

## Programmierung A/B (in Java) – Übung 9

**Thema:** IO Streams, Exceptions, Generics

**Lernziele:** In dieser Übung sollen Sie mit Readern, Writern, generischen Klassen und Exceptions arbeiten

### Aufgabe 1

Schreiben Sie eine Klasse *Student*, welche den *Vornamen*, den *Nachnamen*, die *Universität* und die *Matrikelnummer* des Studenten speichert. All diese Werte sollen im Konstruktor übergeben werden.

Implementieren Sie für die Klasse *Student* das Interface *Comparable* mit dem Typparameter *Student*. Die Implementierung der Interface Methode *compareTo* soll die Studenten anhand ihrer Matrikelnummer vergleichen.

Um die Universitätszugehörigkeit eines Studenten zu speichern, bietet es sich an, den Typ *String* zu benutzen. Diese Vorgehensweise begünstigt jedoch Fehler, da sich der Name einer Universität auf unterschiedliche Weisen als *String* ausdrücken lässt (z.B. „Uni“ statt „Universität“ oder „Duisburg Essen“ anstatt „Duisburg-Essen“). Welche Möglichkeiten bietet Java an, die Universitätszugehörigkeit menschenlesbar und eindeutig zu speichern? Implementieren Sie diese.

Schreiben Sie außerdem eine Klasse *Vorlesung*, die einen *Namen*, eine *Vorlesungsnummer*, und eine Liste von angemeldeten Studenten enthält, und die die Noten jedes Studenten in eine *HashMap* speichert. Implementieren Sie die Methode „**void** anmelden(*Student student*)“, welche einen Studenten in die Liste angemeldeter Studenten einfügt und die Methode **void** „*pruefungDurchfuehren()*“, welche jedem angemeldeten Studenten eine Note gibt (diese Note soll der Einfachheit halber eine zufällige Ganzzahl zwischen 1 und 6 sein, inkl. 1 und 6). Implementieren Sie außerdem die Methode „**float** *getNote*(*Student student*)“, welche die Note eines Studenten liefert. Diese Methode soll eine Exception werfen, wenn der übergebene Student nicht für die Prüfung angemeldet ist. Definieren Sie hierzu den Exceptiontyp *StudentNotFoundException*, der sich von *RuntimeException* ableitet.

### Aufgabe 2

Schreiben Sie die Klasse *Moodle*, die Studenten und Vorlesungen verwaltet. Die Klasse soll die folgenden Methoden anbieten:

```
public void immatrikuliere(Student student);
public void exmatrikuliere(Student student);
public List<Student> getStudenten();
public void vorlesungHinzufuegen(Vorlesung vorlesung);
public List<Vorlesungen> getVorlesungen();
public void pruefungDurchfuehren(Vorlesung vorlesung);
public void schreibeStudentenListe(String dateiName);
public void schreibeVorlesung(Vorlesung vorlesung, String dateiName);
public void leseStudenten(String dateiName);
(Optional) public void leseVorlesung(String dateiName);
```

```
public void immatrikuliere(Student student);
```

Die Methode soll einen Studenten zu Moodle hinzufügen. Hierbei bietet es sich an die Studenten in einer Liste zu verwalten. Die Methode muss sicherstellen, dass keine zwei Studenten die gleiche Matrikelnummer haben. Wird ein Student immatrikuliert, dessen Matrikelnummer schon verwendet wurde, soll eine Fehlermeldung erscheinen (optional können Sie hier auch eine Exception StudentImmatrikulationException vom Typ RuntimeException werfen).

```
public void exmatrikuliere(Student student);
```

Die Methode soll einen Studenten aus Moodle entfernen. Ist der betreffende Student dem System nicht bekannt, soll der Aufruf ignoriert werden.

```
public List<Student> getStudenten();
```

Die Methode soll die Liste aller Studenten zurückgeben, die in Moodle registriert sind.

```
public void vorlesungHinzufuegen(Vorlesung vorlesung);
```

Die Methode soll eine Vorlesung zu Moodle hinzufügen. Ähnlich wie bei der Methode Immatrikuliere Student sollen keine zwei Vorlesungen mit der gleichen Vorlesungsnummer hinzugefügt werden können.

```
public List<Vorlesungen> getVorlesungen();
```

Die Methode soll die Liste aller Vorlesungen zurückgeben, die in Moodle registriert sind.

```
public void pruefungDurchfuehren(Vorlesung vorlesung);
```

Die Methode soll die Prüfung für eine Vorlesung durchführen.

```
public void schreibeStudentenListe(String dateiName);
```

Die Methode soll alle Studenten, die Moodle bekannt sind, in eine Textdatei schreiben. In jeweils einer Zeile sollen die Informationen über einen Studenten stehen. Verwenden Sie dafür folgendes Format: <Vorname>;<Nachname>;<Universität>;<Matrikelnummer>;<VorlesungsNummer1>;<Note1>;<VorlesungsNummer2>;<Note2>...

Bsp:

Max;Mustermann;Duisburg-Essen;123456;2124;1

...

```
public void schreibeVorlesung (Vorlesung vorlesung, String dateiName);
```

Die Methode soll die Ergebnisse einer Prüfung in eine Textdatei schreiben. In der ersten Zeile sollen der Name der Vorlesung und die Vorlesungsnummer stehen. In den folgenden Zeilen sollen die Matrikelnummern der teilgenommenen Studenten und deren Noten stehen. Sortieren Sie die Studenten hierbei nach ihrer Matrikelnummer.

Bsp:

Geometrische Modellierung;2124

123456;1

...

Im dem Fall, dass die Prüfung noch nicht geschrieben ist, sollen nur die Matrikelnummern der angemeldeten Studenten aufgelistet werden.

**public void** leseStudenten(String **dateiName**)

Die Methode soll eine Datei, welche mit `schreibeStudentenListe` geschrieben wurde, lesen können und in der Klasse Moodle speichern. Beim Lesen können die Prüfungsnoten ignoriert werden.

**Optional: public void** leseVorlesung(String **dateiName**)

Die Methode soll eine Datei, welche mit `schreibeVorlesung` geschrieben wurde, lesen können und ein Objekt Vorlesung in Moodle hinzufügen. Studenten sollen automatisch der neu erstellten Klasse hinzugefügt werden, sofern diese Moodle schon bekannt sind.

### Aufgabe 3

Schreiben Sie eine `main` Methode, welche alle Funktionalitäten testet: Mehrere Studenten zu Moodle hinzufügen, Vorlesungen zu Moodle hinzufügen. Studenten zu Vorlesungen hinzufügen, Prüfungen und Studenten schreiben, sowie das Schreiben und Lesen von Studenten und Vorlesungen.

#### Hinweise:

Die Noten können in einer `HashMap<Student, Float>` gespeichert werden und per Zufall generiert werden (`Math.Random`). Nutzen Sie die Methode `parseInt()` der Wrapperklasse `Integer` um aus einem `String` eine `int` Variable zu erhalten. Entsprechende Methoden gibt es für alle Wrapperklassen.