



Technische Universität Darmstadt
 Fachbereich Informatik
 Prof. Dr. Johannes Fürnkranz

Allgemeine Informatik 1 im WS 2006/07

Übungsblatt 12

Bearbeitungszeit: 29.01. bis 04.02.2007

Aufgabe 1: for-Schleife

Neben der **while**-Schleife gibt es in Java noch die **for**-Schleife. Sie hat folgende Form:

```
for (Initialisierung; Bedingung; Weiterschaltung) {
    // ...
}
```

Diese **for**-Schleife:

```
for (int i = 0; i < 5; i++) { // Init, Bedingung, Weiterschaltung
    System.out.println(i);   // Anweisung
}
```

hat den gleichen Effekt wie diese **while**-Schleife:

```
int i = 0; // Initialisierung
while (i < 5) { // Bedingung
    System.out.println(i); // Anweisung
    i++; // Weiterschaltung
}
```

Mit beiden Schleifen werden untereinander die Zahlen 0, 1, 2, 3 und 4 auf dem Bildschirm ausgegeben.

Eine **for**-Schleife lässt sich also immer in eine gleichbedeutende **while**-Schleife übersetzen, ist allerdings etwas kompakter zu schreiben.

Erzeugen Sie in einem **task** einen **Robot karel** bei (1, 1) mit 999 Beepern und Blickrichtung Osten. **karel** bleibt in der ersten Street und soll mittels einer **for**-Schleife in der ersten Avenue zwei Beeper legen, in der zweiten Avenue vier, in der dritten Avenue sechs und so weiter, bis zu 18 Beepern in der neunten Avenue. Sie können zum eigentlichen Ablegen der Beeper innerhalb der **for**-schleife eine **loop**-Schleife oder gerne auch eine zweite **for**-Schleife verwenden.

Aufgabe 2: Arrays

Arrays (Felder) sind vergleichbar mit mathematischen Vektoren. Sie bieten die Möglichkeit, mehrere Variablen unter einem gemeinsamen Namen in Verbindung mit einem Index anzusprechen.

Ist z.B. mathematisch $a = \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix}$, so ist $a_1 = 2$, $a_2 = 3$ und $a_3 = 5$.

In Java fängt der Index allerdings nicht bei 1 an, sondern bei 0, und er steht in eckigen Klammern hinter dem Namen des Arrays: `a[0] = 2` weist dem ersten Array-Element den Wert 2 zu.

Die Deklaration eines Arrays sieht ähnlich aus wie die einer normalen Variablen:

```
datentyp[] arrayname;
```

Also wird ein Array `a`, das `int`-Werte aufnimmt, mit `int[] a;` deklariert, und ein Array, das Roboter beinhaltet, z.B. mit `Robot[] roboter;`.

Die Initialisierung eines Arrays weist Parallelen zur Erzeugung eines Roboters auf: sie geschieht mit `new`. Dabei wird in die eckigen Klammern die Anzahl der Elemente des Arrays geschrieben:

`a = new int[3];` erzeugt ein Array mit den drei Elementen `a[0]`, `a[1]` und `a[2]`. Diese Elemente sind allerdings noch nicht initialisiert und müssen wie jede andere Variable auch erst initialisiert werden. Mit `a[0] = 2; a[1] = 3; a[2] = 5;` entsteht z.B. ein Array, das dem Vektor aus dem obigen Beispiel entspricht.

Mit dieser `for`-Schleife können wir die Elemente des Arrays `a` untereinander auf dem Bildschirm ausgeben:

```
for (int i = 0; i < 3; i++) {  
    System.out.println(a[i]);  
}
```

Da wir nicht immer wissen, wie viele Elemente das Array hat, benutzen wir im Allgemeinen in der Bedingung der `for`-Schleife nicht einen festen Wert, sondern `arrayname.length` (im obigen Beispiel `a.length`) – diese Variable enthält immer die Anzahl der Elemente des Arrays. Die Elemente eines Arrays `a` sind also von 0 bis `a.length - 1` nummeriert.

Erweitern Sie nun die auf der Webseite vorgegebene Datei `uebung12-2.task`. In ihr wird ein Array `a` einer zufälligen Länge erzeugt und mit zufälligen Werten gefüllt.

Benutzen Sie im markierten Bereich eine einzige `for`-Schleife, um folgendes zu tun:

- die einzelnen Arrayelemente untereinander auf dem Bildschirm auszugeben
- die Summe der Werte in der Variablen `summe` zu berechnen
- die Position des maximalen Wertes zu finden und in der Variablen `maxi` zu speichern (dazu müssen Sie den aktuellen Wert mit dem an der Position des bisher maximalen Wertes vergleichen).

Sie helfen sich, wenn Sie zuerst versuchen, die Schleife auf dem Papier zu entwerfen und nachzuvollziehen – in der Klausur haben Sie auch keinen Rechner!

Die Ausgabe der errechneten Werte auf dem Bildschirm ist schon in der Vorgabe enthalten.

Aufgabe 3: Variablen oder Arrayelemente vertauschen

In vielen Fällen ist es nötig, den Inhalt von Variablen vertauschen zu können.

Angenommen, die Variablen `a` und `b` wurden folgendermaßen initialisiert:

```
int a = 42;  
int b = 23;
```

Das Vertauschen funktioniert dann nicht einfach mittels

```
a = b;  
b = a;
```

Dann hätten nämlich beide Variablen den Wert 23. Sie benötigen eine dritte Variable als Hilfe.

Angenommen, wir wollen in einem Array die Elemente `a[i]` und `a[j]` vertauschen, wenn `a[i]` kleiner als `a[j]` ist. Wie sieht der entsprechende Code aus?

```
if (a[i] < a[j]) {  
    _____  
    _____  
    _____  
}
```

Aufgabe 4: Sortieren mit SelectionSort

Gegeben ist eine Datei `uebung12-4.task` (auf unserer Webseite, Adresse siehe oben), die Sie erweitern sollen (**also keine eigene Datei anfangen!**).

Diese Aufgabe verbindet die beiden letzten Aufgaben zu einer sinnvollen Anwendung: einem Sortieralgorithmus. Den Algorithmus **BubbleSort** haben Sie im Programmierprojekt mit Robotern kennen gelernt, in dieser Aufgabe geht es um den etwas effizienteren Algorithmus **SelectionSort**. Die Informatik kennt noch viele weitere, oft bedeutend effizientere (aber auch kompliziertere) Sortieralgorithmen – auf www.sortieralgorithmen.de können Sie mehr darüber erfahren.

In der vorgegebenen Datei wird wieder ein Array `a` zufällig erzeugt und auf dem Bildschirm ausgegeben. Ihre Aufgabe ist es nun, dieses Array aufsteigend zu sortieren. Aufsteigend bedeutet, dass der kleinste Wert des Arrays nach dem Ordnen in `a[0]` steht und der größte in `a[a.length - 1]`.

- a) Implementieren Sie die Methode `print`. Diese soll das Array in einer Zeile und nicht untereinander ausgeben. Dazu geben Sie in einer `for`-Schleife jedes Element mit `System.out.print()` aus, gefolgt von einem Leerzeichen. Nach der Schleife rufen Sie noch einmal `System.out.println()`; auf, um in die nächste Zeile zu springen.

Unser Sortieralgorithmus funktioniert folgendermaßen:

1. Am Anfang ist das Array komplett von Index **0** bis Index **a.length - 1** unsortiert.
2. Wir suchen das größte Element und vertauschen es mit dem letzten Element. Dadurch geht der unsortierte Teil nur noch von **0** bis **a.length - 2**, das letzte Element ist jetzt sortiert.
3. Nun suchen wir im unsortierten Teil das größte Element und vertauschen es mit dem letzten Element im unsortierten Teil des Arrays. Der unsortierte Teil geht nur noch bis **a.length - 3**, die letzten beiden Elemente sind jetzt sortiert.
4. Da in jedem Schritt der sortierte Teil des Arrays um eins größer wird, reichen **a.length - 1** Schritte, um das Array komplett zu sortieren.

<i>Index</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Anfang	42	23	<u>66</u>	7	11	<u>13</u>
<i>nach Schritt 1</i>	<u>42</u>	23	13	7	<u>11</u>	66
<i>nach Schritt 2</i>	11	<u>23</u>	13	<u>7</u>	42	66
<i>nach Schritt 3</i>	11	7	<u>13</u>	23	42	66
<i>nach Schritt 4</i>	<u>11</u>	<u>7</u>	13	23	42	66
<i>nach Schritt 5</i>	7	11	13	23	42	66

weiß: unsortiert

grau: sortiert

unterstrichen: diese Elemente werden vertauscht

- b) Implementieren Sie nun die Methode **sort**. Sie benötigen dazu zwei verschachtelte **for**-Schleifen. Der Index der äußeren Schleife entspricht dem Ende des unsortierten Teils des Arrays und fällt von **a.length - 1** auf **1**. In dieser äußeren **for**-Schleife benutzen Sie wie in Aufgabe 2 eine weitere **for**-Schleife, um den Index des größten Elements zu finden und vertauschen dieses dann wie in Aufgabe 3 mit dem letzten Element des ungeordneten Teils.

Aufgabe 5: Osterdatum

Bis Ostern dauert es zwar noch, aber wie lange genau dauert es eigentlich noch? Und wann ist nächstes oder übernächstes Jahr Ostern? Das lässt sich herausfinden...

Implementieren Sie in der vorgegebenen Datei **uebung12-5.task** den Algorithmus, der zu einem gegebenen Jahr das Datum des Ostersonntags im Format „Ostersonntag: Tag.Monat.Jahr“ auf dem Bildschirm ausgibt.

Den Algorithmus sowie weitere Informationen und Links finden Sie im Internet auf <http://de.wikipedia.org/w/index.php?title=Osterdatum&oldid=25787488>.

Die beiden Zeilen für den Julianischen Kalender müssen Sie natürlich weglassen. Achten Sie darauf, dass alle Zahlen vom Typ **int** sind. Der Modulo-Operator wird in Java durch das Prozentzeichen dargestellt: „a mod b“ entspricht in Java **a % b**.

Viel Spaß (und frohe Ostern...)!