

## Übungsblatt 6

Abgabedatum: 28.11.2021

Die Abgabe Ihrer Lösungen erfolgt vor Ablauf der Abgabefrist digital über die Moodle-Plattform. Erstellen Sie dazu ein PDF-Dokument, das die Lösungen Ihrer schriftlichen Aufgaben enthält. Laden Sie dieses PDF-Dokument und den erarbeiteten Java-Code (.java-Dateien) mit den in den Aufgaben vorgegebenen Namen bei Moodle hoch. Bitte laden Sie die Dateien einzeln hoch, Dateiarhive (z.B. .zip-Dateien) werden nicht akzeptiert.

Sie können maximal **(9 + 1 Punkte)** mit diesem Übungsblatt erreichen.

### Aufgabe 1 (Selection Sort)

2 Punkte

Selection Sort ist ein einfach zu implementierender aber nicht unbedingt effektiver Sortieralgorithmus, der Werte in einem Array sortiert. Das Verfahren funktioniert wie folgt:

- Beginne mit dem ersten Wert im Array an der Stelle  $i = 0$ .
- Finde den kleinsten Wert im restlichen Array (mit  $\text{Index} \geq i$ ).
- Tausche den Wert an Stelle  $i$  mit dem gefundenen kleinsten Wert.
- Erhöhe  $i$  um eins um wiederhole das Verfahren bis das gesamte Array sortiert ist.

Beispiel:

```
i = 0  [6 | 4 | 3 | 1 | 2 | 5]  tausche 6 und 1
i = 1  [1 | 4 | 3 | 6 | 2 | 5]  tausche 4 und 2
i = 2  [1 | 2 | 3 | 6 | 4 | 5]  tausche 3 und 3
i = 3  [1 | 2 | 3 | 6 | 4 | 5]  tausche 6 und 4
i = 4  [1 | 2 | 3 | 4 | 6 | 5]  tausche 6 und 5
i = 5  [1 | 2 | 3 | 4 | 5 | 6]  Array durchlaufen und fertig sortiert
```

1. Schreiben Sie eine Klasse `SelectionSort.java` mit einer `main`-Methode, die den Nutzer nach der Länge eines Arrays fragt und diese dann einliest. Schreiben Sie des Weiteren eine statische Funktion `newArray` die als Eingabe einen `int`-Wert erhält und ein Array zurückliefert, das die Zahlen 1 bis  $n$  aufsteigend enthält. Verwenden Sie für die Funktion die folgende Signatur:

```
private static int[] newArray(int n)
```

2. Schreiben Sie eine Funktion `mixArray`, die die Zahlen im Array mischt und eine Funktion `showArray`, die die Werte des Arrays in einer Zeile in der Konsole ausgibt. Nutzen Sie dazu die beiden folgenden Signaturen:

```
private static void mixArray(int[] array)
```

```
private static void showArray(int[] array)
```

3. Schreiben Sie eine Funktion `selectionSort` die das Array nach dem oben vorgestellten Algorithmus sortiert. Nutzen Sie dabei keine vorgefertigte Sortierfunktion! Verwenden Sie für die Funktion die folgende Signatur:

```
private static void selectionSort(int[] array)
```

## Aufgabe 2 (Game of Life)

2 Punkte

In der Klasse `GameOfLife.java` ist ein Gerüst für Conways Spiel des Lebens gegeben. Der folgende Wikipedia Eintrag beschreibt das Spiel [https://de.wikipedia.org/wiki/Conways\\_Spiel\\_des\\_Lebens](https://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens).

Importieren Sie die beiden Klassen `GameOfLife.java` und `MapView.java` in Ihren Workspace. Die Klasse `GameOfLife.java` simuliert das Spiel und die Klasse `MapView.java` erzeugt ein grafisches Fenster in dem das Spiel dargestellt wird. Vervollständigen Sie die Methode `conwaysRules` so, dass die folgenden 4 Regeln für das Sterben bzw. Leben der Zellen eingehalten werden:

- Eine tote Zelle mit genau drei lebenden Nachbarn wird in der Folgegeneration neu geboren.
- Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit.
- Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt in der Folgegeneration am Leben.
- Lebende Zellen mit mehr als drei lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung.

Jede Zelle wird durch einen boolschen Wert innerhalb des zwei dimensional Arrays `cells` beschrieben. Der boolsche Wert ist dabei genau dann `true`, wenn die Zelle lebt und `false` wenn die Zelle nicht lebt.

## Aufgabe 3 (Minesweeper)

5 + 1 Punkte

Laden Sie sich die Vorgabe `Minesweeper.java` herunter und schauen Sie sich die Klasse, deren Kommentare und deren globalen Variablen gut an und implementieren Sie alle TODOs derart, dass alle Methoden das bewirken, was Ihre Kommentare angeben.

Mehr Informationen zu Minesweeper finden Sie hier <sup>1</sup>

Zusatz: Ändern Sie die `showField`-Methode derart, dass die Anzeige nicht mehr in der System-Konsole, sondern in der vom Lehrstuhl entworfenen Console ausgegeben wird.

```
Welches Feld willst du markieren? Hoehe: 1
Breite: 1
-1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
0 | - | - | - | - | - | - | 0 | 0 | - |
1 | - | 0 | - | - | - | - | - | - | - |
2 | - | 0 | - | 2 | - | - | - | - | - |
3 | - | - | - | - | - | - | - | - | - |
4 | - | - | - | - | - | - | - | - | - |
5 | - | - | - | - | - | - | - | - | - |
6 | 1 | - | - | - | - | - | - | - | - |
7 | - | - | - | - | - | 1 | - | 1 | - |
8 | - | - | - | - | - | - | - | - | 0 |
Welches Feld willst du markieren? Hoehe:
```

<sup>1</sup><https://de.wikipedia.org/wiki/Minesweeper>