

10. Übungsblatt

www.math.tu-berlin.de/Vorlesungen/WS06/ProgMa

Theoretische Aufgaben: Abgabe in der Vorlesung am 16.1.2007

Bei späterer Abgabe werden die erreichten Punkte nur zu 50 % angerechnet.

1. Aufgabe (1+3 Punkte)

- (a) Rechnen Sie die Dezimalzahl 563 ins Zweiersystem um.
- (b) Schreiben Sie einen Algorithmus, der eine Dezimalzahl $z \in \mathbb{N}$ in ein Zahlensystem mit Basis $b \in \mathbb{N}, b > 1$, umrechnet.

2. Aufgabe (1+4 Punkte)

- (a) Rechnen Sie die Dezimalzahl 0.3 ins Zweiersystem um.
- (b) Schreiben Sie einen Algorithmus, der eine Dezimalzahl $z \in [0, 1)$ in ein Zahlensystem mit Basis $b \in \mathbb{N}, b > 1$, umrechnet. Es sollen maximal k Nachkommastellen benutzt werden. Werden mehr Stellen benötigt, so soll der Algorithmus die restlichen Stellen nicht berücksichtigen und abschneiden.

3. Aufgabe (3 Punkte)

Was ist jeweils der Wert von x am Ende der folgenden Programmteile in C++? Warum?

- (a)

```
x=0;
for (i=1; i<=0; i++)
    x=x+1;
```
- (b)

```
x=0;
for (i=1; i<=3; i++)
    x=x+1;
    x=2*x;
```
- (c)

```
x=0;
for (i=1; i<=3; i++);
    x=x+1;
    x=2*x;
```

Hinweis: Der *Inkrementoperator* `++` erhöht den Wert der Variable um 1 (nach Auswertung des Ausdrucks, indem er vorkommt), d.h. `i++` bedeutet `i=i+1`.

4. Aufgabe (3 Punkte)

Welchen Wert hat die Variable `i` am Ende der folgenden Programmteile in C++? Warum?

- (a)

```
int i=0;
double x=2.0;
if (x=1.0)
    i=1;
```
- (b)

```
int i=42;
double x=2.789;
if (x<0.0);
    i=1;
```
- (c)

```
int i=43;
if (1)
    i=1;
```

Programmieraufgabe: Vorführen bis zum 17.1.2007

(a) Implementieren Sie in C++ für die Klasse `bruch`

- wenn Ihre Gruppennummer `coe-1** gerade` ist: die Vergleichsoperatoren `>`, `>=` und `==`.
- wenn Ihre Gruppennummer `coe-1** ungerade` ist: die Vergleichsoperatoren `<`, `<=` und `!=`.
- eine Methode `kuerzen`, die einen Bruch wenn möglich kürzt. Wo sollte diese aufgerufen werden (im Konstruktor, bei jedem Operator usw.)?
- eine Methode, die einen Bruch von der Tastatur einliest. Nutzen Sie dazu den *Inputstream* `cin` aus der *iostream*-Bibliothek. Syntax:

```
#include<iostream>
using namespace std;
...
cin >> variable;
```

(b) Schreiben Sie ein Programm, das zwei Brüche vergleicht und das Ergebnis auf dem Bildschirm ausgibt.

Die Schnittstelle zum Benutzer könnte in Eclipse so aussehen:

