

Java Übungsblatt 1

Am Besten programmieren Sie eine Methode mit den erforderlichen Parametern und testen sie mit entsprechenden Anweisungen in der `main`-Methode. Das *Einlesen* von Werten von der Befehlszeile (=Konsole, Terminal) muss nicht implementiert werden (darf aber).

1 Integer, einfache Programme

Aufgabe 1 (M2.9)

Abstand zwischen Punkten. Schreiben Sie ein Java-Programm, das die x- und y-Koordinaten zweier Punkte einliest und den Abstand zwischen ihnen berechnet und ausgibt. Hinweis: Die ganzzahlige Wurzel einer Zahl x können Sie als `(int)Math.sqrt(x)` berechnen.

* Aufgabe 2 (M2.4)

Ausdrücke. Schreiben Sie einen Java-Ausdruck, der eine Zahl x auf das nächstliegende Vielfache von 100 rundet. Der Wert 149 soll also auf 100 abgerundet und der Wert 150 auf 200 aufgerundet werden.

* Aufgabe 3 (M2.7)

Zeitrechnung. Schreiben Sie ein Java-Programm, das eine Anzahl von Sekunden einliest und in die Anzahl von Stunden, Minuten und Sekunden umrechnet und ausgibt. Die Eingabe 1234 soll also zur Ausgabe von 0:20:34

2 Verzweigungen

Aufgabe 1 (M3.1)

Verzweigungen. Schreiben Sie ein Java-Programm, das drei Werte x , y und z einliest und prüft,

- ☐ ob x , y und z nicht lauter gleiche Werte enthalten,
- ☐ ob x , y und z lauter verschiedene Werte enthalten,
- ☐ ob mindestens zwei Werte gleich sind.

* Aufgabe 2 (M3.2)

Dreiecksbestimmung. Schreiben Sie ein Java-Programm, das die Seitenlängen eines Dreiecks einliest und prüft, ob es ein gleichseitiges, ein gleichschenkeliges, ein rechtwinkeliges, ein sonstiges gültiges oder ein ungültiges Dreieck ist. Ein Dreieck ist ungültig, wenn die Summe zweier Seitenlängen kleiner oder gleich der dritten Seitenlänge ist. Beachten Sie, dass ein Dreieck sowohl rechtwinkelig als auch gleichschenkelig sein kann.

Aufgabe 3 (M3.7)

Sortieren dreier Zahlen. Schreiben Sie ein Programm, das drei Zahlen a, b und c einliest und sie in sortierter Reihenfolge wieder ausgibt. Kommentieren Sie dabei alle if- und else-Zweige durch Assertionen.

*** Aufgabe 4 (M3.10)**

Überlaufsprüfung. Lesen Sie zwei int-Zahlen a und b ein und prüfen Sie, ob bei ihrer Addition ein Überlauf stattfindet, also eine Summe entstehen würde, die größer als $2^{31}-1$ oder kleiner als -2^{31} ist.

*** Aufgabe 5 (M3.11)**

Schnitt zweier Linien. Lesen Sie die Endpunkte zweier horizontaler oder vertikaler Linien in Form ihrer x- und y-Koordinaten ein und prüfen Sie, ob sich die beiden Linien schneiden.

3 Schleifen

Tip: arbeiten Sie mit dem Divisions- und Modulus-Operator (/ , %).

Aufgabe 1 (M4.2)

Anzahl der Ziffern einer Zahl. Schreiben Sie ein Programm, das eine positive ganze Zahl n einliest und feststellt, aus wie vielen Ziffern sie besteht.

Aufgabe 2 (M4.3)

Ziffernsumme einer Zahl. Schreiben Sie ein Programm, das die Ziffernsumme einer Zahl x berechnet und ausgibt. Für 4711 beträgt die Ziffernsumme zum Beispiel 13.

*** Aufgabe 3 (M4.8)**

Codevereinfachungen. Vereinfachen Sie folgende Codestücke:

```
i = 0; // assume j >= 0
while (i != j) i++;

while (a < b) {
    c = a;
    a = b;
    b = c;
}
```

*** Aufgabe 4 (M4.7)**

Primfaktorenzerlegung. Schreiben Sie ein Programm, das eine positive ganze Zahl n einliest und in ihre Primfaktoren zerlegt. Die Zahl 100 besteht zum Beispiel aus den Primfaktoren 2, 2, 5, 5; die Zahl 252 aus den Primfaktoren 2, 2, 3, 3, 7.

4 * Rekursion, Strings

* Aufgabe 5 (M16.1)

Fibonacci-Zahlen. Die Fibonacci-Zahlen spielen bei Wachstumsvorgängen in der Natur eine wichtige Rolle. Sie sind durch folgende Rekursionsformel definiert:

$$\begin{aligned}\text{fib}(0) &= 1 \\ \text{fib}(1) &= 1 \\ \text{fib}(n) &= \text{fib}(n-2) + \text{fib}(n-1) \quad \text{für } n > 1\end{aligned}$$

Schreiben Sie ein Java-Programm, das die Fibonacci-Zahl $\text{fib}(n)$ für ein gegebenes n rekursiv sowie iterativ berechnet. Versuchen Sie, die Laufzeit der beiden Lösungen in Abhängigkeit von n zu schätzen.

Tip: Recherchieren Sie Möglichkeiten zur Zeitstoppung im Internet.

* Aufgabe 6 (M16.4)

Binomialkoeffizient. Implementieren Sie je eine Methode zur rekursiven und zur iterativen Berechnung des Binomialkoeffizienten.

Rekursive Definition

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

$$\binom{n}{0} = 1$$

$$\binom{n}{k} = 0, \text{ für } k > n$$

Iterative Definition

$$\binom{n}{k} = \frac{n!}{k! (n-k)!}$$

* Aufgabe 7 (M16.3)

Spiegeln einer Zeichenfolge. Schreiben Sie ein Programm, das eine Zeichenfolge liest und in umgekehrter Reihenfolge wieder ausgibt. Verwenden Sie kein Array zur Zwischenspeicherung der Zeichenfolge, sondern implementieren Sie die Lösung rekursiv.

Tip: Recherchieren Sie geeignete Methoden der Datentypen `String`, `char` und `StringBuilder`.

* Aufgabe 8

Studieren Sie Quelltext und Dokumentation

(<https://docs.oracle.com/en/java/javase/17/docs/api/index.html>) der Klassen `String` und `StringBuilder` und erklären Sie möglichst genau, was in der Memory passiert beim Erstellen von Strings und StringBuilders und wieso es effizienter sein kann, mit StringBuildern zu arbeiten als mit Strings, wenn diese laufend verändert werden müssen (wie z.B. in Aufgabe 7).