

Praktikum zu  
**Einführung in die Informatik für  
 LogWings und WiMas**  
 Wintersemester 2013/14

**Übungsblatt 8**

Bearbeitungszeit:

6.–10.1.2014

**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 &lt; 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 &lt; b.length; i++) { 10    b[i] = a[i] + 10; 11  } 12  int[] c = new int[b.length]; 13  c = b; </pre>	<table border="1" style="border-collapse: collapse; width: 40px; height: 40px; text-align: center;"> <tr><td>n</td></tr> </table>	n	<table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 40px; height: 40px;"></td> <td style="width: 40px; height: 40px;"></td> <td style="width: 40px; height: 40px;"></td> <td style="width: 40px; height: 40px;"></td> </tr> </table>				
n							
	<table border="1" style="border-collapse: collapse; width: 40px; height: 40px; text-align: center;"> <tr><td>a</td></tr> </table>	a	<table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 40px; height: 40px; text-align: center;">[0]</td> <td style="width: 40px; height: 40px; text-align: center;">[1]</td> <td style="width: 40px; height: 40px; text-align: center;">[2]</td> <td style="width: 40px; height: 40px; text-align: center;">[3]</td> </tr> </table>	[0]	[1]	[2]	[3]
a							
[0]	[1]	[2]	[3]				
	<table border="1" style="border-collapse: collapse; width: 40px; height: 40px; text-align: center;"> <tr><td>b</td></tr> </table>	b	<table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 40px; height: 40px; text-align: center;">[0]</td> <td style="width: 40px; height: 40px; text-align: center;">[1]</td> <td style="width: 40px; height: 40px; text-align: center;">[2]</td> </tr> </table>	[0]	[1]	[2]	
b							
[0]	[1]	[2]					
	<table border="1" style="border-collapse: collapse; width: 40px; height: 40px; text-align: center;"> <tr><td>c</td></tr> </table>	c	<table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 40px; height: 40px; text-align: center;">[0]</td> <td style="width: 40px; height: 40px; text-align: center;">[1]</td> <td style="width: 40px; height: 40px; text-align: center;">[2]</td> </tr> </table>	[0]	[1]	[2]	
c							
[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. Primitive Datentypen werden in Java immer per Wertübergabe (*call by value*) übergeben. 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. 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 bis 38 von Kapitel 5 wurde ein einfacher Sortieralgorithmus beschrieben. Programmieren Sie diesen Algorithmus in einer Java-Methode zur *absteigenden* 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. _____ (a);
```

Die Klasse `Arrays` besitzt außerdem eine Methode zur direkten Ausgabe eines Feldes als Zeichenkette, sodass die Implementierung einer Ausgabe-Schleife überflüssig wird.

- Wie lautet diese Methode?

```
Arrays. _____ (a);
```

- Wie kann man diese Methode in Verbindung mit der Ausgabe nutzen?

```
System.out.println( _____ );
```