



## Übung 8

### Aufgabe 1 :

Gegeben ist folgendes Programm:

```
public class Test {  
    public static void main(String[] args) {  
        double[] a;  
        a = new double[1000];  
        for(int i=0; i<=a.length; i++) {  
            a[i] = i;  
        }  
    }  
}
```

Was würden Sie sehen, wenn Sie dieses Programm übersetzen? Was würden Sie sehen, wenn Sie dieses Programm ausführen? Begründen Sie ihre Antwort.

### Aufgabe 2 :

Gegeben ist folgendes Programm:

```
public class Test {  
    public static void main(String[] args) {  
        double[] a;  
        for(int i=0; i<a.length; i++) {  
            a[i] = a.length - i;  
        }  
    }  
}
```

Was würden Sie sehen, wenn Sie dieses Programm übersetzen? Was würden Sie sehen, wenn Sie dieses Programm ausführen? Begründen Sie ihre Antwort.

### Aufgabe 3 :

Ein regelmäßiger Kettenbruch für gegebene Koeffizienten  $b_0, \dots, b_n$  ist definiert durch:

$$b_0 + \frac{1}{b_1 + \frac{1}{b_2 + \frac{1}{b_3 + \frac{1}{\dots b_{n-1} + \frac{1}{b_n}}}}}$$

Geben Sie eine Methode an, die für ein Feld mit double-Werten  $b_0, \dots, b_n$  den Wert des dadurch gegebenen Kettenbruchs berechnet. Die Anzahl der b-Werte (d.h.  $n$ ) ergibt sich aus der Länge des Feldes.

Ein (nichtregelmäßiger) Kettenbruch für gegebene Koeffizienten  $a_1, \dots, a_n$  und  $b_0, \dots, b_n$  ist definiert durch:

$$b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{\dots b_{n-1} + \frac{a_n}{b_n}}}}}$$

Geben Sie eine weitere Methode an, die für zwei Felder mit double-Werten  $a_1, \dots, a_n$  und  $b_0, \dots, b_n$  den Wert des dadurch gegebenen Kettenbruchs berechnet. Die Anzahl der Werte ergibt sich aus der Länge der Felder. Sie können davon ausgehen, dass das b-Feld einen Wert mehr als das a-Feld enthält.

#### Aufgabe 4 :

Gegeben sei ein Feld von Strings. Sie können testweise in ihrem Programm verwenden:

```
String[] feld = { "Hello", "world" };
```

(Zusatzfrage: Was bewirkt diese Deklaration genau? Machen Sie sich dazu eine Skizze.)

Entwickeln Sie zwei Methoden, die jeweils ein solches Feld übergeben bekommen und ein doppelt so großes Feld anlegen und als Resultat liefern, in dem

1. die Referenz auf das Element an der  $i$ -ten Stelle im übergebenen Feld an der  $2*i$ -ten und  $2*i+1$ -ten Stelle im Resultatfeld steht (Referenzen werden kopiert)
2. der Inhalt des Elements an der  $i$ -ten Stelle (also ein String) im übergebenen Feld in die  $2*i$ -te und  $2*i+1$ -te Stelle im Resultatfeld kopiert (Inhalte werden kopiert). Hinweis: `"" + s`

Geben Sie weiterhin zwei Methoden an, die jeweils für zwei Felder der Länge  $2*n$  für alle  $i = 1, \dots, n$  die beiden Referenzen an der Stelle  $2*i$  und  $2*i+1$  überprüfen

1. auf gleiche Referenz
2. ob sie inhaltsgleich sind

Was sehen Sie als Ergebnis, wenn Sie die Resultatfelder aus den beiden ersten Methoden als Argumente verwenden?

#### Aufgabe 5 (1 Punkt) :

##### Dies ist eine Praktomataufgabe!

Geben Sie in einer Klasse `Klausurnoten` eine Methode

```
public static void verarbeiteKlausurPunktzahl(int[] punkte, int[] punktGrenzen)
```

an, die Statistiken zu Klausurpunkten erstellt. Der Methode wird ein Feld `punkte` übergeben, in dem für alle Teilnehmer die erreichten Punkte stehen (also pro Teilnehmer eine Punktzahl). Weiterhin bekommt die Methode ein Feld `punktGrenzen` übergeben, das die Punktobergrenzen für Noten festlegt. Zur Vereinfachung gibt es nur die ganzen Noten 1,2,3,4,5.

Beispiel für die Eingabe:

- `punkte = {30, 74, 81, 67, 95, 65}`. Die Länge des Feldes entspricht der Anzahl der Teilnehmer an der Klausur, hier also 6. Der erste Teilnehmer hatte 30 Punkte, der zweite 74 usw.
- `punktGrenzen = {59, 69, 79, 89}`, also von 0 bis einschließlich 59 Punkte ist die Note eine 5, zwischen 60 und 69 eine 4, zwischen 70 und 79 eine 3, zwischen 80 und 89 eine 2 und ab 90 eine 1. Das Feld hat also immer die Länge 4.

Aus diesen Angaben soll die Methode nacheinander in jeweils einer Zeile folgende Informationen auf dem Bildschirm ausgeben:

- Anzahl der Teilnehmer
- die beste und schlechteste Note in der Form `x y`
- wieviele Studierende bestanden haben (Note 4 und besser), wieviele nicht bestanden haben (in der Form `x y`)
- Durchschnittspunktzahl

- Ein Histogramm der Noten (wieviele Klausuren eine 1 waren, wieviele eine 2 ...) in der Form **Note Anzahl** für alle Noten 1-5, jeweils in einer Zeile

Die Ausgabe zum obigen Beispiel würde dann also aussehen:

```
6
1 5
5 1
68.66666666666667
1 1
2 1
3 1
4 2
5 1
```

### Aufgabe 6 (1 Punkt) : Dies ist eine Praktomataufgabe!

Schreiben Sie in einer Klasse **vertauschen** zwei Methoden **vertauscheInhalt** bzw. **vertauscheReferenz**, die beide als einziges Argument eine Feld von Strings bekommen und als Ergebnis der Methode jeweils ein **neues** Feld von Strings liefern. Bei beiden Methoden soll der 0-te String des Eingabefeldes an der letzten Stelle des Resultatfeldes stehen, der 1-te String des Eingabefeldes an der vorletzten Stelle,..., der letzte String des Eingabefeldes an der ersten Stelle des Eingabefeldes stehen. Die beiden Methoden unterscheiden sich darin, dass in der Methode **vertauscheInhalt** für den i-ten String ein neuer, aber inhaltsgleicher String erzeugt wird, und in der Methode **vertauscheReferenz** die i-te Referenz in das Ergebnisfeld kopiert wird, also der identische String mit der gleichen Referenz wie der Ursprungsstring.

Damit Sie diesen Unterschied in der Wirkungsweise der beiden Methoden auch selber schon testen können, geben Sie zwei weitere Methoden an:

- `boolean testInhaltGleich(String[] arg1, String[] arg2)`
- `boolean testReferenzGleich(String[] arg1, String[] arg2)`

die zwei gleich lange Felder darauf testen, ob sie inhaltsgleich sind (der Inhalt der beiden i-ten Strings muss gleich sein, aber nicht notwendigerweise die Referenzen) oder auch zusätzlich Referenz-gleich, also die beiden i-ten Feldelemente auf den gleichen String verweisen.

Selber testen können Sie ihre vier Methoden, indem Sie die übergebenen Kommandozeilenargumente als Testfeld nehmen.

Beispielaufrufe für solche Plausibilitätstests in **main**:

```
String [] inhalt = vertauscheInhalt(args);
String [] referenz = vertauscheReferenz(args);

// Tests:
// beide muessen gleiche Inhalte haben
if(!testInhaltGleich(inhalt, referenz)) {
    System.out.println("Fehler: Inhalte stimmen nicht ueberein!");
}
// aber beide muessen unterschiedliche Referenzen haben
if(testReferenzGleich(inhalt, referenz)) {
    System.out.println("Fehler: Referenzen muessen unterschiedlich sein!");
}
```