

Praktikum zu
**Einführung in die Informatik für
 LogWings und WiMas**
 Wintersemester 2011/12

Übungsblatt 8
 Bearbeitungszeit:
 12.–16.12.2011

Aufgabe 8.1 – Aufgaben von Blatt 7

Bitte lösen Sie zunächst die regulären Aufgaben von Blatt 7, die Sie noch nicht bearbeitet haben.

Aufgabe 8.2 – Speicherverwaltung für Felder

In Kapitel 5 und auf Übungsblatt 7 wurden Felder (*Arrays*) eingeführt. Betrachten Sie das untenstehende Programmfragment. Es deklariert und initialisiert eine ganzzahlige Variable *n* und drei Felder von ganzen Zahlen *a*, *b* und *c*.

Gehen Sie den Programmtext nun Zeile für Zeile durch und tragen Sie in das nebenstehende Diagramm ein, wie die einzelnen Speicherzellen im Laufe des Programms belegt und geändert werden.

<pre> 1 int[] a; 2 a = new int[4]; 3 for (int i = 0; i < a.length; i++) { 4 a[i] = i; 5 } 6 int n = a.length - 1; 7 int[] b; 8 b = new int[n]; 9 for (int i = 0; i < b.length; i++) { 10 b[i] = a[i] + 10; 11 } 12 int[] c = new int[b.length]; 13 c = b; </pre>	<div style="border: 1px solid black; width: 40px; height: 40px; margin-bottom: 5px; display: flex; align-items: center; justify-content: center;">n</div> <div style="border: 1px solid black; width: 40px; height: 40px; margin-bottom: 5px; display: flex; align-items: center; justify-content: center;">a</div> <div style="border: 1px solid black; width: 40px; height: 40px; margin-bottom: 5px; display: flex; align-items: center; justify-content: center;">b</div> <div style="border: 1px solid black; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center;">c</div>	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 30px; height: 30px;">[0]</td> <td style="width: 30px; height: 30px;">[1]</td> <td style="width: 30px; height: 30px;">[2]</td> <td style="width: 30px; height: 30px;">[3]</td> </tr> <tr> <td style="width: 30px; height: 30px;">[0]</td> <td style="width: 30px; height: 30px;">[1]</td> <td style="width: 30px; height: 30px;">[2]</td> <td></td> </tr> <tr> <td style="width: 30px; height: 30px;">[0]</td> <td style="width: 30px; height: 30px;">[1]</td> <td style="width: 30px; height: 30px;">[2]</td> <td></td> </tr> </table>	[0]	[1]	[2]	[3]	[0]	[1]	[2]		[0]	[1]	[2]	
[0]	[1]	[2]	[3]											
[0]	[1]	[2]												
[0]	[1]	[2]												

Aufgabe 8.3 – Felder als Parameter

Betrachten Sie noch einmal Aufgabe 7.1; die ganzzahlige Variable *x* wurde dort an die Methode *erhoehe* übergeben. Da Parameter in Java immer per Wertübergabe (*call by value*) übergeben werden, können wir bei primitiven Datentypen wie *int* auch lediglich auf einer Kopie des Wertes einer Variable arbeiten. Bei Feldern ist dies anders: Sie haben in der Vorlesung und in Aufgabe 8.2 gesehen, dass sie mittels Referenzen gespeichert werden. Feldvariablen verweisen also auf Felder, und dieser Verweis wird als Kopie übergeben. Was macht also das folgende Programm? Begründen Sie Ihre Antwort!

```

1 class ArrayParameter {
2     static void erhoehe(int[] a) {
3         for(int i = 0; i < a.length; i++) {
4             a[i] = a[i] + 1;
5         }
6     }
7
8     public static void main(String[] args) {
9         int[] y = new int[1];
10        y[0] = 15;
11        erhoehe(y);
12        System.out.println("Eintrag im Feld y: " + y[0]);
13    }
14 }
                
```

Aufgabe 8.4 – Sortieren eines Ganzzahl-Feldes: selbstprogrammiert

- a) Auf den Folien 17 und 18 von Kapitel 5 wurde ein einfacher Sortieralgorithmus skizziert. Programmieren Sie diesen Algorithmus in einer Java-Methode zur Sortierung eines Ganzzahl-Feldes. Diese Methode soll keinen Rückgabewert und nur ein Ganzzahl-Feld als Parameter haben. Rufen Sie diese Methode dann in der `main`-Methode auf, um das Feld

```
int [] a = {5, -15, 3, 5, 6, 9, -4, 10, 9, 9};
```

zu sortieren. Geben Sie das sortierte Feld danach mit der Ausgabe-Methode aus Aufgabe 7.5 aus.

- b) DrJava stellt einen *Debugger* zur Verfügung. Dabei handelt es sich um ein Werkzeug, mit dem Sie das Verhalten eines Programms zur Laufzeit analysieren können. Insbesondere können Sie sich den Wert von Variablen während des Programmlaufs anzeigen lassen, um so zum Beispiel Fehler zu erkennen. Den Debugger können Sie einschalten, indem Sie im Menü „*Debugger*“ die Option „*Debug Mode*“ auswählen. Mit der Option „*Toggle Breakpoint on Current Line*“ können Sie dann die Stelle im Programm markieren, an der Sie die Diagnose beginnen möchten. Im Debug-Fenster „*Watches*“ können Sie in der „*Name*“-Spalte dann den Namen der zu inspizierenden Variable eintragen. Danach können Sie mithilfe der verschiedenen „*Step*“-Befehle im „*Debugger*“-Menü in Einzelschritten den Ablauf des Programms verfolgen.
- Verwenden Sie den Debugger, um den Ablauf des Programms aus Aufgabe a) im Einzelschritt nachzuvollziehen.
 - Verwenden Sie den Debugger, um den Ablauf des Programms aus Aufgabe 7.2 im Einzelschritt nachzuvollziehen.

Ergänzende Aufgaben

Aufgabe 8.5 – Sortieren eines Ganzzahl-Feldes: mit einer API-Methode

In der Java-API (siehe dazu auch Aufgabe 3.2) gibt es bereits einen Algorithmus zur Sortierung eines Feldes. Suchen zunächst in der „Java-API-Dokumentation“ nach dieser Methode: Öffnen Sie die Webseite <http://docs.oracle.com/javase/7/docs/api/> im Browser. Suchen Sie dann im linken Menü unter „All Classes“ nach dem Eintrag „Arrays“ und klicken Sie darauf; im Hauptfenster öffnet sich daraufhin die Beschreibung der Klasse Arrays. Suchen Sie dann unter „Method Summary“ nach einer Methode, die ein Ganzzahl-Feld sortiert; Sie können auch auf den Methodennamen klicken, um sich eine genauere Beschreibung der Methode anzusehen.

Rufen Sie nun in Ihrer `main`-Methode aus Aufgabe 8.4 die API-Methode auf, um das Feld zu sortieren. Beachten Sie,

- dass Sie eine `import`-Anweisung benötigen, damit die Java-VM die Klasse Arrays findet:

```
import java.util.Arrays;
```

(Diese Anweisung kennen Sie schon von der Benutzung der Klasse Scanner).

- dass Sie die API-Methode mit dem vorangestellten Klassennamen Arrays aufrufen müssen, damit die Java-VM die Methode findet:

```
Arrays.sort(a);
```

Aufgabe 8.6 – Mein Wunschzettel

Weihnachten steht nun kurz vor der Tür, aber Sie waren so mit Ihrem Studium beschäftigt, dass Sie keine Zeit hatten sich über Ihre Wünsche Gedanken zu machen. Nun fragen aber Ihre lieben Verwandten, insbesondere Ihre Eltern, beinahe täglich nach, was sie Ihnen denn schenken könnten. Es wird also höchste Zeit für einen Wunschzettel.

Um das Lernen dabei nicht aus den Augen zu verlieren beschließen Sie, Ihr frisch erworbenes Wissen über Felder anzuwenden, um Ihre Wünsche aufzulisten.

- Erstellen Sie ein Feld von Zeichenketten, in das Sie Ihre Wünsche eintragen werden; es soll am Anfang die Länge 0 haben.
- Programmieren Sie eine Methode, mit der Sie einen neuen Wunsch in Ihr Feld eintragen können. Kopieren Sie dazu die alten Wünsche in ein um 1 längeres Feld und tragen Sie den neuen Wunsch zusätzlich ein. Die Methode soll das neue Feld dann zurückgeben. (Tipp: Statt das Feld von Hand zu kopieren können Sie auch eine Kopier-Methode aus der Klasse Arrays verwenden.)
- Benutzen Sie innerhalb der `main`-Methode diese Methode, um Ihren Wunschzettel zu füllen. Geben Sie den Wunschzettel dann aus. (Tipp: Bei der Ausgabe kann Ihnen eine Methode der Klasse Arrays helfen.)
- Verwenden Sie eine Methode aus der Klasse Arrays, um Ihre Wünsche alphabetisch zu sortieren. Geben Sie den Wunschzettel dann erneut aus.
- Anstatt die Wünsche fest in der `main`-Methode einzugeben, können Sie sie auch mit der Ihnen bekannten Klasse Scanner einlesen: Benutzen Sie die Methode `nextLine()` von Scanner, um einen Wunsch einzulesen und fügen Sie diesen Ihrem Wunschzettel hinzu. Benutzen Sie eine `do-while`-Schleife wie in Aufgabe 4.6, um mehrere Wünsche einzulesen.