

Kontextorientierter Anfangsunterricht in Informatik

Arno Pasternak
Fritz-Steinhoff-Gesamtschule Hagen
Technische Universität Dortmund
Fakultät für Informatik

Abstract: In den letzten Jahren wird wieder über ein *Schulfach Informatik in der Sekundarstufe I* diskutiert. Naheliegender ist, dies in Form einer *Informatik im Kontext* oder – wenn dies nicht möglich – zumindest *kontextorientiert* zu unterrichten. Allerdings muss dann darauf geachtet werden, dass nicht der Kontext, sondern trotz allem die informatischen Inhalte im Mittelpunkt des unterrichtlichen Geschehens stehen. Für viele Kolleginnen und Kollegen erscheint es schwierig, die Themengebiete *Algorithmen und Programmierung* in ein solches Konzept einzubetten. Dabei können als Strukturierungsmittel *Rote Fäden* helfen, die aus Sicht der Schüler einen Zusammenhang der unterschiedlichsten Themen ergeben. Allerdings ist es dann deutlich schwieriger, eine für diesen Unterricht geeignete Programmiersprache auszuwählen. *Tcl/Tk* ist eine für ein derartiges Konzept geeignete Sprache, die in den unterschiedlichsten Kontexten eingesetzt werden kann. Entsprechend dieser Ideen wird ein Unterrichtsprojekt erläutert, das derzeit in der Fritz-Steinhoff-Schule in Hagen durchgeführt wird.

1 Informatik in der Sekundarstufe I

Aus Sicht der Informatik ist eine informatische Bildung als Teil der Allgemeinbildung unstrittig [BFF⁺00],[BFF⁺08]. Die Öffentlichkeit und vor allem die Schüler sind dagegen bei vielen konkreten Inhalten sehr skeptisch. Hier hat sich in den letzten Jahrzehnten ein Wandel vollzogen. Wurde früher *Informatik* häufig fälschlicherweise mit *Programmierung* gleich gesetzt, scheint einem Großteil der Öffentlichkeit und einem Teil der Schüler die *Programmierung* heute fast nebensächlich zu sein. Schließlich programmiert kaum noch jemand – geschweige denn ein Schüler – selbst eine Anwendung, sondern erwirbt diese ausschließlich als fertiges Produkt. Dieses gilt für Schüler der Sekundarstufe II und natürlich umso mehr für Schüler der Sekundarstufe I.

1.1 Informatik im Kontext (in der Sek I)

Allein deshalb funktioniert für das Schulfach Informatik eine aus der Fachsystematik abgeleitete *Abbild-Didaktik* nicht (mehr). Stattdessen bietet sich an, nachvollziehbare Zusammenhänge aus der Lebenswirklichkeit aus Sicht des Schülers als Ausgangspunkt zu

nehmen, wie es beispielsweise im Projekt **Informatik im Kontext**¹ vorgeschlagen wird. Diese Idee ist nicht neu. Da auch die naturwissenschaftlichen Fächer ähnliche Probleme bei der Vermittlung ihrer Fachinhalte erleben, hat dort seit einigen Jahren ein Umdenkungsprozess begonnen, der in Projekten wie *Chemie im Kontext*² [IPF08], *Biologie im Kontext*³ und *Physik im Kontext*⁴ mündete.

Mit dem vorliegenden didaktischen Ansatz sollen Informatikinhalte (möglichst oft) in Kontexten unterrichtet werden. Informatische Inhalte sind dabei das Ziel und nicht das Nebenprodukt.

1.2 Das Konzept der Roten Fäden

Definition der Roten Fäden Es wird ein Strukturierungsmittel benötigt, das über eine Unterrichtseinheit hinaus dem Lehrer die Gewissheit gibt, dass die vermittelten Inhalte einen fachlichen Zusammenhang ergeben, der aus Schülersicht nachvollziehbar ist. Ein solches Strukturierungsmittel stellen die **Roten Fäden** dar [PV09], [PV10].

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.*

[PV09, S.6]

Es ist naheliegend, dass es viele unterschiedliche Fäden konzipiert werden können. Es obliegt dem Fachkollegen, eine Auswahl der (endlich vielen) Fäden vorzunehmen. Solche *Roten Fäden* durchziehen die Bildungsmatrix [PV09, S.4] und sichern dadurch ein Erfüllen der Zellen in den Zeilen und Spalten der Matrix.

Programmierung in der Sekundarstufe I In den Bildungsstandards werden an den verschiedensten Stellen zum Thema **Algorithmen** Forderungen aufgestellt. So wird u.A. im Kapitel *Kompetenzen über alle Jahrgangsstufen* formuliert:

„Schülerinnen und Schüler aller Jahrgangsstufen

¹ <http://www.informatik-im-kontext.de>, Zugriff: 6.6.2011

² <http://www.chik.de>, Zugriff: 6.6.2011

³ <http://bik.ipn.uni-kiel.de/typo3/index.php?id=3>, Zugriff: 6.6.2011

⁴ <http://www.uni-kiel.de/piko/?topic=8>, Zugriff: 6.6.2011

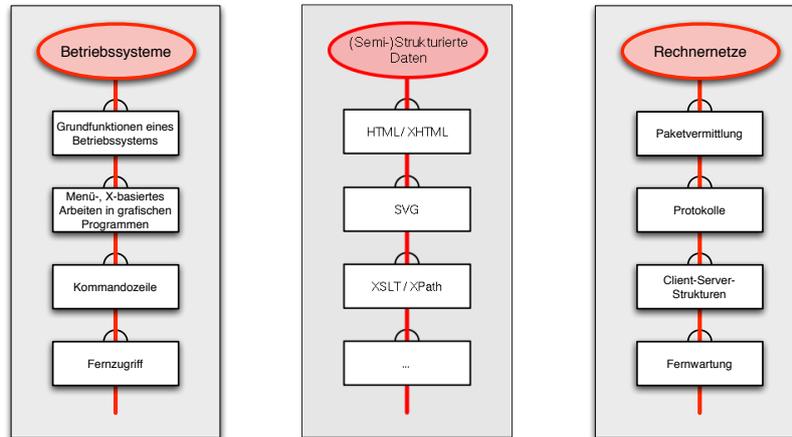


Abbildung 1: Mögliche **Rote Fäden** im Informatik-Unterricht der Sekundarstufe I [PV09, S.7]

- 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“ [BFF⁺08, S.13].

Langjährige Erfahrung als Lehrer legt die Aussage nahe, dass ein Algorithmus zumeist in letzter Konsequenz erst dann voll verstanden ist, wenn er (zumindest potentiell) programmiert werden kann oder programmiert worden ist. Lee meint entsprechend: „*In the beginning, the initial idea is often expressed in such complex terms that it is overlooked by the majority of read’ers. 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.*“ [Le96].

Sind also *Algorithmen* Bestandteil eines Curriculums, dann ist auch *Programmierung* Bestandteil dieses Curriculums. 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“ [Hr08, S.11]? Damit folgt aus den Bildungsstandards Informatik:

Programmierung ist ein unverzichtbarer Teil des Informatikunterrichtes in der Sekundarstufe I.

Programmierung als Roter Faden 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 der Programmierung 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“ [Tu06, S.2].

Dieses Werkzeug soll dann auch als Werkzeug erfahren werden und nicht durch ein *Lernen auf Vorrat* in Form eines Lehrganges in Programmierung welcher Programmiersprache auch immer. Die Programmierung kann dann auch kein Selbstzweck sein. Soll die Schülerin und der 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 dann die Erkenntnisse auf allgemeiner anwendbare Sprachen übertragen werden. Möchten wir, dass die in der Bildungsmatrix angegebenen Ziele verflochten werden, muss auch die Programmierung im Unterricht als ein *Roter Faden* angesehen werden und entsprechend konzipiert werden.

1.3 Werkzeug Programmiersprache

Es gibt sicher viele Programmiersprachen und Programmiersprachenumgebungen, die aus Sicht des Unterrichtes unter den verschiedensten Gründen interessant sein können. Soll die Programmierung allerdings als *Roter Faden* konzipiert werden, ist die Auswahl kleiner. Diese Sprache soll in den verschiedensten Kontexten wie ein *Klebstoff* gute Dienste leisten. „Scripting languages are sometimes referred to as glue languages or system integration languages“ [Ou97], schreibt Ousterhout über derartige Sprachen.

Die von ihm in den 80er Jahren initiierte und später von einer sehr grossen Entwicklergemeinschaft ausgebauten und gepflegten Sprache [Ou95] erfüllt in vielfältiger Hinsicht die Ansprüche, die an eine Unterrichts-Sprache in der Sekundarstufe I gestellt werden können:

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-Scripte lassen sich sehr einfach erstellen.

Es ist daher sicher nicht übertrieben, *Tcl/Tk als ein programmiersprachliches Universalwerkzeug* im gesamten Informatik-Unterricht der Sekundarstufe I anzusehen.

Kontexte im Unterricht Der *Rote Faden Programmierung* realisiert mit der Programmiersprache *Tcl/Tk* muss entsprechend obiger Überlegungen mit den anderen *Roten Fäden* idealerweise in Kontexten verflochten werden.

Ohne Anspruch auf Vollständigkeit sind hier einige Themengebiete aufgelistet, die entsprechend der Ideen der Schüler und Lehrer kontextorientiert unterrichtet werden können: Zeichnen/Malen, Spielen, Multimedia (Sound, Videos, ...), Netze, Web etc..

Aus diesen und in diesen Kontexten müssen entsprechend der Idee der *Roten Fäden* die informatischen Inhalte extrahiert werden. Im folgenden Kapitel wird die Umsetzung dieser Ideen im ersten Jahr Informatik-Unterricht dargestellt.

2 Das Unterrichtsprojekt

Schülerinnen und Schüler an Gesamtschulen in NRW müssen zum Jahrgang 6 ein sogenanntes Wahlpflichtfach [fSuW07, §19] wählen, das sie dann bis zum Ende des Jahrganges 10 belegen. Dieses Fach ist Hauptfach. Zum Schuljahresbeginn 2010/2011 wurde an der Fritz-Steinhoff-Schule eine WP-Gruppe im Jahrgang 6 eingerichtet, in der fachverbindend Informatik und Physik unterrichtet wird.

Es soll gezeigt werden, dass an einer Gesamtschule, deren Schülerinnen und Schüler aus allen sozialen Schichten kommen, die die verschiedensten Bildungsvoraussetzungen mitbringen und weitgehend auch alle Begabungsprofile abdecken, ein Informatikunterricht stattfinden kann, der im Sinne der Bildungsstandards Informatik informatische Bildung für alle ermöglicht. Daher wurde bei der Zusammenstellung darauf geachtet, dass diese Gruppe in jeglicher Hinsicht möglichst heterogen ist. Die Gruppe entspricht mit 27 Schülerinnen und Schülern weitgehend einer typischen Klasse.

Die Schüler haben bisher keinen Unterricht in Informatik erhalten. An der Fritz-Steinhoff-Schule werden allerdings im 5. Jahrgang im Rahmen sogenannter verpflichtender Zertifikatskurse alle Schülerinnen und Schüler mit den Grundlagen der Textverarbeitung und Tabellenkalkulation vertraut gemacht.

Auf der Webseite der Schule steht der Gruppe ein geschützter Bereich zu Verfügung, über den den Schüler – aber auch deren Eltern – alle Materialien zur Verfügung gestellt werden, die im Rahmen dieses Unterrichtes benutzt werden.

3 Der Unterrichtsverlauf

Schüler erfahren heute den Computer im Wesentlichen als *Black-Box*. Zudem ist der Zugang zum Computer im Allgemeinen mit einer grafischen Oberfläche verdeckt. Der Computer als zeichenverarbeitende Maschine wird ihnen nicht mehr deutlich und dadurch auch nicht mehr vermittelt.

Aus Sicht des *Roten Fadens Programmierung* ist daher genau dieses Bewusstsein zu schaffen. Schülerinnen und Schüler sollen daher 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.

3.1 Einheit 1: Buttons und Labels

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 Oberfläche von *Tcl/TK* eingeführt und sie erkennen, dass die Arbeit mit *Tcl/TK* und ihre tägliche Beschäftigung mit dem Computer nicht auseinanderfallen.

Die ersten Anweisungen in der *TCL*-Konsole sind daher:

```
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.

Mit wenig Mühe gestalten die Schüler damit eine Oberfläche, die im Grafikfenster einer üblichen Oberfläche eines heutigen Computer nicht unähnlich ist.

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 aus Labeln oder auch Buttons produziert.

Als Kritik muss angemerkt werden, dass diese Übungen zwar ausreichend umfangreich waren, aber nicht genügend Abwechslung boten. Hier müssen noch mehr alternative Beispiele angeboten werden, die auch mehr den verschiedenen Anforderungsniveaus gerecht werden.

3.2 Einheit 2: Bilder und Webseiten

Bilder exportieren Im weiteren Unterrichtsverlauf wurde der Stundenplan als Bild in eine Webseite integriert. Dazu muss 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. Hierin unterscheidet sich Tcl/Tk wesentlich von anderen interpretierten Sprachen wie beispielsweise *Logo* oder *Python*.

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*⁵ installiert. Mit dem dort enthaltenen Tool *import*⁶ 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.

Die erste Webseite Das Bild im Browser soll mit Überschriften und evtl. Kommentaren versehen werden. Das bietet Anlass, eine erste kleine HTML-Datei zu erzeugen. Diese wird natürlich vorgegeben. Allerdings lassen sich anschliessend mehr oder weniger gezielte Veränderungen vornehmen, sodass die Schülerinnen und Schüler den Aufbau einer HTML-Datei kennen und auch zumindest teilweise schon verstehen lernen.

Die HTML-Datei sieht folgendermassen aus:

```
<html>
<head><title>Stundenplan von Emil</title></head>

<body>
<center>
<h1>Stundenplan von Emil</h1>

```

⁵ Dieses Programmpaket gibt es auch für das Betriebssystem Windows.

⁶ Warum das Tool zum *Exportieren* nun gerade *import* heisst, ist unverständlich.

```
</center>  
</body>  
</html>
```

Mit diesem Exkurs zu HTML haben wir den *Roten Faden (semi-)Strukturierte Daten* aufgegriffen.

3.3 Einheit 3: Dateiorganisation

Um in verschiedenen Computerräumen arbeiten zu können, wurde die Organisation und die Darstellung von Dateien und Dateibäumen auf einem einzelnen Computer und bei uns im Schulnetz behandelt. Besonderen Wert wurde auf die Darstellung von Bäumen gelegt. Die Schüler mussten dabei verschiedene Repräsentationen zeichnerisch darstellen und diese in Kommandos auf der Betriebssystemebene übertragen und auch ausprobieren.

3.4 Einheit 4: Zeichnungen in Tcl/Tk

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.

Dazu ist notwendig, dass sie analysieren, aus welchen Komponenten – in dieser Anfangsphase ausschließlich verschiedenste Strecken – das Bild besteht und dieses in eine Sequenz von fast immer gleichartigen Anweisungen übersetzen. Nebenbei wird auch das Arbeiten in Koordinatensystemen, das sie in der Mathematik kennengelernt haben, wiederholt und vertieft. Einfache Modellierungsansprüche werden bei freien Zeichnungen gefordert.

3.5 Einheit 5: Vektor- und Pixelgrafik

Mit Pixelgrafiken haben die Schüler schon 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. Ideal ist das *xpm*-Format, das eine Präsentation der Pixel im ASCII-Format enthält. Diese Dateien können im Editor manipuliert werden und anschließend von einem Grafik-Betrachter angesehen werden. Mit dieser Einheit befinden wir uns damit im *Roten Faden Multimedia oder Grafik*.

Die Schüler erzeugen eine kleine Grafik in Tcl/Tk und exportieren diese wie schon bekannt in eine Grafik, diesmal aber in das *xpm*-Format. Im Gegensatz zur Darstellung durch geometrische Objekte wie Linien in Tcl/Tk wird im *xpm*-Format jeder Punkt durch ein Zeichen dargestellt.

3.6 Klassenarbeiten

Das Wahlpflichtfach ist an der Gesamtschule Hauptfach. Dementsprechend müssen auch Klassenarbeiten – 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 sicher 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 darstellen. 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.

Es zeigte sich in diesen Klassenarbeiten, dass ein Notenspektrum vergleichbar den Arbeiten in Mathematik erreicht wurde.

Um einen Eindruck von diesen Aufgaben zu erhalten, werden hier einige dieser Aufgaben dargestellt:

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.

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 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?

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*?

4 Vorläufiges Fazit

Nach fast einem Jahr **Informatikunterricht im Jahrgang 6** kann ein erster Versuch der Zusammenfassung erfolgen.

- 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.
- Klassenarbeiten sind sinnvoll zu stellen und ergeben ein für Schulen „normales“ Ergebnis.
- 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.
- Eine Unterstützung seitens des Elternhaus ist oft nicht möglich. Das ist ein Problem und eine Chance zugleich.

Als vorläufiges Fazit kann man formulieren, dass die Schülerinnen und die Schüler ihre Erwartungen mit dem gewählten Fach erfüllt sehen. Es ist deutlich, dass auch in unteren Jahrgängen ein Unterricht möglich ist, in dem die Inhalte der Informatik das angestrebte Ziel darstellen. Dabei ist eine Orientierung an Kontexten aus der Lebenswirklichkeit der Schüler machbar. Die Bildungsstandards Informatik bilden dabei eine Richtschnur. Es bleibt eine spannende Frage, wie mit dem Abschluss dieses Kurses am Ende des 10. Jahrganges in einigen Jahren die Bildungsstandards in Informatik [BFF⁺08] erfüllt werden. Dieses zu überprüfen, bleibt weiteren Untersuchungen vorbehalten.

Literaturverzeichnis

- [BFF⁺00] Norbert Breier, Michael Fothe, Steffen Friedrich, Peter Hubwieser, Bernhard Koerber, Gerhard Röhner, Sigrid Schubert und Monika Seiffert. *Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen*. LOGIN-Verlag, Berlin, fachausschuss 7.3 „informatische bildung in schulen“ der gi. Auflage, 2000, Beilage zur LOGIN Nr. 20(2000) Heft 2
- [BFF⁺08] Thorsten Brinda, Michael Fothe, Steffen Friedrich, Bernhard Koerber, Hermann Puhmann, Gerhard Röhner und Carsten Schulte. *Grundsätze und Standards für die Informatik in der Schule*. LOGIN-Verlag, Berlin, arbeitskreis „bildungsstandards“ der gi. Auflage, 2008, Beilage zur LOGIN Nr. 150/151
- [fSuW07] NRW Ministerium für Schule und Weiterbildung. *Ausbildungs- und Prüfungsordnung Sekundarstufe I, APO-S I*. Ritterbach-Verlag, Frechen, 2007.
- [Hr08] Juraj Hromkovič. *Lehrbuch Informatik : Vorkurs Programmieren, Geschichte und Begriffsbildung, Automatenentwurf*. Studium. Vieweg und Teubner Verlag, Wiesbaden, 1. Auflage, 2008.
- [IPF08] Bernd Ralle Ilka Parchmann und David-S. Di Fuccia. Entwicklung und Struktur der Unterrichtskonzeption Chemie im Kontext. In R. u. a. Demuth, Hrsg., *Chemie im Kontext - Von der Innovation zur nachhaltigen Verbreitung eines Unterrichtskonzepts*, Seiten 9–48, Münster, 2008. Waxmann.
- [Le96] John A. N. Lee. 'Those who forget the lessons of history are doomed to repeat it' : or, Why I study the history of computing. *Annals of the History of Computing, IEEE*, 18(2):54–62, 1996.
- [Ou95] John K. Ousterhout. *Tcl und Tk : Entwicklung grafischer Benutzerschnittstellen für das X-Window-System*. Professional computing. Addison-Wesley Professional Computing, Bonn, 1. Auflage, 1995.
- [Ou97] John K. Ousterhout. Scripting: Higher Level Programming for the 21st Century. *IEEE Computer*, 31:23–30, 1997.
- [PV09] Arno Pasternak und Jan Vahrenhold. Rote Fäden und Kontextorientierung im Informatikunterricht. In Ingo-Rüdiger Peters, Hrsg., *Informatische Bildung in Theorie und Praxis*, Seiten 45–56, Berlin, 2009. LOG IN Verlag.
- [PV10] Arno Pasternak und Jan Vahrenhold. Braided Teaching in Secondary CS Education: Contexts, Continuity, and the Role of Programming. In Tom Cortina und Ellen Walker, Hrsg., *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE 2010)*. ACM, ACM Press, 2010.
- [Tu06] Allen Tucker, Hrsg. *A Model Curriculum for K-12 Computer Science. Final Report of the ACM K-12 Task Force Curriculum Committee*. ACM, New York, 2. Auflage, 2006.