



30. November 2017

## Übungsblatt 7

Abgabe: 13. Dezember 2017

### Abgabe

Ihr erhaltet als Vorgabe *zwei* Dateien: `Felder.java` und `Sortieren.java`.

`Felder.java` Erstere enthält Methoden-Header, die Ihr bitte nicht ändert. Außerdem gibt es wie üblich eine `main` Methode zum Testen. Diese darf auch nicht geändert werden. Dasselbe gilt für die vorgegebenen `print` Methoden für `Felder`. Abzugeben ist dann die bearbeitete `Felder.java`.

`Sortieren.java` Diese Datei enthält das Gerüst einer Klasse incl. einer `main` Methode, welche nicht geändert werden darf. Abzugeben ist die ausgefüllte Datei `Sortieren.java`.

### Aufgaben zu `Felder.java`

Schreibt die folgenden Methoden (je 2 Punkte)!

1. `boomBang` nimmt einen Startwert  $a$  und einen Endwert  $b$ . Die Methode soll ein Feld von Strings returnieren, das wie folgt aufgebaut wird. Für Zahlen  $\geq a$  und  $< b$  werden alle durch 3 teilbaren Zahlen durch "Boom" und alle durch 5 teilbaren Zahlen durch "Bang" ersetzt. Zahlen, die durch 3 und durch 5 teilbar sind werden durch `BoomBang` ersetzt. Alle anderen Zahlen  $i$  werden mit `String.valueOf(i)` in einen String verwandelt. Sie dürfen annehmen, dass  $b > a$ . Beispiele:
  - `boomBang(1,6)` gibt das String Feld {"1", "2", "Boom", "4", "Bang"} zurück.
  - `boomBang(12,16)` gibt das String Feld {"Boom", "13", "14", "BoomBang"} zurück.
2. `geradeUngerade` sortiert ein Feld von Integern so um, dass zuerst alle geraden und dann alle ungeraden Zahlen kommen. Davon abgesehen ist die neue Reihenfolge beliebig. Erzeugen Sie kein neues Feld, sondern arbeiten Sie auf dem gegebenen. Beispiele:
  - `geradeUngerade` von {1,0,1,0,1} ergibt {0,0,1,1,1}
  - `geradeUngerade` von {2,2,2} ergibt {2,2,2}
3. `klumpen` zählt die Anzahl von *Klumpen* in einem Feld. Ein Klumpen ist eine Menge von mindestens 2 direkt aufeinander folgenden identischen Elementen. Zum Beispiel enthält {0,0,0,1,2,3,3,4} zwei Klumpen, und {0,0,0,0,0,0,0,0,0} enthält einen Klumpen.

### Aufgaben zu `Sortieren.java`

Diese Aufgabe erzielt 4 Punkte, wenn sie vollständig gelöst wird. Hierzu erstellen wir eine Klasse mit statischen Methoden, die dazu dient, eine Folge von Zahlen als Kommandozeilenparameter zu übergeben, die schlussendlich sortiert werden. Hierzu sind folgende Methoden zu schreiben:

`int[] einlesen(String[] args)` Diese Methode wandelt ein String Array in ein Feld aus Integern. Hierzu verwende man die Methode `Integer.parseInt(String s)` elementweise

`int[] vertauschen(int[] arr)` Diese Methode betrachtet jeden  $i$ . Wenn das dortige Element größer ist als sein Nachfolger am Index  $i + 1$ , so vertausche die beiden Einträge.

`int[] vertauscheN(int[] arr)` Diese Methode wendet die `vertauschen` Methode so oft nacheinander auf ihr Argument an wie das Argument Einträge hat.

Der Aufruf der kompilierten Klasse erfolgt mit Kommandozeilenparametern, z.B.

```
java Sortieren 4 3 2 1
```