

**Abgabe:** 13.-15. Mai 2008 beim jeweiligen Tutor

## Praktikum Grundlagen der Programmierung

**Themen:** Rekursion und Iteration

### Aufgabe 14 (Ü) Fakultätsfunktion

Die Fakultätsfunktion ist wie folgt definiert:

$$n! = n \cdot (n-1) \cdot \dots \cdot 1, \quad 0! = 1 \quad (n \geq 0)$$

Schreiben Sie ein Programm `fak.java`, das die Fakultätsfunktion auf zwei Varianten berechnet:

- Iterativ in der Funktion `static int fakIt(int n)`
- Rekursiv in der Funktion `static int fakRec(int n)`

### Aufgabe 15 (Ü) Fibonacci-Folge

Die Folge der Fibonacci-Zahlen ist definiert als:

$$n_0 = 0 \quad n_1 = 1 \quad n_i = n_{i-1} + n_{i-2} \quad \text{für } i > 1$$

Schreiben Sie ein Programm, das nach Eingabe des Index  $i$  die Fibonaccizahl  $n_i$  berechnet und ausgibt. Überlegen Sie sich sowohl eine rekursive, als auch eine iterative Lösung. Vergleichen Sie die beiden Lösungen hinsichtlich ihrer Laufzeit.

### Aufgabe 16 (Ü) Quadratwurzel

Das Heron-Verfahren (auch Babylonisches Wurzelziehen genannt) ist ein alter iterativer Algorithmus zur Bestimmung einer rationalen Näherung der Quadratwurzel  $\sqrt{a}$  einer reellen Zahl  $a$ . Die Iterationsvorschrift lautet:

$$x_{n+1} = \frac{1}{2} \cdot \left( x_n + \frac{a}{x_n} \right)$$

Hierbei bezeichnet  $a$  die Zahl, deren Quadratwurzel bestimmt werden soll. Der Startwert  $x_0$  der Iteration kann, solange er nicht gleich Null ist, beliebig festgesetzt werden, wobei zu beachten ist, dass negative Werte gegen die negative Quadratwurzel konvergieren.

Schreiben Sie ein Java-Programm `Wurzel.java`, welches für eine einzulesende Zahl  $a$  die Quadratwurzel  $\sqrt{a}$  näherungsweise, bis auf einen Fehler  $\epsilon$  berechnet und auf dem Bildschirm ausgibt. Geben Sie eine rekursive und eine iterative Lösung an.

**Aufgabe 17 (H) Wiederholung: Arrays**

**(4 Punkte)**

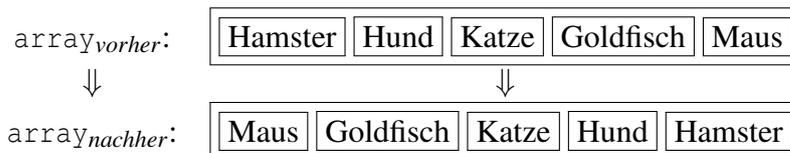
a) **Bestimmung von Minimum und Maximum**

Entwickeln Sie die Funktion `static void printMaxAndMin(int[] array)`, die im Feld `array` sowohl nach der kleinsten, als auch der größten Zahl sucht. Diese beiden Zahlen sollen dann ausgegeben werden.

b) **Umkehrung von Feldern**

Entwickeln Sie die Funktion `static void reverse(String[] array)`, die die Reihenfolge der Wörter im Feld `array` umdreht.

**Beispiel:**



**Aufgabe 18 (H) Quersumme**

**(4 Punkte)**

Schreiben sie ein Programm `Quersumme.java`, das die Quersumme einer von der Konsole einzulesenden Zahl

- a) rekursiv
- b) iterativ

berechnet und anschließend ausgibt.

**Hinweis:** Verwenden Sie den Modulo-Operator.

**Aufgabe 19 (H) Sechs Richtige im Lotto**

**(6 Punkte)**

Der Binomialkoeffizient  $\binom{n}{k}$  gibt an, auf wieviele verschiedene Arten man  $k$  Objekte aus einer Menge von  $n$  verschiedenen Objekten auswählen kann (ohne Zurücklegen und ohne Beachtung der Reihenfolge).

$$\text{Für } 0 \leq k \leq n \text{ gilt: } \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}; \quad \binom{0}{0} = 1; \quad \binom{n}{n} = \binom{n}{0} = 1$$

Schreiben Sie ein Programm `Binom.java`.

- a) Es soll eine Funktion `static long binom(long n, long k)` enthalten, welche  $\binom{n}{k}$  rekursiv berechnet und das Ergebnis zurückgibt.
- b) Erweitern Sie Ihr Programm um eine Funktion `static double sechsRichtige()`, welche die *Wahrscheinlichkeit* zurückgibt, 6 Zahlen aus der Menge  $\{1...49\}$  richtig zu tippen.