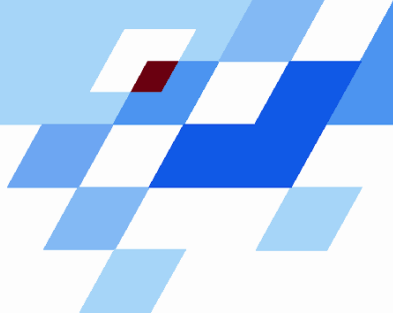

Wintersemester 2006/07

Einführung in die Informatik für Naturwissenschaftler und Ingenieure

(alias **Einführung in die Programmierung)**

(Vorlesung)

Prof. Dr. Günter Rudolph

Fachbereich Informatik

Lehrstuhl für Algorithm Engineering

Bisher:

Prozedurale Programmierung betrachtet, weil

- es in der industriellen Praxis noch viele C-Programme gibt, die nicht auf C++ umgestellt sondern nur gewartet werden:
  - Umstellung wg. Programmgröße zu teuer und
  - Gefahr der Einführung von Programmierfehlern bei Umstellung

  $\Rightarrow$ ein Programmierer muss beide Seiten kennen

- Wesentliche Grundlagen des algorithmischen Programmierens (wie Konstruktionsprinzipien von Funktionen, Rekursionsprinzip) müssen bekannt sein, bevor fruchtbare Auseinandersetzung mit objektorientierten Konzepten von C++ möglich

  $\Rightarrow$ der Blick auf das Wesentliche wird so frei gemacht

Probleme der prozeduralen Sichtweise

```
#include <iostream.h>
using namespace std;

void zeigeKonto(double stand) {
  cout << "Kontostand: " << stand << endl;
}
void zeigeAlter(int alter) {
  cout << "Alter: " << alter << endl;
}
int main() {
  double g = 3129.34;       // Guthaben
  int a = 42;               // Alter
  zeigeKonto(a);
  zeigeKonto(g);
  zeigeAlter(g);
  zeigeAlter(a);
  return 0;
}
```

Ausgabe:

Kontostand: 42

Kontostand: 3129.34

Alter: 3129

Alter: 42

Beobachtung:

- unzulässige Funktionen angewandt auf Daten → semantischer Fehler

- Daten sind ungeschützt: Trennung von Daten und Funktion

Die objektorientierte Sichtweise

natürliche Umgebung → Pflanzen, Autos, Häuser, … (= Objekte der realen Welt)

Objekt =

„Gegenstand, mit dem etwas geschieht oder geschehen soll“ (Duden)

„Gegenstand der Erkenntnis und Wahrnehmung, des Denken und Handelns“ (Brockhaus)

⇒ reale Gegenstände, aber auch von Menschen **gedachte** Gebilde

Modell = Ausschnitt aus realer Welt

Modellbildung →

PKW	
Typ:	Sportwagen
Hersteller:	Bugatti
Farbe:	Froschgrün
Geschwindigkeit:	245 km/h

Anwendungsproblem:

⇒ Modellierung ⇒ Reduzierung auf das „Wesentliche“

„wesentlich“ im Sinne unserer Sicht auf die Dinge bei diesem Problem

→ es gibt verschiedene Sichten auf dieselben Objekte!

⇒ schon bei der Problemanalyse denken im Sinne von

Objekten und ihren Eigenschaften und Beziehungen untereinander

OOP:

- Formulierung eines Modells in Konzepten & Begriffen der realen Welt
- nicht in computertechnischen Konstrukten wie Haupt- und Unterprogramm

Beispiel: Konten

Girokonto	**Sparkonto**
Kontonummer	Kontonummer
Inhaber	Inhaber
Saldo	Saldo
HabenZinsSatz	HabenZinsSatz
SollZinsSatz	
ÜberziehungsZinsSatz	
DispositionsKredit	

gemeinsame Attribute

typisch für alle Ausprägungen der **Klasse** Girokonto

typisch für alle Ausprägungen der **Klasse** Sparkonto

- ein Objekt ist eine Ausprägung / Instanz einer Klasse

Kontonummer	12345
Inhaber	Hugo Hase
Saldo	2344.34
HabenZinsSatz	1.5

Kontonummer	34321
Inhaber	Maria Kron
Saldo	7654.12
HabenZinsSatz	1.675

→ zwei Instanzen der Klasse **Sparkonto**

- auf den Konten lassen sich Operationen ausführen:
  anlegen, einzahlen, auszahlen, anzeigen, auflösen, …

→ Methoden, die sich auf ein Objekt anwenden lassen

Synonyme: Elementfunktionen, Klassenfunktionen

engl.: *methods, member functions*

Klasse = Beschreibung von **Eigenschaften** und **Operationen**

⇒ Eine Klasse ist also die Beschreibung des Bauplans (Konstruktionsvorschrift) für konkrete (mit Werten belegte) Objekte

⇒ Eine Klasse ist **nicht** das Objekt selbst

⇒ Ein Objekt ist eine **Instanz** / Ausprägung einer Klasse

Zusammenfassung von Daten / Eigenschaften und Operationen …

… kennen wir bereits von den abstrakten Datentypen (ADT)!

Zugriff auf Daten nur über Operationen der Klasse;
man sagt auch: dem Objekt wird eine Nachricht geschickt:

Objektname.Nachricht(Daten)

- **Klasse:**
  Beschreibung einer Menge von Objekten mit
  gemeinsamen Eigenschaften und Verhalten.
  Ist ein Datentyp!

- **Objekt:**
  Eine konkrete Ausprägung, eine Instanz, ein Exemplar der Klasse.
  Belegt Speicher!
  Besitzt Identität!
  Objekte tun etwas; sie werden als Handelnde aufgefasst!

- **Methode / Klassenfunktion:**
  Beschreibt das Verhalten eines Objektes.
  Kann als spezielle Nachricht an das Objekt aufgefasst werden.