

zu Kapitel 6

Aufgabe 1

(Kalenderwoche2.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.)

Anders als in einer vorherigen Übungsaufgabe sollen Sie die Lösung aber ohne switch-Anweisung, dafür aber mit einem 12-elementigen Array, das Sie mit der Anzahl der Tage jeweils eines Monat initialisiert haben, lösen:

```
int[] arr = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

Aufgabe 2

(Temperatur.java)

Schreiben Sie ein Programm, das für eine Woche jeweils die Tageshöchsttemperatur einliest (also 7 Temperaturwerte) und die mittlere Tageshöchsttemperatur sowie die Wochenhöchsttemperatur ausgibt.

Aufgabe 3

(Bubbelsort.java)

- a) Schreiben Sie eine Funktion sort(), welche ein übergebenes Integer-Array der Größe nach sortiert.
- b) Testen Sie Ihre Funktion mit einer geeigneten main()-Funktion, in der Sie ein beliebiges (unsortiertes) Integer-Array anlegen, damit die Funktion sort() aufrufen und danach das sortierte Array zur Kontrolle ausgeben.

Hinweis: Kurzbeschreibung eines Sortieralgorithmus (findet man in der Literatur unter dem Namen Bubblesort):

Sortieren Sie immer paarweise über alle Elemente eines Arrays mit n Elementen (also erst array[0] und array[1], dann array[1] und array[2] usw. bis array[n-2] und array[n-1]). Stellen Sie fest, ob dabei tatsächlich (mindestens) ein Vertauschen stattgefunden hat. Wenn ja, dann fangen Sie wieder von vorne an. Das machen Sie so oft, bis Sie das erste Mal beim paarweisen Sortieren über das ganze Array keinen einzigen Tausch mehr vornehmen mussten. Dann ist das Array sortiert.

Aufgabe 4

(Pascal.java)

Das Pascal'sche Dreieck ist so aufgebaut, dass die Elemente in einer Reihe immer genau die Summe der direkt darüber stehenden Elemente sind. Die Ränder haben den Wert 1, also:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

usw.

Erstellen Sie ein Programm, dass ein Pascal'sches Dreieck mit fest vorgegeben Tiefe (z.B. ist die Tiefe obigen Beispiels 6) berechnet und ausgibt.

Hinweis: Die Ausgabe kann einfach zeilenweise erfolgen. (Alles andere ist dann doch zu trickreich.)

Ausgabe:

```
1
1 1
1 2 1
1 3 3 1
usw.
```