet.

Der Aufruf:

```
leinwand.create_line (20, 30, 200, 350, fill="blue")
```

zeichnet eine Strecke vom Punkt  $A(20/30)$  bis zum Punkt  $B(200/350)$  in der Farbe *blau*.

Dabei liegt der *Ursprungspunkt*  $O(0/0)$  ist – anders als in der Mathematik üblich – in der linken oberen Ecke der Leinwand.

---

Bei allen folgenden Aufgaben gilt, dass eine Leinwand mit:

```
from Tkinter import *
```

```
grafikfenster = Tk()
```

```
leinwand = Canvas(grafikfenster, width=350, height=500, background="yellow")
```

```
leinwand.pack()
```

erzeugt wird.

**Wichtiger Hinweis:** Erstelle in Deinem Heft immer zuerst eine Zeichnung, bevor Du aufschreibst, welche Anweisungen Du in Python eingeben willst.

---

## Aufgabe 1

Zeichne auf der Leinwand eine Strecke, die von ganz rechts nach ganz links genau in der Mitte der Leinwand verläuft.

## Aufgabe 2

Zeichne auf der Leinwand eine Strecke, die von ganz oben nach ganz unten genau in der Mitte der Leinwand verläuft.

## Aufgabe 3

Zeichne beide Strecken auf eine Leinwand.

## Aufgabe 4

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine waagerechte Strecke mit dem Pixelabstand 20 von der oberen Kante sein, die zweite soll ebenfalls eine waagerechte Strecke mit dem Pixelabstand 20 von der unteren Leinwandkante sein.

## Aufgabe 5

Zeichne auf der Leinwand zwei Strecken:

Die erste soll eine senkrechte Strecke mit dem Pixelabstand 20 von der linken Kante sein, die zweite soll ebenfalls eine senkrechte Strecke mit dem Pixelabstand 20 von der rechten Leinwandkante sein.



## **Aufgabe 6**

Zeichne auf der Leinwand drei waagerechte Strecken:

Die Abstände von der oberen und der unteren Kante und zwischen den Strecken sollen alle gleich sein.

## **Aufgabe 7**

Zeichne auf der Leinwand drei senkrechte Strecken:

Die Abstände von der oberen und der unteren Kante und zwischen den Strecken sollen alle gleich sein.

## **Aufgabe 8**

Zeichne auf der Leinwand ein beliebiges Dreieck.

## **Aufgabe 9**

Zeichne auf der Leinwand ein Dreieck mit einer Seitenlänge von 300 Pixelbreiten und einer Seitenlänge mit 200 Pixelbreiten. Die Länge der dritten Seite ist beliebig.

```
print "Wir basteln uns unseren Stundenplan"  
print "Version 3"
```

```
from Tkinter import *  
grafikfenster = Tk()
```

```
montag = Frame(grafikfenster)  
montag.pack(side=LEFT)
```

```
montag.tag = Label(montag, text="Montag", width=15, height=2)  
montag.tag.pack()  
montag.stunde1 = Label(montag, text="DE", background="red", width=15)  
montag.stunde1.pack()  
montag.stunde2 = Label(montag, text="MA", background="orange", width=15)  
montag.stunde2.pack()  
montag.stunde3 = Label(montag, text="MA", background="orange", width=15)  
montag.stunde3.pack()  
montag.stunde4 = Label(montag, text="EN", background="pink", width=15)  
montag.stunde4.pack()  
montag.stunde5 = Label(montag, text="AS", background="purple", width=15)  
montag.stunde5.pack()  
montag.stunde6 = Label(montag, text="MU", background="burlywood", width=15)  
montag.stunde6.pack()
```

```
dienstag = Frame(grafikfenster)  
dienstag.pack(side=LEFT)
```

```
dienstag.tag = Label(dienstag, text="Dienstag", width=15, height=2)  
dienstag.tag.pack()  
dienstag.stunde1 = Label(dienstag, text="GL", background="green", width=15)  
dienstag.stunde1.pack()  
dienstag.stunde2 = Label(dienstag, text="SP", background="yellow", width=15)  
dienstag.stunde2.pack()  
dienstag.stunde3 = Label(dienstag, text="DE", background="red", width=15)  
dienstag.stunde3.pack()  
dienstag.stunde4 = Label(dienstag, text="KU", background="white", width=15)  
dienstag.stunde4.pack()  
dienstag.stunde5 = Label(dienstag, text="M-F", background="orange", width=15)  
dienstag.stunde5.pack()  
dienstag.stunde6 = Label(dienstag, width=15)  
dienstag.stunde6.pack()
```

```
mittwoch = Frame(grafikfenster)  
mittwoch.pack(side=LEFT)
```

```

mittwoch.tag = Label(mittwoch, text="Mittwoch", width=15, height=2)
mittwoch.tag.pack()
mittwoch.stunde1 = Label(mittwoch, text="EN", background="pink", width=15)
mittwoch.stunde1.pack()
mittwoch.stunde2 = Label(mittwoch, text="MA", background="orange", width=15)
mittwoch.stunde2.pack()
mittwoch.stunde3 = Label(mittwoch, text="SP", background="yellow", width=15)
mittwoch.stunde3.pack()
mittwoch.stunde4 = Label(mittwoch, text="WP", background="blue", width=15)
mittwoch.stunde4.pack()
mittwoch.stunde5 = Label(mittwoch, text="OS", background="purple", width=15)
mittwoch.stunde5.pack()
mittwoch.stunde6 = Label(mittwoch, text="KU", background="white", width=15)
mittwoch.stunde6.pack()

```

```

donnerstag = Frame(grafikfenster)
donnerstag.pack(side=LEFT)

```

```

donnerstag.tag = Label(donnerstag, text="Donnerstag", width=15, height=2)
donnerstag.tag.pack()
donnerstag.stunde1 = Label(donnerstag, text="MA", background="orange", width=15)
donnerstag.stunde1.pack()
donnerstag.stunde2 = Label(donnerstag, text="GL", background="green", width=15)
donnerstag.stunde2.pack()
donnerstag.stunde3 = Label(donnerstag, text="DE", background="red", width=15)
donnerstag.stunde3.pack()
donnerstag.stunde4 = Label(donnerstag, text="NW", background="cyan", width=15)
donnerstag.stunde4.pack()
donnerstag.stunde5 = Label(donnerstag, text="ZK", background="grey", width=15)
donnerstag.stunde5.pack()
donnerstag.stunde6 = Label(donnerstag, text="REL", background="brown", width=15)
donnerstag.stunde6.pack()

```

```

freitag = Frame(grafikfenster)
freitag.pack(side=LEFT)

```

```

freitag.tag = Label(freitag, text="Freitag", width=15, height=2)
freitag.tag.pack()
freitag.stunde1 = Label(freitag, text="DE", background="red", width=15)
freitag.stunde1.pack()
freitag.stunde2 = Label(freitag, text="WP", background="blue", width=15)
freitag.stunde2.pack()
freitag.stunde3 = Label(freitag, text="EN", background="pink", width=15)

```

```
freitag.stunde3.pack()
freitag.stunde4 = Label(freitag, text="SP", background="yellow", width=15)
freitag.stunde4.pack()
freitag.stunde5 = Label(freitag, text="AS", background="purple", width=15)
freitag.stunde5.pack()
freitag.stunde6 = Label(freitag, width=15)
freitag.stunde6.pack()

# mainloop()
```



## 11.7 Fragebögen

### Schülerfragebogen

Liebe Schülerin, lieber Schüler,

An der *Technischen Universität Dortmund* beschäftigen wir uns mit dem Informatik-Unterricht in der Sekundarstufe I, also in den Jahrgängen 5 bis 10 in der Schule. Um den Unterricht untersuchen zu können, benötigen wir Daten von Schülern, die Informatikunterricht haben bzw. hatten und Daten von Schülern, die bisher keinen Informatik-Unterricht hatten.

Daher bitten wir jetzt auch Dich, an diesem Test teilzunehmen.

Du erhältst diesen Fragebogen dreimal: einmal vor Beginn der Testreihe, einmal am Ende der Reihe und noch einmal nach ein paar weiteren Monaten. Damit wollen wir herausfinden, wie sich Dein Kenntnisstand und vielleicht auch Deine Einstellung zur Informatik im Lauf der Zeit verändern.

Lies die Fragen bitte genau durch und antworte ehrlich. Wenn Du Dir bei einer Antwort nicht sicher bist, dann kreuze die Antwort an, die Deiner Meinung nach die Beste ist. Achte darauf, dass Du keine Frage auslässt. Einige Fragen klingen zuerst sehr ähnlich. Sie sind aber dennoch immer verschieden oder beziehen sich auf etwas Anderes. Da Schüler verschiedene Auffassungen vertreten können, wird jeder eventuell etwas Anderes ankreuzen. Es gibt also keine richtigen und falschen Antworten. Wenn Dir etwas unklar ist, kannst Du nachfragen.

Auf dem Fragebogen muss unter Anderem Deine persönliche Codenummer stehen, die Dein Lehrer bzw. Deine Lehrerin eingetragen hat. Deine Codenummer bleibt für die gesamte Dauer des Projekts gleich. Deine Antworten werden natürlich vertraulich behandelt, das heißt: sie werden nicht an außenstehende Personen weitergegeben. Sie werden natürlich auch nicht zur Ermittlung einer Schulnote verwendet.

Herzlichen Dank fürs Mitmachen!

---

Schule:

Klasse:  Alter:  Codenummer:

Geschlecht:    Junge     Mädchen

## Schülerfragebogen \*\*\* 1. Halbzeit

Zuerst möchten wir wissen, was Du bereits aus der *Informatik* kennst bzw. weißt. Kreuze also entsprechend *Ja* oder *Nein* oder *Weiss nicht* an.

Da Du bisher keinen oder nur wenig Informatik-Unterricht hattest, ist es überhaupt nicht schlimm, wenn Du einige oder auch viele Fragen mit *Weiss nicht* beantwortest.

		Ja	Nein	Weiss nicht
1	Das <i>Internet</i> ist eine <i>Zusammenfassung</i> von vielen <i>Netzwerken</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Mit Hilfe einer <i>bedingten Anweisung</i> kann der Computer beispielsweise unterschiedlich reagieren, wenn eine eingegebene Zahl grösser oder kleiner bzw. gleich 7 ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Ich kenne <i>mehrere</i> Betriebssysteme.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Das <i>Internet</i> ist ein einzelnes <i>Netzwerk</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Mehrere Anweisungen können in einem Computerprogramm unter einem Namen zusammengefasst werden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Ich weiss, was eine <i>Wiederholungsanweisung</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Ich kenne das <i>JPG-Format</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Ich weiss, was <i>Anwendersoftware</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<i>Windows XP</i> ist eine <i>Anwendersoftware</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	<i>LINUX</i> ist <i>Anwendersoftware</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	Ein <i>Wert</i> kann in einer Variablen gespeichert werden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	Ich weiss, was eine <i>bedingte Anweisung</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	<i>Windows 7</i> ist ein <i>Betriebssystem</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	Ich weiss, was eine <i>Prozedur</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	Ein GIF-Bild ist eine <i>Pixelgrafik</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	Ich weiss, was <i>Netzwerke</i> sind.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	<i>WORD</i> ist ein <i>Betriebssystem</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	Ich weiss, was eine <i>Shell</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	Ich weiss, was ein <i>Algorithmus</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	Ich weiss, was eine <i>Vektorgrafik</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21	Um die Zahlen 1 bis 100 zu summieren, kann man im Programm eine <i>Wiederholungsanweisung</i> verwenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22	Ich weiss, was eine <i>Variable</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23	Ich weiss, was ein <i>Betriebssystem</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
24	Ich weiss, was eine <i>Kommandozeile</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
25	Eine <i>eindeutige Handlungsvorschrift</i> wird als <i>Algorithmus</i> bezeichnet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
26	Ich weiss, was eine <i>Pixelgrafik</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
27	Ich weiss, was <i>GUIs</i> sind.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
28	Ich weiss, was ein <i>Verzeichnisbaum</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
29	Ich weiss, was ein <i>Verzeichnis</i> bzw. ein <i>Ordner</i> ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Schülerfragebogen \*\*\* 2. Halbzeit

Die folgenden Fragen lassen sich oft nicht eindeutig mit *Ja* oder *Nein* beantworten. Daher hast Du hier mehrere Möglichkeiten zur Auswahl.

Wenn Du Dich nicht ganz entscheiden kannst, kreuze die Antwort an, die Deiner Position am nächsten ist.

		stimmt gar nicht	stimmt nicht	weder ja noch nein	stimmt	stimmt völlig	weiss nicht
1	In der Informatik ist Webseiten-Gestaltung sehr wichtig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Ein Programmtext sollte sehr gut lesbar sein.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Für viele Aufgaben gibt es mehr als ein Programm.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Programmierung ist in Informatik sehr wichtig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Es geht in der Informatik sehr oft um Algorithmen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Computer-Handbücher sind gelegentlich sehr nützlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Ich kenne mich sehr gut mit Tabellenkalkulation aus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Ich kenne mich mit sehr gut Computer-Spielen aus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Wenn ein Programm das tut, was es soll, ist es egal, wie es konstruiert wurde.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Es ist bei jedem Programm sehr wichtig, die aktuellste Softwareversion zu haben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	Bevor ich an den Computer zum Programmieren gehe, durchdenke ich das Problem erst vollständig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	Installieren von Software ist ein sehr wichtiger Teil von Informatik.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	Ich kenne mich sehr gut mit dem Internet aus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	Ich kann sehr gut programmieren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	Ich kenne mich sehr gut mit Textverarbeitung aus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	Programmieren ist sehr leicht.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	Ein Computerprogramm ist oft nicht einfach zu erstellen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	Informatiker müssen die meiste Zeit programmieren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	Für ein Problem gibt es nur ein mögliches Programm.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	In der Informatik sind Programmiersprachen sehr wichtig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



## Schülerfragebogen \*\*\* Nachspielzeit

Und jetzt ist nach Deiner Meinung gefragt. Du hast wieder mehrere Antwortmöglichkeiten zur Auswahl.

Wenn Du Dich nicht ganz entscheiden kannst, kreuze auch hier die Antwort an, die Deiner Meinung am nächsten ist.

		stimmt gar nicht	stimmt nicht	weder ja noch nein	stimmt	stimmt völlig	weiss nicht
1	Informatik ist für Jungen und Mädchen genauso gut geeignet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Ich weiss noch nicht, was ich später machen will.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Mein Lehrer hat mir Informatik als Fach empfohlen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Ich beschäftige mich täglich mit Computern.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Ich beschäftige mich mehrmals wöchentlich mit Computern.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Die Arbeit am Computer beim Programmieren ist schwierig und anstrengend.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Informatik kann man nur lernen, wenn man jede Stunde aufmerksam ist.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Jungen sind in Informatik besser als Mädchen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Ein Informatiker sollte ein guter Team-Worker sein.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Ich möchte mehr können als nur am Computer zu spielen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	Ich habe Informatik gewählt, weil meine Eltern das wollten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	Ein Informatiker ist ein Fachidiot.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	Ich interessiere mich für Computer-Spiele.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	In Informatik haben Jungen mehr Erfolg als Mädchen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	Es ist wichtig, Basiskenntnisse in Programmieren zu haben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	Ich möchte später irgend etwas mit Informatik oder Computern machen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	Ich lerne die meisten Sachen erst, wenn ich sie am Computer ausprobiert habe.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	Ich habe Informatik wegen des Lehrers gewählt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	Die Arbeit am Computer beim Programmieren macht Spass.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	Mädchen sind genauso erfolgreich wie Jungen in Informatik.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21	Mädchen sind besser in Informatik als Jungen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Schülerfragebogen \* Elfmeterschiessen

Liebe Schülerin, lieber Schüler,

An der *Technischen Universität Dortmund* beschäftigen wir uns mit dem Informatik-Unterricht in der Sekundarstufe I, also in den Jahrgängen 5 bis 10 in der Schule. Um den Unterricht untersuchen zu können, benötigen wir Daten von Schülern, die Informatikunterricht haben bzw. hatten und Daten von Schülern, die bisher keinen Informatik-Unterricht hatten.

Daher bitten wir jetzt auch Dich, an diesem Test teilzunehmen. Du erhältst einen Fragebogen dieser Art dreimal: einmal vor Beginn der Testreihe, einmal am Ende der Reihe und noch einmal nach ein paar weiteren Monaten. Damit wollen wir herausfinden, wie sich Dein Kenntnisstand im Lauf der Zeit verändert.

Die folgenden Aufgaben entstammen den *Biber-Wettbewerben* der Jahre 2007 bis 2009. Lies die Fragen bitte genau durch und antworte ernsthaft. Bei den Aufgaben ist immer genau eine Antwort richtig. Achte darauf, dass Du keine Frage auslässt.

Auf dem Fragebogen muss unter Anderem Deine persönliche Codenummer stehen, die Dein Lehrer bzw. Deine Lehrerin eingetragen hat. Deine Codenummer bleibt für die gesamte Dauer des Projekts gleich. Deine Antworten werden natürlich vertraulich behandelt, das heisst: sie werden nicht an außenstehende Personen weitergegeben. Sie werden natürlich auch nicht zur Ermittlung einer Schulnote verwendet.

Herzlichen Dank fürs Mitmachen!

---

Schule:

Klasse/Kurs:

Alter:

Codenummer:

Geschlecht:

Junge

Mädchen

## A1: Bibers Geheimcode

Biber möchte seinem Freund, dem Hasen, geheime Nachrichten zukommen lassen. Die beiden haben sich dafür einen Geheimcode ausgedacht. Mit dem werden ihre Nachrichten verschlüsselt, damit niemand mitlesen kann.

Bei ihrem Geheimcode bleiben die *Vokale* (A, E, I, O, U) und die *Satzzeichen* unverändert.

Die *Konsonanten* werden durch den jeweils folgenden Konsonanten im Alphabet ersetzt. Z wird dabei durch B ersetzt.

Wie lautet Bibers Nachricht

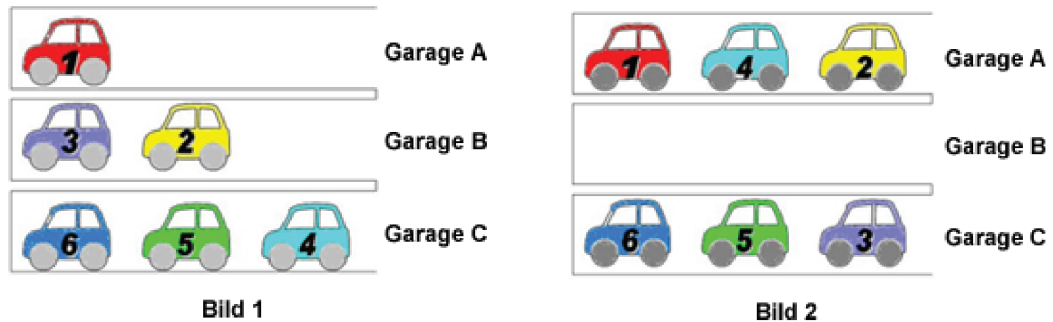
**HALB ACHT IM WALD**

im Geheimcode?

- HELB ECHT OM WELD**
- JEMC EDJV ON XEMF**
- GAKZ ABGS IL VAKC**
- JAMC ADJV IN XAMF**
  
- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.**
- Ich habe die Aufgabenstellung nicht verstanden.**

## A2: Umparken

Im Hotel Biber werden die Autos der Gäste von einem Angestellten geparkt, Herrn Krause. Das ist auch gut so, weil das Hotel nur drei Garagen hat, in denen die Autos hintereinander geparkt werden müssen. In jede Garage passen maximal drei Autos. Herr Krause kennt die Fahrgewohnheiten der Gäste gut und parkt die Autos am liebsten so wie in Bild 1 gezeigt. Dann muss er nur selten umparken, wenn er einem Gast seinen Wagen bringen soll.



Am Wochenende war Herr Krause im Urlaub und Herr Brandt hat ihn vertreten. Aber als Herr Krause wiederkommt, findet er die Autos in den Garagen vor wie in Bild 2 gezeigt.

Jetzt will Herr Krause schnell die richtige Reihenfolge wiederherstellen. Leider hat er keinen weiteren Parkplatz zur Verfügung, so dass immer nur das vorderste Auto aus einer Garage X in eine andere Garage Y (so weit hinein wie möglich) umparken kann.

Diese Operation nennen wir **Umparken (X, Y)**.

**In welcher Reihenfolge muss Herr Krause die Autos umparken, um seinen bevorzugten Parkzustand (Bild 1) wieder herzustellen?**

- Umparken (C, B), Umparken (A, C), Umparken (A, B)
- Umparken (C, B), Umparken (A, B), Umparken (A, C)
- Umparken (A, B), Umparken (C, B), Umparken (A, C)
- Umparken (B, C), Umparken (C, B), Umparken (A, B)
  
- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.
- Ich habe die Aufgabenstellung nicht verstanden.

## A3: Hervorgehobenes

Du willst dem Biber mitteilen, wie er bestimmte Wörter in einem Text hervorheben soll den er für dich in eine hübsche Holztafel nagt.

Das geht mit Hilfe besonderer Marken:

- Wörter zwischen den Marken `<F>` und `</F>` soll er in **Fettschrift** nagen.
- Wörter zwischen den Marken `<K>` und `</K>` soll er in *Kursivschrift* nagen.
- Wörter zwischen den Marken `<SCH>` und `</SCH>` soll er in Schreibmaschinenschrift nagen.

Was kommt heraus, wenn der Biber den folgenden Text nagt, den du mit einigen Marken versehen hast?

`<SCH>Der <F>Informatik-Biber <K>macht</K></F><K>viel</K> Spaß!</SCH>`

- Der Informatik-Biber *macht viel Spaß!*
- Der **Informatik-Biber** *machtviel* Spaß!
- Der **Informatik-Biber** *macht viel* Spaß!
- Der **Informatik-Biber** macht viel Spaß!
  
- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.**
- Ich habe die Aufgabenstellung nicht verstanden.**

## A4: Dateisuche

Im Computer kann man nach Dateien suchen, auch wenn man nur einen Teil ihres Namens benutzt.

Nimm an, folgende vier Dateien sind vorhanden:

1. **nmas.jpg**
2. **astmp.jpg**
3. **mdmtexas.png**
4. **nmtast.jpg**

Wenn du mit *\*.jpg* suchst, erhältst du eine Liste mit den Dateien *nmas.jpg*, *astmp.jpg* und *nmtast.jpg*.

Eine Suche mit *?????.jpg* liefert die Datei *astmp.jpg*.

Die Suche *\*s??.\** findet *keine Datei* mit einem passenden Namen.

**Welche Datei findest du mit *\*???as.\** ?**

- nmas.jpg**
- astmp.jpg**
- nmtast.jpg**
- mdmtexas.png**
  
- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.**
- Ich habe die Aufgabenstellung nicht verstanden.**

## A5: Links um!

Du hast einen Spielzeugroboter, der folgende Befehle auf Zuruf ausführen kann:

<i>Befehl</i>	<i>Bedeutung</i>
<b>Vor!</b>	<b>Der Roboter fährt 10 cm nach vorne.</b>
<b>Rechts!</b>	<b>Der Roboter dreht sich nach rechts. (um 90 Grad, also einen Viertelkreis).</b>

Du möchtest nun den Roboter so bewegen, dass er am Ende um 90 Grad (einen Viertelkreis) nach links gedreht ist.

Mit welcher Befehlsfolge kannst du das erreichen?

- Vor! Vor!**
- Rechts! Rechts!**
- Rechts! Rechts! Rechts!**
- Vor! Rechts! Vor!**
  
- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.**
- Ich habe die Aufgabenstellung nicht verstanden.**

## A6: Film digital

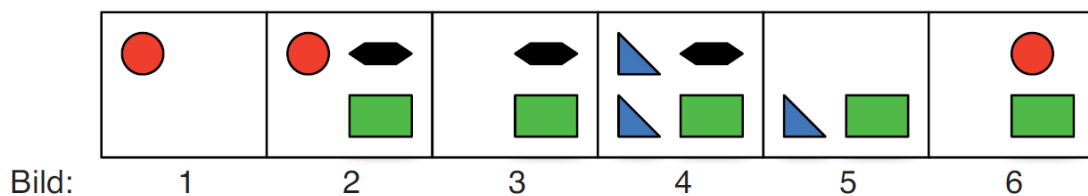
Bei digital gespeicherten Filmen spielt es eine Rolle, wie stark sich aufeinander folgende Bilder des Films unterscheiden.

Der *Unterschied* zwischen zwei aufeinander folgenden Bildern A und B lässt sich wie folgt berechnen:

Die Anzahl der Objekte in Bild A, die nicht in Bild B sind,  
plus die Anzahl der Objekte in Bild B, die nicht in Bild A sind.

Die *Unterschiedssumme eines Films* ist dann die *Summe aller Unterschiede* zwischen je zwei aufeinander folgenden Bildern.

Was ist die Unterschiedssumme der folgenden sechs Bilder?

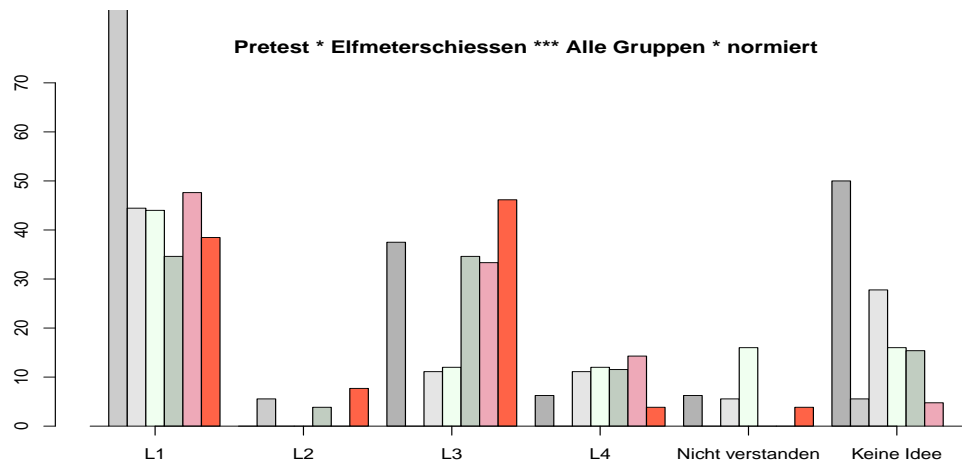


- 8**
- 9**
- 11**
- 13**
  
- Ich habe die Aufgabenstellung zwar verstanden, weiss aber nicht, wie diese Aufgabe zu lösen ist.**
- Ich habe die Aufgabenstellung nicht verstanden.**

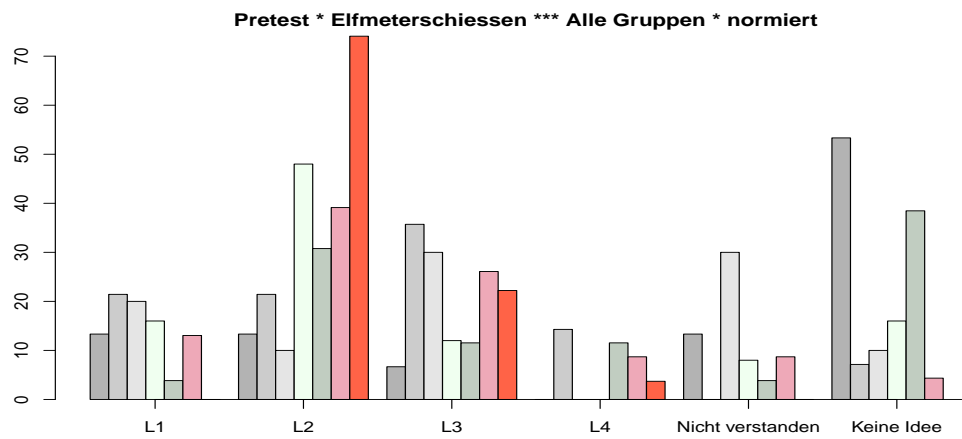




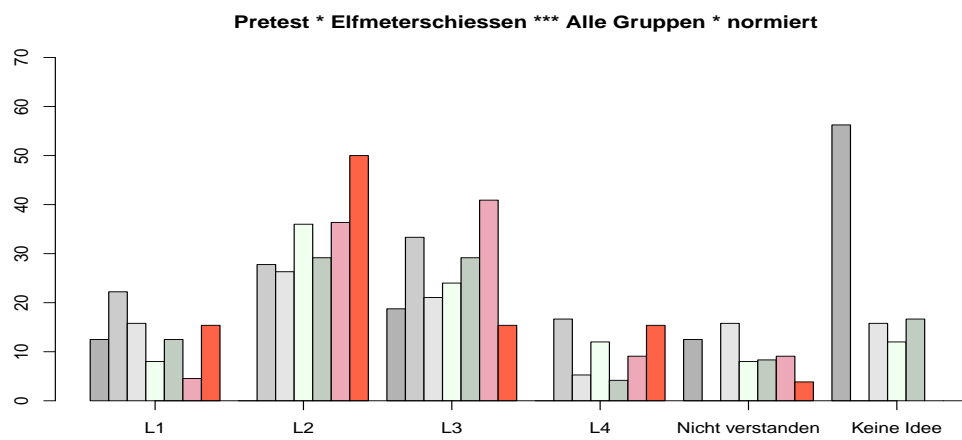
# 11.8 Auswertung Biber-Fragen



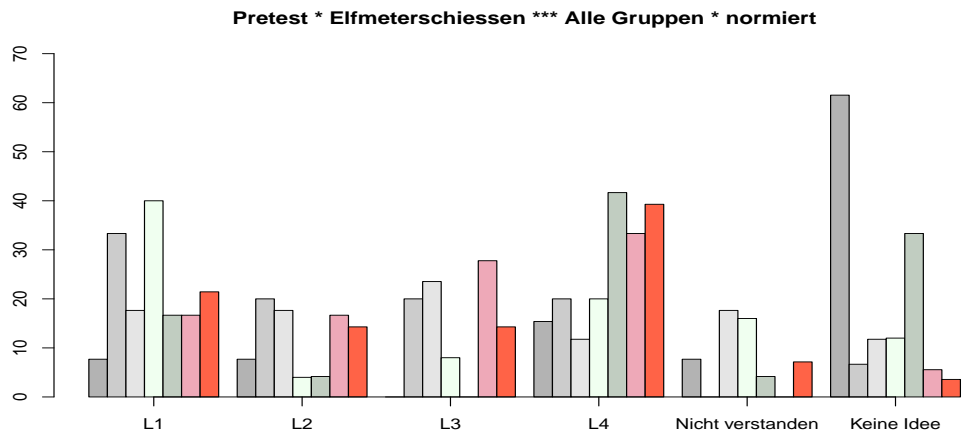
**A1: Bibers Geheimcode (Loesung: L4)**



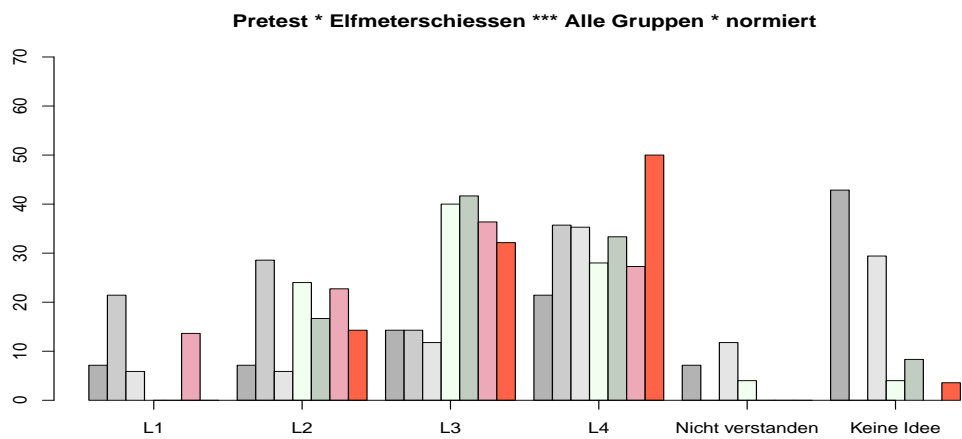
**A2: Umparken (Loesung: L2)**



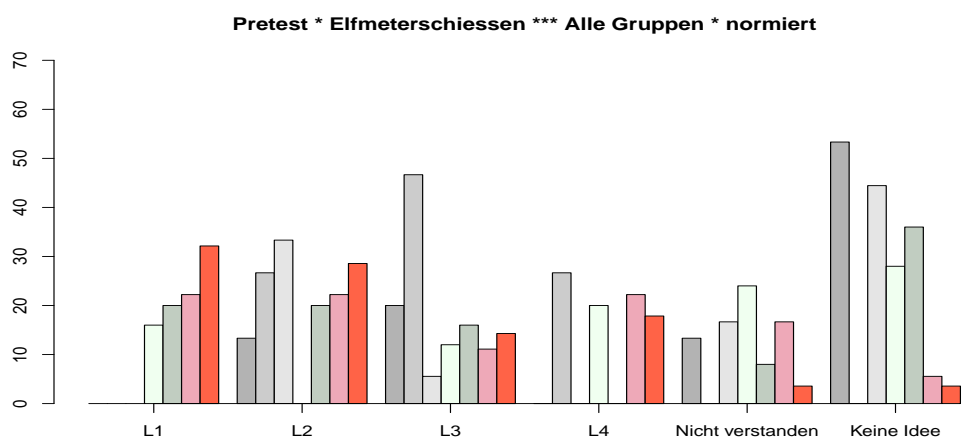
**A3: Hervorgehobenes (Loesung: L2)**



**A4: Dateisuche (Loesung: L4)**

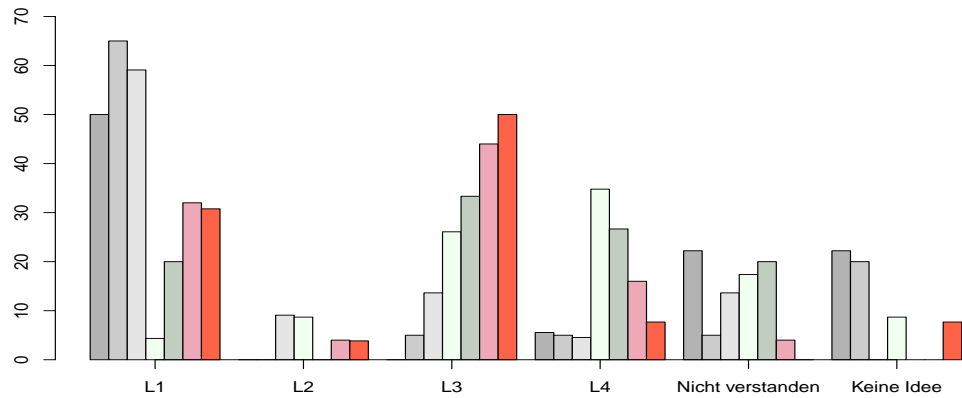


**A5: Links um! (Loesung: L3)**



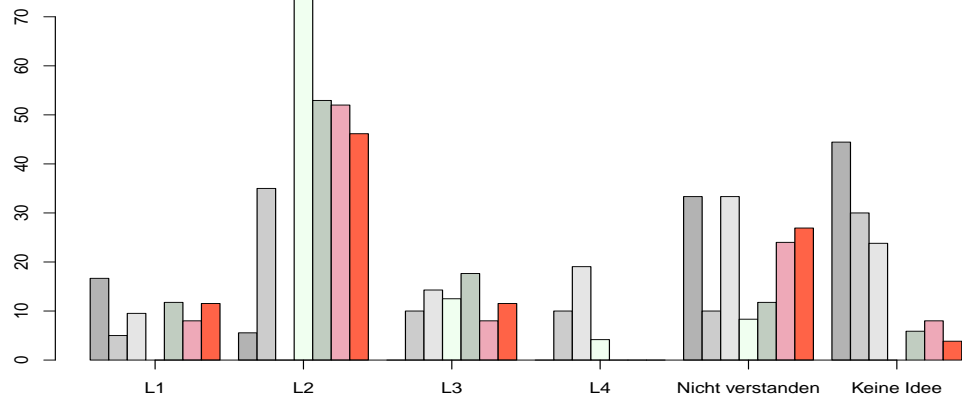
**A6: Film digital (Loesung: L2)**

Endtest \* Elfmeterschiessen \*\*\* Alle Gruppen \* normiert



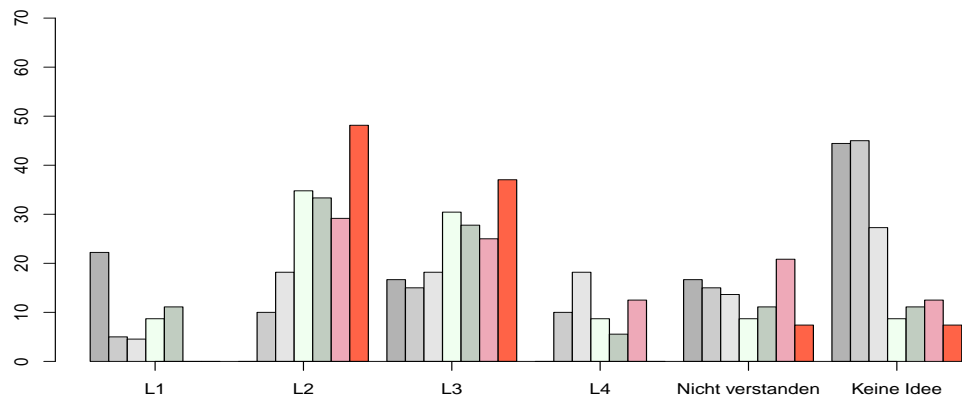
A1: Bibers Geheimcode (Loesung: L4)

Endtest \* Elfmeterschiessen \*\*\* Alle Gruppen \* normiert



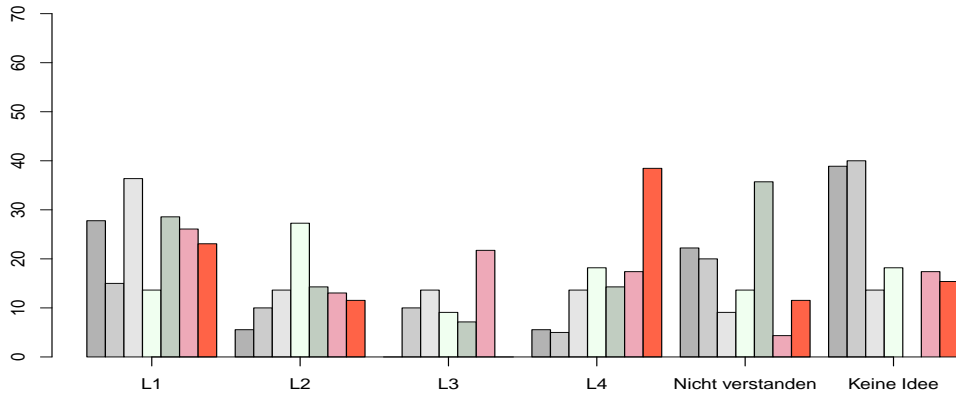
A2: Umparken (Loesung: L2)

Endtest \* Elfmeterschiessen \*\*\* Alle Gruppen \* normiert



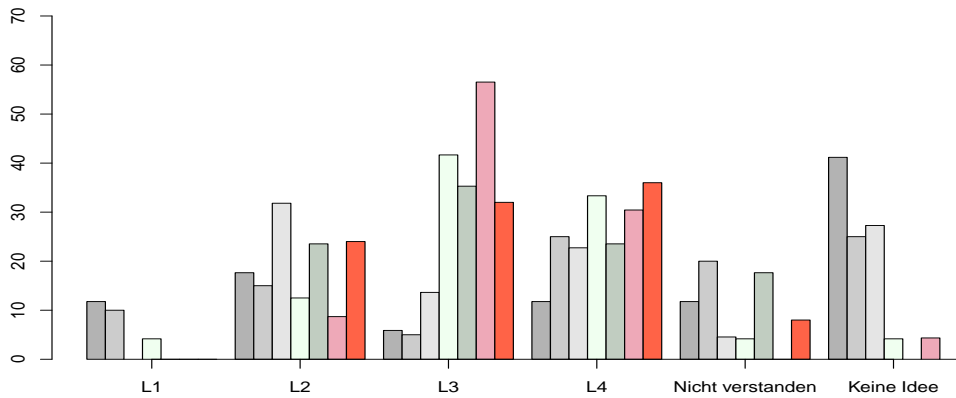
A3: Hervorgehobenes (Loesung: L2)

Endtest \* Elfmeterschiessen \*\*\* Alle Gruppen \* normiert



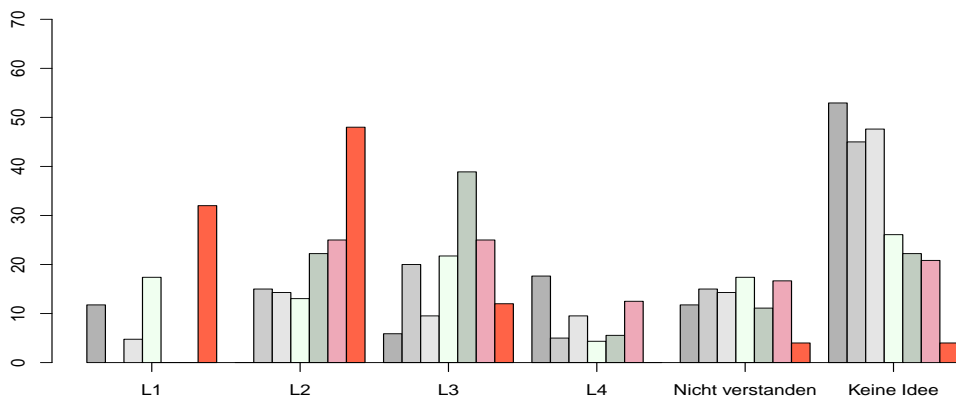
A4: Dateisuche (Loesung: L4)

Endtest \* Elfmeterschiessen \*\*\* Alle Gruppen \* normiert



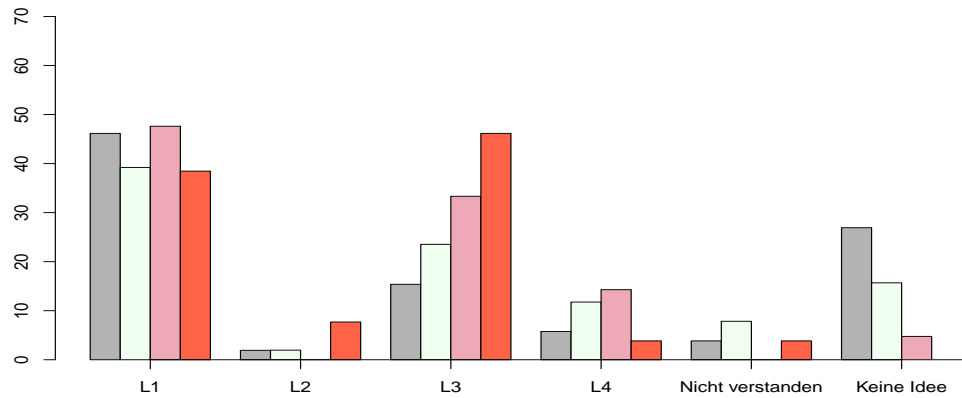
A5: Links um! (Loesung: L3)

Endtest \* Elfmeterschiessen \*\*\* Alle Gruppen \* normiert



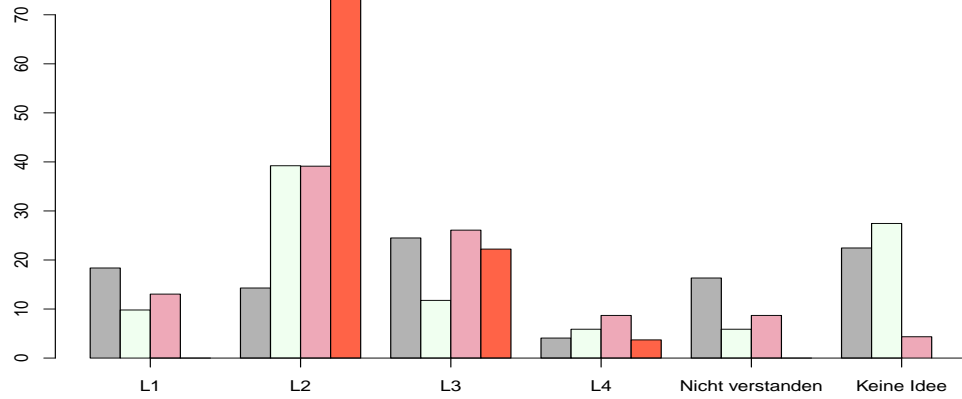
A6: Film digital (Loesung: L2)

Pretest \* Elfmeterschiesen \*\*\* Aller Schulen \* normiert



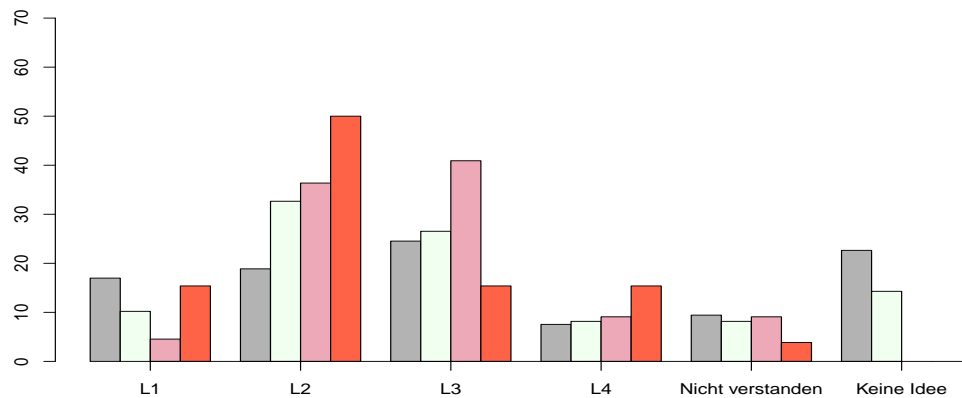
A1: Bibers Geheimcode (Loesung: L4)

Pretest \* Elfmeterschiesen \*\*\* Aller Schulen \* normiert

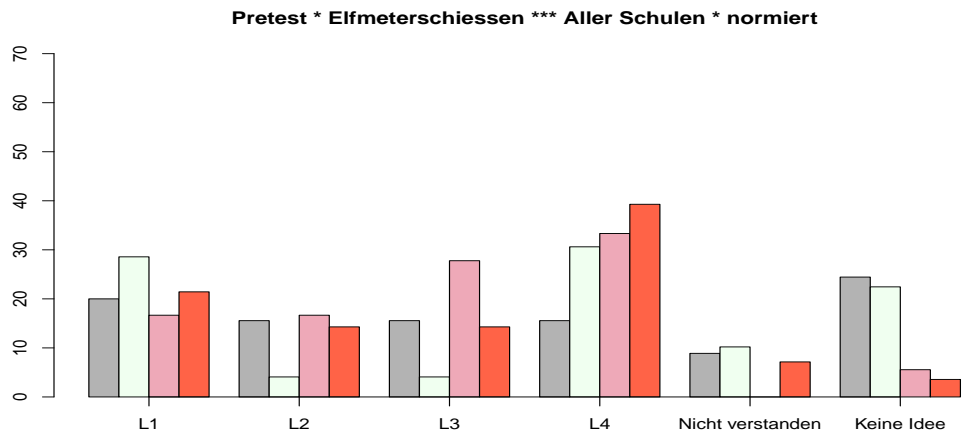


A2: Umparken (Loesung: L2)

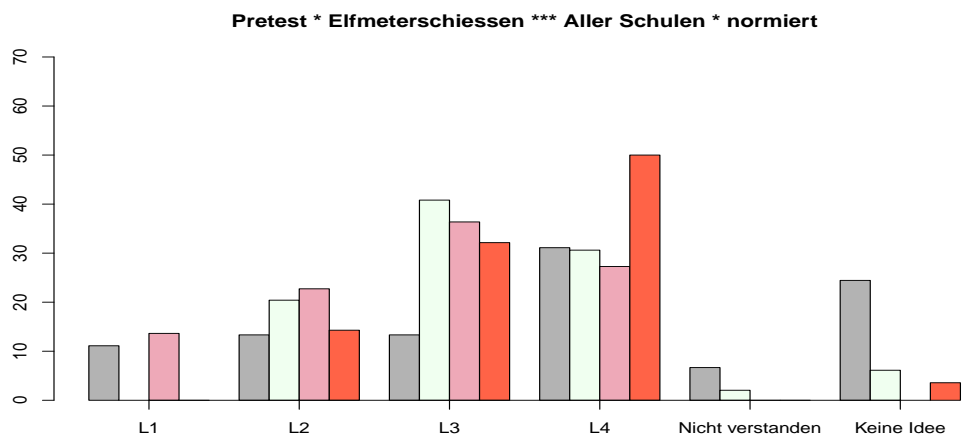
Pretest \* Elfmeterschiesen \*\*\* Aller Schulen \* normiert



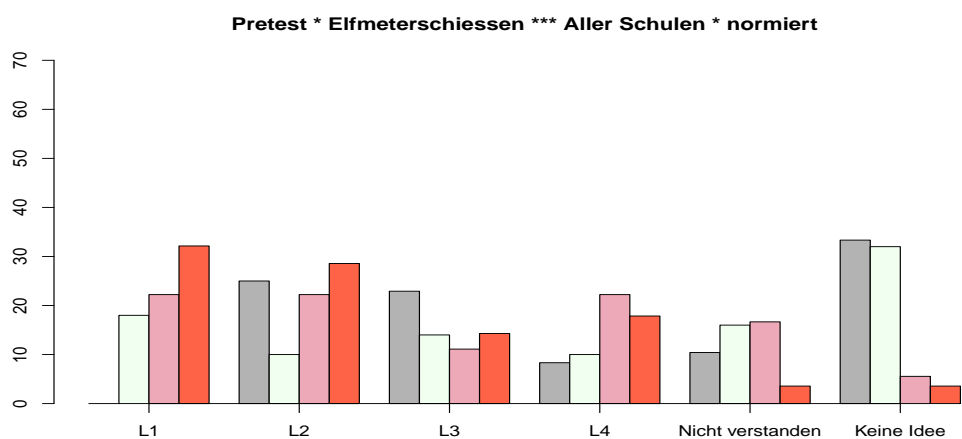
A3: Hervorgehobenes (Loesung: L2)



**A4: Dateisuche (Loesung: L4)**

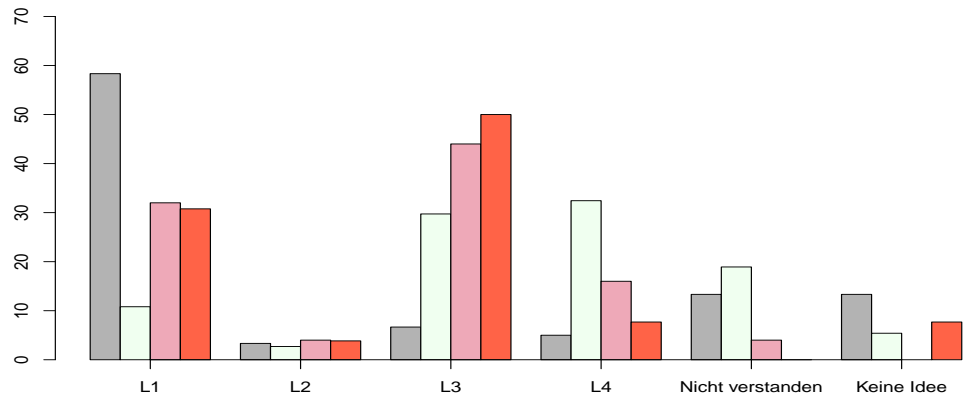


**A5: Links um! (Loesung: L3)**



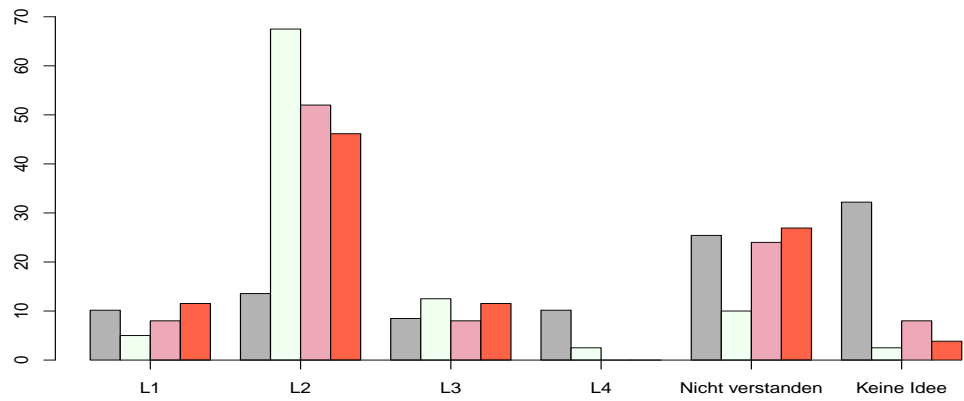
**A6: Film digital (Loesung: L2)**

Endtest \* Elfmeterschiessen aller Schulen \* normiert



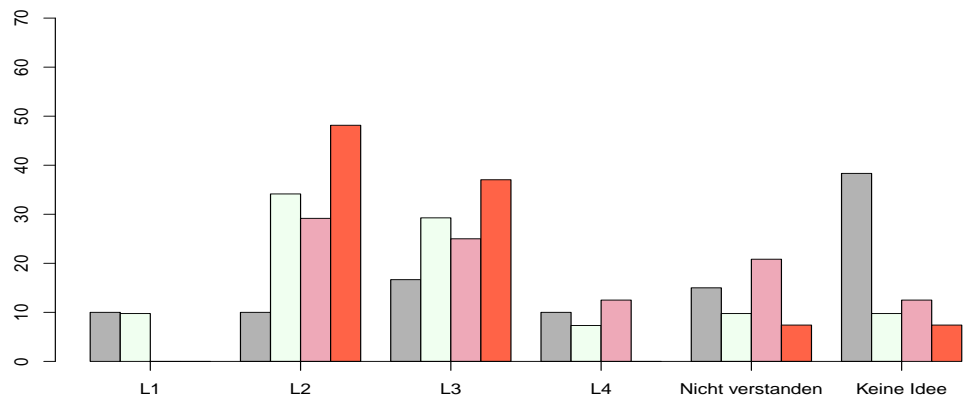
A1: Bibers Geheimcode (Loesung: L4)

Endtest \* Elfmeterschiessen aller Schulen \* normiert



A2: Umparken (Loesung: L2)

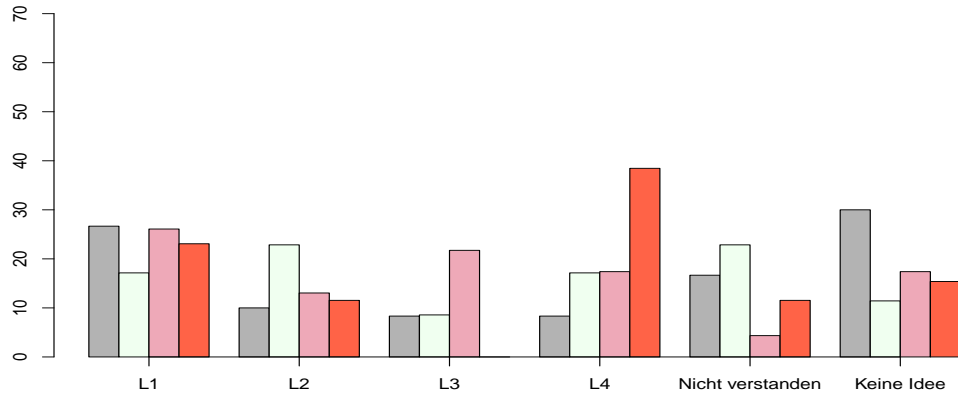
Endtest \* Elfmeterschiessen aller Schulen \* normiert



A3: Hervorgehobenes (Loesung: L2)

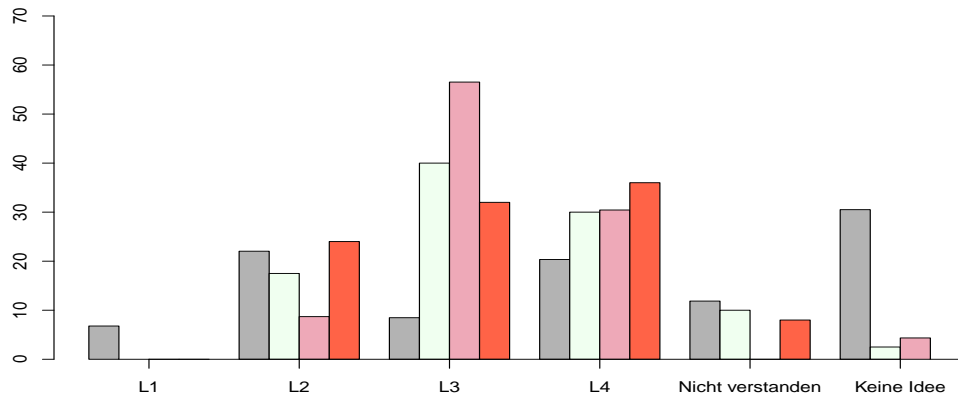


Endtest \* Elfmeterschiessen aller Schulen \* normiert



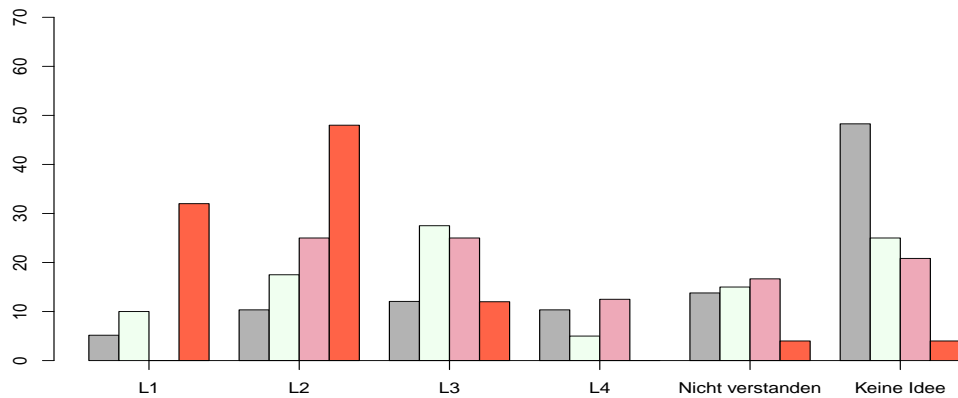
A4: Dateisuche (Loesung: L4)

Endtest \* Elfmeterschiessen aller Schulen \* normiert



A5: Links um! (Loesung: L3)

Endtest \* Elfmeterschiessen aller Schulen \* normiert

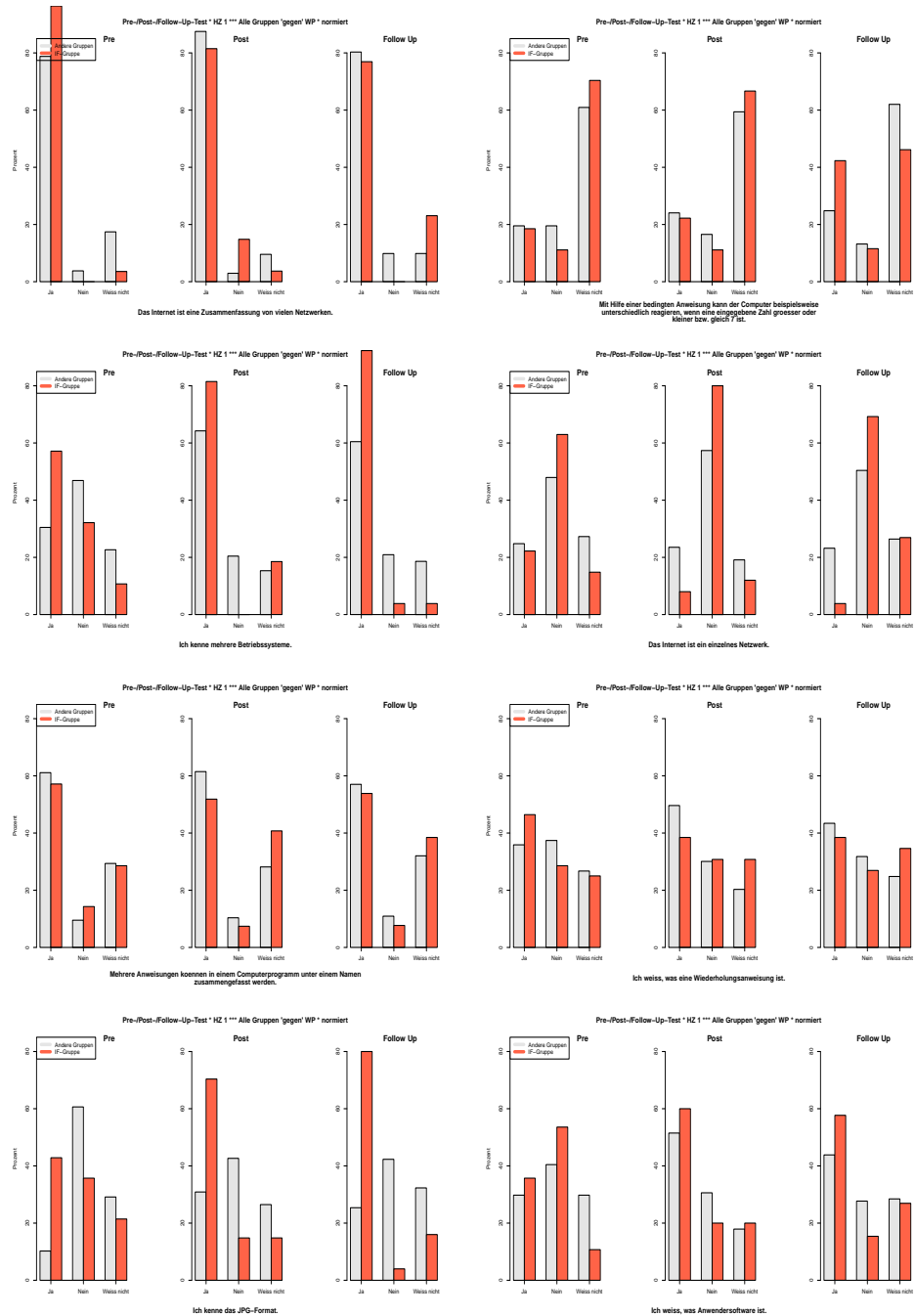


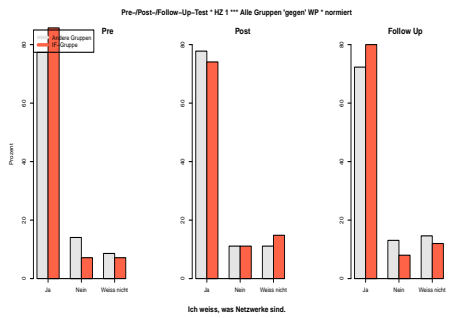
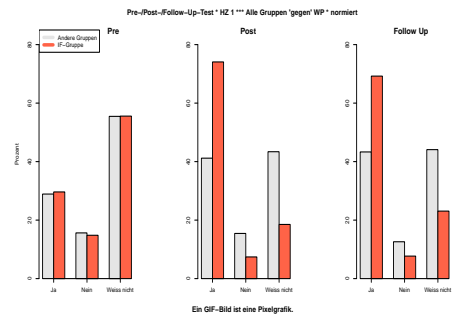
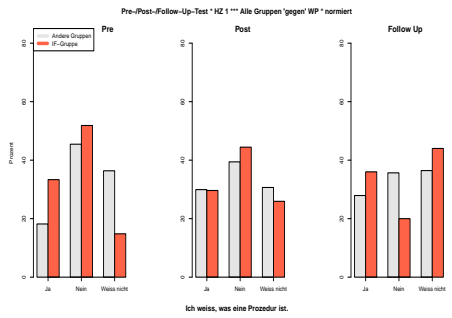
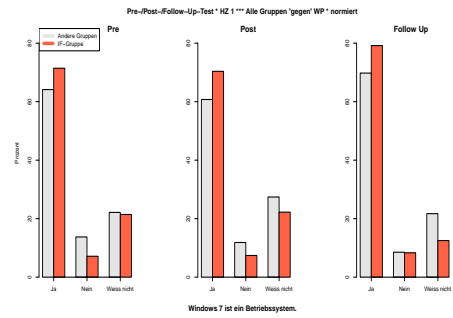
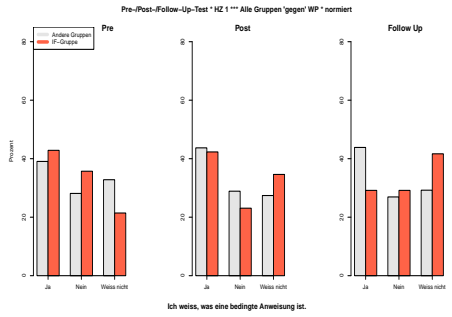
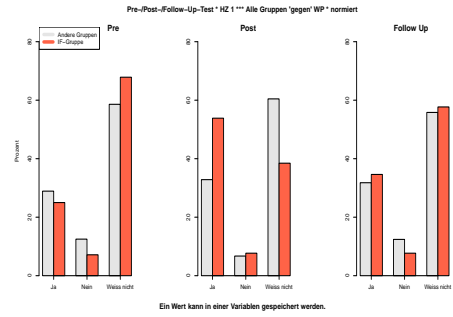
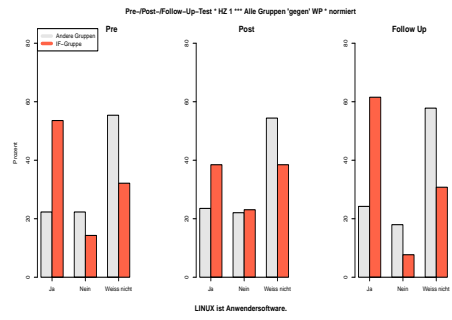
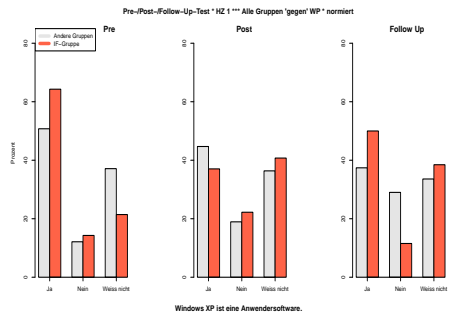
A6: Film digital (Loesung: L2)

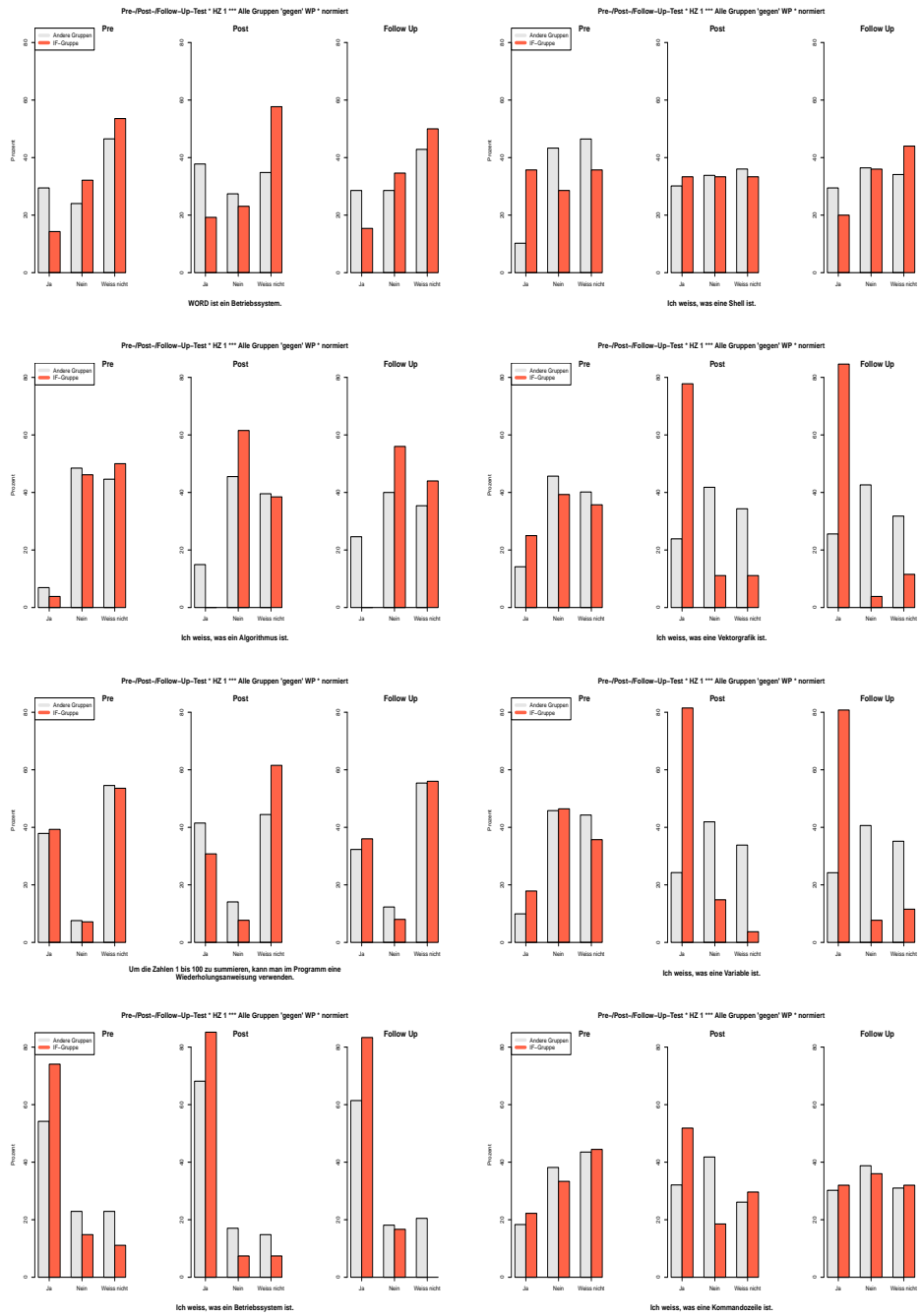
# 11.9 Follow Up

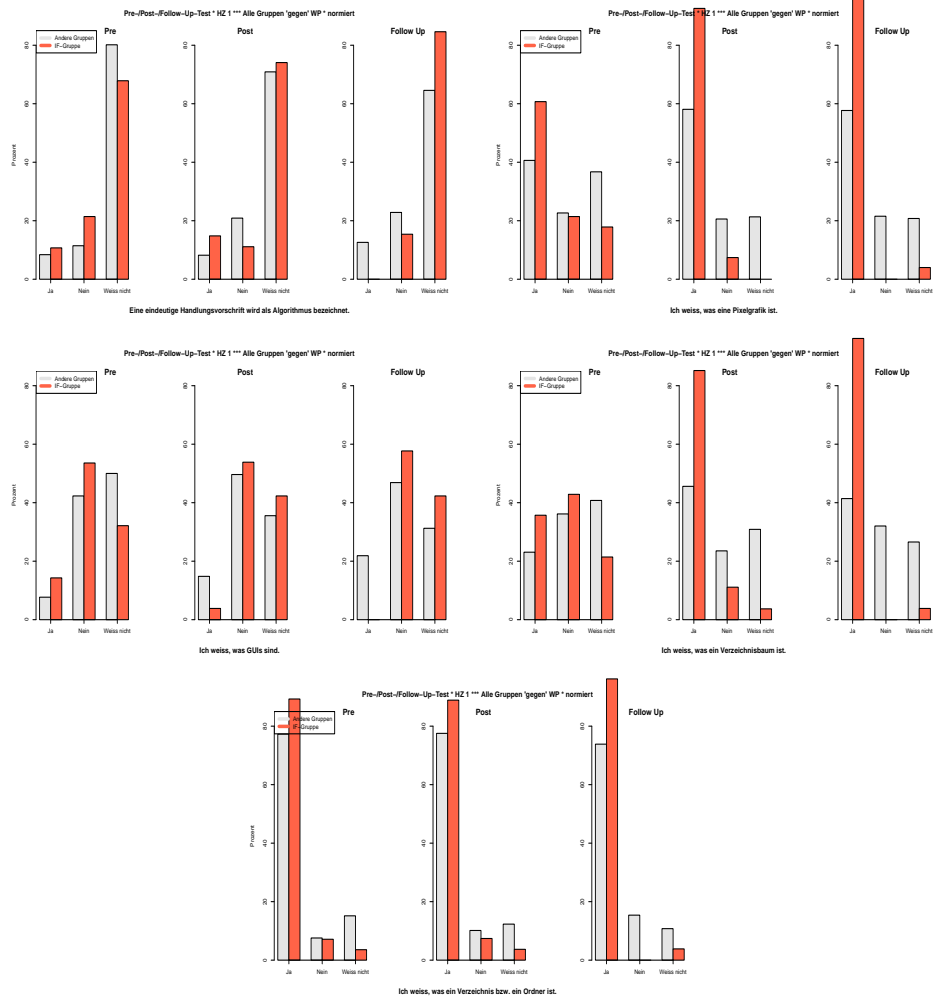
## Vergleich Pre - / Post - /Follow-Up - Test Halbzeit1

### Informatik-Gruppe im Vergleich zu allen anderen Gruppen



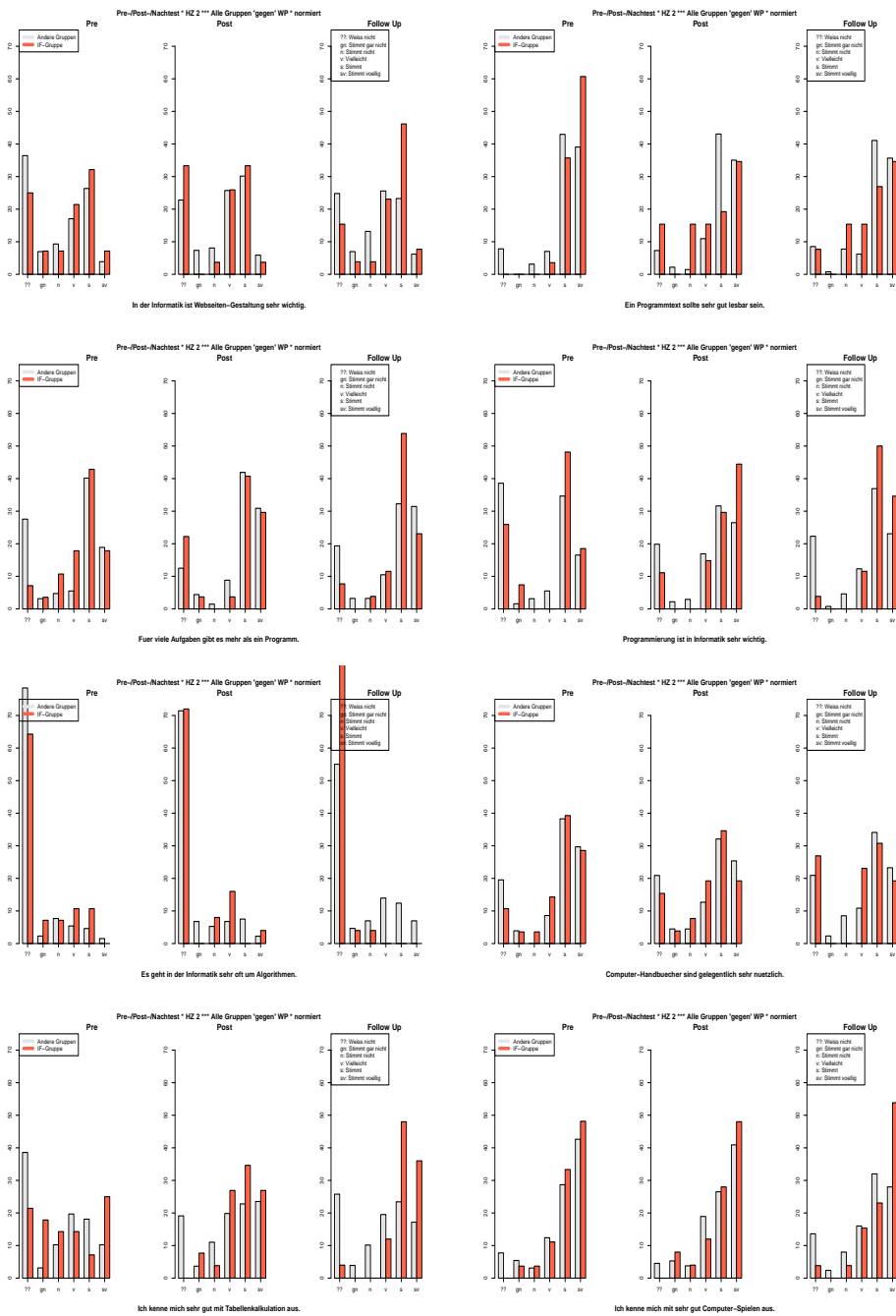


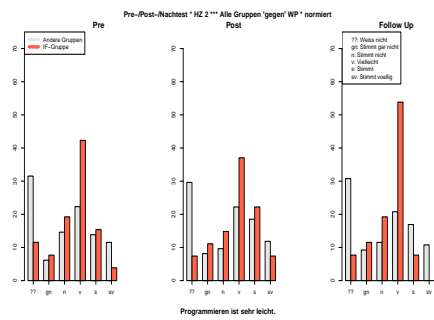
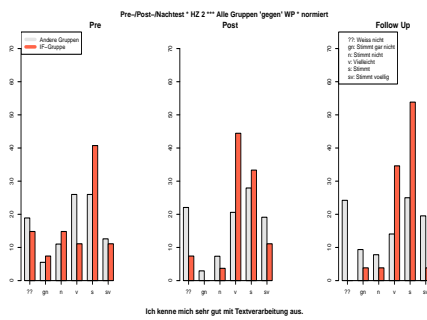
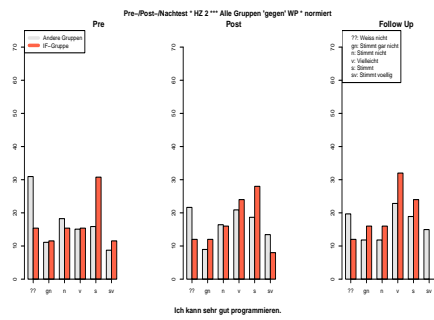
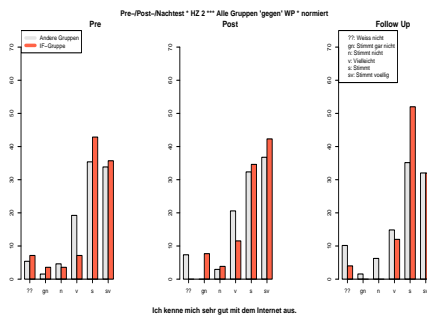
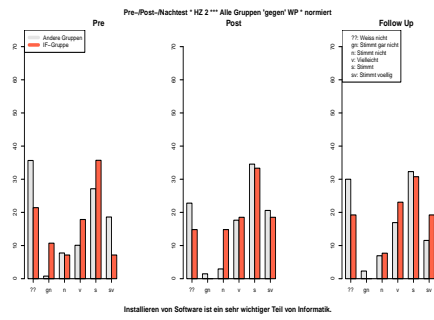
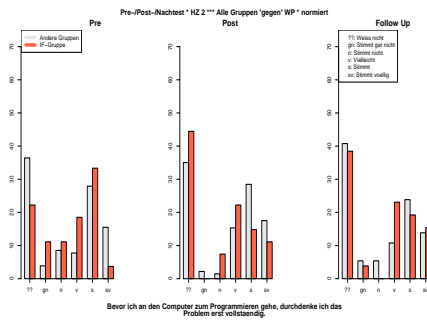
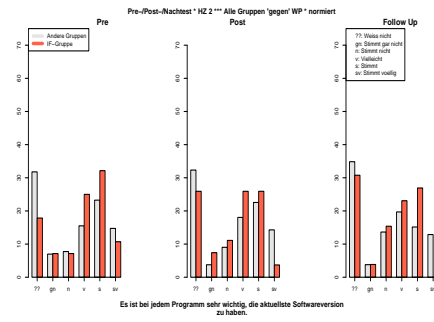
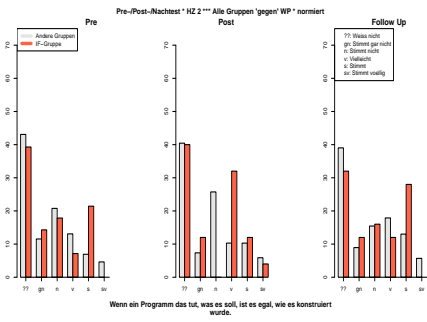


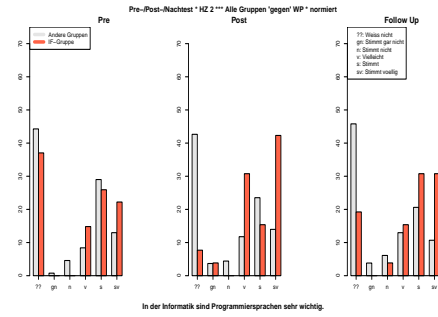
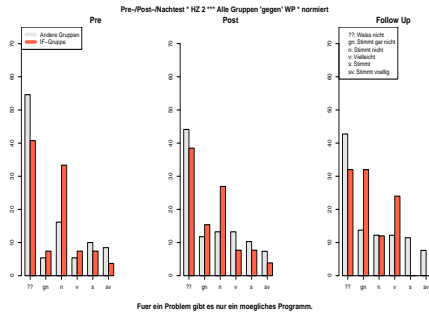
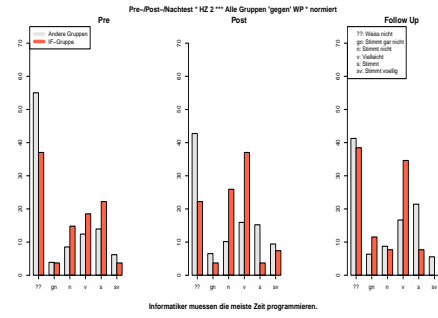
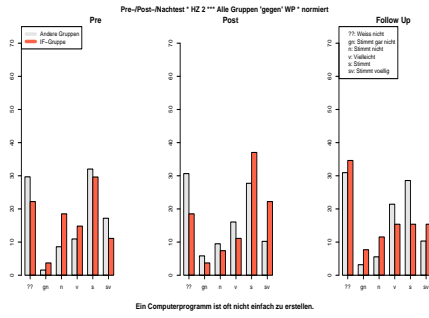


# Vergleich Pre - / Post - / Follow-Up - Test Halbzeit 2

## Informatik-Gruppe im Vergleich zu allen anderen Gruppen



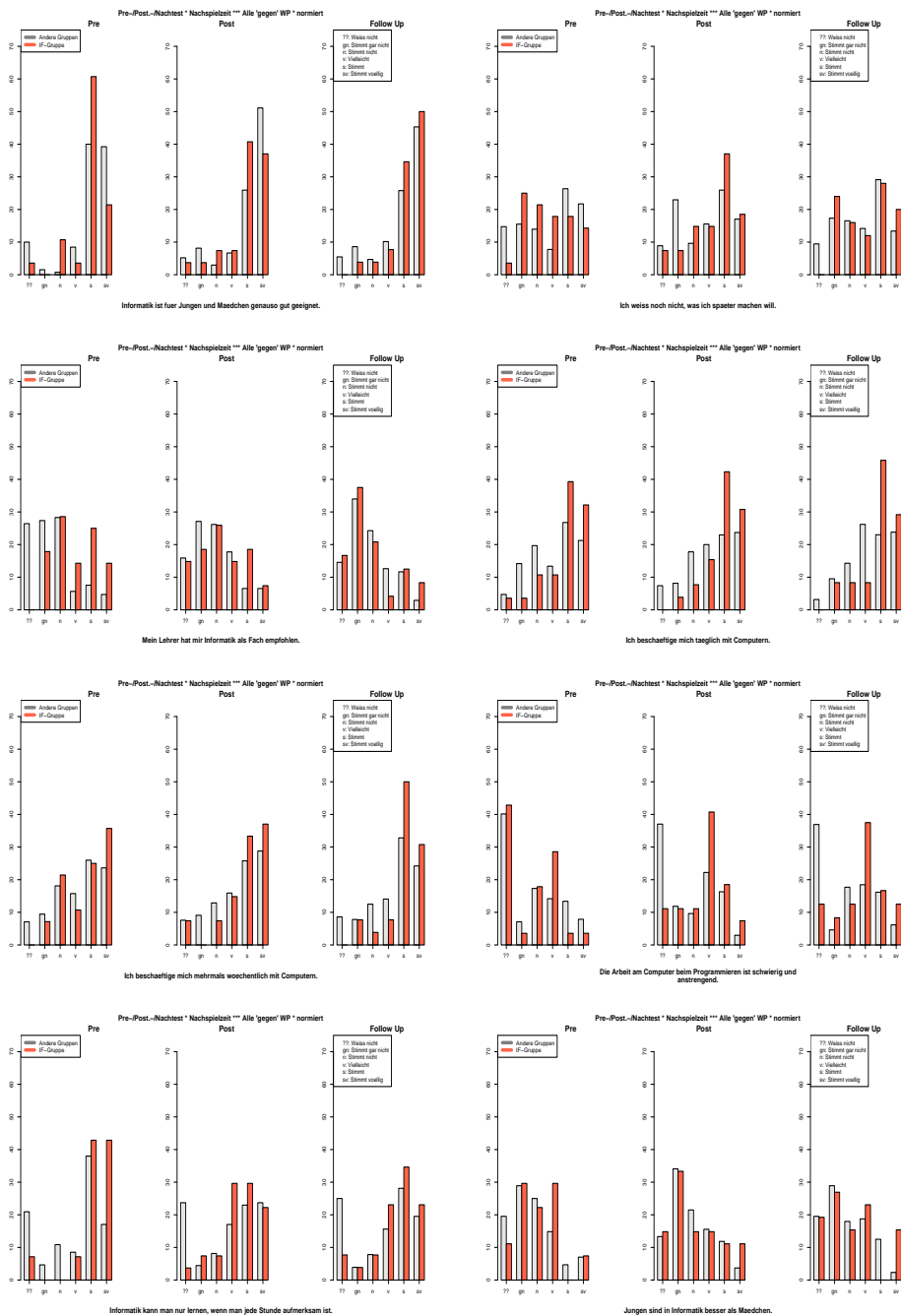


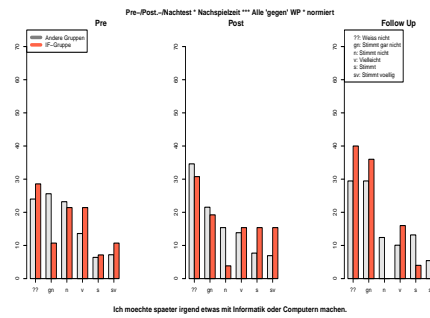
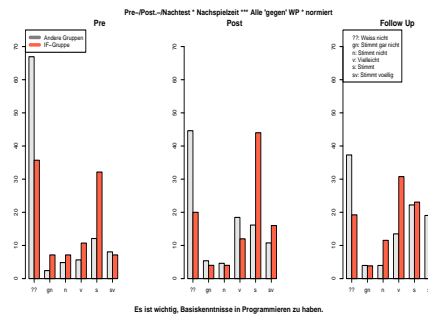
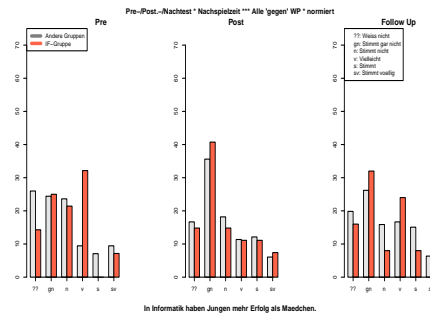
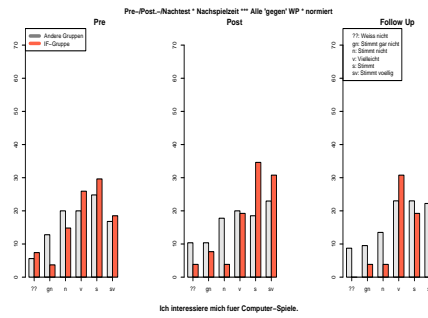
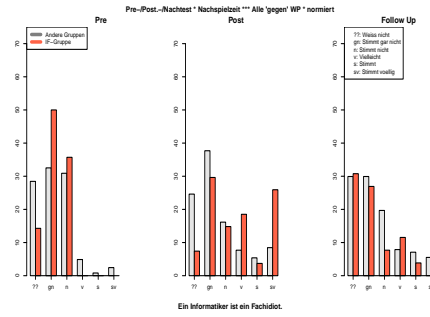
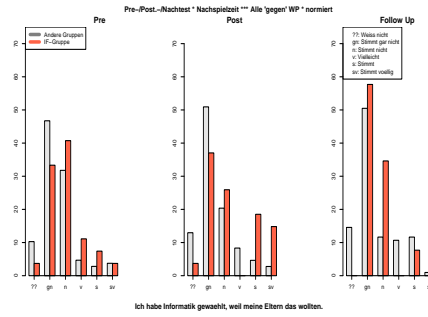
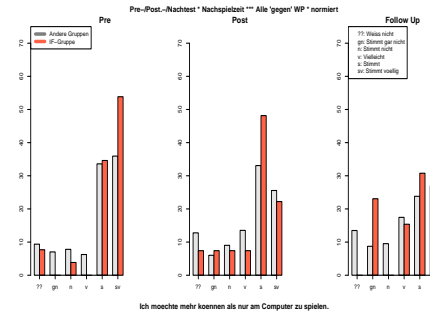
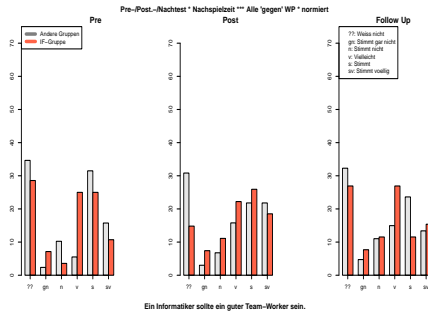


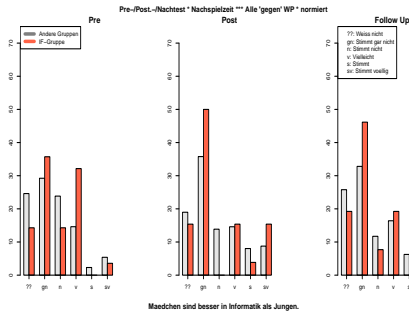
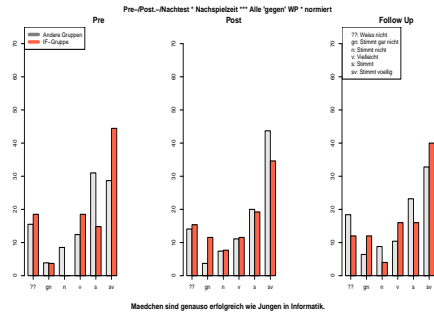
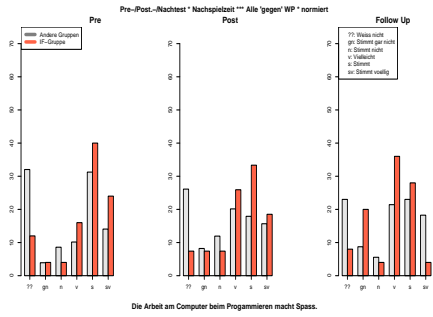
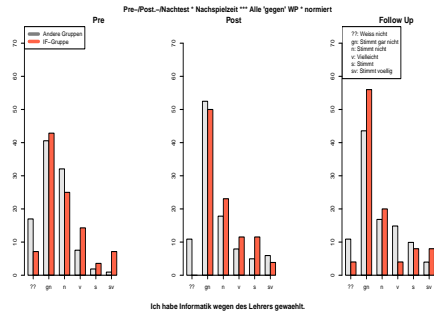
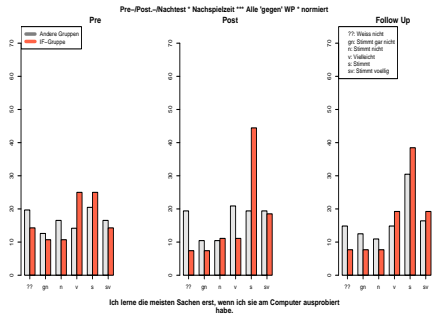


# Vergleich Pre - / Post - / Follow-Up - Test Nachspielzeit

## Informatik-Gruppe im Vergleich zu allen anderen Gruppen







## 11.10 Think Aloud

### 11.10.1 Beispielhafte Interviews

#### Transkription eines Schüler-Interviews an der Fritz-Steinhoff-Gesamtschule in Hagen

Interviewer Holger Danielsik, Datum: 15. Juli 2011

##### Setting

Zur Feststellung von Schülerwissen in der Sekundarstufe 1 wurden einige Interviews mit Schüler der Jahrgangsstufe 6 durchgeführt. Die Interviews in der Fritz-Steinhoff-Gesamtschule Hagen wurden von Holger in einem Abstellraum der Schule durchgeführt. Als Aufnahmegerät diente ein H2-Handy Recorder. Die aufgezeichnete Länge des aktuellen Interviews betrug 8:19 Minuten.

##### Dateien und Verzeichnisse

Der Interviewpartner stimmte der Tonaufnahme zu:

*Holger:* ... irgendwelche Sachen frage, die ihr auch noch gar nicht hattet, dann sagst du einfach und fertig.

*Interviewter:* Okay.

*Holger:* Die erste Frage wäre, was eine Datei ist, am Computer?

*Interviewter:* Eine Datei ist so..., wenn das..., ist halt, ähm wie soll ich das erklären, ähm... es ist sowas, wie ein Bild, nur halt so mit Wörter, also so kann ich nicht so erklären, so.

*Holger:* Aha, ja. Gibt es halt auf jeden Fall so ein paar am Computer.

*Interviewter:* Ja.

*Holger:* Ähm, was ist denn ein Verzeichnis?

*Interviewter:* Verzeichnis ist so etwas wie so ein... wie ein Ordner, wo man etwas reinkopieren kann.

*Holger:* Mhm..., da sind dann mehrere Sachen drin?

*Interviewter:* Ja.

*Holger:* Was zum Beispiel?

*Interviewter:* Mhm... zum Blei... zum Beispiel so..., wenn man irgendwie etwas... auf einer Leinwand oder so etwas haben möchte, das kann man da ja reinkopieren. Zum Beispiel ein Haus oder so.

*Holger:* Und was ist ein Unterschied zwischen einem Verzeichnis und einer Datei?

*Interviewter:* Eine Datei ist... mhm... etwas... so ein Verzeichnis ist... eigentlich ein Ordner

und eine Datei kann da rein. Also es ist etwas, wo irgendetwas passiert.

*Holger:* Genau. Ja, hast du Verzeichnisstruktur schon einmal gehört?

*Interviewter:* Mhm, nee.

*Holger:* Verzeichnisbaum?

*Interviewter:* Ja... das war mit dem Baum, was wir jetzt machen, ne? Kann das sein?

*Holger:* Ich weiß gerade nicht, was ihr jetzt macht, deshalb... Ähm... Ich würde dir dann einfach mal diese beiden Bilder hier zeigen.

*Interviewter:* Ja.

*Holger:* Und du guckst mal, was ist hier der Unterschied... Oder was ist da gleich? Erst einmal.

*Interviewter:* Da sind halt, gleich ist, dass da eigentlich alle... alle Wörter auch da sind, wie hier unten, also... die gleichen Wörter, also.

*Holger:* Aber was ist unterschiedlich?

*Interviewter:* Dass das „findet“ hier eingekreist ist und hier das „Informatik“ blau... umrandet.

*Holger:* Ja... was noch?

*Interviewter:* Ähm... Dass hier, sind die Ordner alle geöffnet und hier sind sie noch alle zu.

*Holger:* Hmm, woran siehst du das?

*Interviewter:* Also hier sind... (gemurmelt-unverständlich- 2:30) Da ist... da ist ja einmal auf Informatik geklickt und da war im Ordner noch andere Ordner drin.

*Holger:* (zustimmendes) Mhm.

*Interviewter:* Also hier „der“, „findet“, „sucht“, „wer“.

*Holger:* Ja, und hier?

*Interviewter:* Äh.. ja das weiß ich, glaube ich, noch nicht.

*Holger:* Ja, da ist in Informatik auch erst einmal der „wer“-Ordner drin.

*Interviewter:* Ja?... Die sind, glaube ich, da ist man auf „findet“ gegangen, aber ich glaube, dass die das so,... aber nee, ich weiß es nicht.

*Holger:* Verwirrend.

*Interviewter:* Ja.

*Holger:* Na gut, dann packen wir die weg.

## **Textverarbeitung**

3:13

*Holger:* Dann eine äh ganz andere Sache. Ihr habt Texte am Computer geschrieben.

*Interviewter:* Ja.

*Holger:* Wenn du jetzt so einen Text schreiben sollst. Mit welchem Programm schreibst du den denn?

*Interviewter:* Mhm, es kommt darauf an, jetzt, mit welchem Programm wir schreiben sollten, aber das sollte man an so einem Text-Dingens ähm... schreiben, also am...

*Holger:* Welches... oder welche kennst du denn da? Womit man Texte schreiben kann? Welche Programme?

*Interviewter:* Äh... Text... ähm.. wie heißt das... ähm... mir fällt gerade, mir fallen gerade keine ein. Mhm... wie hießen die?... Weiß ich gerade nicht.

*Holger:* Ja, aber da kann man dann einen Text so schreiben. Und wenn du eine Zeile geschrieben hast und du willst in die nächste Zeile, was machst du dann?

*Interviewter:* Dann drücke ich, also Eingabe und dann kann ich eigentlich schon in der nächsten weiterschreiben.

*Holger:* Genau. Und dann hast du den Text irgendwann so geschrieben.

*Interviewter:* Ja.

*Holger:* Und dann willst du, aber, dass der dann lieber so aussieht. Wie der untere. Ist der gleiche Text, ne, sieht ja ein bisschen anders aus. Was machst du denn da?

*Interviewter:* Da muss man das halt verschieben, das kann ja eigentlich so bleiben und dann kann man das in der Mitte so äh versuchen zu verschieben. Mit der Leertaste, glaube ich, nach hier vorne. Und dann kannst du das, kann man das gleich so nach rechts schieben.

*Holger:* Das ist eine Möglichkeit. Kennst du da noch eine andere?

*Interviewter:* Äh... (verneinendes) mhm.

*Holger:* Nee? Na gut.

## **Programmierung**

4:50

*Holger:* Kannst du mir denn erzählen, was ein Programm ist?

*Interviewter:* Ein Programm ist, äh... ein Programm ist zum Beispiel, wenn ähm... wenn man, da kann man etwas starten, das heißt so, so wie so ein Spiel auch, sowas halt etwas spielen oder ähm... ja ähm... also ein Programm halt öffnen. Mhm... kann ich nicht so recht erklären.

*Holger:* Mhm, kannst du mir denn sagen, was ein Anwendungsprogramm ist, vielleicht?

*Interviewter:* Mhm..., da kann man... das ist so glaube ich das Gleiche nur das man das dann anwenden kann, also...

*Holger:* Also etwas mit machen kann?

*Interviewter:* Ja.

*Holger:* Mhm. Und was ist ein Betriebssystem?

*Interviewter:* Ein Betriebssystem ist zum Beispiel so ein Arbeitsspeicher, glaube ich, ne? So etwas.

*Holger:* Ja, ein bisschen größer...

*Interviewter:* Externer?

*Holger:* Ja, also Windows ist zum Beispiel ein Betriebssystem.

*Interviewter:* Achso.

*Holger:* Was macht denn so ein Betriebssystem? Was hat das für Aufgaben?

*Interviewter:* Da... da war... da kann man halt vieles machen. Man kann ähm... halt Programme dann auch dadrauf machen. Man kann da das Öffnen und ähm... so ungefähr.

*Holger:* Gut, ja dann, äh... was ist eine Variable?

*Interviewter:* Ähm.. Variable... ähm... ähm... weiß ich gerade nicht, ähm...(13 Sekunden Pause) weiß ich nicht, halt...

*Holger:* Ja, kein Problem. Eine Anweisung?

*Interviewter:* Eine Anweisung ist, wenn man etwas schreibt und das dann ausgeführt wird, ist. Also man schreibt... irgendetwas, also halt eine Anweisung und wenn der das dann ausführt, dann ist das sozusagen eine Anweisung, also wenn man, ähm...

*Holger:* Ja, also eine Aufgabe für einen Computer?

*Interviewter:* Ja genau.

## **Bildverarbeitung**

7:04

*Holger:* Und dann Graphik, was ist das?

*Interviewter:* Eine Graphik ist zum Beispiel... eine... ein Bild auf dem Computer, so etwas.

*Holger:* Kennst du da irgendwie... Dateiformat oder so. Also eine Endung, die so eine Graphik hat?

*Interviewter:* Nö.

*Holger:* Nee. Oder hast du Pixelgraphik schon einmal gehört?

*Interviewter:* Jaja, das sind so kleine Vierecke, die.. die ähm... die kann man dann... daraus kann man dann ähm... Bilder machen, halt.

*Holger:* Wenn man die nebeneinander packt?

*Interviewter:* Ja, genau.

*Holger:* Die Vierecke sind... haben unterschiedliche Farben? Und wenn du die alle zusammenpackst, dann hast du ein großes Bild?

*Interviewter:* Ja.

*Holger:* Genau. Hast du Rastergraphik schon einmal gehört?

*Interviewter:* Was für ein Ding?

*Holger:* Äh Rastergraphik? Oder Vektorgraphik?

*Interviewter:* Vektor kommt mir bekannt vor, aber das andere nicht. Vektor, ähm... was war das noch einmal?... Das weiß ich gerade auch nicht. Die andere habe ich auf jeden Fall nichts drüber gehört.

*Holger:* Okay. Ja dann sind wir auch durch. Vielen Dank.

*Interviewter:* Okay.

[Ende bei 8:19 Minuten]



# Transkription eines Schüler-Interviews an der Fritz-Steinhoff-Gesamtschule in Hagen

Interviewer Wolfgang Paul, Datum: 15. Juli 2011

## Setting

Zur Feststellung von Schülerwissen in der Sekundarstufe 1 wurden einige Interviews mit Schüler der Jahrgangsstufe 6 durchgeführt. Die Interviews in der Fritz-Steinhoff-Gesamtschule Hagen wurden von Wolfgang in einem Materialraum der Schule durchgeführt. Als Aufnahmegerät diente ein H2-Handy Recorder. Die aufgezeichnete Länge des aktuellen Interviews betrug 4:13 Minuten.

## Dateien und Verzeichnisse

Der Interviewpartner stimmte der Tonaufnahme zu:

*Wolfgang:* Gut! (lautes Geräusch vom Mirko) Ähm, als erstes würde ich dich fragen: Was ist eine Datei? Datei auf dem Computer...

*Interviewter:* A...a...irgendwie wahrscheinlich so das Hintergrundbild oder so...

*Wolfgang:* Mh!

*Interviewter:* Vielleicht auch so eine Datei, wenn man was installieren will vielleicht.

*Wolfgang:* Mh!

*Interviewter:* Sonst weiß ich nicht so richtig viel mehr.

*Wolfgang:* Okay! Äh...weißt du, was ein Ordner ist oder ein Verzeichnis?

*Interviewter:* Ordner ist so, wenn man irgendwas in den Ordner reinton will, damit das dadrin gespeichert bleibt und nicht sofort weggeht.

*Wolfgang:* Ähm, weißt du, was ein Ordnerbaum oder eine Ordnerstruktur ist?

*Interviewter:* Mh! (schüttelt den Kopf) Das weiß ich nicht.

*Wolfgang:* Okay! Ähm, ich habe hier zwei Bildchen...Ich werde dir erstmal nur eins zeigen und dich bitten...Guck dir das mal an und sag mir, was du da siehst...Ob das für dich eine Bedeutung hat und, was das bedeutet.

*Interviewter:* (überlegt) (lacht) Verstehe ich irgendwie nicht.

*Wolfgang:* Okay! Ähm...das sagen wir mal, ist der Inhalt von deiner Festplatte auf dem Computer. Da sind ja Sachen gespeichert, wie du gesagt hast...Dateien und Ordner...und dieses Symbol ist ein Zeichen für einen Ordner...

*Interviewter:* Mh!

*Wolfgang:* Das sind also alles Ordner...und das daneben...Hast du eine Idee, was das dann sein könnte?

*Interviewter:* Die Informatik.

*Wolfgang:* Okay, aber das ist ein Ordner und daneben steht „Informatik“. Dann ist das wahrscheinlich?

*Interviewter:* Ein Informatik-Ordner.

*Wolfgang:* Genau, der Name des Ordners. Der Ordner heißt „Informatik“, der (zeigt auf das Papier) heißt „der“, der (zeigt auf das Papier) heißt „findet“, der (zeigt auf das Papier) heißt „sucht“, der (zeigt auf das Papier) heißt „wer“.

*Interviewter:* (lautes Niesen)

*Wolfgang:* Und, ähm... wenn jetzt sozusagen das (eine Lehrerin betritt den Raum und grüßt)... Hallo! ... Das hier drunter steht ... Kannst du mir sagen, was das bedeutet, dass diese Ordner unter dem hier stehen?

*Interviewter:* Hm...(überlegt) (Hintergrundgeräusche)

*Wolfgang:* Nicht stören lassen!

*Interviewter:* Äh...keine Ahnung.

*Wolfgang:* Nee? Okay! Kennst...kannst du hier (zeigt auf das Papier) einen Unterschied erkennen...zwischen den beiden? Und zwischen dem, wie die hier (zeigt auf das Papier) stehen und dem, wie die da (zeigt auf das Papier) stehen?

*Interviewter:* Ja! Da kann ich einen Unterschied herausfinden.

*Wolfgang:* Okay, was ist der Unterschied?

*Interviewter:* Ähm, das erste ist, also dass hier (zeigt auf das Papier) noch sowas ist und da (zeigt auf das Papier)...vier (meint die Minute). Dass das nur „findet“ also so angeklickt ist und nur da „Informatik“.

*Wolfgang:* Mh!

*Interviewter:* Und, ähm...dass die nicht alle so richtig in der Reihenfolge stehen.

*Wolfgang:* Okay! Und ähm...macht das einen Unterschied, dass...oder, was bedeutet das, wenn die untereinander stehen...von links nach rechts runter?

*Interviewter:* Keine Ahnung.

## **Textverarbeitung**

2:38

*Wolfgang:* Okay! Alles klar. Ihr hatte ja dann am Computer auch mal Texte schreiben müssen...selber...in der Fünf.

*Interviewter:* Oh, das habe ich noch nie gemacht am Computer...Texte zu schreiben.

*Wolfgang:* Mh! Okay. Ähm, wenn jetzt, sagen wir mal, dein Lehrer mit dir...zu dir kommt mit so einem Zettel wie ich jetzt und sagt „Schreib das am Computer mit einem Programm

deiner Wahl“...Kannst du mir sagen, ob du da...da irgendwas kennst, mit dem du das machen könntest?

*Interviewter:* Mh! (schüttelt den Kopf)

*Wolfgang:* Nee? Okay!

## **Programmierung**

3:02

*Wolfgang:* Alles klar, dann ist das...Ähm, kannst du mir sagen, was ein Programm ist...ganz allgemein?

*Interviewter:* Programm ist, glaube ich, so wie Internet und so.

*Wolfgang:* Mh! Und was auf dem Computer selber vielleicht ein Programm...Kennst du da irgendwas?

*Interviewter:* Öh...nein. Ich bin ja nicht so oft am Computer!

*Wolfgang:* Okay!

*Interviewter:* (laute Hintergrundgeräusche) Weil wir ziehen ja jetzt bald auch um und der Computer ist abgebaut und den wieder neu dranzubauen, ja, das dauert ein bisschen länger.

*Wolfgang:* Okay! Kannst du mir vielleicht sagen, was eine Variable ist?

*Interviewter:* (sehr laut) Das weiß ich nicht!

*Wolfgang:* Eine Anweisung?

*Interviewter:* He?

*Wolfgang:* Eine Anweisung...am Computer?

*Interviewter:* Keine Ahnung.

*Wolfgang:* Okay! Ähm, Betriebssystem?

*Interviewter:* Betriebs...system...! Hö! Keine Ahnung, nur ich kenne mich fast so gut mit Computerspielen aus. Das weiß ich.

*Wolfgang:* Okay (lacht)! Weißt du, was ein Anwendungsprogramm ist? Hast du das schonmal gehört, den Begriff?

*Interviewter:* (schüttelt den Kopf)

## **Bildverarbeitung**

3:55

*Wolfgang:* Okay! Ähm, weißt du, was eine Grafik oder eine Grafikdatei ist?

*Interviewter:* Hm...nein!

*Wolfgang:* Okay! Pixelgrafik...vielleicht schonmal gehört?

*Interviewter:* Nicht so richtig.

*Wolfgang:* Okay! Vektorgrafik? Das ist die letzte Frage, dann sind wir auch durch.

*Interviewter:* Nee...

*Wolfgang:* Okay! Alles klar! Dankeschön!

*Interviewter:* Fertig schon?

*Wolfgang:* Fertig, mh!

# Transkription eines Schüler-Interviews an der Realschule Halden in Hagen

Interviewer Holger Danielsik, Datum: 12. Juli 2011

## Setting

Zur Feststellung von Schülerwissen in der Sekundarstufe 1 wurden einige Interviews mit Schüler der Jahrgangsstufe 6 durchgeführt. Die Interviews in der Realschule Hagen-Halden wurden von Holger in einem Besprechungsraum der Schule durchgeführt. Als Aufnahmegerät diente ein H2-Handy Recorder. Die aufgezeichnete Länge des aktuellen Interviews betrug 6:16 Minuten.

## Dateien und Verzeichnisse

Der Interviewpartner stimmte der Tonaufnahme zu:

*Holger:* ... damit ich hinterher weiß, was gesagt wurde, ne. Ich würde von dir gerne als erstes wissen: Eine Datei, was ist das am Computer?

*Interviewter:* Ähm, das ist... eine... (lacht) das sind so verschiedene Programme. Also eine Datei, da sind so verschiedene Sachen drauf. Zum Beispiel Photos oder ähm... Videos oder andere Sachen.

*Holger:* Und Verzeichnis, hast du das schon mal gehört? Oder einen Ordner?

*Interviewter:* Ein Ordner, das habe ich schon mal gehört, aber ich komme da jetzt nicht drauf, was das war.

*Holger:* Dann gebe ich dir mal einen Tipp. Und zwar habe ich hier so zwei Sachen. Äh, da sind Ordner drauf. Hast du das schon mal gesehen am Computer?

*Interviewter:* Ja.

*Holger:* Ja, kannst du mir einen Unterschied zwischen den beiden Sachen erzählen? Oder sagen, was da gleich ist?

*Interviewter:* Ähm, ähm.... äh... also hier stehen die alle untereinander und hier stehen sie alle quer. Und... hier in dieser Reihenfolge und da sind sie ein bisschen gedreht worden.

*Holger:* Mhm, anders angeordnet, genau. Ja, also hier kann man ja lesen „Wer sucht, der findet“. Weißt du, warum die hier in der Reihenfolge stehen?

*Interviewter:* Nein.

*Holger:* Nein, na gut. Weißt du, warum eine Sache immer blau ist.

*Interviewter:* Das ist ähm... weil das der ganze Ordner ist? Also...

*Holger:* Also eigentlich, weil ich den einfach gerade angeklickt habe - deshalb.

*Interviewter:* Ach so.

*Holger:* Gar nichts Wildes.

## **Textverarbeitung**

2:18

*Holger:* Na, eine ganz andere Sache ist Textverarbeitung, oder - Texte habt ihr geschrieben am Computer auch, ne?

*Interviewter:* Hm... Texte noch nicht, aber so Tabellen gezeichnet...

*Holger:* Ach, Tabellen, ja... Jetzt habe ich aber so einen Text. Mit welchem Programm würdest du den denn schreiben, wenn du den irgendwie auf den Computer dann kriegen möchtest.

*Interviewter:* (12 Sekunden Denkpause) Komm ich gerade nicht drauf.

*Holger:* Womit habt ihr die Tabellen gemacht?

*Interviewter:* Mit ähm... da war da so ein Zeichen „Tabelle zeichnen“ und da mussten wir drauf klicken und dann ähm... kam da... mussten wir die Zeichen... also mit der Maus mussten wir dann die ähm... Linien ziehen. Dann wurden das Tabellen.

*Holger:* Ach so, dann auch Basteln? Ja, aber so... so einen Text in den Computer habt ihr schon mal... habt ihr noch gar nicht gemacht?

*Interviewter:* Nein.

*Holger:* Na gut, dann äh... gucken wir mal. Was haben wir denn dazu noch? Ich glaube dann... hm, packen wir das mal wieder weg.

## **Programmierung**

3:45

*Holger:* Dann: Kannst du mir denn sagen, was ein Programm ist?

*Interviewter:* Ähm, ein Programm ist so zum Beispiel... also ähm... das... also Firefox ähm... oder Google also äh... Google Earth oder so andere Sachen noch.

*Holger:* Also alles im Internet? Oder gibt es da auch andere Sachen?

*Interviewter:* Auch andere Sachen.

*Holger:* Ja, weißt du denn, was ein Betriebssystem ist?

*Interviewter:* Nicht so.

*Holger:* Nicht so, hast du eine Idee, was das sein könnte?

*Interviewter:* Ähm, damit der Computer also laufen kann?

*Holger:* Mhm, ein Beispiel dafür kennst du nicht? Einen Programmname?

*Interviewter:* Nein.

*Holger:* Mhm, also du hast bestimmt eins gehört, wenn ich dir jetzt sage, dass Windows ein Betriebssystem wäre.

*Interviewter:* Ah ja.

*Holger:* Also, das läuft ja so die ganze Zeit. Aber ähm... hast du von der Variable schon mal was gehört?

*Interviewter:* Ja?

*Holger:* Variable?

*Interviewter:* Nein.

*Holger:* Oder Anweisung?

*Interviewter:* Anweisung ähm... dass man etwas macht, also... (lacht)

*Holger:* Also ein Anweisung am Computer oder für den Computer eigentlich.

*Interviewter:* Also, dass der irgendetwas programmiert oder ausführt.

*Holger:* Dass der irgendetwas ausführt oder machen muss, genau.

## **Bildverarbeitung**

5:26

*Holger:* So, als letztes habe ich noch eine Frage nach einer Grafik. Was eine Grafik ist, weißt du?

*Interviewter:* Ja, ich meine, das wäre dann... wie hell der Computer sein muss, also der Bildschirm.

*Holger:* Ja, das könnte auch, aber... so ein Bild, könnte auch einfach eine Grafik sein. Also so... so etwas.

*Interviewter:* Wie... wie groß das dann sein soll, oder?

*Holger:* Nein, einfach nur so eine Zeichnung am Computer, die nennt man schon Grafik. Ähm, aber Vektorgrafik oder Pixelgrafik, hast du auch noch nicht gehabt?

*Interviewter:* Nein, auch noch nicht gehabt.

*Holger:* Dann kommt das bestimmt noch. Kein Problem, dann wäre ich auch soweit schon durch.

*Interviewter:* Mhm.

*Holger:* Vielen Dank.

*Interviewter:* Ja, ok.

[Ende bei 6:16 Minuten]

# Transkription eines Schüler-Interviews an der Hauptschule Vorhalle in Hagen

Interviewer Jörn Gödel, Datum: 28. Juli 2011

## Setting

Zur Feststellung von Schülerwissen in der Sekundarstufe 1 wurden einige Interviews mit Schüler der Jahrgangsstufe 5 und 6 durchgeführt. Die Interviews in der Hauptschule Hagen-Vorhalle wurden von Jörn in einem Informatikraum der Schule durchgeführt. Als Aufnahmegerät diente ein H2-Handy Recorder. Die aufgezeichnete Länge des aktuellen Interviews betrug 6:04 Minuten.

## Einleitungsphase

Der Interviewpartner stimmte der Tonaufnahme zu:

*Jörn:* Ok, los geht es mit dem fünften Interview. Danke, dass du mitmachst.

*Interviewter:* Ja.

*Jörn:* Ähm ja, fangen wir direkt mit den Fragen an. Was, glaubst du, ist eine Datei?

*Interviewter:* (Denkpause) Weiß ich nicht.

*Jörn:* Keine Ahnung, ok. Ähm... weißt du, was ein Verzeichnis ist?

*Interviewter:* (Denkpause) Das hatte wir einmal in Informatik gemacht.

*Jörn:* Weißt du noch, was das war?

*Interviewter:* (Denkpause)

*Jörn:* Weißt du nicht mehr?

*Interviewter:* Nein.

*Jörn:* Kannst du etwas mit dem Begriff Ordner anfangen? Den schon mal gehört?

*Interviewter:* (Denkpause)

*Jörn:* Auch noch nicht. Ok, dann kommen wir direkt zum ersten Bild. Und zwar sind die beiden relativ ähnlich. Die musst du getrennt betrachten quasi, ne? So, ne? Ähm, siehst du Unterschiede oder Gemeinsamkeiten bei denen? (Pause) Oder, mal ganz vorneweg: Weißt du, was es heißt, warum hier so eine Mappe ist? Was uns das sagt?

*Interviewter:* (Denkpause)

*Jörn:* Keine Ahnung. Naja, egal. Siehst du irgendwelche Gemeinsamkeiten oder Unterschiede?

*Interviewter:* Ja.

*Jörn:* Ja? Sag mal, was ist denn gemeinsam, eine Gemeinsamkeit?



*Interviewter:* Zum Beispiel hier steht „der“ und da auch.

*Jörn:* Mhm.

*Interviewter:* Hier „wer“ und da auch. Da „Informatik“.

*Jörn:* Genau.

*Interviewter:* Und suchen und finden.

*Jörn:* Mhm, ist rechts auch suchen?

*Interviewter:* Nein.

*Jörn:* Ok, ist nicht. Ähm naja, hast du irgend eine Ahnung, warum der hier... die beiden hier schwarz sind, „findet“ und „Informatik“?

*Interviewter:* (Denkpause)

*Jörn:* Nicht? Ähm, weißt du, warum „der“ hier ganz oben steht und „sucht“ als Vorletztes? Sind die irgendwie geordnet, oder sind die da... blindlings reingeschrieben?

*Interviewter:* (lacht) (Denkpause) Hm, das weiß ich auch nicht.

*Jörn:* Macht ja nix. Dann kommen wir zu dem Unteren hier. Das sind beides Texte. Wenn du jetzt am Computer einen Text schreiben möchtest, wie machst du das?

*Interviewter:* Ich gehe auf Excel rein...

*Jörn:* ...auf Excel...

*Interviewter:* ...oder auf Word. (Hier und im Folgenden immer als World ausgesprochen)

*Jörn:* Word geht auch. Kennst du noch andere Programme, mit denen man Texte schreiben kann?

*Interviewter:* PowerPoint.

*Jörn:* Noch irgendeins?

*Interviewter:* Microsoft. (wirklich Mikro deutsch ausgesprochen. . .)

*Jörn:* Microsoft.

*Interviewter:* Mehr weiß ich nicht.

*Jörn:* Ok. Ähm, diese Texte hier, die bestehen ja aus einzelnen Wörtern, ne?

*Interviewter:* Mhm.

*Jörn:* Woraus bestehen denn die Wörter?

*Interviewter:* (Denkpause) Weiß nicht.

*Jörn:* Auch... weißt du nicht. Ähm, wenn du die beiden Texte betrachtest, dann stellst du fest, dass die eigentlich gleich sind, außer dass die alle links am Rand stehen und hier sind die so eingerückt, ne? Also hier sind es sechs Zeichen, hier sind es auch sechs. Nur, die stehen halt viel weiter links, hier unten. Jetzt nehmen wir mal an: Du hast den oberen Text, möchtest aber

den unteren haben. Wie machst du das?

*Interviewter:* Ich ordne die einfach.

*Jörn:* Mhm? Also was genau würdest du dann machen?

*Interviewter:* Ich würde erst einmal diese Xe da zählen und die in der Reihenfolge.

*Jörn:* Die zählen und in der Reihenfolge dann neu sortieren? Ok, gibt es noch eine andere Möglichkeit, wie du es zum Beispiel schaffst, diese zwei Xe weiter nach rechts zu schieben? Also du willst die ja quasi in die Mitte haben, ne? Das ist ja hier genau die Mitte - wie kriege ich das hier in die Mitte rein?

*Interviewter:* Da muss man erst auf Start in äh... Ex..., nein Word macht man das. Da stehen da solche Texte, so Striche. Da gibt es so Mitte.

*Jörn:* Genau.

*Interviewter:* Da muss man drauf klicken, dann kommt das auf die Mitte drauf.

*Jörn:* Genau. Das genau, haben wir gesucht. Gibt es noch eine andere Möglichkeit, wie man die weiter nach links kriegt äh... rechts?

*Interviewter:* Weiß ich nicht.

*Jörn:* Gibt es keine Andere. Ok, dann habe ich noch ein paar letzte Fragen für dich, und zwar: Was, glaubst du, ist ein Programm?

*Interviewter:* (Denkpause) Ja, ein Programm ist so wie zum Beispiel Facebook und so, glaube ich.

*Jörn:* Mhm.

*Interviewter:* Und Google, SchülerVZ und so.

*Jörn:* Mhm, also alles was aus dem Internet kommt, ist ein Programm? Ok. Gibt es noch Programme, für die man kein Internet braucht? Oder brauchen alle Programme Zugang zum Internet?

*Interviewter:* (Denkpause) Das weiß ich nicht.

*Jörn:* Ist äh... Word und Excel kennst du ja. Ist das ein Programm?

*Interviewter:* Ja.

*Jörn:* Sind auch Programme? Ok, ähm... weißt du, was eine Variable ist?

*Interviewter:* Nein.

*Jörn:* Weißt du, was eine Anweisung ist? Gar keine Ahnung, was eine Anweisung ist? Hast du schon mal den Begriff Betriebssystem gehört?

*Interviewter:* Ja.

*Jörn:* Kannst du ein Beispiel für ein Betriebssystem nennen?

*Interviewter:* Ich habe wieder alles vergessen.

*Jörn*: Macht ja nix. Gar... gar keine Ahnung? Ok, na dann sind wir schon fertig. Ok, vielen Dank.  
[Ende nach 6:04 Minuten]



# 11.10.2 Zusammenfassung 'Think Aloud' nach dem Posttest

Zusammenfassung Think Aloud 1. Runde Dateien	Was ist eine Datei? 3u 8q 5v 0k 0u 3q 4v 0k 1u 5q 12v 0k 4u 4q 5v 0k	Was ist ein Verzeichnis? 9u 3q 0v 4k 1u 1q 2v 3k 5u 4q 8v 1k 10u 2q 1v 0k	Was ist ein Ordner? 1u 4q 6v 3k 1u 2q 4v 0k 0u 2q 6v 10k 0u 3q 7v 3k	Unterschied zwischen Datei und Ordner 6u 0q 2v 8k 0u 0q 0v 7k 1u 1q 2v 14k 0u 1q 1v 11k	Unterschied zwischen Verzeichnis und Ordner 8u 0q 1v 7k 0u 0q 1v 6k 3u 2q 0v 13k 1u 0q 0v 12k	Was ist eine Verzeichnisstruktur? 2u 0q 0v 14k 1u 1q 1v 4k 4u 1q 6v 6k 7u 3q 2v 1k	Unterschiede Verzeichnisstrukturen 6u 9q 0v 1k 4u 0q 2v 0k 8u 5q 3v 2k 10u 2q 0v 1k	Gemeinsamkeiten Verzeichnisstrukturen 0u 6q 3v 7k 1u 3q 1v 2k 6u 3q 1v 8k 3u 4q 0v 6k
Zusammenfassung Think Aloud 1. Runde Begriffe	Was ist ein Programm? 3u 8q 4v 0k 0u 1q 5v 0k 1u 0q 1v 1k 6u 2q 5v 0k	Was ist eine Variable? 13u 0q 0v 3k 5u 0q 0v 1k 5u 7q 8v 1k 13u 0q 0v 0k	Was ist eine Anweisung? 11u 2q 0v 3k 1u 1q 3v 1k 1u 2q 14v 1k 6u 3q 4v 0k	Was ist ein Betriebssystem? 7u 7q 1v 1k 0u 2b 4v 0k 3u 5q 9v 0k 4u 1q 0v 7k	Windows ist ein Beispiel für ein Betriebssystem 0u 0q 0v 16k 1u 0q 1v 4k 6u 1b 5v 6k 4u 0b 2v 6k	Was ist ein Anwendungsprogramm 4u 2q 1v 9k 3u 1q 0v 2k 6u 0b 8v 5k 7u 1q 3v 2k		
Zusammenfassung Think Aloud 1. Runde Grafik	Was ist eine Grafik? 0u 0q 0v 16k 1u 1q 0v 4k 0u 1q 6v 12k 6u 2q 1v 3k	Was ist ein Grafcormat? 0u 0q 0v 16k 0u 0q 0v 6k 5u 4q 5v 6k 5u 1q 1v 6k	Was ist ein Pixel? 0u 0q 0v 16k 0u 0q 1v 5k 1u 2q 0v 2k 3u 5q 7v 1k 1u 1q 1v 10k	Was ist eine Pixelmatrix? 0u 0q 0v 16k 4u 0q 0v 2k 6u 5q 7v 1k 9u 0q 0v 1k				
Zusammenfassung Think Aloud 1. Runde Textverarbeitung	Womit würdest du diese Texte schreiben? 0u 1q 15v 0k 0u 1q 5v 0k 4u 4b 10v 1k 3u 4q 6v 0k	Wie zentriert man einen Text automatisch? 7u 0q 3v 6k 2u 0q 2v 1k 3u 0q 10v 5k 6u 3q 2v 2k	Wie zentriert man einen Text von Hand? 2u 3q 7v 4k 0u 0q 2v 4k 1u 1q 6v 10k 0u 0q 6v 7k	Woraus bestehen die Wörter? 0u 1q 0v 15k 0u 0q 0v 6k 5u 0q 5v 9k 0u 0q 3v 10k	Wie macht man einen Zeilenumbruch? 0u 0q 3v 13k 0u 0q 0v 6k 0u 0q 8v 11k 2u 0q 3v 8k	Wie kommt man von dem ersten eingegebenen Text 0u 1q 0v 15k 0u 0q 0v 6k 1u 3q 2v 13k 1u 0q 2v 10k	Kennst du noch andere Textverarbeitungsprogramme? 6u 0q 9v 1k 3u 0q 2v 1k 6u 2q 8v 3k 1u 0q 3v 9k	Was machen diese Programme anders? 1u 1q 1v 13k 0u 0q 1v 5k 0u 1q 5v 12k 1u 0q 3v 9k

# FSG Hagen

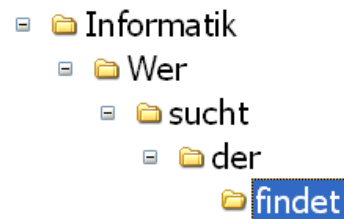
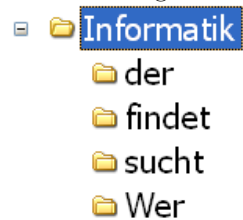
28. Juni 2011

## Dateien und Verzeichnisse

- Was ist eine Datei?
- Was ist ein Verzeichnis?
- Was ist eine Verzeichnisstruktur/baum?

## Verzeichnisstrukturen

Bilder aus Bildungsstandards S. 27



Erläutere die Unterschiede und Gemeinsamkeiten!

## Texte

Betrachte die beiden „Schein-Texte“:

Die Trichter

Zwei Trichter wandeln durch die Nacht.  
Durch ihres Rumpfs verengten Schacht  
fließt weißes Mondlicht  
still und heiter  
auf ihren  
Waldweg  
u. s.  
w.

Die Trichter

Zwei Trichter wandeln durch die Nacht.  
Durch ihres Rumpfs verengten Schacht  
fließt weißes Mondlicht  
still und heiter  
auf ihren  
Waldweg  
u. s.  
w.

- Wie bzw. womit würdest Du diese Texte auf dem Computer erfassen/schreiben?
- Woraus bestehen diese Wörter?
- Wie kommt man von dem ersten eingegebenen Text am einfachsten zum unteren Text?
- Kennst Du neben der Textverarbeitung aus dem Schulunterricht noch andere Programme, mit denen Texte erfasst werden können?

# HS Vorhalle

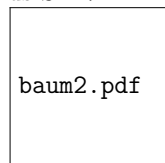
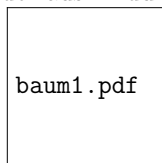
28. Juni 2011

## Dateien und Verzeichnisse

- Was ist eine Datei?
- Was ist ein Verzeichnis?
- Was ist eine Verzeichnisstruktur/baum?

## Verzeichnisstrukturen

Bilder aus Bildungsstandards S. 27



Erläutere die Unterschiede und Gemeinsamkeiten!





# 11.10.3 Zusammenfassung 'Think Aloud' nach dem Posttest und Follow-Up-Test

Zusammenfassung Think Aloud 1. und 2. Runde Dateien	Legende: U: unbekannt/nicht gewusst/ richtig G: gehört/teilweise bekannt/ richtig V: verstanden/richtig U: nicht gefragt	Was ist ein Ordner?	Was ist ein Verzeichnis?	Was ist eine Datei?	Was ist eine Variable?	Was ist eine Anweisung?	Was ist ein Betriebssystem?	Was ist ein Betriebssystem?	Unterschied zwischen Datei und Ordner	Unterschied zwischen Verzeichnis und Ordner	Was ist eine Verzeichnisstruktur?	Unterschiede Verzeichnisstrukturen	Gemeinsamkeiten Verzeichnisstrukturen
HSV		Was ist eine Datei?	Was ist ein Verzeichnis?	3u 8g 5v 0k	9u 3g 0v 4k	1u 4g 6v 5k	6u 0g 2v 8k	8u 0g 1v 7k	2u 0g 0v 14k	6u 1g 0v 1k	0u 6g 3v 7k		
RSH		0u 3g 4v 0k	1u 1g 2v 3k	1u 2g 4v 0k	0u 0g 0v 7k	0u 0g 1v 6k	0u 0g 1v 6k	0u 0g 1v 6k	1u 1g 1v 4k	4u 0g 2v 0k	1u 3g 1v 2k		
FSG WP		1u 5g 12v 0k	5u 4g 8v 1k	0u 2g 6v 10k	1u 1g 2v 14k	3u 2g 0v 13k	3u 2g 0v 13k	3u 2g 0v 13k	4u 1g 6v 6k	8u 5g 3v 2k	6u 3g 1v 8k		
FSG 6.2		4u 4g 5v 0k	10u 2g 1v 0k	0u 3g 7v 3k	0u 1g 1v 11k	1u 0g 0v 12k	1u 0g 0v 12k	1u 0g 0v 12k	7u 3g 2v 1k	10u 2g 0v 1k	3u 4g 0v 6k		
<b>2. Runde Dateien</b>													
HSV		5u 2g 6v	7u 1g 1v 4k	1u 3g 7v 2k	1u 3g 7v 2k	1u 3g 7v 2k	1u 3g 7v 2k	1u 3g 7v 2k	5u 0g 2v 6k	6u 1g 0v 6k	0u 3g 4v 6k		
RSH		0u 1g 3v	0u 0g 2v 2k	1u 0g 3v	1u 0g 3v	1u 0g 3v	1u 0g 3v	1u 0g 3v	1u 0g 3v	1u 0g 2v 1k	4k		
FSG		1u 6g 7v	3u 3g 6v 1k	1u 1g 6v 5k	1u 1g 6v 5k	1u 1g 6v 5k	1u 1g 6v 5k	1u 1g 6v 5k	4u 1g 5v 3k	1u 5g 6v 1k	1u 2g 3v 7k		
FSG Kont		1u 4g 6v	3u 2g 2v 4k	0u 2g 6v 3k	0u 2g 6v 3k	0u 2g 6v 3k	0u 2g 6v 3k	0u 2g 6v 3k	2u 1g 2v 6k	7u 2g 1v 1k	3u 0g 3v 5k		
<b>1. Runde Begriffe</b>													
HSV		Was ist ein Programm?	Was ist eine Variable?	Was ist eine Anweisung?	Was ist ein Betriebssystem?	Was ist ein Betriebssystem?	Was ist ein Betriebssystem?	Windows ist ein Beispiel für ein Betriebssystem.	Was ist ein Betriebssystem?	Was ist ein Anwendungsprogramm?			
RSH		3u 8g 4v 0k	13u 0g 0v 3k	11u 2g 0v 3k	7u 7g 1v 1k	0u 0g 0v 16k	0u 0g 0v 16k	0u 0g 0v 16k	4u 2g 1v 9k	3u 1g 0v 2k			
FSG WP		0u 1g 5v 0k	5u 0g 0v 1k	1u 1g 3v 1k	0u 2b 4v 0k	1u 0g 1v 4k	1u 0g 1v 4k	1u 0g 1v 4k	3u 1g 0v 2k	6u 0b 8v 5k			
FSG 6.2		1u 6g 11v 1k	5u 7g 8v 1k	1u 2g 14v 1k	3u 5g 9v 0k	6u 1b 5v 6k	6u 1b 5v 6k	6u 1b 5v 6k	7u 1g 3v 2k				
<b>2. Runde Begriffe</b>													
HSV		4u 3g 4v 2k	9u 2g 0v 2k	5u 1g 3v 4k	5u 4g 1v 3k	0u 1g 0v 12k	0u 1g 0v 12k	0u 1g 0v 12k					
RSH		1u 1g 2v	4u	0u 1g 3v	1u 0g 3v	4k	4k	4k					
FSG		1u 4g 7v 1k	9u 0g 2v 2k	3u 2g 7v 1k	4u 2g 6v 1k	1u 1g 1v 10k	1u 1g 1v 10k	1u 1g 1v 10k					
FSG Kont		4u 2g 4v 2k	9u 2g	7u 3g 1v	9u 1g 1v	11k	11k	11k					
<b>1. Runde Grafik</b>													
HSV		Was ist eine Grafik?	Was ist ein Grafikformat?	Was ist ein Pixel?	Was ist eine Pixelgrafik?	Was ist eine Vektorgrafik?	Was ist eine Vektorgrafik?	Was ist eine Vektorgrafik?					
RSH		0u 0g 0v 16k	0u 0g 0v 16k	0u 0g 0v 16k	0u 0g 0v 16k	0u 0g 0v 16k	0u 0g 0v 16k	0u 0g 0v 16k					
FSG WP		1u 1g 0v 4k	0u 0g 0v 6k	0u 0g 1v 5k	1u 2g 0v 3k	4u 0g 0v 2k	4u 0g 0v 2k	4u 0g 0v 2k					
FSG 6.2		0u 1g 6v 12k	5u 4g 5v 6k	1u 2g 6v 10k	3u 5g 7v 4k	6u 5g 7v 1k	6u 5g 7v 1k	6u 5g 7v 1k					
		6u 2g 1v 3k	5u 1g 1v 6k	1u 1g 1v 10k	5u 2g 2v 4k	9u 0g 0v 1k	9u 0g 0v 1k	9u 0g 0v 1k					





## 11.10.4 Zusammenfassung 'Think Aloud' am Ende des Schuljahres 2011/2012

422

<b>Zusammenfassung Think Aloud WP If 1. Bis 3. Runde</b>	<b>Dateien</b>							
	Was ist eine Datei?	Was ist ein Verzeichnis?	Was ist ein Ordner?	Unterschied zwischen Datei und Ordner	Unterschied zwischen Verzeichnis und Ordner			
FSG 1. Runde	1u 5g 12v 0k	5u 4g 8v 1k	0u 2g 6v 10k	1u 1g 2v 14k	3u 2g 0v 13k			
FSG 2. Runde	1u 6g 7v	3u 3g 6v 1k	1u 1g 6v 5k					
FSG 3. Runde	1u 5g 15v	4u 4g 6v 8k	2u 3g 13v 4k					
<b>Begriffe</b>								
	Was ist ein Programm?	Was ist eine Variable?	Was ist eine Anweisung?	Was ist ein Betriebssystem?	Windows ist ein Beispiel für ein Betriebssystem. Welche Aufgaben?			
FSG 1. Runde	1u 6g 11v 1k	5u 7g 8v 1k	1u 2g 14v 1k	3u 5g 9v 0k	6u 1b 5v 6k			
FSG 2. Runde	1u 4g 7v 1k	9u 0g 2v 2k	3u 2g 7v 1k	4u 2g 6v 1k	1u 1g 1v 10k			
FSG 3. Runde		13u 7g 2v	4u 2g 16v					
<b>Grafik</b>								
	Was ist eine Grafik?	Was ist ein Grafikformat?	Was ist ein Pixel?	Was ist eine Pixelgrafik?	Was ist eine Vektorgrafik?			
FSG 1. Runde	0u 1g 6v 12k	5u 4g 5v 6k	1u 2g 6v 10k	3u 5g 7v 4k	6u 5g 7v 1k			
FSG 2. Runde			0u 1g 3v 9k	2u 4g 6v 1k	9u 1g 0v 3k			
FSG 3. Runde		1u Rest k	4u 6g 7v 5k	6u 1g 12v 3k	12u 3g 0v 6k			
<b>Textverarbeitung</b>								
	Womit würdest du diese Texte schreiben?	Wie zentriert man einen Text automatisch?	Wie zentriert man einen Text von Hand?	Woraus bestehen die Wörter?	Wie macht man einen Zeilenumbruch?			
FSG 1. Runde	4u 4b 10v 1k	3u 0g 10v 5k	1u 1g 6v 10k	5u 0g 5v 9k	0u 0g 8v 11k			
FSG 2. Runde	2u 2g 9v	2u 0g 7v 4k	1u 0g 4v 8k					
FSG 3. Runde	0u 1g 5v 16k	7u 1g 11v 3k	1 u 2 g 8v 11k					





## Umfrage 4 \* Halbzeit 2 \*\* FSG Klasse 7.2

Nr;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20  
1311;0;5;0;0;3;2;4;-1;-1;0;0;3;-1;5;0;-1;-1;0;2;0  
1312;4;5;5;5;4;4;5;5;-1;1;-1;0;5;1;1;1;5;2;1;4  
1313;4;4;4;0;0;5;3;5;2;3;0;1;5;1;1;0;0;0;0;0  
1314;3;3;4;0;0;0;2;5;0;0;0;0;4;0;2;0;0;0;0;0  
1315;4;5;-1;4;3;5;2;3;0;3;0;5;3;3;0;3;4;0;5;3  
1316;5;5;5;5;0;3;5;5;0;0;0;0;5;5;0;1;-1;3;4;5  
1317;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1;-1  
1318;4;4;5;4;3;4;5;5;3;2;3;4;5;-1;5;4;3;3;5;5  
1319;4;4;0;0;2;4;5;5;2;0;2;4;5;5;5;4;3;0;2;0  
1320;0;5;5;4;0;4;4;5;4;3;5;5;5;5;5;4;4;3;1;5  
1321;1;4;4;0;4;4;4;-1;-1;4;0;0;0;4;4;4;4;4;0;0  
1322;0;4;0;4;5;2;4;2;4;3;0;5;0;4;-1;3;4;-1;2;5  
1323;0;3;3;0;0;1;0;1;1;5;5;0;1;3;1;3;-1;-1;2;4  
1324;0;5;4;5;0;4;4;4;2;0;5;5;4;2;3;0;4;4;2;2  
1325;0;4;0;0;0;4;3;2;0;0;0;0;3;2;0;0;0;0;0;0  
1326;4;4;4;4;4;4;4;3;2;4;5;5;3;3;4;4;2;3;0;4  
1327;0;4;5;4;0;5;0;4;0;5;0;3;4;2;-1;3;0;0;0;3  
1328;3;4;5;5;0;5;4;4;0;4;4;4;4;4;3;4;4;-1;1;4  
1329;-1;0;0;0;0;3;3;5;0;0;0;0;4;0;0;5;0;0;0;0  
1330;1;4;4;2;2;4;4;4;3;0;4;0;4;4;4;0;4;-1;0;0  
1331;5;5;0;5;5;5;5;0;3;5;0;3;5;1;1;1;0;0;0;5  
1332;0;0;5;4;0;3;5;2;0;0;0;0;2;2;5;0;0;0;0;0  
1333;0;5;5;5;0;5;5;4;0;5;0;5;4;5;5;5;2;0;3;0  
1334;4;4;5;4;0;5;4;4;0;4;0;0;5;2;4;2;4;0;0;0  
1335;0;5;0;0;0;4;0;-1;2;0;5;0;4;0;0;0;0;0;2;0  
1336;0;4;4;5;0;4;5;5;0;0;5;0;5;5;5;5;4;0;0;0  
1337;5;5;5;0;0;5;0;5;1;0;0;0;4;1;1;0;4;0;1;0  
1350;4;4;4;4;0;0;2;1;0;0;0;4;3;0;2;2;4;0;0;0

## Umfrage 4 \* Nachspielzeit \*\* RSH Klasse 7a

Nr;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21  
3611;5;1;1;4;4;0;0;0;5;4;3;-1;5;1;0;0;1;1;0;5;5  
3612;5;1;2;5;5;2;5;1;4;3;1;1;3;0;5;1;4;1;2;4;1  
3613;1;2;3;4;5;0;5;4;3;2;1;2;3;4;5;0;5;4;3;2;1  
3614;-1  
3615;-1  
3616;5;5;2;3;4;2;3;2;3;5;1;1;4;4;4;4;4;1;3;0;3  
3617;5;1;1;1;4;0;5;1;5;1;1;1;5;1;0;1;2;1;3;5;1  
3618;5;2;2;5;5;4;4;1;4;5;1;1;4;1;5;5;4;1;4;5;1  
3619;3;3;0;4;4;4;0;0;0;0;0;0;-1;3;2;4;4;4;4;4;4  
3620;5;4;1;2;4;3;4;2;5;5;2;2;2;2;4;0;4;2;4;4;3  
3621;-1  
3622;5;1;1;5;1;0;0;3;5;5;1;4;5;1;5;1;3;1;0;5;1  
3623;4;3;4;5;5;4;3;0;3;5;3;1;4;0;4;4;4;2;3;-1;0  
3624;5;1;1;2;2;0;5;3;0;5;1;1;4;3;4;1;0;1;0;3;3  
3625;4;2;2;4;4;2;3;3;4;4;1;2;3;3;4;1;4;1;1;4;3  
3626;5;3;3;3;5;3;3;3;1;2;3;1;3;4;4;4;5;1;5;3;3  
3627;5;4;3;5;4;3;4;3;4;5;1;2;2;3;4;3;3;3;4;3;3  
3628;5;4;5;4;3;3;3;3;4;2;3;0;2;3;-1;1;1;2;3;-1;1  
3629;5;0;1;1;3;0;0;0;3;3;1;3;5;0;5;1;0;1;0;0;0  
3630;5;1;1;2;2;0;5;1;5;1;1;2;1;1;0;2;4;1;0;5;1  
3631;1;4;2;1;3;2;4;0;2;3;1;4;0;3;5;3;4;3;4;2;4  
3632;5;1;1;5;5;2;4;1;5;2;1;1;5;1;0;4;5;1;0;5;1  
3633;5;1;1;3;4;1;1;-1;-1;5;1;1;5;1;5;3;4;1;5;5;1  
3634;-1  
3635;4;4;4;3;4;3;4;2;4;4;2;2;3;2;0;3;4;2;3;4;2  
3650;5;3;2;5;5;3;3;2;4;5;1;1;2;3;5;5;3;1;5;2;2



## Umfrage 4 \* Elfmeterschiessen \*\* RSH Klasse 7c

Nr;1;2;3;4;5;6  
3511;5;5;2;3;2;5  
3512;1;2;2;5;4;2  
3513;4;2;2;3;3;2  
3514;1;1;5;4;2;5  
3515;0;0;0;0;0;0  
3516;0;6;6;6;6;0  
3517;1;0;0;6;6;6  
3518;0;0;0;0;0;0  
3519;0;0;0;0;0;0  
3520;0;0;0;0;0;3  
3521;5;5;0;5;5;0  
3522;0;0;0;0;0;0  
3523;4;5;1;2;2;2  
3524;4;2;1;2;2;2  
3525;4;2;2;4;3;1  
3526;3;2;2;0;0;2  
3527;3;5;1;5;4;6  
3528;0;0;1;0;0;0  
3529;0;0;0;0;0;0  
3530;0;0;0;0;0;0  
3533;0;0;0;0;0;0  
3534;1;2;2;3;2;2  
3535;0;0;0;0;0;0  
3536;1;5;2;2;3;5  
3537;0;0;0;0;0;0  
3538;5;2;5;3;3;5  
3539;3;5;6;4;4;3  
3550;4;3;3;0;0;0  
3551;0;3;3;5;2;6  
3552;1;1;5;4;4;3

## Umfrage 4 \* Elfmeterschiessen \*\* HSV Klasse 6a

Nr; 1; 2; 3; 4; 5; 6  
2111; 1; 6; 6; 6; 2; 6  
2112; 6; 6; 6; 6; 6; 4  
2113; 1; 1; 4; 1; 6; 4  
2114; 5; 5; 6; 5; 5; 6  
2115; 6; 5; 5; 6; 1; 1  
2116; 1; 5; 5; 5; 5; 6  
2117; 6; 5; 3; 1; 1; 5  
2118; 6; 6; 3; 6; 6; 6  
2119; 5; 2; 6; 6; 6; 6  
2120; -1; -1; -1; -1; -1; -1  
2121; 1; 6; 6; 6; 4; 6  
2122; -1; -1; -1; -1; -1; -1  
2123; -1; -1; -1; -1; -1; -1  
2124; 5; 5; 6; 1; 5; 3  
2125; 5; 6; 2; 1; 3; 5  
2126; 5; 5; 5; 6; 2; 6  
2127; -1; -1; -1; -1; -1; -1  
2128; 1; 6; 6; 2; 1; 6  
2150; 6; 5; 3; 1; 5; 6  
2151; 1; 6; 6; 5; 5; 6  
2153; 1; 6; 6; 6; 6; 6  
2160; 5; 3; 2; 4; 2; 4  
2161; 2; 5; 3; 2; 5; 3  
2162; 6; 5; 5; 3; 2; 5

## Umfrage 4 \* Elfmeterschiessen \*\* HSV Klasse 7a

Nr;1;2;3;4;5;6

2301;-1;-1;-1;-1;-1;-1

2302;2;5;1;5;4;4

2303;6;5;5;5;5;6

2304;-1;-1;-1;-1;-1;-1

2305;-1;-1;-1;-1;-1;-1

2306;-1;-1;-1;-1;-1;-1

2307;2;2;2;6;2;2

2308;3;2;2;3;2;3

2309;6;6;2;1;6;6

2310;1;2;2;2;2;2

2311;-1;-1;-1;-1;-1;-1

2312;1;2;5;5;3;6

2313;-1;-1;-1;-1;-1;-1

2314;1;6;2;1;6;6

2315;1;6;5;0;1;4

2316;3;3;2;2;4;3

2317;5;6;6;5;6;6

2318;5;5;6;5;5;5

2319;6;6;6;6;6;6

2320;1;2;1;4;3;4

2330;6;5;5;6;6;5

2340;0;5;5;2;3;4

2341;5;6;6;6;6;5

2342;6;6;6;6;6;5

## Umfrage 4 \* Elfmeterschiessen \*\* HSV Klasse 7b

Nr; 1; 2; 3; 4; 5; 6  
2211; 1; 2; 2; 5; 2; 6  
2212; 3; 3; 3; 4; 2; 6  
2213; 1; 3; 2; 1; 3; 1  
2214; 1; 6; 3; 5; 5; 6  
2215; 5; 2; 2; 5; 5; 6  
2216; 5; 6; 2; 6; 3; 6  
2217; -1; -1; -1; -1; -1; -1  
2218; 1; 5; 5; 1; 4; 6  
2219; -1; -1; -1; -1; -1; -1  
2220; 6; 2; 6; 6; 6; 6  
2221; -1; -1; -1; -1; -1; -1  
2222; 2; 5; 2; 3; 3; 2  
2223; 1; 6; 2; 6; 4; 6  
2224; 1; 6; 1; 5; 1; 4  
2225; 1; 2; 3; 1; 1; 1  
2226; 4; 5; 3; 1; 4; 3  
2227; 1; 2; 4; 1; 4; 4  
2228; -1; -1; -1; -1; -1; -1  
2229; -1; -1; -1; -1; -1; -1  
2230; 1; 6; 5; 1; 4; 1  
2250; 3; 3; 3; 3; 4; 5  
2251; 1; 5; 2; 1; 4; 1  
2260; 1; 5; 1; 1; 4; 3  
2261; 1; 2; 4; 2; 4; 2  
2262; 1; 5; 5; 1; 4; 4  
2263; 1; 1; 1; 1; 1; 6

## 11.12 Skripte in R

Die Auswertung der Daten erfolgte mit dem statistischen Programmsystem *R*. Für die Auswertung wurden zahlreiche Skripte erstellt, die teilweise ähnlich sind. Für die verschiedenartigen Auswertungen werden hier einige beispielhafte Skripte angegeben.

### 11.12.1 Beispiel einer Auswertung der 'Halbzeit 1'

```
gruppe <- "Endtest * Halbzeit 1 *** Aller Schulen * normiert"

sink("daten_alle_schulen.txt")
sink("daten_alle_schulen_kurzform2_urform.txt")

normieren <- function (frage_daten) {
  ja <- frage_daten == "j"
  nein <- frage_daten == "n"
  unklar <- frage_daten == "w"
  gesamt <- sum(sum(ja), sum(nein), sum(unklar))

  # Normierung bezueglich Prozent
  ja <- sum(ja)/gesamt *100
  nein <- sum(nein)/gesamt*100
  unklar <- sum(unklar)/gesamt *100

  return(balken=c(ja,nein,unklar))
}

fragentext <- read.table("fragestellung.txt")

datendatei_hsv1 <- "daten_hsv1.dat"
daten_hsv1 <- read.table(datendatei_hsv1,header=TRUE,sep=";")
datendatei_hsv2 <- "daten_hsv2.dat"
daten_hsv2 <- read.table(datendatei_hsv2,header=TRUE,sep=";")
datendatei_hsv3 <- "daten_hsv3.dat"
daten_hsv3 <- read.table(datendatei_hsv3,header=TRUE,sep=";")
datendatei_rsh1 <- "daten_rsh1.dat"
daten_rsh1 <- read.table(datendatei_rsh1,header=TRUE,sep=";")
datendatei_rsh2 <- "daten_rsh2.dat"
daten_rsh2 <- read.table(datendatei_rsh2,header=TRUE,sep=";")

datendatei_fsg1 <- "daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1,header=TRUE,sep=";")
datendatei_fsg2 <- "daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2,header=TRUE,sep=";")

#attach(daten)
fragenmax <- 30

fragenr <- 1

pdf("daten_alle_schulen.pdf",width=10,height=6)

while (fragenr < fragenmax) {

frage_daten_hsv1 <- daten_hsv1[fragenr+1]
frage_daten_hsv2 <- daten_hsv2[fragenr+1]
frage_daten_hsv3 <- daten_hsv3[fragenr+1]
```

```

frage_daten_rsh1 <- daten_rsh1[fragenr+1]
frage_daten_rsh2 <- daten_rsh2[fragenr+1]

frage_daten_fsg1 <- daten_fsg1[fragenr+1]
frage_daten_fsg2 <- daten_fsg2[fragenr+1]

fragestellung <- fragentext[fragenr,1]
print ("-----")
print ("")
print (fragestellung)

frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_rest <- rbind(frage_daten_rsh,frage_daten_hsv,frage_daten_fsg2)

par(mar=c(7,5,4,4))

balken_hsv=normieren(frage_daten_hsv)
balken_rsh=normieren(frage_daten_rsh)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)
balken_rest=normieren(frage_daten_rest)

print ("WP-Gruppe:")
#print (summary(frage_daten_fsg1))
print (balken_fsg1)
print ("")

print ("Rest:")
#print (summary(frage_daten_rest))
print (balken_rest)
print ("")

grafik.daten <- rbind(balken_hsv,balken_rsh,balken_fsg2,balken_fsg1)

barplot (grafik.daten, beside=TRUE,,main=gruppe, col=c("grey70","palegreen","pink2","tomato"))
xaxt="n"

axis(1, at=seq(3,15,5),label=c("Ja","Nein","Weiss nicht"))
title(strwrap(fragestellung),outer=FALSE,line=-24)
fragenr <- fragenr + 1
}
dev.off()

```

## 11.12.2 Beispiel einer Auswertung der 'Halbzeit 2'

```
gruppe <- "Endtest * Halbzeit 2 *** Alle Gruppen 'gegen' WP * normiert"

normieren <- function (frage_daten) {
  null <- frage_daten == 0
  eins <- frage_daten == 1
  zwei <- frage_daten == 2
  drei <- frage_daten == 3
  vier <- frage_daten == 4
  fuenf <- frage_daten == 5

  gesamt <- sum(null)+sum(eins)+sum(zwei)+sum(drei)+sum(vier)+sum(fuenf)

  # Normierung bezueglich Prozent
  null <- sum(null)/gesamt *100
  eins <- sum(eins)/gesamt*100
  zwei <- sum(zwei)/gesamt *100
  drei <- sum(drei)/gesamt *100
  vier <- sum(vier)/gesamt*100
  fuenf <- sum(fuenf)/gesamt *100

  return(balken=c(null,eins,zwei,drei,vier,fuenf))
}

fragentext <- read.table("fragestellung.txt")

datendatei_hsv1 <- "daten_hsv1.dat"
daten_hsv1 <- read.table(datendatei_hsv1,header=TRUE,sep=";")
datendatei_hsv2 <- "daten_hsv2.dat"
daten_hsv2 <- read.table(datendatei_hsv2,header=TRUE,sep=";")
datendatei_hsv3 <- "daten_hsv3.dat"
daten_hsv3 <- read.table(datendatei_hsv3,header=TRUE,sep=";")
datendatei_rsh1 <- "daten_rsh1.dat"
daten_rsh1 <- read.table(datendatei_rsh1,header=TRUE,sep=";")
datendatei_rsh2 <- "daten_rsh2.dat"
daten_rsh2 <- read.table(datendatei_rsh2,header=TRUE,sep=";")

datendatei_fsg1 <- "daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1,header=TRUE,sep=";")
datendatei_fsg2 <- "daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2,header=TRUE,sep=";")

#attach(daten)
fragenmax <- 21

fragenr <- 1

pdf("daten_wp_gegen_alle.pdf",width=10,height=6)

while (fragenr < fragenmax) {
  frage_daten_hsv1 <- daten_hsv1[fragenr+1]
  frage_daten_hsv2 <- daten_hsv2[fragenr+1]
  frage_daten_hsv3 <- daten_hsv3[fragenr+1]

  frage_daten_rsh1 <- daten_rsh1[fragenr+1]
  frage_daten_rsh2 <- daten_rsh2[fragenr+1]

  frage_daten_fsg1 <- daten_fsg1[fragenr+1]
  frage_daten_fsg2 <- daten_fsg2[fragenr+1]
```

```

fragestellung <- fragentext[fragenr,1]

frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh,frage_daten_fsg2,frage_daten_fsg2)

par(mar=c(7,5,4,4))

balken_fsg1=normieren(frage_daten_fsg1)
balken_rest=normieren(frage_daten_alle_ohne_wp)

grafik.daten <- rbind(balken_rest,balken_fsg1)

barplot(grafik.daten, beside=TRUE,,main=gruppe, col=c("grey50","tomato"))
xaxt="n"

axis(1, at=seq(2,18.5,3),label=c("Weiss nicht","gar nicht","nicht","weder noch","stimmt","voellig"))
title(strwrap(fragestellung),outer=FALSE,line=-24)
fragenr <- fragenr + 1
}
dev.off()

```

## 11.12.3 Beispiel einer Auswertung der 'Nachspielzeit'

```

gruppe <- "Endtest * Nachspielzeit *** FSG nach Geschlecht normiert"

normieren <- function (frage_daten) {
  eins <- frage_daten == "1"
  zwei <- frage_daten == "2"
  drei <- frage_daten == "3"
  vier <- frage_daten == "4"
  fuenf <- frage_daten == "5"
  null <- frage_daten == "0"
  gesamt <- sum(sum(eins),sum(zwei),sum(drei),sum(vier),sum(fuenf),sum(null))

  # Normierung bezueglich Prozent
  eins <- sum(eins)/gesamt *100
  zwei <- sum(zwei)/gesamt*100
  drei <- sum(drei)/gesamt *100
  vier <- sum(vier)/gesamt *100
  fuenf <- sum(fuenf)/gesamt*100
  null <- sum(null)/gesamt *100

  return(balken=c(null,eins,zwei,drei,vier,fuenf))
}

normieren_geschlecht <- function (frage_daten, stamm_daten, geschlecht) {
  eins <- frage_daten == "1" & stamm_daten == geschlecht
  zwei <- frage_daten == "2" & stamm_daten == geschlecht
  drei <- frage_daten == "3" & stamm_daten == geschlecht
  vier <- frage_daten == "4" & stamm_daten == geschlecht
  fuenf <- frage_daten == "5" & stamm_daten == geschlecht
  null <- frage_daten == "0" & stamm_daten == geschlecht
  gesamt <- sum(sum(eins),sum(zwei),sum(drei),sum(vier),sum(fuenf),sum(null))

  # Normierung bezueglich Prozent
  eins <- sum(eins)/gesamt *100
  zwei <- sum(zwei)/gesamt*100
  drei <- sum(drei)/gesamt *100
  vier <- sum(vier)/gesamt *100
  fuenf <- sum(fuenf)/gesamt*100

```



```

null <- sum(null)/gesamt *100

return(balken=c(null,eins,zwei,drei,vier,fuenf))
}

fragentext <- read.table("fragestellung.txt")

datendatei_fsg1 <- "daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1,header=TRUE,sep=";")
datendatei_fsg2 <- "daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2,header=TRUE,sep=";")

stammdatei_fsg1 <- "../stammdaten/daten_fsg1.dat"
stamm_fsg1 <- read.table(stammdatei_fsg1,header=TRUE,sep=";")
stammdatei_fsg2 <- "../stammdaten/daten_fsg2.dat"
stamm_fsg2 <- read.table(stammdatei_fsg2,header=TRUE,sep=";")

#attach(daten)
fragenmax <- 22

fragenr <- 1
gruppe <- "Pretest * Nachspielzeit *** FSG nach Geschlecht normiert"

pdf("daten_normiert.pdf",width=10,height=6)

while (fragenr < fragenmax) {
frage_daten_fsg1 <- daten_fsg1[fragenr+1]
frage_daten_fsg2 <- daten_fsg2[fragenr+1]

fragestellung <- fragentext[fragenr,1]

fsg1_0 <- frage_daten_fsg1 == 0
fsg1_0_m <- frage_daten_fsg1 == 0 & stamm_fsg1[3] == "m"
fsg1_0_w <- frage_daten_fsg1 == 0 & stamm_fsg1[3] == "w"
fsg1_1 <- frage_daten_fsg1 == 1
fsg1_1_m <- frage_daten_fsg1 == 1 & stamm_fsg1[3] == "m"
fsg1_1_w <- frage_daten_fsg1 == 1 & stamm_fsg1[3] == "w"
fsg1_2 <- frage_daten_fsg1 == 2
fsg1_2_m <- frage_daten_fsg1 == 2 & stamm_fsg1[3] == "m"
fsg1_2_w <- frage_daten_fsg1 == 2 & stamm_fsg1[3] == "w"
fsg1_3 <- frage_daten_fsg1 == 3
fsg1_3_m <- frage_daten_fsg1 == 3 & stamm_fsg1[3] == "m"
fsg1_3_w <- frage_daten_fsg1 == 3 & stamm_fsg1[3] == "w"
fsg1_4 <- frage_daten_fsg1 == 4
fsg1_4_m <- frage_daten_fsg1 == 4 & stamm_fsg1[3] == "m"
fsg1_4_w <- frage_daten_fsg1 == 4 & stamm_fsg1[3] == "w"
fsg1_5 <- frage_daten_fsg1 == 5
fsg1_5_m <- frage_daten_fsg1 == 5 & stamm_fsg1[3] == "m"
fsg1_5_w <- frage_daten_fsg1 == 5 & stamm_fsg1[3] == "w"

fsg2_0 <- frage_daten_fsg2 == 0
fsg2_0_m <- frage_daten_fsg2 == 0 & stamm_fsg2[3] == "m"
fsg2_0_w <- frage_daten_fsg2 == 0 & stamm_fsg2[3] == "w"
fsg2_1 <- frage_daten_fsg2 == 1
fsg2_1_m <- frage_daten_fsg2 == 1 & stamm_fsg2[3] == "m"
fsg2_1_w <- frage_daten_fsg2 == 1 & stamm_fsg2[3] == "w"
fsg2_2 <- frage_daten_fsg2 == 2
fsg2_2_m <- frage_daten_fsg2 == 2 & stamm_fsg2[3] == "m"
fsg2_2_w <- frage_daten_fsg2 == 2 & stamm_fsg2[3] == "w"
fsg2_3 <- frage_daten_fsg2 == 3
fsg2_3_m <- frage_daten_fsg2 == 3 & stamm_fsg2[3] == "m"
fsg2_3_w <- frage_daten_fsg2 == 3 & stamm_fsg2[3] == "w"
fsg2_4 <- frage_daten_fsg2 == 4
fsg2_4_m <- frage_daten_fsg2 == 4 & stamm_fsg2[3] == "m"

```

```

fsg2_4_w <- frage_daten_fsg2 == 4 & stamm_fsg2[3] == "w"
fsg2_5 <- frage_daten_fsg2 == 5
fsg2_5_m <- frage_daten_fsg2 == 5 & stamm_fsg2[3] == "m"
fsg2_5_w <- frage_daten_fsg2 == 5 & stamm_fsg2[3] == "w"

par(mar=c(7,5,4,4))

balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

balken_fsg1_m=normieren_geschlecht(frage_daten_fsg1, stamm_fsg1[3], "m")
balken_fsg1_w=normieren_geschlecht(frage_daten_fsg1, stamm_fsg1[3], "w")
balken_fsg2_m=normieren_geschlecht(frage_daten_fsg2, stamm_fsg2[3], "m")
balken_fsg2_w=normieren_geschlecht(frage_daten_fsg2, stamm_fsg2[3], "w")

grafik.daten <- rbind(balken_fsg1, balken_fsg1_m, balken_fsg1_w, balken_fsg2, balken_fsg2_m, balken_fsg2_w)

barplot(grafik.daten, beside=TRUE, ,main=gruppe, col=c("darkgrey", "blue", "coral", "grey", "lightblue", "lightgrey"),

xaxt="n"
axis(1, at=seq(4,42,7), label=c("Weiss nicht", "gar nicht", "nicht", "weder noch", "stimmt", "voellig"))

mtext("(grau: alle * blau: Jungen * rot: Maedchen **** links WP, rechts KG)",3,0)
mtext("Anzahl der Schueler in %",2,2)
title(strwrap(fragestellung), outer=FALSE, line=-24)
fragenr <- fragenr + 1

}
dev.off()

```

## 11.12.4 Beispiel einer Auswertung des 'Elfmeterschiessens'

```

gruppe <- "Endtest * Elfmeterschiessen *** FSG nach Geschlecht normiert"

normieren <- function (frage_daten) {
  eins <- frage_daten == 1
  zwei <- frage_daten == 2
  drei <- frage_daten == 3
  vier <- frage_daten == 4
  fuenf <- frage_daten == 5
  sechs <- frage_daten == 6

  gesamt <- sum(eins)+sum(zwei)+sum(drei)+sum(vier)+sum(fuenf)+sum(sechs)

  # Normierung bezueglich Prozent
  eins <- sum(eins)/gesamt*100
  zwei <- sum(zwei)/gesamt *100
  drei <- sum(drei)/gesamt *100
  vier <- sum(vier)/gesamt*100
  fuenf <- sum(fuenf)/gesamt *100
  sechs <- sum(sechs)/gesamt *100

  return(balken=c(eins,zwei,drei,vier,fuenf,sechs))
}

normieren_geschlecht <- function (frage_daten, stamm_daten, geschlecht) {
  eins <- frage_daten == 1 & stamm_daten == geschlecht
  zwei <- frage_daten == 2 & stamm_daten == geschlecht
  drei <- frage_daten == 3 & stamm_daten == geschlecht
  vier <- frage_daten == 4 & stamm_daten == geschlecht
  fuenf <- frage_daten == 5 & stamm_daten == geschlecht
  sechs <- frage_daten == 6 & stamm_daten == geschlecht

```

```

gesamt <- sum(sum(eins), sum(zwei), sum(drei), sum(vier), sum(fuenf), sum(sechs))

# Normierung bezueglich Prozent
eins <- sum(eins)/gesamt *100
zwei <- sum(zwei)/gesamt*100
drei <- sum(drei)/gesamt *100
vier <- sum(vier)/gesamt *100
fuenf <- sum(fuenf)/gesamt*100
sechs <- sum(sechs)/gesamt *100

return(balken=c(eins, zwei, drei, vier, fuenf, sechs))
}

fragentext <- read.table("fragestellung.txt")

datendatei_fsg1 <- "daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1, header=TRUE, sep=";")
datendatei_fsg2 <- "daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2, header=TRUE, sep=";")

stammdatei_fsg1 <- "../stammdaten/daten_fsg1.dat"
stamm_fsg1 <- read.table(stammdatei_fsg1, header=TRUE, sep=";")
stammdatei_fsg2 <- "../stammdaten/daten_fsg2.dat"
stamm_fsg2 <- read.table(stammdatei_fsg2, header=TRUE, sep=";")

#attach(daten)
fragenmax <- 7
fragenr <- 1

pdf("daten_normiert.pdf", width=10, height=6)

while (fragenr < fragenmax) {
frage_daten_fsg1 <- daten_fsg1[fragenr+1]
frage_daten_fsg2 <- daten_fsg2[fragenr+1]

frage_daten_fsg <- rbind(frage_daten_fsg1, frage_daten_fsg2)
stamm_fsg <- rbind(stamm_fsg1, stamm_fsg2)

fragestellung <- fragentext[fragenr,1]

par(mar=c(7, 5, 4, 4))

balken_fsg=normieren(frage_daten_fsg)
balken_fsg_m=normieren_geschlecht(frage_daten_fsg, stamm_fsg[3], "m")
balken_fsg_w=normieren_geschlecht(frage_daten_fsg, stamm_fsg[3], "w")

grafik.daten <- rbind(balken_fsg, balken_fsg_m, balken_fsg_w)

barplot(grafik.daten, beside=TRUE, , main=gruppe, col=c("grey", "lightblue", "lightcoral"))

xaxt="n"
axis(1, at=seq(2.5, 24, 4), label=c("L1", "L2", "L3", "L4", "Nicht verstanden", "Keine Idee"))

mtext("(grau: alle * blau: Jungen * rot: Maedchen)", 3, 0)
mtext("Anzahl der Schueler in %", 2, 2)
title(strwrap(fragestellung), outer=FALSE, line=-24)

fragenr <- fragenr + 1
}
dev.off()

```

## 11.12.5 Beispiel einer Vergleichsauswertung

```
gruppe <- "Pre-/Post-/Nachtest * HZ 1 *** RSH/HSV und 6.2'gegen' WP * normiert"

normieren <- function (frage_daten) {
  ja <- frage_daten == "j"
  nein <- frage_daten == "n"
  unklar <- frage_daten == "w"
  gesamt <- sum(sum(ja), sum(nein), sum(unklar))

  # Normierung bezueglich Prozent
  ja <- sum(ja)/gesamt *100
  nein <- sum(nein)/gesamt*100
  unklar <- sum(unklar)/gesamt *100

  return(balken=c(ja, nein, unklar))
}

fragentext <- read.table("fragestellung.txt")

# Daten Pretest
datendatei_hsv1 <- "pretest/daten_hsv1.dat"
daten_hsv1 <- read.table(datendatei_hsv1, header=TRUE, sep=";")
datendatei_hsv2 <- "pretest/daten_hsv2.dat"
daten_hsv2 <- read.table(datendatei_hsv2, header=TRUE, sep=";")
datendatei_hsv3 <- "pretest/daten_hsv3.dat"
daten_hsv3 <- read.table(datendatei_hsv3, header=TRUE, sep=";")
datendatei_rsh1 <- "pretest/daten_rsh1.dat"
daten_rsh1 <- read.table(datendatei_rsh1, header=TRUE, sep=";")
datendatei_rsh2 <- "pretest/daten_rsh2.dat"
daten_rsh2 <- read.table(datendatei_rsh2, header=TRUE, sep=";")

datendatei_fsg1 <- "pretest/daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1, header=TRUE, sep=";")
datendatei_fsg2 <- "pretest/daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2, header=TRUE, sep=";")

# Daten Zwischentest
datendatei_hsv4 <- "zwischentest/daten_hsv1.dat"
daten_hsv4 <- read.table(datendatei_hsv4, header=TRUE, sep=";")
datendatei_hsv5 <- "zwischentest/daten_hsv2.dat"
daten_hsv5 <- read.table(datendatei_hsv5, header=TRUE, sep=";")
datendatei_hsv6 <- "zwischentest/daten_hsv3.dat"
daten_hsv6 <- read.table(datendatei_hsv6, header=TRUE, sep=";")
datendatei_rsh3 <- "zwischentest/daten_rsh1.dat"
daten_rsh3 <- read.table(datendatei_rsh3, header=TRUE, sep=";")
datendatei_rsh4 <- "zwischentest/daten_rsh2.dat"
daten_rsh4 <- read.table(datendatei_rsh4, header=TRUE, sep=";")

datendatei_fsg3 <- "zwischentest/daten_fsg1.dat"
daten_fsg3 <- read.table(datendatei_fsg3, header=TRUE, sep=";")
datendatei_fsg4 <- "zwischentest/daten_fsg2.dat"
daten_fsg4 <- read.table(datendatei_fsg4, header=TRUE, sep=";")

# Daten Endtest
datendatei_hsv7 <- "endtest/daten_hsv1.dat"
daten_hsv7 <- read.table(datendatei_hsv7, header=TRUE, sep=";")
datendatei_hsv8 <- "endtest/daten_hsv2.dat"
daten_hsv8 <- read.table(datendatei_hsv8, header=TRUE, sep=";")
datendatei_hsv9 <- "endtest/daten_hsv3.dat"
daten_hsv9 <- read.table(datendatei_hsv9, header=TRUE, sep=";")
datendatei_rsh5 <- "endtest/daten_rsh1.dat"
daten_rsh5 <- read.table(datendatei_rsh5, header=TRUE, sep=";")
datendatei_rsh6 <- "endtest/daten_rsh2.dat"
daten_rsh6 <- read.table(datendatei_rsh6, header=TRUE, sep=";")
```

```

datendatei_fsg5 <- "endtest/daten_fsg1.dat"
daten_fsg5 <- read.table(datendatei_fsg5,header=TRUE,sep=";")
datendatei_fsg6 <- "endtest/daten_fsg2.dat"
daten_fsg6 <- read.table(datendatei_fsg6,header=TRUE,sep=";")

fragenmax <- 30

fragenr <- 1

hoehe <- 6
pdf("daten_wp_62_rshhsv.pdf",width=10,height=hoehe)

while (fragenr < fragenmax) {

# Einstellung fuer die Grafik
fragestellung <- fragentext[fragenr,1]
par(mar=c(4,3.3,3,2)+.5)
par(bty="o")
par(mfrow=c(1,2))

# Aufbereitung Daten Pretest
frage_daten_hsv1 <- daten_hsv1[fragenr+1]
frage_daten_hsv2 <- daten_hsv2[fragenr+1]
frage_daten_hsv3 <- daten_hsv3[fragenr+1]

frage_daten_rsh1 <- daten_rsh1[fragenr+1]
frage_daten_rsh2 <- daten_rsh2[fragenr+1]

frage_daten_fsg1 <- daten_fsg1[fragenr+1]
frage_daten_fsg2 <- daten_fsg2[fragenr+1]

balken_hsv1=normieren(frage_daten_hsv1)
balken_hsv2=normieren(frage_daten_hsv2)
balken_hsv3=normieren(frage_daten_hsv3)
balken_rsh1=normieren(frage_daten_rsh1)
balken_rsh2=normieren(frage_daten_rsh2)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

# Darstellung Daten Pretest
frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh,frage_daten_hsv)

balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)
balken_rest=normieren(frage_daten_alle_ohne_wp)

grafik.daten <- rbind(balken_rest,balken_fsg2,balken_fsg1)

barplot(grafik.daten, beside=TRUE,ylim=c(0,75),las="1",ylab="Prozent", col=c("grey90","grey80","tomato"))
xaxt="n"
axis(1, at=seq(2.5,11.5,4),label=c("Ja","Nein","Weiss nicht"))
legend("topleft",c("KG extern","KG intern","RF"),col=c("grey90","grey80","tomato"),lwd=c(6,6))

# Aufbereitung Daten Endtest
frage_daten_hsv1 <- daten_hsv7[fragenr+1]
frage_daten_hsv2 <- daten_hsv8[fragenr+1]
frage_daten_hsv3 <- daten_hsv9[fragenr+1]

frage_daten_rsh1 <- daten_rsh5[fragenr+1]
frage_daten_rsh2 <- daten_rsh6[fragenr+1]

frage_daten_fsg1 <- daten_fsg5[fragenr+1]
frage_daten_fsg2 <- daten_fsg6[fragenr+1]

```

```

balken_hsv1=normieren(frage_daten_hsv1)
balken_hsv2=normieren(frage_daten_hsv2)
balken_hsv3=normieren(frage_daten_hsv3)
balken_rsh1=normieren(frage_daten_rsh1)
balken_rsh2=normieren(frage_daten_rsh2)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

# Darstellung Daten Endtest
frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh,frage_daten_hsv)

balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)
balken_rest=normieren(frage_daten_alle_ohne_wp)

grafik.daten <- rbind(balken_rest,balken_fsg2,balken_fsg1)

barplot(grafik.daten, beside=TRUE,ylim=c(0,75),las="1",ylab="Prozent", col=c("grey90","grey80","tomato"),
xaxt="n"
axis(1, at=seq(2.5,11.5,4),label=c("Ja","Nein","Weiss nicht"))

title(strwrap(fragestellung),outer=TRUE,line=-29,cex.main=1.2)
title(strwrap(gruppe),outer=TRUE,line=-2.5)
title("
Pre
Post",cex.main=1.4,outer=TRUE,line=-5)

# Darstellung Daten Pretest2
frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh,frage_daten_hsv)

balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)
balken_rest=normieren(frage_daten_alle_ohne_wp)

grafik.daten <- rbind(balken_rest,balken_fsg2,balken_fsg1)

xaxt="n"
axis(1, at=seq(2.5,11.5,4),label=c("Ja","Nein","Weiss nicht"))

# Darstellung Daten Endtest2
frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh,frage_daten_hsv)

balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)
balken_rest=normieren(frage_daten_alle_ohne_wp)

grafik.daten <- rbind(balken_rest,balken_fsg2,balken_fsg1)

xaxt="n"
title("
Pre
Post",cex.main=1.4,outer=TRUE,line=-5)

fragenr <- fragenr + 1
}
dev.off()

```

## 11.12.6 Variablen-Auswertung

```
#gruppe <- "Signikanztest * HZ 1 *** RSH/HSV und 6.2'gegen' WP * normiert"
gruppe <- ""
normieren <- function (frage_daten) {
  ja <- frage_daten == "j"
  nein <- frage_daten == "n"
  unklar <- frage_daten == "w"
  gesamt <- sum(sum(ja), sum(nein), sum(unklar))

  # Normierung nicht bezueglich Prozent
  ja <- sum(ja)
  nein <- sum(nein)
  unklar <- sum(unklar)

  return(c(ja, nein, unklar))
}

fragentext <- read.table("fragestellung_en.txt")

# Daten Pretest
datendatei_hsv1 <- "pretest/daten_hsv1.dat"
daten_hsv1 <- read.table(datendatei_hsv1, header=TRUE, sep=";")
datendatei_hsv2 <- "pretest/daten_hsv2.dat"
daten_hsv2 <- read.table(datendatei_hsv2, header=TRUE, sep=";")
datendatei_hsv3 <- "pretest/daten_hsv3.dat"
daten_hsv3 <- read.table(datendatei_hsv3, header=TRUE, sep=";")
datendatei_rsh1 <- "pretest/daten_rsh1.dat"
daten_rsh1 <- read.table(datendatei_rsh1, header=TRUE, sep=";")
datendatei_rsh2 <- "pretest/daten_rsh2.dat"
daten_rsh2 <- read.table(datendatei_rsh2, header=TRUE, sep=";")

datendatei_fsg1 <- "pretest/daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1, header=TRUE, sep=";")
datendatei_fsg2 <- "pretest/daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2, header=TRUE, sep=";")

# Daten Endtest
datendatei_hsv7 <- "endtest/daten_hsv1.dat"
daten_hsv7 <- read.table(datendatei_hsv7, header=TRUE, sep=";")
datendatei_hsv8 <- "endtest/daten_hsv2.dat"
daten_hsv8 <- read.table(datendatei_hsv8, header=TRUE, sep=";")
datendatei_hsv9 <- "endtest/daten_hsv3.dat"
daten_hsv9 <- read.table(datendatei_hsv9, header=TRUE, sep=";")
datendatei_rsh5 <- "endtest/daten_rsh1.dat"
daten_rsh5 <- read.table(datendatei_rsh5, header=TRUE, sep=";")
datendatei_rsh6 <- "endtest/daten_rsh2.dat"
daten_rsh6 <- read.table(datendatei_rsh6, header=TRUE, sep=";")

datendatei_fsg5 <- "endtest/daten_fsg1.dat"
daten_fsg5 <- read.table(datendatei_fsg5, header=TRUE, sep=";")
datendatei_fsg6 <- "endtest/daten_fsg2.dat"
daten_fsg6 <- read.table(datendatei_fsg6, header=TRUE, sep=";")

fragenmax <- 30

fragenr <- 1

datei <- file("signifikanz_alles.txt", "w")
cat("Signifikanz einiger Fragen", "Chi-Quadrat-Test", "", "erstellt mit R", "Arno, 27.8.2011", "", "",
    file = datei, sep = "\n")

while (fragenr < fragenmax) {
```

```

fragestellung <- fragentext[fragenr,1]
cat(format(fragestellung), "\n", file = datei)

# Aufbereitung Daten Pretest
frage_daten_hsv1 <- daten_hsv1[fragenr+1]
frage_daten_hsv2 <- daten_hsv2[fragenr+1]
frage_daten_hsv3 <- daten_hsv3[fragenr+1]

frage_daten_rsh1 <- daten_rsh1[fragenr+1]
frage_daten_rsh2 <- daten_rsh2[fragenr+1]

frage_daten_fsg1 <- daten_fsg1[fragenr+1]
frage_daten_fsg2 <- daten_fsg2[fragenr+1]

balken_hsv1=normieren(frage_daten_hsv1)
balken_hsv2=normieren(frage_daten_hsv2)
balken_hsv3=normieren(frage_daten_hsv3)
balken_rsh1=normieren(frage_daten_rsh1)
balken_rsh2=normieren(frage_daten_rsh2)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

#Daten Pretest
frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh,frage_daten_hsv)

fsg1=normieren(frage_daten_fsg1)
fsg2=normieren(frage_daten_fsg2)
rest=normieren(frage_daten_alle_ohne_wp)

pre_fsg1_fsg2_tabelle <- matrix(c(fsg1,fsg2),3,2)
dimnames(pre_fsg1_fsg2_tabelle)<- list( c("Ja","Nein","Weiss nicht"),c("fsg1","fsg2" ) )
pretest_fsg1_fsg2 <- chisq.test(pre_fsg1_fsg2_tabelle)

pre_fsg1_rest_tabelle <- matrix(c(fsg1,rest),3,2)
dimnames(pre_fsg1_rest_tabelle)<- list( c("Ja","Nein","Weiss nicht"),c("fsg1","rest" ) )
pretest_fsg1_rest <- chisq.test(pre_fsg1_rest_tabelle)

cat("Pretest: BT/CG int: ", format(pretest_fsg1_fsg2), "\n", file = datei)
cat("Pretest: BT/CG ext: ", format(pretest_fsg1_rest), "\n", file = datei)

# Aufbereitung Daten Endtest
frage_daten_hsv1 <- daten_hsv7[fragenr+1]
frage_daten_hsv2 <- daten_hsv8[fragenr+1]
frage_daten_hsv3 <- daten_hsv9[fragenr+1]

frage_daten_rsh1 <- daten_rsh5[fragenr+1]
frage_daten_rsh2 <- daten_rsh6[fragenr+1]

frage_daten_fsg1 <- daten_fsg5[fragenr+1]
frage_daten_fsg2 <- daten_fsg6[fragenr+1]

balken_hsv1=normieren(frage_daten_hsv1)
balken_hsv2=normieren(frage_daten_hsv2)
balken_hsv3=normieren(frage_daten_hsv3)
balken_rsh1=normieren(frage_daten_rsh1)
balken_rsh2=normieren(frage_daten_rsh2)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

#Daten Endtest
frage_daten_hsv <- rbind(frage_daten_hsv1,frage_daten_hsv2,frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1,frage_daten_rsh2)

```



```

frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh, frage_daten_hsv)

fsg1=normieren(frage_daten_fsg1)
fsg2=normieren(frage_daten_fsg2)
rest=normieren(frage_daten_alle_ohne_wp)

post_fsg1_fsg2_tabelle <- matrix(c(fsg1, fsg2), 3, 2)
dimnames(post_fsg1_fsg2_tabelle) <- list( c("Ja", "Nein", "Weiss nicht"), c("fsg1", "fsg2" ))
posttest_fsg1_fsg2 <- chisq.test(post_fsg1_fsg2_tabelle)

post_fsg1_rest_tabelle <- matrix(c(fsg1, rest), 3, 2)
dimnames(post_fsg1_rest_tabelle) <- list( c("Ja", "Nein", "Weiss nicht"), c("fsg1", "rest" ))
posttest_fsg1_rest <- chisq.test(post_fsg1_rest_tabelle)

cat("Posttest: BT/CG int: ", format(posttest_fsg1_fsg2), "\n", file = datei)
cat("Posttest: BT/CG ext: ", format(posttest_fsg1_rest), "\n", file = datei)
cat("\n", file = datei)

fragenr <- fragenr + 1

}
cat("\n", "\n", "Dat wars!\n", file = datei)
close(datei)

```

## 11.12.7 Jungen/Mädchen-Auswertung

```
#gruppe <- "Signikanztest * NSZ *** Maedchen/Jungen"
gruppe <- ""

normieren <- function (frage_daten) {
  eins <- frage_daten == "1"
  zwei <- frage_daten == "2"
  drei <- frage_daten == "3"
  vier <- frage_daten == "4"
  fuenf <- frage_daten == "5"
  null <- frage_daten == "0"
  gesamt <- sum(sum(eins), sum(zwei), sum(drei), sum(vier), sum(fuenf), sum(null))

  # Normierung nicht bezueglich Prozent
  eins <- sum(eins)
  zwei <- sum(zwei)
  drei <- sum(drei)
  vier <- sum(vier)
  fuenf <- sum(fuenf)
  null <- sum(null)

  return(c(eins, zwei, drei, vier, fuenf, null))
}

normieren_geschlecht <- function (frage_daten, stamm_daten, geschlecht) {
  eins <- frage_daten == "1" & stamm_daten == geschlecht
  zwei <- frage_daten == "2" & stamm_daten == geschlecht
  drei <- frage_daten == "3" & stamm_daten == geschlecht
  vier <- frage_daten == "4" & stamm_daten == geschlecht
  fuenf <- frage_daten == "5" & stamm_daten == geschlecht
  null <- frage_daten == "0" & stamm_daten == geschlecht
  gesamt <- sum(sum(eins), sum(zwei), sum(drei), sum(vier), sum(fuenf), sum(null))

  # Normierung nicht bezueglich Prozent
  eins <- sum(eins)
  zwei <- sum(zwei)
  drei <- sum(drei)
  vier <- sum(vier)
  fuenf <- sum(fuenf)
  null <- sum(null)

  return(c(eins, zwei, drei, vier, fuenf, null))
}

fragentext <- read.table("fragestellung_en.txt")

# Daten Pretest
datendatei_hsv1 <- "pretest/daten_hsv1.dat"
daten_hsv1 <- read.table(datendatei_hsv1, header=TRUE, sep=";")
datendatei_hsv2 <- "pretest/daten_hsv2.dat"
daten_hsv2 <- read.table(datendatei_hsv2, header=TRUE, sep=";")
datendatei_hsv3 <- "pretest/daten_hsv3.dat"
daten_hsv3 <- read.table(datendatei_hsv3, header=TRUE, sep=";")
datendatei_rsh1 <- "pretest/daten_rsh1.dat"
daten_rsh1 <- read.table(datendatei_rsh1, header=TRUE, sep=";")
datendatei_rsh2 <- "pretest/daten_rsh2.dat"
daten_rsh2 <- read.table(datendatei_rsh2, header=TRUE, sep=";")

datendatei_fsg1 <- "pretest/daten_fsg1.dat"
daten_fsg1 <- read.table(datendatei_fsg1, header=TRUE, sep=";")
datendatei_fsg2 <- "pretest/daten_fsg2.dat"
daten_fsg2 <- read.table(datendatei_fsg2, header=TRUE, sep=";")

# Daten Endtest
datendatei_hsv7 <- "endtest/daten_hsv1.dat"
```

```

daten_hsv7 <- read.table(datendatei_hsv7, header=TRUE, sep="; ")
datendatei_hsv8 <- "endtest/daten_hsv2.dat"
daten_hsv8 <- read.table(datendatei_hsv8, header=TRUE, sep="; ")
datendatei_hsv9 <- "endtest/daten_hsv3.dat"
daten_hsv9 <- read.table(datendatei_hsv9, header=TRUE, sep="; ")
datendatei_rsh5 <- "endtest/daten_rsh1.dat"
daten_rsh5 <- read.table(datendatei_rsh5, header=TRUE, sep="; ")
datendatei_rsh6 <- "endtest/daten_rsh2.dat"
daten_rsh6 <- read.table(datendatei_rsh6, header=TRUE, sep="; ")

datendatei_fsg5 <- "endtest/daten_fsg1.dat"
daten_fsg5 <- read.table(datendatei_fsg5, header=TRUE, sep="; ")
datendatei_fsg6 <- "endtest/daten_fsg2.dat"
daten_fsg6 <- read.table(datendatei_fsg6, header=TRUE, sep="; ")

# Stammdaten
stammdatei_fsg1 <- "../stammdaten/daten_fsg1.dat"
stamm_fsg1 <- read.table(stammdatei_fsg1, header=TRUE, sep="; ")
stammdatei_fsg2 <- "../stammdaten/daten_fsg2.dat"
stamm_fsg2 <- read.table(stammdatei_fsg2, header=TRUE, sep="; ")

fragenmax <- 30
fragenr <- 1

datei <- file("signifikanz_boys_girls.txt", "w")
cat("Signifikanz einiger Fragen", "Chi-Quadrat-Test", "", "erstellt mit R", "Arno, 28.8.2011", "", "",
    file = datei, sep = "\n")

while (fragenr < fragenmax) {

fragestellung <- fragentext[fragenr,1]
cat(format(fragestellung), "\n", file = datei)

# Aufbereitung Daten Pretest
frage_daten_hsv1 <- daten_hsv1[fragenr+1]
frage_daten_hsv2 <- daten_hsv2[fragenr+1]
frage_daten_hsv3 <- daten_hsv3[fragenr+1]

frage_daten_rsh1 <- daten_rsh1[fragenr+1]
frage_daten_rsh2 <- daten_rsh2[fragenr+1]

frage_daten_fsg1 <- daten_fsg1[fragenr+1]
frage_daten_fsg2 <- daten_fsg2[fragenr+1]

balken_hsv1=normieren(frage_daten_hsv1)
balken_hsv2=normieren(frage_daten_hsv2)
balken_hsv3=normieren(frage_daten_hsv3)
balken_rsh1=normieren(frage_daten_rsh1)
balken_rsh2=normieren(frage_daten_rsh2)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

#Daten Pretest
frage_daten_hsv <- rbind(frage_daten_hsv1, frage_daten_hsv2, frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1, frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh, frage_daten_hsv)

fsg1=normieren_geschlecht(frage_daten_fsg1, stamm_fsg1[3], "w")
fsg2=normieren_geschlecht(frage_daten_fsg2, stamm_fsg2[3], "w")

pre_fsg1_fsg2_tabelle <- matrix(c(fsg1, fsg2), 6, 2)
dimnames(pre_fsg1_fsg2_tabelle) <- list( c("an", "n", "p", "r", "ar", "?"), c("fsg1", "fsg2" ))
pretest_fsg1_fsg2 <- chisq.test(pre_fsg1_fsg2_tabelle)

```

```

cat("Maedchen Pretest: BT/CG int: ", format(pretest_fsg1_fsg2), "\n", file = datei)

fsg1=normieren_geschlecht (frage_daten_fsg1, stamm_fsg1[3], "m")
fsg2=normieren_geschlecht (frage_daten_fsg2, stamm_fsg2[3], "m")

pre_fsg1_fsg2_tabelle <- matrix(c(fsg1, fsg2), 6, 2)
dimnames(pre_fsg1_fsg2_tabelle) <- list( c("an", "n", "p", "r", "ar", "?"), c("fsg1", "fsg2" ))
pretest_fsg1_fsg2 <- chisq.test(pre_fsg1_fsg2_tabelle)

cat("Jungen Pretest: BT/CG int: ", format(pretest_fsg1_fsg2), "\n", file = datei)

# Aufbereitung Daten Endtest
frage_daten_hsv1 <- daten_hsv7[fragenr+1]
frage_daten_hsv2 <- daten_hsv8[fragenr+1]
frage_daten_hsv3 <- daten_hsv9[fragenr+1]

frage_daten_rsh1 <- daten_rsh5[fragenr+1]
frage_daten_rsh2 <- daten_rsh6[fragenr+1]

frage_daten_fsg1 <- daten_fsg5[fragenr+1]
frage_daten_fsg2 <- daten_fsg6[fragenr+1]

balken_hsv1=normieren(frage_daten_hsv1)
balken_hsv2=normieren(frage_daten_hsv2)
balken_hsv3=normieren(frage_daten_hsv3)
balken_rsh1=normieren(frage_daten_rsh1)
balken_rsh2=normieren(frage_daten_rsh2)
balken_fsg1=normieren(frage_daten_fsg1)
balken_fsg2=normieren(frage_daten_fsg2)

#Daten Endtest
frage_daten_hsv <- rbind(frage_daten_hsv1, frage_daten_hsv2, frage_daten_hsv3)
frage_daten_rsh <- rbind(frage_daten_rsh1, frage_daten_rsh2)
frage_daten_alle_ohne_wp <- rbind(frage_daten_rsh, frage_daten_hsv)

fsg1=normieren_geschlecht (frage_daten_fsg1, stamm_fsg1[3], "w")
fsg2=normieren_geschlecht (frage_daten_fsg2, stamm_fsg2[3], "w")

post_fsg1_fsg2_tabelle <- matrix(c(fsg1, fsg2), 6, 2)
dimnames(post_fsg1_fsg2_tabelle) <- list( c("an", "n", "p", "r", "ar", "?"), c("fsg1", "fsg2" ))
posttest_fsg1_fsg2 <- chisq.test(post_fsg1_fsg2_tabelle)

cat("Maedchen Posttest: BT/CG int: ", format(pretest_fsg1_fsg2), "\n", file = datei)

fsg1=normieren_geschlecht (frage_daten_fsg1, stamm_fsg1[3], "m")
fsg2=normieren_geschlecht (frage_daten_fsg2, stamm_fsg2[3], "m")

post_fsg1_fsg2_tabelle <- matrix(c(fsg1, fsg2), 6, 2)
dimnames(post_fsg1_fsg2_tabelle) <- list( c("an", "n", "p", "r", "ar", "?"), c("fsg1", "fsg2" ))
posttest_fsg1_fsg2 <- chisq.test(post_fsg1_fsg2_tabelle)

cat("Jungen Posttest: BT/CG int: ", format(pretest_fsg1_fsg2), "\n", file = datei)

fragenr <- fragenr + 1
}
cat("\n", "\n", "Dat wars!\n", file = datei)
close(datei)

```



## 11.13 Lehrpläne an der HS Vorhalle und der RS Halden

### 11.13.1 Lehrplan Informatik an der Hauptschule Vorhalle (Auszug)



## Ganztagshauptschule Vorhalle

Gemeinschaftshauptschule der Stadt Hagen – Sekundarstufe I

Vossacker 23 - 58089 Hagen

02331/301531 - 02331/305013

eMail: ghs-vorhalle@gmx.de

---

## Schulinterner Lehrplan für das Fach Informatik

### Einleitung

Informations- und Kommunikationstechnologien sowie die globale Vernetzung verändern kontinuierlich unsere Arbeitswelt. Die Fähigkeit Informatiksysteme sachgerecht, selbstständig und kreativ zu beherrschen ist eine Schlüsselqualifikation, über die SchülerInnen heute in unserer Gesellschaft verfügen müssen. Die GHS Vorhalle sieht es als Aufgabe an, diese Zusammenhänge durchschaubar zu machen und ein breites Spektrum an Einsatz- und Verwendungsmöglichkeiten zu vermitteln.

### Ausgangslage

Die GHS Vorhalle verfügt über einen komplett ausgestatteten Informatikraum mit 15 Arbeitsplätzen und einen weiteren mit 6 Arbeitsplätzen, der aber auch als Differenzierungsraum genutzt wird.

Das Fach Informatik wird seit Beginn des Schuljahres 2007/08 durchgängig in allen Jahrgangsstufen mit 2 Wochenstunden (in der Regel Doppelstunden) von 4 KollegInnen unterrichtet.

### Perspektiven und Zielsetzungen

Das angestrebte Ziel ist die Erlangung von Medienkompetenz.

Die SchülerInnen sollen:

- Kompetenzen im Umgang mit dem Computer erwerben
- neue Formen des Lehrens und Lernens erproben und üben
- ihre Lerndefizite durch Lernprogramme aufarbeiten
- verschiedene Anwendungsprogramme sinnvoll, themenbezogen, kreativ und sicher nutzen
- vielfältige Informationen selbstständig recherchieren können
- ihre Mediennutzung kritisch hinterfragen
- individuell gefördert werden mit Unterstützung von Textverarbeitungs- und Lernprogrammen
- zur Selbstständigkeit und Selbsttätigkeit erzogen werden

**Jahrgang 5/6**

<b>Themen</b>	<b>Inhalte</b>
Betriebssystem	<p>Erlernen der Windows-Oberfläche anhand des Malprogramms Paint  Rechner starten und herunterfahren.  Programme starten und beenden.  Grundbegriffe von Windows (Menübedienung, Symbole, Fenster-Elemente)  Elementare Fenstertechniken (Verschieben, Größe ändern usw.)  Bilder (Dokumente) speichern und öffnen.  Bilder (Dokumente) drucken.</p>
PC-Arbeitsplatz	<p>Hauptbestandteile eines Computers  Ein- und Ausgabegeräte  Speichermedien  Tastatur</p>
Textverarbeitung	<p>Grundeinstellungen  Texteingabe  Textkorrektur  Markieren von Text  Zeichenformatierungen  Absatzformatierungen (Textausrichtung)  Texte umstellen  Dokumente öffnen, speichern und drucken  Einfache Tabellen erstellen (z. B. Stundenplan)</p>
Internet	<p>Einsatzmöglichkeiten  Arbeiten mit dem Internet-Explorer (Suchmaschinen für Kinder, Favoriten verwalten, Recherchieren im Internet)  Gefahren im Internet (Chatten, Kostenfallen...) evtl. mit externem Partner</p>
Lernprogramme	<p>Einführung in Förderprogramme</p>





## 11.13.2 Lehrplan Informatik an der Realschule Halden (Auszug)

### **1. Unterrichtsangebote an der Realschule Halden**

Das Fach Informatik wird an der Realschule Halden als Neigungsschwerpunkt in den Klassen 7, 8, 9 und 10, sowie in Form einer AG in den Klassen 9 und 10 angeboten und unterrichtet. Die Klasse 6 macht einen PC-Führerschein (TIT-Kurs).

In dem Kurs der Jahrgangsstufe 7 und 8 werden 6 Klassenarbeiten, in denen der Jahrgangsstufen 9 und 10 werden jeweils 5 Klassenarbeiten geschrieben, wobei 3 auf das erste Halbjahr und 2 auf das zweite entfallen. Projektarbeiten können schriftliche Klassenarbeiten ersetzen.

### **2. Ziele des Informatikunterrichts**

Der Informatikunterricht dient in erster Linie dazu, die Schüler und Schülerinnen mit den neuen Technologien vertraut zu machen. Dabei müssen im Unterricht die Vorkenntnisse der Schülerschaft sowie die Ausstattung und – damit verbunden – die Möglichkeiten der Schule berücksichtigt werden.

### **3. Themenbereiche des Informatikunterrichts**

Die Themen des Informatikunterrichts sind folgender Themenbereichen zuzuordnen (vgl. Richtlinien und Lehrpläne des Landes NRW):

- I Vernetzte Information und Kommunikation, neue Medien
- II Prozessdatenverarbeitung und Automatisierung
- III Anwendungs- und Programmiersystem
- IV Modellbildung, Simulation, Künstliche Intelligenz
- V Algorithmik, Hardware

Der Schwerpunkt in den AGs liegt im Bereich der Anwendungs- und Programmiersysteme, wobei insbesondere der Umgang mit der „Bürosoftware“ (Textverarbeitung, Datenverarbeitung und Tabellenkalkulation) im Vordergrund steht.

Im Neigungsschwerpunkt liegen für die Jahrgangsstufe 7, 8 und 9 die Schwerpunkte in den Themenbereichen III, I und V und für die Klassen 10 in den Themenbereich II und IV.

Im PC-Führerschein ist das Erlangen von Computer-Grundkenntnissen das Ziel. Dies soll Grundlage sein, um den Computer auch in anderen Fächern zu nutzen.

### **4. Aspekte des fächerübergreifenden Unterrichts**

Anbindungen an Inhalte anderer Fächer ergeben sich aus der Unterrichtsarbeit und werden zurzeit in Mathematik, Sozialwissenschaft, Deutsch, Kunst und Musik gezielt abgesprochen und durchgeführt.

## 5. Unterrichtsinhalte für die Jahrgangsstufe 6 – PC-Führerschein



# Computer-Führerschein der Klasse 6

Name: \_\_\_\_\_

Datum: \_\_\_\_\_

Lernbereich	Erfolgreich
<p>1. <i>Sich anmelden</i></p> <ul style="list-style-type: none"> <li>• erste Anmeldung im Computerraum der Schule</li> <li>• Benutzername eingeben, neues Passwort finden</li> <li>• Kennenlernen des Desktop</li> <li>• Bedienung der Maus</li> </ul> <p>2. <i>Sich abmelden</i></p> <ul style="list-style-type: none"> <li>• Computer ordnungsgemäß „herunterfahren“</li> </ul> <p>3. <i>Ordner anlegen</i></p> <ul style="list-style-type: none"> <li>• Dateien verschieben</li> </ul>	<p>✓</p> <p>✓</p> <p>✓</p>
<p>4. <i>Bildverarbeitung</i></p> <ul style="list-style-type: none"> <li>• Erstellen eines Mandalas</li> <li>• Weihnachtsbilder</li> <li>• Raumplanung</li> </ul>	<p>✓</p>
<p>5. <i>Lernen im Internet</i></p> <ul style="list-style-type: none"> <li>• Internet-Ralleys</li> <li>• Sicherheit im Netz und Möglichkeiten des Internets</li> <li>• Arbeiten mit Suchmaschinen</li> </ul>	<p>✓</p>
<p>6. <i>Texte schreiben</i></p> <ul style="list-style-type: none"> <li>• Werkzeuge zur Textgestaltung ausprobieren in einem Tiersteckbrief</li> <li>• Einen Comic erstellen</li> <li>• Erstellen eines Stundenplans</li> </ul>	<p>✓</p>
<p>7. <i>Tabellenkalkulation</i></p> <ul style="list-style-type: none"> <li>• Auswerten eines Schätzwettbewerbs</li> <li>• Erstellen eines Kreis- und Säulendiagramms</li> <li>• Rechnung erstellen</li> </ul>	<p>✓</p>
<p>8. <i>Homepage</i></p> <ul style="list-style-type: none"> <li>• Designen und Erstellen einer eigenen Internet-Seite</li> </ul>	<p>✓</p>

# Literaturverzeichnis

- [ABZ07] ALBRECHT BEUTELSPACHER, M.A.Z. ; ZSCHIEGNER, M.A.: *Diskrete Mathematik für Einsteiger: Mit Anwendungen in Technik und Informatik. Mit Lösungen der Übungsaufgaben.* Vieweg+Teubner Verlag, 2007
- [ADH<sup>+</sup>00] ABROSIMOV, Leonid I. ; DEUTSCHMANN, Jörg ; HORN, Werner ; REIF, Holger ; RESCHKE, Dietrich ; SCHILLER, Jochen ; SEITZ, Jochen ; KRÜGER, Gerhard (Hrsg.) ; RESCHKE, Dietrich (Hrsg.): *Lehr- und Übungsbuch Telematik.* Leipzig : Fachbuchverlag, 2000. – ISBN 3–446–21053–9
- [Ae11] ALT, Tina ; ET.AL., Bernd C.: *Selbstverständlich Physik: Lehrbuch für die gymnasiale Oberstufe.* Berlin, Mannheim : Duden-Schulbuchverlag, 2011
- [Aeb69] AEBLI, Hans: Die geistige Entwicklung als Funktion von Anlage, Reifung, Umwelt- und Erziehungsbedingungen. In: *Begabung und Lernen*[RA69],
- [Aeb76] AEBLI, Hans: *Grundformen des Lehrens: Eine allgemeine Didaktik auf Kognitionspsychologischer Grundlage.* 9. stark erweiterte und umgearbeitete Auflage. Klett, 1976 (Pädagogik - Einführung)
- [Aeb98] AEBLI, Hans: *Zwölf Grundformen des Lehrens: Eine Allgemeine Didaktik auf psychologischer Grundlage. Medien und Inhalte didaktischer Kommunikation, der Lernzyklus.* 10. Auflage. Klett-Cotta, 1998
- [Aig06] AIGNER, Martin: *Diskrete Mathematik.* Vieweg+Teubner Verlag, 2006 (Vieweg Studium: Aufbaukurs Mathematik). – ISBN 9783834800848
- [AK05] ALBERTS, Gerard ; KANDERS, Rainer: Ik liet de kinderen wél iets leren, Interview met P.M. van Hiele. In: *Nieuw Archief voor Wiskunde* 5 (2005), Nr. 3, S. 247–251
- [AKA<sup>+</sup>00] ANDERSON, Lorin W. ; KRATHWOHL, David R. ; AIRASIAN, Peter W. ; CRUIKSHANK, Kathleen A. ; MAYER, Richard E. ; PINTRICH, Paul R. ; RATHS, James ; WITTRICK, Merlin C.: *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition.* 2. Allyn & Bacon, 2000
- [AKHK<sup>+</sup>90] ALTERMANN-KÖSTER, Marita ; HOLTAPPELS, Heinz G. ; KANDERS, Michael ; PFEIFFER, Hermann ; DE WITT, Claudia: *Bildung über Computer?* Weinheim : Juventa Verlag, 1990

- [Arb60] ARBEITSGEMEINSCHAFT DEUTSCHER LEHRERVERBÄNDE: Der ‚Bremer Plan‘ zur Neugestaltung des Deutschen Schulwesens. In: *Theorie und Geschichte der Bildungsreform: Eine Quellensammlung von Comenius bis zur Gegenwart*, 1984[Kem84], S. 195–209
- [Arb08] ARBEITSKREIS „BILDUNGSSTANDARDS“ DER GI: *Grundsätze und Standards für die Informatik in der Schule*. Berlin : LOGIN-Verlag, 2008. – Beilage zur LOGIN Nr. 150/151  
Im Internet [Stand: 2012-04-11]:  
[http://www.sn.schule.de/istandard/docs/...bildungungsstandards\\_2008.pdf](http://www.sn.schule.de/istandard/docs/...bildungungsstandards_2008.pdf)
- [Arl81] ARLT, Wolfgang (Hrsg.): *Datenverarbeitung/Informatik im Bildungsbereich*. Bd. 4: *Informatik als Schulfach*. München : R.Oldenbourg Verlag, 1981
- [Asc83] ASCHERSLEBEN, Karl (Hrsg.): *Didaktik*. Stuttgart : Verlag W. Kohlhammer, 1983
- [Ass65] ASSER, Günter: *Einführung in die mathematische Logik, Teil 1*. Zürich und Frankfurt a.M. : Verlag Harri Deutsch, 1965
- [Bac59] BACKUS, J.W.: *The syntax and Semantics of the Proposed International Algebraic Language of the Zuerich ACM-GRAMM conference*. Paris : Proc. Int. Conf. Inf. Processing, 1959. – 125–132 S.
- [BAL10] BELL, Tim ; ANDREAE, Peter ; LAMBERT, Lynn: Computer Science in New Zealand high schools. In: *Proceedings 12th Australasian Computing Education Conference (ACE 2010)*. Darlinghurst, Australia, Australia : Australian Computer Society, Inc., 2010 (ACE 2010). – ISBN 978–1–920682–84–2, 15–22
- [Bau90] BAUMANN, Rüdiger: *Didaktik der Informatik, 1. Auflage*. Stuttgart : Ernst Klett Verlag, 1990
- [Bau96] BAUMANN, Rüdiger: *Didaktik der Informatik, 2. Auflage*. Stuttgart : Ernst Klett Verlag, 1996
- [Bau98] BAUMANN, Rüdiger: Projekte im Mathematikunterricht — geht denn das? Was der Mathematik- vom Informatikunterricht lernen kann. In: *LOG IN* 18 (1998), Nr. 2, S. 33–45
- [Bau07] BAUER, Friedrich L. (Hrsg.): *40 Jahre Informatik in München: 1967 – 2007, Festschrift*. München : Fakultät für Informatik, 2007
- [Bau12] BAUER, Friedrich L.: Die ALGOL-Verschöörung. In: *Informatik Spektrum* 35 (2012), Nr. 2, S. 141–149
- [BBWJH03] BRENT B. WELCHAND JONES, K. ; HOBBS, J.: *Practical Programming in Tcl and Tk*. Prentice Hall, 2003 (Practical Programming in Tcl/Tk)

- [BCS96] BYRNE, Michael D. ; CATRAMBONE, Richard ; STASKO, John T.: *Do Algorithm Animations Aid Learning?* 1996. – Technical Report GIT-GVU-96-18
- [BD09] BADER, Franz ; DORN, Friedrich: *Physik Band 1 Gymnasium NRW*. Hannover : Schroedel Verlag GmbH, 2009 (Bd. 5–6)
- [BD10] BADER, Franz ; DORN, Friedrich: *Physik Gymnasium II*. Braunschweig : Schroedel Verlag GmbH, 2010
- [BDF<sup>+</sup>00] BUETTNER, Y. ; DUCHÂTEAU, C. ; FULFORD, C. ; HOGENBIRK, P. ; KENDALL, M. ; MOREL, R. ; WEERT, T. van: *Information and communication technology in secondary education: A curriculum for schools*. Paris : UNESCO, 2000
- [Bec05] BECKMANN, Astrid: Informatische Aspekte im Mathematikunterricht — Möglichkeiten und Chancen. In: KORTENKAMP, Ulrich (Hrsg.) ; WEIGAND, Hans-Georg (Hrsg.) ; WETH, Thomas (Hrsg.): *Informatische Ideen im Mathematikunterricht \* Bericht über die 23. Arbeitstagung des Arbeitskreises ‚Mathematikunterricht und Informatik‘ in der Gesellschaft für Didaktik der Mathematik e.V. vom 23. bis 25. September 2005 in Dillingen an der Donau*, Gesellschaft für Didaktik der Mathematik, 2005, S. 8–15
- [Ben62] BENSE, Max: *Theorie der Texte*. University Microfilms, 1962. – Original: University of Michigan
- [BFH<sup>+</sup>94] BRUNS, Martin ; FÖRSTER, Frank ; HERGET, Wilfried ; HISCHE, Horst ; KÖRNER, Henning ; PRUZINA, Manfred ; WINKELMANN, Bernard ; WOLFF, Klaus P.: *Stellungnahme zur Forderung des »Fakultätentages Informatik«, Informatik als obligatorisches Fach in der Sekundarstufe II einzurichten*. 1994
- [BFM<sup>+</sup>10] BRANDHOFER Gerhard (Hrsg.) ; FUTSCHEK Gerald (Hrsg.) ; MICHEUZ Peter (Hrsg.) ; REITER Anton (Hrsg.) ; SCHODER Karl (Hrsg.): *25 Jahre Schulinformatik in Österreich — Zukunft mit Herkunft*. Bd. 271. Wien : Österreichische Computergesellschaft, 2010
- [BH06] BOVET, Gislinde ; HUWENDIEK, Volker: *Leitfaden Schulpraxis: Pädagogik und Psychologie für den Lehrberuf*. 4. Cornelsen Scriptor, 2006
- [BHW85] BOSLER, Ulrich ; HAMPE, Wolfgang ; WANKE, Ilona ; WEERT, Tom J.: *Grundbildung Informatik*. Stuttgart : J.B. Metzlersche Verlagsbuchhandlung, 1985
- [Bie97] BIERMANN, Alan W.: *Great Ideas in Computer Science, 2nd Edition: A Gentle Introduction*. Cambridge (USA MA), London : MIT Press, 1997
- [Bik90] BIKOS, Konstantinos: *Informationstechnologische Grundbildung als Curriculum-Problem der Achtziger Jahre*. Frankfurt am Main : Peter Lang, 1990 (Europäische Hochschulschriften: Pädagogik)

- [Bit56] BITTORF, Wilhelm: *Automation, die zweite industrielle Revolution*. Darmstadt : Leske-Verlag, 1956 (Lebendige Wirtschaft)
- [BK78] BORMANN, Manfred ; KLINGER, Hermann: *Untersuchung zur Entwicklung formal-operationaler Strukturen und physikspezifischer Schemata bei Schülern der Sekundarstufe*. Stuttgart : Klett, 1978
- [BK97] BRAUN, Hans-Joachim ; KAISER, Walter ; KÖNIG, Wolfgang (Hrsg.): *Propyläen Technikgeschichte, 5. Band: Energiewirtschaft, Automatisierung, Information*. Berlin : Ullstein Buchverlag GmbH, 1997
- [Bla00] BLANKERTZ, Herwig: *Theorien und Modelle der Didaktik*. Juventa, 2000 (Grundfragen der Erziehungswissenschaft)
- [Blo56] BLOOM, Benjamin S.: *Taxonomy of educational objectives: the classification of educational goals*. D. McKay, 1956 (Taxonomy of Educational Objectives: The Classification of Educational Goals Bd. 1)
- [Bod76] BODENDIEK, Rainer: Graphentheorie in den Sekundarstufen. In: LEPPIG, Manfred (Hrsg.): *Neue Aspekte zum Mathematikunterricht der Sekundarstufen I und II*, Schöningh, 1976, S. 22–51
- [Bog67] BOGEN, Hans J.: *Knaurs Buch der modernen Biologie*. München, Zürich : Droemer Knauer, 1967
- [Bor06] BORTZ, Nicola Jürgen ; D. Jürgen ; Döring: *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler : mit ... 87 Tabellen*. 4., überarb. Aufl. Heidelberg : Springer Medizin Verl., 2006 (Springer-Lehrbuch). – ISBN 3–540–33305–3 ; 978–3–540–33305–0
- [BOS77] BAUERSFELD, Heinrich (Hrsg.) ; OTTE, Michael (Hrsg.) ; STEINER, Hans G. (Hrsg.): *Informatik im Unterricht der Sekundarstufe II: Grundfragen, Probleme und Tendenzen mit Bezug auf allgemeinbildende und berufsqualifizierende Ausbildungsgänge*. Bielefeld : Universität Bielefeld, 1977 (Schriftenreihe des IDM (Institut für Didaktik der Mathematik) 15 (Band I) und 16 (Band II)). – Arbeitstagung: Bielefeld 12.–14. September 1977
- [Bre82] BRENNER, Rul Anton ; G. Anton ; Gunzenhäuser: *Informatik : didaktische Materialien für Grund- u. Leistungskurse*. 1. Aufl. Stuttgart : Klett, 1982. – ISBN 3–12–922851–9. – Literaturverz. S. 149 - 155
- [Bre03] BREIER, Norbert: *Quo vadis Informatikunterricht, und was kann die Informatik für die künftigen Informatiklehrer leisten*. 2003. – <http://www.erzwiss.uni-hamburg.de/personal/breier/vortrag271003.pdf>
- [Bre05] BREIER, Norbert: Informatik im Fächerkanon allgemein bildender Schulen - Überlegungen zu einem informationsorientierten didaktischen Ansatz. In: FRIEDRICH, Steffen (Hrsg.): *INFOS Bd. 60, GI*, 2005 (LNI), S. 67–78

- [Bru60] BRUNER, Jerome S.: The Process of Education. In: *Harvard University Press* (1960)
- [Bru70] BRUNER, Jerome S.: *Der Prozeß der Erziehung*. Berlin : Berlin-Verlag, 1970 (Sprache und Lernen 4)
- [Bru09] BRUDERER, Herbert: *Wie viel Informatik brauchen Mittelschulen?* Zürich, 2009
- [Bud08] BUDDE, J.: *Mathematikunterricht und Geschlecht: empirische Ergebnisse und pädagogische Ansätze*. Bundesministerium für Bildung und Forschung (BMBF), Referat Bildungsforschung, 2008
- [Bun73] BUND-LÄNDER-KOMMISSION FÜR BILDUNGSPLANUNG: *Bildungsgesamtplan Band 1*. Stuttgart : Klett, 1973
- [Bun10] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG ; SCHELHOWE, Heidi (Hrsg.): *Kompetenzen in einer digital geprägten Kultur: Medienbildung für die Persönlichkeitsentwicklung, für die gesellschaftliche Teilhabe und für die Entwicklung von Ausbildungs- und Erwerbsfähigkeit*. Berlin : BMBF, 2010 (Serie: Bildung: Ideen zünden!)
- [Bun12] BUNDESAGENTUR FÜR ARBEIT: *Der Arbeitsmarkt für Akademikerinnen und Akademiker in Deutschland – Naturwissenschaften/Informatik*. Bundesagentur für Arbeit, 2012
- [Cap92a] CAPURRO, Rafael: Die Herausforderung der Informatik für die praktische Philosophie. In: COY, Wolfgang (Hrsg.) u. a.: *Sichtweisen der Informatik*, 1992, S. 343–353
- [Cap92b] CAPURRO, Rafael: Zur Frage der professionellen Ethik. In: [SHDK93], S. 121–140
- [CGM<sup>+</sup>89] COMER, D. E. ; GRIES, David ; MULDER, Michael C. ; TUCKER, Allen ; TURNER, A. J. ; YOUNG, Paul R.: Computing as a discipline. In: *Communications of the ACM* 32 (1989), Januar, Nr. 1, S. 9–23
- [CH10] COHEN, Avi ; HABERMAN, Bruria: CHAMSA: five languages citizens of an increasingly technological world should acquire. In: *ACM Inroads* 1 (2010), Dezember, Nr. 4, S. 54–57
- [Cho56] CHOMSKY, Noam: Three models for the description of language. In: *IRE Transactions on Information Theory* 2 (1956), S. 113–124
- [Cho67] CHOMSKY, Noam: A Review of B. F. Skinner's Verbal Behavior. In: *Readings in the Psychology of Language* (1967)
- [Cho73] CHOMSKY, Noam: *Aspekte der Syntax-Theorie*. Frankfurt : Suhrkamp, 1973

- [Chr85] CHRISTENSEN, Børge R: *Strukturierte Programmierung mit COMAL 80, 2., verbesserte Auflage*. München : Oldenbourg-Verlag, 1985
- [CKW03] CHARLTON, Michael ; KÄPPLER, Christoph ; WETZEL, Helmut: *Einführung in die Entwicklungspsychologie*. Beltz, 2003 (Beltz Studium : Entwicklung und Lernen)
- [Cod70] CODD, Edgar F.: A Relational Model of Data for Large Shared Data Banks. In: *Communications of the ACM* 13 (1970), Nr. 6, S. 377–387
- [Cod72] CODD, Edgar F.: Relational Completeness of Data Base Sublanguages. In: RUSTIN, R. (Hrsg.): *Database Systems*. San Jose, California, 1972, S. 65–98
- [Com12a] COMPUTING AT SCHOOL WORKING GROUP: *Computer Science: A curriculum for schools*. British Computer Society, 2012
- [Com12b] COMPUTING AT SCHOOL WORKING GROUP: *Computer Science as a school subject – Seizing the opportunity*. British Computer Society, 2012
- [Coo40] COOLIDGE, J.L.: *A History of Geometrical Methods*. Dover, 1940 (Dover Phoenix Editions)
- [CS86] CLAUS, Volker ; SCHWILL, Andreas: *Schülerduden Informatik*. 1. Dudenverlag, 1986 (Schülerduden / Schülerduden)
- [CS97] CHURCHLAND, Patricia S. ; SEJNOWSKI, Terrence J.: *Grundlagen zur Neuroinformatik und Neurobiologie: The Computational Brain in deutscher Sprache: übers. von Steffen Hölldobler und Claudia Hölldobler*. Braunschweig, Wiesbaden : Vieweg, 1997 (Computational intelligence)
- [Cub71] CUBE, Felix von: *Was ist Kybernetik?: Grundbegriffe, Methoden, Anwendungen*. Deutscher Taschenbuch Verlag, 1971
- [Dah65] DAHRENDORF, Ralf: *Bildung ist Bürgerrecht. Plädoyer für eine aktive Bildungspolitik*. Hamburg : Nannen-Verlag, 1965
- [DCP09] DANN, Wanda P. ; COOPER, Stephen ; PAUSCH, Randy: *Learning to program with Alice*. 2. Pearson Prentice Hall, 2009
- [DD87] DREYFUS, Hubert L. ; DREYFUS, Stuart E.: *Künstliche Intelligenz. Von den Grenzen der Denkmaschine und dem Wert der Intuition*. Hamburg : Rowohlt, 1987
- [DDH72] DAHL, Ole-Johan (Hrsg.) ; DIJKSTRA, Edsger W. (Hrsg.) ; HOARE, Charles Antony R. (Hrsg.): *Structured programming*. London, UK, UK : Academic Press Ltd., 1972
- [DEa08] DIEHL, Bardo ; ERB, Roger ; ALL, Harri H.: *Physik Oberstufe. Gesamtband*. Hannover : Cornelsen Verlag, 2008



- [Del80] DELLER, Hellmut: *Informatik in der Sekundarstufe II : zur Grundlegung der Informatik als Unterrichtsfach*. 1. Aufl. Frankfurt am Main [u.a.] : Diesterweg, 1980. – ISBN 3–425–07040–1
- [Den03] DENNING, Peter J.: Great Principles of Computing. In: *Communications of the ACM* 46 (2003), Nr. 11, S. 15–20
- [Der77] DER NIEDERSÄCHSISCHE KULTUSMINISTER: *Handreichungen für Lernziele, Kurse und Projekte im Sekundarbereich II. Für das mathematisch-naturwissenschaftlich-technische Aufgabenfeld 4.Folge C*. Hannover : Presse und Informationsdienst KM NDS, 1977
- [Der81a] DER KULTUSMINISTER DES LANDES NORDRHEIN-WESTFALEN: *Richtlinien für die gymnasiale Oberstufe in Nordrhein-Westfalen Informatik*. Köln : Greven Verlag, 1981 (Schriftenreihe: „Die Schule in Nordrhein-Westfalen“, Nr. 4725)
- [Der81b] DER KULTUSMINISTER DES LANDES NORDRHEIN-WESTFALEN: *Richtlinien für die gymnasiale Oberstufe in Nordrhein-Westfalen Mathematik*. Köln : Greven Verlag, 1981 (Schriftenreihe: „Die Schule in Nordrhein-Westfalen“, Nr. 4720)
- [Der81c] DER KULTUSMINISTER DES LANDES NORDRHEIN-WESTFALEN: *Richtlinien für die gymnasiale Oberstufe in Nordrhein-Westfalen Physik*. Köln : Greven Verlag, 1981 (Schriftenreihe: „Die Schule in Nordrhein-Westfalen“, Nr. 4721)
- [Der03] DER NIEDERSÄCHSISCHE KULTUSMINISTER: *Rahmenrichtlinien für die Integrierte Gesamtschule: Mathematik*. Hildesheim : Niedersächsisches Landesinstitut für Schulentwicklung und Bildung (NLI), 2003
- [Deu59] DEUTSCHER AUSSCHUSS FÜR DAS ERZIEHUNGS UND BILDUNGSWESEN: Rahmenplan zur Umgestaltung und Vereinheitlichung des allgemeinen öffentlichen Schulwesens. In: *Theorie und Geschichte der Bildungsreform: Eine Quellensammlung von Comenius bis zur Gegenwart*, 1984[Kem84], S. 187–195
- [Deu70] DEUTSCHER BILDUNGSRAT: *Strukturplan für das Bildungswesen, 2. Auflage*. Stuttgart : Ernst Klett Verlag, 1970 (Empfehlungen der Bildungskommission)
- [DFKT78] DRESCH, Peter ; FROBEL, Gunter ; KOSCHORRECK, Hans-Jürgen ; TEUFEL, J. ; HAUF, Annemarie (Hrsg.) ; STURM, Lothar (Hrsg.): *Kursmaterialien Informatik*. Paderborn : Gesamtschule Berger Feld, Gelsenkirchen – FEoLL (Forschungs-und Entwicklungszentrum objektivierter Lehr- und Lernverfahren), 1978
- [DFN03] DONOHUE, William ; FERGUSON, Kyle E. ; NAUGLE, Amy E.: The structure of the cognitive revolution: An examination from the philosophy of science. In: *Behav Anal* 26 (2003), Nr. 1, S. 85–110

- [Dij59] DIJKSTRA, Edsger W.: A Note on Two Problems in Connexion with Graphs. In: *Numerische Mathematik* 1 (1959), Nr. 1, S. 269–271
- [Dij68] DIJKSTRA, Edsger W.: Go To Statement Considered Harmful. In: *Communications of the ACM* 11 (1968), Nr. 3, S. 147–148
- [Dij69] DIJKSTRA, Edsger W.: *Structured programming*. August 1969. – circulated privately – <http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD268.PDF> – last visited 8<sup>th</sup> December 2007
- [DMSW07] DUIT, Reinders ; MIKELSKIS-SEIFERT, Silke ; WODZINSKI, Christoph: Physics in Context – A program for Improving Physics Instruction in Germany. In: PINTÓ, Roser (Hrsg.) ; COUSO, Digna (Hrsg.): *Contributions from Science Education Research*. Springer Netherlands, 2007, S. 119–130
- [DO11] DIETZ, Alexander ; OPPERMAN, Frank: *Planspiel 'Datenschutz 2.0'*. Berlin : LOGIN-Verlag, 2011. – Beilage zur LOGIN Nr. 31(2011) Heft 171
- [Dui10] DUIT, Reinders: *Physik im Kontext: Konzepte, Ideen, Materialien für effizienten Physikunterricht; Kontextorientierung, neue Lehr-Lern-Kultur, naturwissenschaftliche Denk- und Arbeitsweisen, moderne Physik und Technologie*. Seelze : Friedrich, 2010
- [EBK07] ESSER, Ruth ; BARKOWSKI, Hans ; KRUMM, Hans J.: *Bausteine für Babylon: Sprache, Kultur, Unterricht- : Festschrift zum 60. Geburtstag von Hans Barkowski*. Iudicium, 2007
- [Eng77] ENGEL, Arthur: *Elementarmathematik vom algorithmischen Standpunkt*. Klett, 1977 (Klett Studienbücher : Mathematik)
- [Eng04] ENGBRING, Dieter: *Informatik im Herstellungs- und Nutzungskontext. Ein technikbezogener Zugang zur fachübergreifenden Lehre*. Paderborn, Universität Paderborn, Diss., 2004
- [Eng11] ENGBRING, Dieter: Was ist/kann/soll Informatikunterricht? In: *INFOS*, 2011, S. 97–106
- [EP10a] ENGBRING, Dieter ; PASTERNAK, Arno: Einige Anmerkungen zum Begriff IniK. In: DIETHELM, Ira (Hrsg.) ; DÖRGE, Christina (Hrsg.) ; HILDEBRANDT, Claudia (Hrsg.) ; SCHULTE, Carsten (Hrsg.): *DDI* Bd. 168, GI, 2010 (LNI), S. 119–124
- [EP10b] ENGBRING, Dieter ; PASTERNAK, Arno: INIK — Versuch einer Begriffsbestimmung. In: BRANDHOFER Gerhard (Hrsg.) ; FUTSCHEK Gerald (Hrsg.) ; MICHEUZ Peter (Hrsg.) ; REITER Anton (Hrsg.) ; SCHODER Karl (Hrsg.): *25 Jahre Schulinformatik in Österreich — Zukunft mit Herkunft* Bd. 271. Wien : Österreichische Computergesellschaft, 2010. – ISBN 9783854032711, S. 100–115

- [Fac76] FACHAUSSCHUß AUSBILDUNG DER GI: Zielsetzung und Inhalte des Informatikunterrichts. In: ZDM[Arl81], S. 35–43. – Nachdruck
- [FB96] FLOYD, R.W. ; BEIGEL, R.: *Die Sprache der Maschinen*. International Thomson Publishing, 1996. – ISBN 9783826602160
- [FFFK01] FELLEISEN, Matthias ; FINDLER, Robert B. ; FLATT, Matthew ; KRISHNAMURTH, Shriram: *How to Design Programs: An Introduction to Programming and Computing*. MIT Press, 2001
- [Fis83] FISCHLER, Helmut: Projektunterricht und Lehrerbildung. In: *LOG IN 3* (1983), Nr. 2, S. 21–25
- [Fis01] FISCHER, Ernst P.: *Die andere Bildung: was man von den Naturwissenschaften wissen sollte*. Ullstein, 2001
- [Fly12] FLYNT, C.: *Tcl/Tk: A Developer's Guide*. Elsevier Science & Technology, 2012 (The Morgan Kaufmann Series in Software Engineering and Programming). – ISBN 9780123847171
- [FMV90] FISCHER, G. ; MATHEMATIKER-VEREINIGUNG, Deutsche: *Ein Jahrhundert Mathematik 1890 - 1990*. Vieweg, 1990 (Dokumente Zur Geschichte der Mathematik). – ISBN 9783528063269
- [For92] FORNECK, Hermann-Josef: *Bildung im informationstechnischen Zeitalter. Untersuchung der fachdidaktischen Entwicklung der informationstechnischen Bildung*. Aarau : Sauerlaender, 1992. – ISBN 3–7941–3622–5
- [For08] FORD, Jerry L.: *Scratch Programming for Teens*. Course Technology, 2008 (For Teens)
- [Fre73] FREUDENTHAL, Hans: *Mathematik als pädagogische Aufgabe*. Stuttgart : Ernst Klett Verlag, 1973 (Klett Studienbücher Bd. 1)
- [Fre83] FREY, Karl: Die sieben Komponenten der Projektmethode – mit Beispielen aus dem Schulfach Informatik. In: *LOG IN 3* (1983), Nr. 2, S. 16–20
- [Fre10] FREIERMUTH, Karin ; STEFFEN, Juraj Hromkovič ; Lucia Keller ; B. (Hrsg.): *Einführung in die Kryptologie : Lehrbuch für Unterricht und Selbststudium*. Wiesbaden : Vieweg+Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden, 2010 (SpringerLink : Bücher)
- [Fre11a] FREIE UND HANSESTADT HAMBURG, BEHÖRDE FÜR SCHULE UND BERUFSBILDUNG: *Bildungsplan Gymnasium (Jahrgangsstufen 5 bis 11), Lernbereich Naturwissenschaften und Technik*. Hamburg : Landesinstitut für Lehrerbildung und Schulentwicklung, 2011

- [Fre11b] FREIE UND HANSESTADT HAMBURG, BEHÖRDE FÜR SCHULE UND BERUFSBILDUNG: *Bildungsplan Stadtteilschule (Jahrgangsstufen 5 bis 11), Lernbereich Naturwissenschaften und Technik*. Hamburg : Landesinstitut für Lehrerbildung und Schulentwicklung, 2011
- [Fri90] FRIEDRICHS, Jürgen: *Methoden empirischer Sozialforschung*. VS Verlag für Sozialwissenschaften, 1990
- [Fri03] FRIEDRICH, Steffen: Informatik und PISA - vom Wehe zum Wohl der Schul-informatik. In: HUBWIESER, Peter (Hrsg.): *INFOS* Bd. 32, GI, 2003 (LNI), S. 133–144
- [FSR05] FUCCIA, David D. ; SCHWARZ, Wolfgang ; RALLE, Bernd: Lehrer planen gemeinsam – Ein Einblick in die Arbeit des Projektes Chemie im Kontext. In: *MNU* 58 (2005), Nr. 7, S. 388–393
- [FW75] FEILMEIER, Manfred ; WACKER, Hansjörg: *Numerische Mathematik: Ein Lehr- und Arbeitsbuch*. Bayerischer Schulbuch-Verlag, 1975 (bsv Mathematik Bd. 1). – ISBN 9783762730934
- [GE09] GUZDIAL, Mark ; ERICSON, Barbara: *Introduction to Computing & Programming in Python: A Multimedia Approach*. Prentice Hall, 2009
- [Ges09] GESELLSCHAFT FÜR FACHDIDAKTIK E.V. (GFD): Mindeststandards am Ende der Pflichtschulzeit. In: *Theo-Web. Zeitschrift für Religionspädagogik* 8 (2009), S. 151–159
- [Gie82] GIEDION, Sigfried: *Die Herrschaft der Mechanisierung*. Hamburg : Europäische Verlagsanstalt, 1982
- [Gie98] GIESECKE, Hermann: *Pädagogische Illusionen: Lehren aus 30 Jahren Bildungspolitik*. Klett-Cotta, 1998
- [Gie02] GIESECKE, Hermann: *Funkmanuskripte*. 2002 <http://www.hermann-giesecke.de/funk.htm>
- [Gie05] GIESECKE, Hermann: 'Humankapital' als Bildungsziel? Grenzen ökonomischen Denkens für das pädagogische Handeln. In: *Neue Sammlung* 45 (2005), Nr. 3, S. 377–389
- [Gil93] GILBERT, Walter: Das Genom — eine Zukunftsvision. In: *Der Supercode*[Kev93], S. 95–108
- [GK<sup>+</sup>98] GREHN, Joachim ; KRAUSE, Joachim u. a.: *Metzler Physik: Gesamtband*. 3. Auflage. Schroedel, 1998
- [GK<sup>+</sup>07] GREHN, Joachim ; KRAUSE, Joachim u. a.: *Metzler Physik: Gesamtband*. 4. Auflage. Schroedel, 2007

- [Gör98] GÖRKE, Winfried (Hrsg.) ; Fakultätentag Informatik (Veranst.): *25 Jahre Fakultätentag Informatik – 1973–1998*. Fakultätentag Informatik, 1998
- [Gut08] GUTZMER, August [. (Hrsg.): *Die Tätigkeit der Unterrichtskommission der Gesellschaft Deutscher Naturforscher und Ärzte : Gesamtbericht enthaltend die Vorverhandlungen auf den Versammlungen in Cassel und Breslau sowie die seitens der Kommission den Versammlungen in Meran, Stuttgart und Dresden unterbreiteten Reformvorschläge*. Leipzig : Teubner, 1908
- [Häd10] HÄDER, M.: *Empirische Sozialforschung: Eine Einführung*. 2. VS Verlag für Sozialwissenschaften, 2010
- [Haf88] HAFENBRAK, Bernd: *COMAL im Unterricht: Beiträge zum Computereinsatz in der Schule*. Deutscher Studien-Verlag, 1988 (Schriftenreihe der Pädagogischen Hochschule Heidelberg)
- [Ham50] HAMMING, R. W.: Error Detecting and Error Correcting Codes. In: *Bell System Technical Journal* 29 (1950), Nr. 2, S. 147–160
- [Har97] HARRISON, Mark: *Tcl/Tk tools*. O'Reilly, 1997 (O'Reilly Series)
- [Häu76] HÄUSSLER, Peter: Ansätze zu einem integrierten Curriculum Naturwissenschaften. In: *Curriculum Naturwissenschaften*, 1976, S. 36–84
- [HCD11] HABERMAN, Bruria ; COHEN, Avi ; DAGIENE, Valentina: The beaver contest: attracting youngsters to study computing. In: *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*. New York, NY, USA : ACM, 2011 (ITiCSE '11). – ISBN 978–1–4503–0697–3, 378–378
- [HD00] HUNDHAUSEN, Christopher D. ; DOUGLAS, Sarah A.: Using Visualizations to Learn Algorithms: Should Students Construct Their Own, or View an Expert's? In: *VL*, 2000, S. 21–28
- [Hen68] HENNINGSEN, J.: *Atome, Algen, Automaten: Futurologie in der Schule*. Westermann, 1968 (Pädagogisches Forum)
- [Her87] HERTZ, Heinrich: Über sehr schnelle elektrische Schwingungen. In: *Annalen der Physik* 267 (1887), S. 421–448
- [Her71] HERTZ, Heinrich: *Über sehr schnelle elektrische Schwingungen: Vier Arbeiten(1887-1889)*. Deutsch Harri GmbH, 1971 (Ostwalds Klassiker). – ISBN 9783817132515
- [Het48] HETZER, Hildegard: *Kind und Jugendlicher in der Entwicklung*. Wolfenbütteler Verlagsanstalt, 1948 (Arbeitsbücher für die Lehrerbildung)

- [Hey96] HEYMANN, Hans W.: *Allgemeinbildung und Mathematik*. Weinheim [u.a.] : Beltz, 1996 (Studien zur Schulpädagogik und Didaktik ; 13)
- [HHG78] HIELE, Pierre Marie v. ; HIELE-GELDORF, Dina v.: Die Bedeutung der Denkebenen im Unterrichtssystem nach der deduktiven Methode. In: *Didaktik der Mathematik* (1978), S. 127–139
- [Hie76] HIELE, Pierre Marie v.: Wie kann man im Mathematikunterricht den Denkstufen Rechnung tragen? In: *Educational Studies in Mathematics* 7 (1976), S. 157–169
- [Hie86] HIELE, Pierre Marie v.: *Structure and insight: a theory of mathematics education*. Orlando : Academic Press, 1986 (Developmental psychology series)
- [Hir87] HIRSCHI, Rainer: *Beiträge zur Geschichte des Physikunterrichts*. Frankfurt : Verlag Peter Lang, 1987 (Allgemeine und spezielle Didaktik, Bd. 4)
- [HM98] HARRISON, Mark ; MCLENNAN, Michael: *Effektiv Tcl/Tk programmieren*. Bonn [u.a.] : Addison-Wesley, 1998 (Professionelle Programmierung)
- [HM01] HEPP, Ralph ; MÜNZINGER, Wolfgang: Projektorientierter Physikunterricht. In: *Unterricht Physik* 12 (2001), Nr. 63/64, S. 4–7
- [Hoa62] HOARE, Charles Antony R.: Quicksort. In: *The Computer Journal* 5 (1962), Nr. 1, S. 10–16
- [Hoo93] HOOD, Leroy: Biologie und Medizin im 21. Jahrhundert. In: *Der Supercode*[Kev93], S. 156–183
- [Hop82] HOPFIELD, John J.: Neural networks and physical systems with emergent collective computational abilities. In: *Proceedings of the National Academy of Sciences of the United States of America* 79 (1982), Nr. 8, S. 2554–2558
- [Hro06] HROMKOVIČ, Juraj: *Sieben Wunder der Informatik : eine Reise an die Grenze des Machbaren mit Aufgaben und Lösungen*. 1. Wiesbaden : Teubner Verlag, 2006 (Lehrbuch Informatik). – ISBN 3–8351–0078–5 ; 978–3–8351–0078–7
- [Hro08] HROMKOVIČ, Juraj: *Lehrbuch Informatik : Vorkurs Programmieren, Geschichte und Begriffsbildung, Automatenentwurf*. 1. Wiesbaden : Vieweg und Teubner Verlag, 2008 (Studium). – ISBN 978–3–8348–0620–8
- [Hro10] HROMKOVIČ, Juraj: *Theoretische Informatik*. Vieweg Verlag, Friedr. & Sohn Verlagsgesellschaft mbH, 2010
- [Hro11] HROMKOVIČ, Juraj: *Berechenbarkeit: Logik, Argumentation, Rechner und Assembler, Unendlichkeit, Grenzen der Automatisierbarkeit; Lehrbuch für Unterricht und Selbststudium*. Vieweg+Teubner Verlag, 2011 (Vieweg + Teubner Studium)

- [Hub44] HUBER, Franz: *Allgemeine Unterrichtslehre*. Leipzig : Julius Klinkhardt Verlagsbuchhandlung, 1944
- [Hub07] HUBWIESER, Peter (Hrsg.): *Didaktik der Informatik : Grundlagen, Konzepte, Beispiele*. 3. Auflage. Berlin, Heidelberg : Springer-Verlag, 2007 (eXamen.press). – ISBN 978–3–540–72478–0. – In: Springer-Online
- [Hum36] HUMBOLDT, Wilhelm v.: *Über die Verschiedenheit des menschlichen Sprachbaues: Und ihren Einfluss auf die geistige Entwicklung des Menschengeschlechts*. Druckerei der Königlichen Akademie der Wissenschaften, 1836
- [Hum05] HUMBERT, Ludger: *Didaktik der Informatik: mit praxiserprobtem Unterrichtsmaterial*. Vieweg+Teubner Verlag, 2005 (Leitfäden der Informatik)
- [Huß07] HUSSMANN, Stephan [. (Hrsg.): *Kombinatorische Optimierung erleben : in Studium und Unterricht*. 1. Aufl. Wiesbaden : Vieweg, 2007 (Mathematik erleben-Mathematik für das Lehramt). – ISBN 3–528–03216–2 ; 978–3–528–03216–6
- [Ihm08] IHM, Maximilian (Hrsg.): *Gaius Suetonius Tranquillus: De vita Caesarum*. Stuttgart : B.G. Teubner, 1908
- [INF84] Informatik als Herausforderung an Schule und Ausbildung : GI-Fachtagung, Berlin, 8. - 10. Oktober 1984. Berlin : Springer, 1984 (Informatik-Fachberichte ; 90). – ISBN 3–540–13869–2 ; 0–387–13869–2
- [IPDF08] ILKA PARCHMANN, Bernd R. ; DI FUCCIA, David-S.: Entwicklung und Struktur der Unterrichtskonzeption Chemie im Kontext. In: DEMUTH, R. u. a. (Hrsg.): *Chemie im Kontext - Von der Innovation zur nachhaltigen Verbreitung eines Unterrichtskonzepts*. Münster : Waxmann, 2008, S. 9–48
- [ISC<sup>+</sup>09] ISBELL, Charles L. ; STEIN, Lynn A. ; CUTLER, Robb ; FORBES, Jeffrey ; FRASER, Linda ; IMPAGLIAZZO, John ; PROULX, Viera ; RUSS, Steve ; THOMAS, Richard ; XU, Yan: (Re)defining computing curricula by (re)defining computing. In: *SIGCSE Bull.* 41 (2009), Nr. 4, S. 195–207. <http://dx.doi.org/http://doi.acm.org/10.1145/1709424.1709462>. – DOI <http://doi.acm.org/10.1145/1709424.1709462>. – ISSN 0097–8418
- [Jan92] JANICH, Peter: Zur Konstitution der Informatik als Wissenschaft. In: *Informatik und Philosophie*, 1992, S. 53–68
- [JM07] JAHNKE, T. ; MEYERHÖFER, W.: *Pisa& Co: Kritik eines Programms*. Franzbecker KG, 2007
- [Jud93] JUDSON, Horace F.: Eine Geschichte der Wissenschaft un der Technologie der Genkartierung und -sequenzierung. In: *Der Supercode*[Kev93], S. 48–91

- [JW74] JENSEN, Kathleen ; WIRTH, Niklaus ; HANSEN, P. B. (Hrsg.) ; GRIES, D. (Hrsg.) ; MOLER, C. (Hrsg.) ; SEEGMÜLLER, G. (Hrsg.) ; WIRTH, N. (Hrsg.): *PASCAL user manual and report*. New York, NY, USA : Springer-Verlag New York, Inc., 1974. – ISBN 3–540–06950–X
- [KAH<sup>+</sup>10a] KLIEME, Eckhard ; ARTELT, Cordula ; HARTIG, Johannes ; JUDE, Nina ; KÖLLER, Olaf ; PRENZEL, Manfred ; SCHNEIDER, Wolfgang ; STANAT, Petra: *PISA 2009: Bilanz nach einem Jahrzehnt*. Waxmann Verlag GmbH, 2010
- [KAH<sup>+</sup>10b] KLIEME, Eckhard ; ARTELT, Cordula ; HARTIG, Johannes ; JUDE, Nina ; KÖLLER, Olaf ; PRENZEL, Manfred ; SCHNEIDER, Wolfgang ; STANAT, Petra: *PISA 2009: Bilanz nach einem Jahrzehnt - Zusammenfassung*. Waxmann Verlag GmbH, 2010
- [Kem84] KEMPER, H.: *Theorie und Geschichte der Bildungsreform: Eine Quellensammlung von Comenius bis zur Gegenwart*. Forum Academicum in der Verlagsgruppe Athenäum, Hain, Hanstein, 1984 (Hochschulschriften Erziehungswissenschaft)
- [Kev93] KEVLES, Daniel J.: *Der Supercode: Die genetische Karte des Menschen*. Frankfurt a.M., Wien : Büchergilde Gutenberg, 1993
- [KH05] KASTNER, Wolfgang ; HELGE, Gerhard: *Informatik: Aufgaben und Lösungen*. Springer, 2005 (Springers Lehrbücher der Informatik)
- [KL10] KONIDARI, Eleni ; LOURIDAS, Panos: When students are not programmers. In: *ACM Inroads* 1 (2010), March, S. 55–60
- [Kla58a] KLAFKI, Wolfgang: Bildung und Erziehung im Spannungsfeld von Vergangenheit, Gegenwart und Zukunft. In: *Studien zur Bildungstheorie und Didaktik*[Kla64], S. 9–24
- [Kla58b] KLAFKI, Wolfgang: Didaktische Analyse als Kern der Unterrichtsvorbereitung. In: *Studien zur Bildungstheorie und Didaktik*[Kla64], S. 126–154
- [Kla59] KLAFKI, Wolfgang: Kategoriale Bildung. In: *Studien zur Bildungstheorie und Didaktik*[Kla64], S. 25–45
- [Kla64] KLAFKI, W.: *Studien zur Bildungstheorie und Didaktik*. Beltz, 1964 (Reihe Pädagogik)
- [Kla71] KLAFKI, Wolfgang: Restaurative Schulpolitik 1945–1950 in Westdeutschland. In: *Theorie und Geschichte der Bildungsreform: Eine Quellensammlung von Comenius bis zur Gegenwart*[Kem84], S. 173–186
- [Kli07] KLIEME, ECKHARD AND AVENARIUS, HERMANN AND BLUM, WERNER AND DÖBRICH, PETER AND GRUBER, HANS AND PRENZEL, MANFRED AND



REISS, KRISTINA AND RIQUARTS, KURT AND ROST, JÜRGEN AND TEN-ORTH, HEINZ-ELMAR AND VOLLMER, HELMUT J. ; BMBF (Hrsg.): *Zur Entwicklung nationaler Bildungsstandards. Eine Expertise*. Bonn : BMBF – Referat Öffentlichkeitsarbeit, 2007

- [KMK72] KMK, Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland: *Vereinbarung zur Gestaltung der gymnasialen Oberstufe in der Sekundarstufe II*. 1972
- [KMK04a] KMK, Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland: *Bildungsstandards im Fach Deutsch für den mittleren Bildungsabschluss - Beschluss vom 4.12.2003*. Neuwied : Wolters Kluwer Deutschland - Luchterhand, 2004 (Reihe: „Beschlüsse der Kultusministerkonferenz“)
- [KMK04b] KMK, Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland: *Bildungsstandards im Fach Mathematik für den mittleren Bildungsabschluss - Beschluss vom 4.12.2003*. Neuwied : Wolters Kluwer Deutschland - Luchterhand, 2004 (Reihe: „Beschlüsse der Kultusministerkonferenz“)
- [KMK05a] KMK, Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland: *Bildungsstandards im Fach Biologie für den mittleren Bildungsabschluss - Beschluss vom 16.12.2004*. Neuwied : Wolters Kluwer Deutschland - Luchterhand, 2005 (Reihe: „Beschlüsse der Kultusministerkonferenz“)
- [KMK05b] KMK, Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland: *Bildungsstandards im Fach Chemie für den mittleren Bildungsabschluss - Beschluss vom 16.12.2004*. Neuwied : Wolters Kluwer Deutschland - Luchterhand, 2005 (Reihe: „Beschlüsse der Kultusministerkonferenz“)
- [KMK05c] KMK, Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland: *Bildungsstandards im Fach Physik für den mittleren Bildungsabschluss - Beschluss vom 16.12.2004*. Neuwied : Wolters Kluwer Deutschland - Luchterhand, 2005 (Reihe: „Beschlüsse der Kultusministerkonferenz“)
- [Kno11] KNOBELSDORF, Maria: *Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung*. Berlin, Freie Universität, Diss., 2011
- [Köh21] KÖHLER, Wolfgang: *Intelligenzprüfungen an Menschenaffen*. Springer, 1921
- [Köh59] KÖHLER, Wolfgang: *The mentality of apes*. Vintage Books, 1959
- [Kon10] KONRAD, Klaus: Lautes Denken. In: MEY, Günter (Hrsg.) ; MRUCK, Katja (Hrsg.): *Handbuch Qualitative Forschung in der Psychologie*. VS Verlag für Sozialwissenschaften, 2010. – ISBN 978–3–531–92052–8, S. 476–490

- [Krä92] KRÄMER, Sybille: Operative Schriften als Geistestechnik. Zur Vorgeschichte der Informatik. In: [SHDK93], S. 69–83
- [KSS77] KRESS, K. ; SCHUSTER, D. ; SCHUSTER, S.: *Digitale Elektronik und Computer*. Diesterweg, 1977. – ISBN 9783425053257
- [Küm90a] KÜMMEL, Hartmut: *Schüler programmieren in COMAL*. Paderborn : Schöningh, 1990
- [Küm90b] KÜMMEL, Hartmut: *Schüler programmieren in COMAL \* Lösungen*. Paderborn : Schöningh, 1990
- [Kur93] KURZWEIL, Ray: *KI: Das Zeitalter der künstlichen Intelligenz*. Carl Hanser Verlag, 1993
- [KW97] KÖNIG, Wolfgang ; WEBER, Wolfhard ; KÖNIG, Wolfgang (Hrsg.): *Propyläen Technikgeschichte, 4. Band: Netzwerke, Stahl und Strom*. Berlin : Ullstein Buchverlag GmbH, 1997
- [Led11] LEDER, Jennifer: *Informatik im Kontext, Ecommerceverhalten Jugendlicher und ihre Erwartungen an den Informatikunterricht*. Universität Potsdam, 2011. – Masterarbeit
- [Lee96] LEE, John A. N.: ' Those who forget the lessons of history are doomed to repeat it' : or, Why I study the history of computing. In: *Annals of the History of Computing, IEEE* 18 (1996), Nr. 2, S. 54–62
- [Leh85] LEHMANN, Eberhardt: *Projektarbeit im Informatikunterricht*. Stuttgart : B.G. Teubner, 1985
- [Lew10] LEWIS, Colleen M.: How Programming Environment Shapes Perception, Learning and Goals: Logo vs. Scratch. In: *SIGCSE '10: Proceedings of the 41st ACM technical symposium on Computer science education*. New York, NY, USA : ACM, 2010, S. 346–350
- [LF97] LÖWIS, Martin von ; FISCHBECK, Nils: *Das Python-Buch*. Bonn : Addison-Wesley, 1997
- [Lie49] LIETZMANN, Walther: *Schulreform und mathematischer Unterricht*. Heidelberg : Quelle [und] Meyer, 1949
- [Lös02] LÖSCHER, C.: *Globalisierung*. Berlin : Cornelsen, 2002 (Globalisierung / hrsg. von Christel Löscher)
- [LS92] LANOUILLE, William ; SILARD, Bela A.: *Genius in the shadows: a biography of Leo Szilard : the man behind the bomb*. C. Scribner's Sons, 1992
- [Mar92] MARTENS, Ekkehard: Computerethik. In: [SHDK93], S. 141–154

- [Mau96] MAURER, R.: *HTML und CGI-Programmierung: Dynamische WWW-Seiten erstellen*. Heidelberg : dpunkt Verlag für digitale Technologie GmbH, 1996
- [May08] MAYER, H.O.: *Interview und schriftliche Befragung: Entwicklung, Durchführung und Auswertung*. Oldenbourg, 2008. – ISBN 9783486586695
- [McC60] MCCARTHY, John: Recursive functions of symbolic expressions and their computation by machine, Part I. In: *Commun. ACM* 3 (1960), Nr. 4, S. 184–195
- [McC63] MCCARTHY, John: A Basis For A Mathematical Theory Of Computation. In: *Computer Programming and Formal Systems*, North-Holland, 1963, S. 33–70
- [Med09] MEDIENPÄDAGOGISCHER FORSCHUNGSVERBUND SÜDWEST (Hrsg.): *JIM-Studie 2009*. Stuttgart, 2009 <http://www.mpfs.de/fileadmin/JIM-pdf09/JIM-Studie2009.pdf>
- [Med10] MEDIENPÄDAGOGISCHER FORSCHUNGSVERBUND SÜDWEST (Hrsg.): *KIM-Studie 2010. Kinder + Medien Computer + Internet. Basisuntersuchung zum Medienumgang 6- bis 13-Jähriger in Deutschland*. Stuttgart, 2010 <http://www.mpfs.de/index.php?id=192>
- [Men27] MENGER, Karl: Zur allgemeinen Kurventheorie. In: *Fund. Math* 10 (1927), Nr. 95-115, S. 5
- [Mer08] MERZYN, Gottfried: *Naturwissenschaften, Mathematik, Technik- immer unbeliebter?: Die Konkurrenz von Schulfächern um das Interesse der Jugend im Spiegel vielfältiger Untersuchungen*. Schneider Verlag GmbH, 2008
- [Mes05] MESCHÉDE, Dieter: *Gerthsen Physik*. 23. Auflage. Berlin : Springer, 2005
- [MG03] MESCHÉDE, Dieter ; GERTHSEN, Christian: *Gerthsen Physik*. 22. Auflage. Springer, 2003 (Springer-Lehrbuch). – ISBN 9783540026228
- [Mil03] MILLER, George A.: The cognitive revolution: a historical perspective. In: *TRENDS in Cognitive Sciences* 7 (2003), Nr. 3, S. 141–144
- [Min93] MINSKY, Marvin: Gedanken zur Künstlichen Intelligenz. In: *KI: Das Zeitalter der künstlichen Intelligenz* (1993), S. 214–220
- [Min99a] MINISTERIUM FÜR SCHULE UND WEITERBILDUNG, WISSENSCHAFT UND FORSCHUNG DES LANDES NORDRHEIN-WESTFALEN: *Sekundarstufe II Gymnasium/Gesamtschule Richtlinien und Lehrpläne Informatik*. Frechen : Ritterbach Verlag, 1999 (Schriftenreihe: „Schule in NRW“, Nr. 4725)
- [Min99b] MINISTERIUM FÜR SCHULE UND WEITERBILDUNG, WISSENSCHAFT UND FORSCHUNG DES LANDES NORDRHEIN-WESTFALEN: *Sekundarstufe II Gymnasium/Gesamtschule Richtlinien und Lehrpläne Mathematik*. Frechen : Ritterbach Verlag, 1999 (Schriftenreihe: „Schule in NRW“, Nr. 4720)

- [Min99c] MINISTERIUM FÜR SCHULE UND WEITERBILDUNG, WISSENSCHAFT UND FORSCHUNG DES LANDES NORDRHEIN-WESTFALEN: *Sekundarstufe II Gymnasium/Gesamtschule Richtlinien und Lehrpläne Physik*. Frechen : Ritterbach Verlag, 1999 (Schriftenreihe: „Schule in NRW“, Nr. 4721)
- [Min11] MINISTERIUM FÜR SCHULE UND WEITERBILDUNG, WISSENSCHAFT UND FORSCHUNG DES LANDES NORDRHEIN-WESTFALEN: *Kernlehrplan für die Gesamtschule Sekundarstufe I in Nordrhein-Westfalen Naturwissenschaften Biologie, Chemie, Physik*. 2011
- [Min12] MINISTERIUM FÜR SCHULE UND WEITERBILDUNG, WISSENSCHAFT UND FORSCHUNG DES LANDES NORDRHEIN-WESTFALEN: *Bereinigte Amtliche Sammlung Der Schulvorschriften Nordrhein-Westfalen*. Frechen : Ritterbach Verlag, 2012
- [MK99] MITRA, Sugata ; KHAS, Haus: Minimally Invasive Education for Mass Computer Literacy. In: *CSI Communications* 32 (1999), S. 1216
- [ML10] MILLER, Bradley N. ; L.RANUM, David: *Python Programming in Context*. Jones & Bartlett Learning, 2010
- [Mni07] MNICH, Jasmin N.: *Wege in die Chemie - Eindrücke und Erfahrungen von Chemiestudentinnen und -studenten*. Universität Bielefeld, 2007. – Masterarbeit
- [MNU05] *Bericht der Unterrichtskommission der Gesellschaft Deutscher Naturforscher und Ärzte über ihre bisherige Tätigkeit*. Sonderausg. Leipzig : Vogel, 1905
- [MP69] MINSKY, Marvin ; PAPERT, Seymour: *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969
- [MP90] MCCULLOCH, Warren ; PITTS, Walter: A logical calculus of the ideas immanent in nervous activity. In: *Bulletin of Mathematical Biology* 52 ((1943),1990), S. 99–115
- [MS06] MAGENHEIM, Johannes ; SCHULTE, Carsten: Social, ethical and technical issues in informatics–An integrated approach. In: *Education and Information Technologies* 11 (2006), Oktober, Nr. 3-4, S. 319–339
- [MS<sup>+</sup>11] MEYER, Lothar (Hrsg.) ; SCHMIDT, Gerd-Dietrich (Hrsg.) u. a.: *Duden Physik Gymnasiale Oberstufe*. Berlin, Mannheim : Duden Schulbuch-Verlag, 2011
- [Mün76] MÜNZINGER, Wolfgang: Möglichkeiten projektorientierten Mathematikunterrichts. In: *Geisler et.Al.: Projektorientierter Unterricht* (1976), S. 76–83
- [Mut67] MUTH, Jakob: *Von acht bis eins: Situationen aus dem Schulalltag und ihre didaktische Dimension*. Essen : Verlage Neue Deutsche Schule, 1967

- [Nau63] NAUR, Peter: Revised Report on the Algorithmic Language ALGOL 60. In: *Numerische Mathematik* (1963), Nr. 4, S. 420–454
- [Neu91] NEUMANN, János von: *Die Rechenmaschine und das Gehirn*. Oldenbourg, 1991 (Scientia nova). – ISBN 9783486452266
- [Nie95] NIEVERGELT, Jürg: Welchen Wert haben theoretische Grundlagen für die Berufspraxis? Gedanken zum Fundament des Informatikturms. In: *Informatik Spektrum* 18 (1995), Nr. 6, S. 342–344
- [NRW85] NRW KULTUSMINISTER: *Schriftenreihe des Kultusministers*. Bd. 43: *Rahmenkonzept Neue Informations- und Kommunikationstechnologien in der Schule – Zielvorstellungen, Maßnahmen und Entwicklungsstand* –. Köln : Greven, 1985
- [NRW10] NRW MINISTERIUM FÜR SCHULE UND WEITERBILDUNG: *Ausbildungs- und Prüfungsordnung Sekundarstufe I, APO-S I*. NRW Ministerium für Schule und Weiterbildung, 2010 [www.schulministerium.nrw.de/BP/Schulrecht/APOen/APO\\_SI.pdf](http://www.schulministerium.nrw.de/BP/Schulrecht/APOen/APO_SI.pdf)
- [NTC00] NTCM: *Principles and Standards for School Mathematics*. USA : Reston, 2000. – National Council of Teachers of Mathematics
- [Nyq02] NYQUIST, H.: Certain topics in telegraph transmission theory. In: *Proceedings of the IEEE* 90 (2002), feb, Nr. 2, S. 280 –305. – Wiederdruck aus: Transactions of the American Institute of Electrical Engineers. Vol. 47, April 1928, S. 617–644
- [Oer07] OERTER, Rolf: Zur Psychologie des Spiels. In: *Psychologie & Gesellschaftskritik* (2007), Nr. 4, S. 7–32
- [OJ09] OUSTERHOUT, John K. ; JONES, Ken: *Tcl and the Tk Toolkit*. 2. Addison Wesley, 2009 (Addison-Wesley professional computing series)
- [Old11] OLDENBURG, Reinhard: *Mathematische Algorithmen im Unterricht: Mathematik aktiv erleben durch Programmieren*. Vieweg+Teubner Verlag, 2011
- [Ous95] OUSTERHOUT, John K.: *Tcl und Tk : Entwicklung grafischer Benutzerschnittstellen für das X-Window-System*. 1. Bonn : Addison-Wesley Professional Computing, 1995 (Professional computing). – ISBN 3–89319–793–1
- [Ous97] OUSTERHOUT, John K.: Scripting: Higher Level Programming for the 21st Century. In: *IEEE Computer* 31 (1997), S. 23–30
- [Pap85] PAPERT, Seymour: *Gedankenblitze: Kinder, Computer und neues Lernen*. Rowohlt, 1985
- [Pas96] PASTERNAK, Arno: Thesen zur aktuellen didaktischen Diskussion. In: *FifF-Kommunikation* 2/96 (1996), S. 9–10. – <http://pasternak.in.hagen.de/if/koenigstein/grundbildung.htm>

- [Pas09] PASTERNAK, Arno: Kryptologie im Informatikunterricht der Sekundarstufe I. (2009). – <http://ls11-www.cs.uni-dortmund.de/people/paster/forschung/kryptologie/kryptologie.pdf>
- [Pat81] PATTIS, R.E.: *Karel the robot: a gentle introduction to the art of programming*. Wiley, 1981
- [Phi12] PHILLIPS, Pat (Hrsg.): *Special Issue – Computer Science K8: Building a Strong Foundation*. New York : CSTA, 2012
- [Pia74] PIAGET, Jean: *Der Aufbau der Wirklichkeit beim Kinde*. Ernst Klett, 1974
- [Pia00] PIAGET, J.: *Psychologie der Intelligenz*. Klett-Cotta, 2000
- [Pic65] PICHT, Georg: *Die Deutsche Bildungskatastrophe*. 2. München : Deutscher Taschenbuch Verlag, 1965. – 1. Auflage 1964, Walter-Verlag Olten
- [Pos36] POST, Emil L.: Finite Combinatory Processes-Formulation 1. In: *The Journal of Symbolic Logic* 1 (1936), Nr. 3, S. pp. 103–105
- [PSH09] POHL, Wolfgang ; SCHLÜTER, Kirsten ; HEIN, Hans-Werner: Informatik-Biber: Informatik-Einstieg und mehr. In: *INFOS 2009, 13. GI-Fachtagung Informatik und Schule, FU Berlin. GI-Lectures Notes in Informatics*. Berlin, Germany : Gesellschaft für Informatik GI, September 2009, S. 38–49
- [PT80] PILZ, Eva ; TIESLER, Bärbel ; GORNY, Peter (Hrsg.): *Informatik in der Sekundarstufe II - am Beispiel einer Unterrichtseinheit*. Oldenburg : Zentrum für pädagogische Berufspraxis Universität Oldenburg, 1980
- [Puh04] PUHLMANN, Hermann: Informatische Literalität nach dem PISA-Muster und ihre Operationalisierung durch Test-Items. In: *Informatica Didactica* 6 (2004)
- [PV09] PASTERNAK, Arno ; VAHRENHOLD, Jan: Rote Fäden und Kontextorientierung im Informatikunterricht. In: PETERS, Ingo-Rüdiger (Hrsg.): *Informatische Bildung in Theorie und Praxis*. Berlin : LOG IN Verlag, 2009. – ISBN 978-3-980-5540-7-7, S. 45–56
- [PV10] PASTERNAK, Arno ; VAHRENHOLD, Jan: Braided Teaching in Secondary CS Education: Contexts, Continuity, and the Role of Programming. In: CORTINA, Tom (Hrsg.) ; WALKER, Ellen (Hrsg.) ; ACM (Veranst.): *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE 2010)* ACM, ACM Press, 2010, S. 204–208
- [PV12] PASTERNAK, Arno ; VAHRENHOLD, Jan: Design and evaluation of a braided teaching course in sixth grade computer science education. In: KING, Laurie A. S. (Hrsg.) ; MUSICANT, David R. (Hrsg.) ; CAMP, Tracy (Hrsg.) ; TYMANN, Paul T. (Hrsg.): *SIGCSE*, ACM, 2012. – ISBN 978-1-4503-1098-7, S. 45–50

- [RA69] ROTH, Heinrich ; AEBLI, Hans: *Begabung und Lernen*. Klett, 1969 (Deutscher Bildungsrat)
- [Rai98] RAINES, Paul: *Tcl, Tk : kurz & gut*. Cambridge [u.a.] : O'Reilly, 1998
- [Rau08] RAU, Franco: *Informatik im Kontext — Web 2.0 — Anwendungen*. Universität Potsdam, 2008. – Bachelorarbeit
- [Rei06] REICH, Kersten: *Konstruktivistische Didaktik: Lehr- und Studienbuch mit Methodenpool*. Beltz, 2006 (Pädagogik und Konstruktivismus)
- [RFHN10] RASCH, Björn ; FRIESE, Malten ; HOFMANN, Wilhelm J. ; NAUMANN, Ewald: *Springer-Lehrbuch Bachelor*. Bd. 1.: *Quantitative Methoden*. 3., erw. Aufl. Heidelberg : Springer, 2010. – ISBN 978-3-642-05271-2
- [RNH04] REICHERT, R. ; NIEVERGELT, J. ; HARTMANN, W.: *Programmieren Mit Kara: Ein Spielerischer Zugang Zur Informatik*. Springer, 2004 (EXamen. press Series)
- [Rob67] ROBINSOHN, Saul B.: *Bildungsreform als Revision des Curriuculum*. Neuwied : Luchterhand Verlag, 1967
- [RR88] RATHFELDER, Heinz ; REINELT, Günther: *Computergrafik mit COMAL*. Frankfurt a.M. : M. Diesterweg, 1988
- [RT99] RAINES, Paul ; TRANTER, Jeff: *Tcl/Tk in a Nutshell*. O'Reilly Media, Incorporated, 1999 (In a Nutshell)
- [Rut56] RUTISHAUSER, Heinz.: *Automatische Rechenplanfertigung bei programmgesteuerten Rechenmaschinen*. Birkhäuser, 1956 (Mitteilungen aus dem Institut für Angewandte Mathematik an der Eidgenössischen Technischen Hochschule in Zürich)
- [RZ85] ROLFF, Hans-Günter ; ZIMMERMANN, Peter: *Neue Medien und Lernen*. Weinheim : Beltz Verlag, 1985
- [Sch79] SCHAUER, Helmut: *Pascal für Anfänger*. R. Oldenbourg, 1979 (Fortbildung durch Selbststudium). – ISBN 9783702901332
- [Sch86] SCHNUER, Günther: *Die deutsche Bildungskatastrophe: 20 Jahre nach Picht, Lehren und Lernen in Deutschland*. Busse Seewald, 1986
- [Sch92] SCHEFE, Peter: Informatik und Philosophie. Eine Einführung. In: [SHDK93], S. 1–9
- [Sch93] SCHWILL, Andreas: Fundamentale Ideen der Informatik. In: *Zentralblatt für Didaktik der Mathematik* 25 (1993), Nr. 1, S. 20–31

- [Sch02] SCHWANITZ, Dietrich: *Bildung - Alles, was man wissen mus.* 10. Auflage. Goldmann Verlag, 2002
- [Sch09a] SCHECKER, Horst: Rezension Gottfried Merzyn: Naturwissenschaften, Mathematik, Technik — immer unbeliebter? Die Konkurrenz von Schulfächern um das Interesse der Jugend im Spiegel vielfältiger Untersuchungen. Hohengehren: Schneider 2008, 158 S., 31. Abb. In: *PhyDid A - Physik und Didaktik in Schule und Hochschule* 8 (2009), Nr. 2
- [Sch09b] SCHLÜTER, Kirsten: Eine Studie zu den Merkmalen der Aufgabenschwierigkeit am Beispiel eines Informatik-Schülerwettbewerbs, Erster Teil: Aufgabenklassifizierung. In: *INFOS 2009, 13. GI-Fachtagung Informatik und Schule, FU Berlin. GI-Lectures Notes in Informatics.* Berlin, Germany : Gesellschaft für Informatik GI, September 2009, S. 38–49
- [Sch10] SCHLÜTER, Kirsten: Eine Studie zu den Merkmalen der Aufgabenschwierigkeit am Beispiel eines Informatik-Schülerwettbewerbs, Zweiter Teil: Empirische Aufgabenanalyse. In: *DDI 2010, Diethelm, Ira ; Dörge, Christina ; Hildebrandt, Claudia ; Schulte, Carsten (Hrsg.) : Didaktik der Informatik - Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik (6. Workshop der GI-Fachgruppe ‚Didaktik der Informatik‘, Oldenburg, 16.-17. September 2010).* Oldenburg, Germany : Gesellschaft für Informatik GI, September 2010, S. 69–80
- [Sch11] SCHOBERT, Nicole: *Konstruktivismus - Bedeutung für die Schule.* München : GRIN Verlag, 2011
- [SD08] SCHULTE, Carsten ; DÖRGE, Christina: Digitale Artefakte und Schlüsselkompetenzen im Informatikunterricht der Sekundarstufe I. In: BRINDA, Torsten (Hrsg.) ; FOTHE, Michael (Hrsg.) ; HUBWIESER, Peter (Hrsg.): *DDI Bd. 135, GI, 2008 (LNI), S. 13–22*
- [Sea90] SEARLE, John R.: Is the Brain a Digital Computer? In: *Proceedings and Addresses of the American Philosophical Association* 64 (1990), Nr. 3, S. 21–37
- [Sea92] SEARLE, John R.: Ist das Gehirn ein Digitalcomputer? In: [SHDK93], S. 211–232
- [Sen83] SENFTLEBEN, Dietrich: *Programmieren mit Logo.* Würzburg : Vogel-Buchverlag, 1983
- [SGEHV05] STEPHENSON, Chris ; GAL-EZER, Judith ; HABERMAN, Bruria ; VERNO, Anita: *The New Educational Imperative: Improving High School Computer Science Education.* 2005
- [Sha48] SHANNON, Claude E.: A Mathematical Theory of Communication. In: *The Bell Systems Technical Journal* 27 (1948), S. 379–423



- [SHDK93] SCHEFE, Peter (Hrsg.) ; HASTEDT, Heiner (Hrsg.) ; DITTRICH, Yvonne (Hrsg.) ; KEIL, Geert (Hrsg.): *Informatik und Philosophie [Dagstuhl, 21.-25. September 1992]*. BI Wissenschaftsverlag, 1993 . – ISBN 3-411-16521-9
- [SHG90] SCHÖNEBURG, Eberhard ; HANSEN, Nikolaus ; GAWELCZYK, Andreas: *Neuronale Netzwerke.: Einführung, Überblick und Anwendungsmöglichkeiten*. Markt-&-Technik-Verlag, 1990
- [Ski57] SKINNER, Burrhus F.: *Verbal behavior*. Appleton-Century-Crofts, 1957 (Century psychology series)
- [SL<sup>+</sup>11] SCHEUNGRAB, Christian ; LORBER, Vasco u. a. ; STAATSIINSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (Hrsg.): *Computeralgebrasysteme (CAS)im Mathematikunterricht des Gymnasiums Jahrgangsstufe 10*. München : Staatsinstitut für Schulqualität und Bildungsforschung, 2011
- [SSW10] STEHLIK, Mark ; STEPHENSON, Chris ; WILSON, Cameron: Running on empty: The Failure to Teach K-12 Computer Science in the Digital aAge. In: *EMS magazine* 39 (2010), Nr. 9, 74. <http://www.acm.org/runningonempty/fullreport.pdf>
- [Sta10] STARRUSS, Isabelle: *Synopse zum Informatikunterricht in Deutschland*. 2010. – Bachelorarbeit
- [Ste57] STEINBUCH, Karl: Informatik: Automatische Informationsverarbeitung. In: *SEG Nachrichten* (1957), Nr. 4, S. 171–176
- [Ste69] STEINBUCH, Karl: *Die Informierte Gesellschaft: Geschichte und Zukunft der Nachrichtentechnik*. Rowohlt, 1969 (Rororo Sachbuch)
- [Ste78] STEINBUCH, Karl: *Maßlos informiert: die Enteignung unseres Denkens*. Herbig, 1978
- [Ste93] STEINMÜLLER, Wilhelm: *Informationstechnologie und Gesellschaft: Einführung in die angewandte Informatik*. Darmstadt : Wissenschaftliche Buchgesellschaft, 1993
- [Ste09] STECHERT, Peer: *Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht*. Siegen, Universtit/ä Siegen, Diss., 2009
- [SZ78] SCHULZ-ZANDER, Renate: Analyse curricularer Ansätze für das Schulfach Informatik. In: ARLT, Wolfgang (Hrsg.): *EDV-Einsatz in Schule und Ausbildung : Modelle u. Erfahrungen*. München : Oldenbourg, 1978, S. 40–49
- [SZ86] SCHULZ-ZANDER, Renate: *Auswirkungen von Programmiersprachen auf das Problemlöseverhalten von Schülern: Theoretische Analyse und empirische Feldstudie*, Universität Hamburg, Diss., 1986

- [Tae83] TAEGER, Jürgen: *Die Volkszählung*. Rowohlt, 1983 (Rororo aktuell)
- [Tan03] TANENBAUM, Andrew S.: *Computernetzwerke*. 4., überarb. A. Pearson Studium, 2003. – ISBN 3827370469
- [TBAA09] TAUB, Rivka ; BEN-ARI, Mordechai ; ARMONI, Michal: The effect of CS unplugged on middle-school students' views of CS. In: *SIGCSE Bulletin* 41 (2009), Nr. 3, S. 99–103
- [TDJ+06] TUCKER, Allen ; DEEK, Fadi ; JONES, Jill ; MCCOWAN, Dennis ; STEPHENSON, Chris ; Verno, Anita ; TUCKER, Allen (Hrsg.): *A Model Curriculum for K-12 Computer Science. Final Report of the ACM K-12 Task Force Curriculum Committee*. 2. Auflage. New York : ACM, 2006. – 1. Auflage: Oktober 2003 – Im Internet [Stand: 2012-04-11] – <http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf> –
- [Tet92] TETENS, Holms: Informatik und die Philosophie des Geistes. In: [SHDK93], S. 175–123
- [Tom05] TOMASELLO, Michael: *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, 2005. – ISBN 9780674017641
- [TT89] TINNEFELD, Marie-Theres ; TUBIES, Helga: *Datenschutzrecht*. R. Oldenbourg, 1989
- [Tuc11] TUCKER, Allen (Hrsg.): *CSTA K-12 Computer Science Standards Revised 2011*. New York : CSTA, 2011. – Draft: March 2011 – Im Internet [Stand: 2012-04-11] – <http://csta.acm.org/includes/Other/CSTASTandardsReview2011.pdf> –
- [Tur36] TURING, Alan M.: On Computable Numbers, with an application to the Entscheidungsproblem. In: *Proceedings of the London Mathematical Society* 42 (1936), S. 230–265
- [Tur50] TURING, Alan M.: *Computing Machinery and Intelligence*. 1950. – One of the most influential papers in the history of the cognitive sciences: <http://cogsci.umn.edu/millennium/final.html>
- [Tur84] TURKLE, Sherry: *Die Wunschmaschine : Vom Entstehen der Computerkultur*. Reinbek bei Hamburg : Rowohlt, 1984
- [Voh00] VOHNS, Andreas: *Das Messen als fundamentale Idee im Mathematikunterricht der Sekundarstufe I*. Universität- Gesamthochschule Siegen, 2000. – Masterarbeit
- [Vol97] VOLMERT, Johannes: *Grundkurs Sprachwissenschaft: Eine Einführung in die Sprachwissenschaft für Lehramtsstudiengänge*. 2. Auflage. W. Fink, 1997 (UTB. Uni-Taschenbücher)

- [Wag68] WAGENSCHHEIN, Martin: *Verstehen lehren*. Beltz, 1968 (Beltz Taschenbuch)
- [Wag71] WAGENSCHHEIN, Martin: *Die pädagogische Dimension der Physik*. Braunschweig : Westermann, 1971 (Grundthemen der pädagogischen Praxis)
- [Wal07] WALL, Kurt: *Tcl and Tk Programming for the Absolute Beginner*. Boston, MA, United States : Course Technology Press, 2007
- [WC53] WATSON, James D. ; CRICK, Francis H.: Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. In: *Nature* 171 (1953), April, Nr. 4356, S. 737–738
- [Web00] WEBSTER, Tim ; FRANCIS, Alex (Hrsg.): *Tcl/tk für Dummies : gegen den täglichen Frust mit Tcl/Tk*. Bonn : mitp, 2000. – ISBN 3–8266–2872–1
- [Wee07] WEEGER, Moritz: *Synopse zum Informatikunterricht in Deutschland*. Technische Universität Dresden, 2007. – Bachelorarbeit
- [Weg05] WEGENER, Ingo: *Theoretische Informatik: eine algorithmenorientierte Einführung*. Teubner, 2005 (Leitfäden der Informatik). – ISBN 9783835100336
- [Wei78] WEIZENBAUM, Joseph: *Die Macht der Computer und die Ohnmacht der Vernunft*. Frankfurt : Suhrkamp, 1978
- [Wei01] WEINERT, Franz: Vergleichende Leistungsmessung in Schulen - eine umstrittene Selbstverständlichkeit. In: *Leistungsmessung in Schulen* (2001), S. 17–31
- [Wen07] WENTZEL, Wenka: *Ingenieurin statt Germanistin und Tischlerin statt Friseurin? Evaluationsergebnisse zum GirlsDay Mädchen-Zukunftstag*. Bielefeld. Bielefeld : Kompetenzzentrum Technik-Diversity-Chancengleichheit e.V., 2007
- [Whi58] WHITEHEAD, Alfred N.: *Eine Einführung in die Mathematik*. Lehnen, 1958 (Dalp-Taschenbücher)
- [Wie48] WIENER, Norbert: *Cybernetics: Or, Control and Communication in the Animal and the Machine*. Mit Press, 1948 (M.I.T. paperback series)
- [Wie68] WIENER, Norbert: *Kybernetik: Regelung und Nachrichtenübertragung in Lebewesen und Maschine*. Rowohlt, 1968 (rororo Wissen)
- [Win96] WINSLOW, Leon E.: Programming pedagogy —— a psychological overview. In: *SIGCSE Bull.* 28 (1996), September, S. 17–22
- [Win06] WING, Jeannette M.: Computational Thinking. In: *Communications of the ACM* 49 (2006), Nr. 3, S. 33–35
- [Win08] WINTERHOFF, Michael: *Warum unsere Kinder Tyrannen werden: Oder: Die Abschaffung der Kindheit*. Gütersloher Verlag-Haus, 2008

- [Wir71] WIRTH, Niklaus: The Programming Language Pascal. In: *Acta Informatica* 1 (1971), Nr. 1, S. 35–63
- [Wir72] WIRTH, Niklaus: The Programming Language Pascal (Revised Report / ETH Zürich. 1972 (Berichte der Fachgruppe Computerwissenschaften – 5). – Forschungsbericht. – <http://www.fh-jena.de/~kleine/history/languages/Wirth-PascalRevisedReport.pdf> – last visited 15<sup>th</sup> March 2012
- [Wit63] WITTENBERG, Alexander I.: *Bildung und Mathematik: Mathematik als exemplarisches Gymnasialfach*. Stuttgart : E. Klett, 1963 (Erziehungswissenschaftliche Bücherei)
- [Wit03] WITTEN, Helmut: Allgemeinbildender Informatikunterricht? Ein neuer Blick auf H. W. Heymanns Aufgaben allgemeinbildender Schulen. In: HUBWIESER, Peter (Hrsg.): *Informatik und Schule – Informatische Fachkonzepte im Unterricht INFOS 2003 – 10. GI-Fachtagung 17.–19. September 2003, München* Bd. 32. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, September 2003 (GI-Edition – Lecture Notes in Informatics – Proceedings P 32), S. 59–75
- [Wit07] WITTKÉ, Gregor: *Kompetenzerwerb und Kompetenztransfer bei Arbeitssicherheitsbeauftragten*. Berlin, Freie Universität, Diss., 2007
- [Yah10] YAHYA, Adnan H.: The inteaction between high school curriculum and first year college courses: the case of computing. In: *Proceedings of the 41st ACM technical symposium on Computer science education*. New York, NY, USA : ACM, 2010 (SIGCSE '10). – ISBN 978–1–4503–0006–3, 406–410
- [ZG04] ZIMBARDO, Philip G. ; GERRIG, Richard J.: *Psychologie. Eine Einführung*. 16. aktualisierte Auflage. Pearson Studium, 2004
- [Zim80] ZIMMERMANN, Hubert: OSI reference model - The ISO model of architecture for open systems interconnection. In: *IEEE Transactions on Communications* 28 (1980), April, Nr. 4, S. 425–432
- [Zim11] ZIMMERMANN, Gero: *Die Philosophie des Geistes im Spiegel der Informatik und der Komplexitätstheorie*. Tectum Verlag, 2011 (Interdisziplinäre Perspektiven in der Philosophie)
- [Zus72] ZUSE, Konrad: *Der Plankalkül*. GMD-Forschungszentrum Informationstechnik, 1972 (Berichte der Gesellschaft für Mathematik und Datenverarbeitung 63)
- [Zus10] ZUSE, Konrad: *Der Computer - mein Lebenswerk*. 5., unveränd. Aufl. Berlin : Springer, 2010. – ISBN 978–3–642–12095–4

