

# 9. Übungsblatt

Abgabe Theorie bis zum 17.1.06

[www.math.tu-berlin.de/Vorlesungen/WS05/ProgMa](http://www.math.tu-berlin.de/Vorlesungen/WS05/ProgMa)

## 1. Aufgabe

(3 Punkte)

Zeichnen Sie ein UML-Diagramm, das die Beziehungen zwischen den folgenden Begriffen, interpretiert als Klassen und Objekte, darstellt:

*Datentyp, Ganze Zahl, Gleitpunktzahl, short, int, char, float, char zeichen, int zaehler, float pi, Standard-C++-Datentyp, selbstdefinierter Datentyp.*

## 2. Aufgabe

(5 Punkte)

Welche der folgenden Aussagen sind richtig, welche nicht?

- Ein Objekt ist ein Exemplar einer Klasse.
- Eine Klasse ist ein Objekt.
- Bei der Vererbung erbt die übergeordnete Klasse alle Eigenschaften der Unterklasse.
- Eine Objekt ist eine Klasse.
- Eine Unterklasse erbt nur bestimmte, vom Benutzer wählbare Eigenschaften von ihrer Oberklasse.
- Ein Objekt ist eine Instanz einer Klasse.
- Eine Unterklasse ist eine Klasse, die durch Vererbung aus einer anderen Klasse entsteht.
- Das Neue an der Objektorientierung ist die klare Trennung zwischen Daten und Funktionen, so dass beliebige Funktionen jederzeit auf globale Daten zugreifen können.
- Eine Unterklasse erbt alle Eigenschaften von ihrer Oberklasse.
- Eine Klasse ist ein Exemplar eines Objektes.

## 3. Aufgabe

(8 Punkte)

Jemand kommt zu Ihnen und sagt: "Ich möchte gerne ein Programm haben, das verschiedene geometrische Figuren auf dem Bildschirm anzeigt. Ich dachte an Kreise, Rechtecke und Dreiecke. Die Position des Schwerpunkts soll als  $x$  und  $y$ -Koordinate eingegeben werden können. Zusätzlich soll man beim Kreis den Radius, bei Rechteck und Dreieck die Seitenlängen einzeln setzen können."

Zeichnen Sie ein UML-Diagramm, das diese Beziehungen darstellt. Dabei sollen ebenfalls die Attribute, eventuell vorhandene Zusicherungen und Methoden (Name, evtl. Parameter und Rückgabewert) dargestellt werden.

## 8. Programmieraufgabe

(Vorführen bis zum 24.1.06)

- Laden Sie die Dateien `bruch.h` und `bruch.cc` von der Homepage.
- Implementieren Sie einen oder mehrere Konstruktoren, so dass man einen Bruch wahlweise
  - ohne Parameter (dann initialisiert mit 0/1),
  - mit einem Parameter (dann als Zähler interpretiert mit Nenner =1)
  - oder mit zwei Parametern (interpretiert als Zähler und Nenner)

erzeugen kann.

- Implementieren Sie eine Methode `read`, die einen Bruch von der Tastatur einliest. Stellen Sie sicher, dass der Nenner  $\neq 0$  ist.
- Implementieren Sie die Operatoren `+`, `*`, `<` für die Klasse `Bruch`.