

Aufgabe 6 – Schnittstellen

6.1 Game of Life mit Schnittstellen

Die Aufgabe zum „Game of Life“ hat einen ziemlich Design-Fehler: Die Klasse des Grundgerüsts gibt die Klasse, in der Ihre Logik implementiert wurde, vor. Schöner wäre es, wenn das Grundgerüst lediglich eine Schnittstelle vorgibt, die die Logik implementieren muss. Leider waren Ihnen zum damaligen Zeitpunkt Schnittstellen zumindest aus der Vorlesung noch nicht bekannt. Bauen Sie Ihre Lösung so um, dass die Klasse `GameOfLifeApplication` eine Schnittstelle vorgibt, die die Logik implementiert. In `GameOfLifeApplication` soll dann eine Referenz auf das Logik-Objekt übergeben werden. Leider geht das in der JavaFX-Lösung nur durch eine Klassenmethode in `GameOfLifeApplication`.

6.2 Pflichtaufgabe: Sortierung von Studierenden

Die folgende Klasse für Studierende ist vorgegeben:

```
public class Student {  
    private int matriculationNumber;  
    private String firstname;  
    private String lastname;  
  
    // Getter, Setter und Konstruktoren vorhanden.  
}
```

Erstellen Sie eine `ArrayList`, in der Sie einige unterschiedliche Studierenden-Objekte eintragen. Mit der Methode

```
Collections.sort(List<T> list, Comparator<? super T> c)
```

können Sie Datenstrukturen der Collections-API sortieren. Um Ihre `Student`-Objekte in der `ArrayList` sortieren zu können, schreiben Sie eine `Comparator`-Klasse. Die Sortierung soll anhand des Nachnamens erfolgen. Sind die Nachnamen gleich, dann berücksichtigen Sie auch die Vornamen. Sind diese auch gleich, dann entscheidet die Matrikelnummer.