

Praktikum zu
**Einführung in die Informatik für
LogWings und WiMas**
Wintersemester 2011/12

Übungsblatt 7

Bearbeitungszeit:

5.–9.12.2011

Aufgabe 7.1 – Call by value

Was macht das folgende Programm? Tut es das, was der Programmierer anscheinend wollte? Begründen Sie Ihre Antwort!

```

1 class CallByValue {
2     public static void erhoehe(int a) {
3         a = a + 1;
4     }
5
6     public static void main(String [] args) {
7         int x = 15;
8         erhoehe(x);
9         System.out.println("x=" + x);
10    }
11 }
```

Aufgabe 7.2 – „Ganz einfache“ Rekursion

- Schreiben Sie eine rekursive Methode, die die Zahlen 1 bis n in *umgekehrter Reihenfolge* (von n bis 1) ausgibt. n ist ein Parameter der Methode.
- Verändern Sie die Methode so, dass die Zahlen aufsteigend (von 1 bis n) ausgegeben werden.

Aufgabe 7.3 – Rekursion: Potenzen (Eine typische Klausuraufgabe)

Sollten Sie Aufgabe 6.6 noch nicht bearbeitet haben, so holen Sie dies jetzt nach. In der Aufgabe wurden Potenzen mit Hilfe von Schleifen berechnet. Potenzen lassen sich auch rekursiv berechnen. Schreiben Sie eine Methode `hoch`, die folgenden Ansatz verwendet und vergleichen sie mit der Lösung von Aufgabe 6.7.

Um die Potenz b^e auszurechnen, kann man folgende Funktion benutzen ($b \in \mathbb{R}$ und $e \in \mathbb{N}$, mit b Basis und e Exponent):

$$\text{hoch}(b, e) = \begin{cases} 1 & \text{falls } e = 0 \\ (\text{hoch}(b, e/2))^2 & \text{falls } e \text{ gerade} \\ b \cdot \text{hoch}(b, e - 1) & \text{sonst} \end{cases}$$

Machen Sie sich an einigen Beispielen klar, dass diese Funktion tatsächlich funktioniert.

Ignorieren Sie bei Ihrer Implementation vorerst den Fall, dass der Exponent auch negativ sein kann.

Aufgabe 7.4 – Quersumme

- Entwickeln Sie eine Methode `letzteZiffer`, die die letzte Ziffer einer als Parameter übergebenen ganzen Zahl z zurückliefert (im Dezimalsystem). Das Ergebnis ergibt sich als der Rest bei einer Division durch 10.
- Entwickeln und testen Sie eine Methode `quersummeRekursiv`, die *rekursiv* die Quersumme einer Zahl z berechnet. Sie dürfen die Methode `letzteZiffer` als Hilfsmethode benutzen.

Aufgabe 7.5 – Iccanobif

Analysieren Sie folgendes Programm (ohne es auszuprobieren). Was ist die Ausgabe?

```
1 class Iccanobif {
2     public static void main(String [] args) {
3         int m = 10;
4         int[] zahlen = new int[m];
5         zahlen[0] = 0;
6         zahlen[1] = 1;
7         for (int n = 2; n < m; n++) {
8             zahlen[n] = zahlen[n-1] + zahlen[n-2];
9         }
10        arrayAusgeben(zahlen);
11    }
12
13    public static void arrayAusgeben(int [] a) {
14        for (int i = 0; i < a.length; i++) {
15            System.out.println("Wert an Index " + i + " ist " + a[i]);
16        }
17    }
18 }
```

Aufgabe 7.6 – Arrays: Quadratzahlen

Schreiben Sie ein Programm „Quadrate“, das Folgendes tut:

- Es wird eine Variable vom Typ „Feld (Array) von ganzen Zahlen“ angelegt.
- Für das Feld wird ein Speicher von 20 ganzen Zahlen reserviert.
- In jedes Element des Feldes wird (in einer Schleife) das Quadrat seines Index geschrieben, d. h. dem Element mit Index i wird der Wert i^2 zugewiesen.
- In einer weiteren Schleife werden die Elemente des Feldes der Reihe nach ausgegeben.

Ergänzende Aufgaben

Aufgabe 7.7 – Negative Exponenten

Verbessern Sie das Programm aus der Aufgabe 7.3 so, dass auch negative Exponenten zulässig sind.

Aufgabe 7.8 – Eingabeverarbeitung

In dieser Aufgabe soll eine Eingabe unbekannter Länge verarbeitet werden. Die Eingabe besteht aus einer Folge ganzer Zahlen, die durch die Folge $-1, -1, -1$ abgeschlossen wird, welche sonst nicht vorkommt. Ein Beispiel:

1, 2, 4, 0, -1 , 0, -4 , 0, 0, 3, 4, $-1, -1, -1$

Diese Werte sollen wie folgt verarbeitet werden:

- a) Negative Zahlen werden nicht wieder ausgegeben, also aus der Folge „gelöscht“.
- b) In der *verbleibenden* Zahlenfolge wird für jede Gruppe von zwei oder mehr aufeinanderfolgenden Nullen eine einzige Null ausgegeben.
- c) Die übrigen Zahlen werden ausgegeben.

Für das Beispiel wäre die korrekte Ausgabe damit:

1, 2, 4, 0, 3, 4

Die Eingabe soll Zahl für Zahl durch den Benutzer erfolgen, Sie müssen sich also in ihrem Programm jeweils merken, in welchem Zustand bzgl. der obigen Fälle sie sich jeweils zum Eingabezeitpunkt befinden. Um zu erkennen, ob Sie sich gerade in einer Nullfolge befinden, und auch für das Zählen aufeinanderfolgender Eingaben von -1 können Sie zum Beispiel jeweils eine eigene Variable benutzen.