technische universität dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand

Praktikum zu

Einführung in die Informatik für LogWings, WiMas und MedPhys

Wintersemester 2020/21

Übungsblatt 8

Besprechung:
18.–22.1.2021 (KW 3)

Vorbereitende Aufgaben

Aufgabe 8.1: Manuelle Sortierung

Gegeben ist folgendes Array:

7	12	3	2	21	9

In der Vorlesung (Kapitel 5.1) haben Sie den Algorithmus **Selectionsort** zum Sortieren von Arrays kennengelernt. Sortieren Sie das Array nun aufsteigend nach diesem Algorithmus verfahrend. Notieren Sie dabei **jede** Tauschoperation:

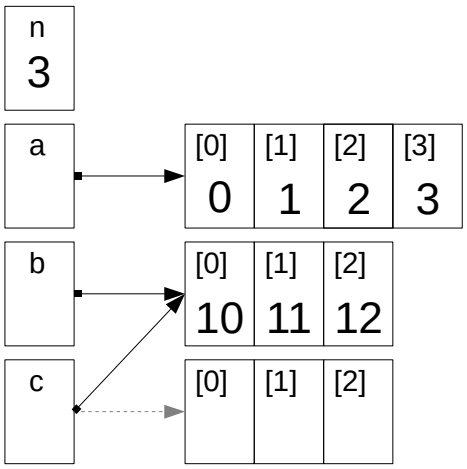
3	**12**	**7**	**2**	**21**	**9**
2	**12**	**7**	**3**	**21**	**9**
2	**7**	**12**	**3**	**21**	**9**
2	**3**	**12**	**7**	**21**	**9**
2	**3**	**7**	**12**	**21**	**9**
2	**3**	**7**	**9**	**21**	**12**
2	**3**	**7**	**9**	**12**	**21**

Präsenzaufgaben

Aufgabe 8.2: Speicherverwaltung für Felder

In dieser Aufgabe wollen wir uns mit der Speicherverwaltung im Zusammenhang mit Arrays beschäftigen. 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 zeilenweise durch und tragen Sie in das nebenstehende Diagramm ein, wie die einzelnen Speicherzellen im Laufe des Programms belegt und geändert werden.

```java
int n;
int[] a;
a = new int[4];
for (int i = 0; i < a.length; i++) {
    a[i] = i;
}
n = a.length - 1;
int[] b;
b = new int[n];
for (int i = 0; i < b.length; i++) {
    b[i] = a[i] + 10;
}
int[] c = new int[b.length];
c = b;
```



Aufgabe 8.3: Arrays als Parameter

In dieser Aufgabe wollen wir lernen, Arrays als Parameter zu nutzen. Legen Sie dazu eine Klasse **UseArrays** an.

a) Schreiben Sie eine Funktion **printArray**, die den Inhalt eines übergebenen **int**-Array ausgibt. Der Funktionskopf soll wie folgt aussehen: `public static void printArray(int[] array)`

Lösung

```java
public static void printArray(int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        System.out.print(array[i] + ", ");
    }
    System.out.println(array[array.length-1]);
}
```

b) Implementieren Sie eine Funktion mit dem Namen **swap**, die ein **int**-Array, sowie zwei Indizes entgegen nimmt und die Elemente im Array an den entsprechenden Indizes miteinander tauscht. Die Funktion besitzt keine Rückgabe.

Lösung

```java
public static void swap(int[] array, int index1, int index2) {
    int tmp = array[index1];
    array[index1] = array[index2];
    array[index2] = tmp;
}
```

c) Verwenden Sie anschließend die Funktion **swap** in einer **main**-Methode, um das erste mit dem letzten Element eines Arrays zu tauschen. Geben Sie das Array vor und nach dem Aufruf von **swap** mit der Funktion **printArray** aus.

Lösung

```java
public static void main(String[] args) {
    int[] array = {4, 5, 7, 2, 9, 1};
    printArray(array);
    swap(array, 0, array.length-1);
    printArray(array);
}
```

d) Was fällt Ihnen auf, wenn Sie das Ergebnis mit den Erkenntnissen zu primitiven Datentypen als Parameter aus Aufgabe 6.5 vergleichen?

Anders als primitive Werte können Array-Elemente durchaus auch in einer Funktion vertauscht und verändert werden, ohne dass die Änderungen verloren gehen (Call by reference)

Aufgabe 8.4: Bibliothek erstellen

In dieser Aufgabe wollen wir eine Klasse programmieren, mit der Sie Arrays mit verschiedenen Eigenschaften erzeugen können. Erstellen Sie eine neue Klasse namens **ArrayGenerator**. Diese Klasse soll keine **main**-Methode erhalten! Implementieren Sie die folgenden beiden Funktionen in dieser Klasse.

a) Die Funktionen **generateAscendingArray** soll ein **int**-Array der Länge **length** erzeugen und es aufsteigend mit den Werten von 0 bis **length**−1 befüllen.

Lösung

```java
public static int[] generateAscendingArray(int length) {
    int[] result = new int[length];
    for (int i = 0; i < result.length; i++) {
        result[i] = i;
    }
    return result;
}
```

b) Die Funktion **generateRandomArray** soll ein zufällig gefülltes Array mit der Länge **length** erzeugen.

Dafür müssen Sie die Klasse **Random** aus dem Paket **java.util** importieren. Verwenden Sie dafür: `import java.util.Random;` noch vor der Definition der Klasse. Anschließend können Sie mit `Random randomNumbers = new Random();` einen neuen Zufallsgenerator erzeugen und danach mit `randomNumbers.nextInt(n)` eine zufällige Zahl aus dem halboffenen Intervall $[0, n)$ erhalten. Verwenden Sie als n die Länge des erzeugten Arrays.

Lösung

```java
public static int[] generateRandomArray(int length) {
    Random randomNumbers = new Random();
```

```
    int[] result = new int[length];
    for (int i = 0; i < result.length; i++) {
        result[i] = randomNumbers.nextInt(length);
    }
    return result;
}
```

Komplettlösung für die Klasse ArrayGenerator

```
import java.util.Random;

public class ArrayGenerator {
    public static int[] generateAscendingArray(int length) {
        int[] result = new int[length];
        for (int i = 0; i < result.length; i++) {
            result[i] = i;
        }
        return result;
    }

    public static int[] generateRandomArray(int length) {
        Random randomNumbers = new Random();
        int[] result = new int[length];
        for (int i = 0; i < result.length; i++) {
            result[i] = randomNumbers.nextInt(length);
        }
        return result;
    }
}
```

Aufgabe 8.5: Bibliothek verwenden

In dieser Aufgabe wollen wir nun die Funktionen der soeben definierten Klasse verwenden. Eine Klasse ohne **main**-Methode, wie Ihre **ArrayGenerator**-Klasse, nennt man auch **Bibliothek**. Bibliotheken besitzen meistens sehr hilfreiche Funktionen. Diese nützlichen Funktionen aus der eben geschriebenen Bibliothek wollen wir nun in der Klasse **UseArrays** benutzen. Anschließend erweitern wir die Klasse **UseArrays** noch um zwei weitere Funktionen.

a) Verwenden Sie in der **main**-Methode der Klasse **UseArrays** die soeben von Ihnen definierten Funktionen, um ein aufsteigendes und ein zufälliges Array der Länge 10 auszugeben.

Lösung

```
printArray(ArrayGenerator.generateAscendingArray(10));
printArray(ArrayGenerator.generateRandomArray(10));
```

b) Erweitern Sie nun die Klasse **UseArrays** um die Funktion **arrayMin**, die das Minimum eines übergebenen **int**-Arrays zurückgibt. Rufen Sie diese Funktion für ein zufällig generiertes Array auf.

Lösung

```java
public static int arrayMin(int[] array) {
    int min = array[0];
    for (int i = 1; i < array.length; i++) {
        if (array[i] < min) {
            min = array[i];
        }
    }
    return min;
}
```

c) Schreiben Sie die Funktion **average**, die den Mittelwert eines übergebenen **int**-Arrays berechnet und als **double** zurückgibt. Rufen Sie auch diese Funktion für ein zufällig generiertes Array auf.

Lösung

```java
public static double average(int[] array) {
    double sum = 0;
    for (int i = 0; i < array.length; i++) {
        sum = sum + array[i];
    }
    return sum / array.length;
}
```

Komplettlösung für die Klasse UseArrays

```java
public class UseArrays {
    // gehört zu Aufgabe 8.4a:
    public static void printArray(int[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            System.out.print(array[i] + ", ");
        }
        System.out.println(array[array.length-1]);
    }

    // gehört zu Aufgabe 8.4b:
    public static void swap(int[] array, int index1, int index2) {
        int tmp = array[index1];
        array[index1] = array[index2];
        array[index2] = tmp;
    }

    // gehört zu Aufgabe 8.6b:
    public static int arrayMin(int[] array) {
```

```
19          int min = array[0];
20
21          for (int i = 1; i < array.length; i++) {
22              if (array[i] < min) {
23                  min = array[i];
24              }
25          }
26          return min;
27      }
28
29      // gehört zu Aufgabe 8.6c:
30      public static double average(int[] array) {
31          double sum = 0;
32
33          for (int i = 0; i < array.length; i++) {
34              sum = sum + array[i];
35          }
36          return sum / array.length;
37      }
38
39      public static void main(String[] args) {
40          int[] array = {4, 5, 7, 2, 9, 1};
41
42          // gehört zu Aufgabe 8.4:
43          printArray(array);
44          swap(array, 0, array.length-1);
45          printArray(array);
46          // gehört zu Aufgabe 8.6:
47          printArray(ArrayGenerator.generateAscendingArray(10));
48          printArray(ArrayGenerator.generateRandomArray(10));
49
50          System.out.println(arrayMin(
51              ArrayGenerator.generateRandomArray(10)));
52          System.out.println(average(
53              ArrayGenerator.generateRandomArray(10)));
54      }
55  }
```