

## zu Kapitel 4

### Aufgabe 1

(Boolean.java)

Geben Sie die Ausgabe untenstehenden Programms an. Überlegen Sie erst und schreiben Sie die Ergebnisse Ihrer Überlegungen auf. Bringen Sie danach dieses und dann ein äquivalentes Programm, das aber durch eine sinnvolle Klammersetzung lesbarer geworden ist, zum Laufen und vergleichen Sie mit Ihren Überlegungen.

```
/* Boolean.java */

public class Boolean
{
    public static void main (String[] args)
    {
        int m = 1, n = -1;
        boolean a = true, b = false;

        System.out.println(a && !b);
        System.out.println(!a || b && a || !b);
        System.out.println(!a || !b || m < n);
        System.out.println(m > n || m == n && m > n );
    }
}
```

### Aufgabe 2

(Schaltjahr.java)

Schreiben Sie ein Programm, das für eine Jahreszahl ausgibt, ob es sich um ein Schaltjahr handelt oder nicht.

Hinweis:

Jahreszahlen, die durch 4 teilbar sind, sind Schaltjahre (z.B. 2004). Das gilt jedoch nicht, wenn die Jahre durch 100 teilbar sind (z.B. war 1900 kein Schaltjahr). Sind die Jahre jedoch zusätzlich durch 400 teilbar, handelt es sich wieder um Schaltjahre (z.B. 2000).

### Aufgabe 3

(Sort.java)

Schreiben Sie ein Programm, das drei beliebige Zahlen der Größe nach sortiert ausgibt. Testen nicht vergessen!

### Aufgabe 4

(QuadGleichung.java)

Schreiben Sie ein Programm, das für drei beliebige ganze Zahlen a, b, c, die Lösung(en) der Gleichung  $ax^2 + bx + c = 0$  berechnet. x sei dabei vom Typ float. Es sollen alle möglichen Fälle behandelt werden.

Testen Sie das Programm mit folgenden Werten (zur Überprüfung sind die zu erwartenden Lösungen mit angegeben):

(1, -8, 15) → zwei Lösungen: 3.0 und 5.0  
(9, 6, 1) → eine Lösung: -0,3333...  
(3, 2, 7) → keine reellen Lösungen  
(0, 0, 0) → unendlich viele Lösungen  
(0, 3, -2) → eine Lösung: 0.6666...  
(0, 0, 5) → keine Lösungen

Hinweis:

Die Wurzel berechnet sich mittels Math.sqrt(). Beispiel:

```
x = Math.sqrt(y);
```

Danach steht in x die Wurzel von y.

## **Aufgabe 5**

(Kalenderwoche.java)

Schreiben Sie ein Programm, das für ein gegebenes Datum bestehend aus Tag und Monat ausrechnet, in der wievielten Woche des Jahres das Datum liegt. (Nehmen Sie der Einfachheit halber an, dass es sich um kein Schaltjahr handelt und das Jahr mit dem Tag anfängt, mit dem auch die Woche beginnt. Es liegt also genau eine komplette Woche zu Beginn des Jahres.)

## **Aufgabe 6**

(Wurzelberechnung.java)

Berechnen Sie die Wurzel einer vorgegeben Zahl auf 10 Stellen genau (und zwar nicht unter Verwendung von Math.sqrt() sondern nur unter Verwendung der vier Grundrechenarten).

Hinweis:

Der Heron-Algorithmus funktioniert wie folgt:

Es muss gelten  $x^2 = (\sqrt{a})^2 = a$ .

Oder geometrisch: wir haben  $\sqrt{a}$  gefunden, wenn wir ein Quadrat mit Flächeninhalt a bilden können. Das können wir aber nicht, deshalb bilden wir näherungsweise ein Rechteck mit Flächeninhalt a:

$$x_0 = a, x_0' = 1$$

Es ist offensichtlich, dass eine Kante zu lang und eine Kante zu kurz ist. Der Mittelwert der beiden Kanten dürfte also näher an dem gesuchten Wert x liegen. Wir bilden mit diesem Mittel ein neues Rechteck mit Flächeninhalt a:

$$x_1 = (x_0 + 1)/2, x_1' = a/x_1$$

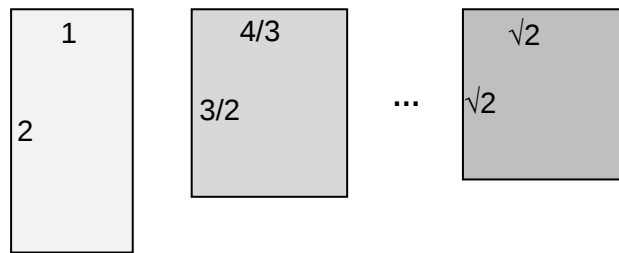
Das jetzige Rechteck kommt unserem gesuchten Quadrat schon näher, trotzdem ist wieder eine Seite zu lang und eine zu kurz. Damit dürfte der Mittelwert der beiden Seiten ein besserer Näherungswert für unser gesuchtes Quadrat sein. Wir erhalten:

$$x_2 = (x_1 + a/x_1)/2, x_2' = a/x_2$$

Ab jetzt sind die weiteren Schritte klar: Unser gesuchtes Quadrat wird sukzessive durch immer bessere Rechtecke angenähert. Für jeden weiteren Schritt wird dabei die Kantenlänge einer Kante als Mittelwert der vorherigen Kanten bestimmt. Allgemein erhalten wir:

$$x_n = (x_{n-1} + a/x_{n-1})/2$$

Wenn die gewünschte Genauigkeit erreicht ist, kann die Berechnung beendet werden.



Anschauliche Darstellung für  $a=2$

## Aufgabe 7

(Tabelle.java)

Schreiben Sie ein Programm, welches das große Einmaleins in Tabellenform ausgibt. Halten Sie sich dabei an folgendes Format der Ausgabe:

```
10:
---
1 x 10 = 10
2 x 10 = 20
:
10 x 10 = 100
=====
```

```
11:
---
1 x 11 = 11
2 x 11 = 22
:
10 x 11 = 110
=====
```

```
12:
---
: usw.
```

## Aufgabe 8

(Wochentag.java)

Eine weitere Funktion eines Kalenders ist die Angabe des Wochentags zu einem gegebenen Datum. In elektronischen Kalendern wird i.d.R. der Wochentag aus einem bestimmten Anfangsdatum, ab dem der Kalender gilt, berechnet.

Berechnen Sie für ein beliebiges Datum bestehend aus Tag, Monat und Jahr den Wochentag (also Montag, Dienstag, usw.). Verwenden Sie hierbei, dass der 1.1.1980 ein Dienstag war.

Hinweis:

Teile früherer Übungsaufgaben dürfen dabei durchaus verwendet werden!

