

Einführung in die Informatik

Institut für Eingebettete Systeme/Echtzeitsysteme | Wintersemester 2020/21

Julian Bestler, Nicolas Zellner, Valentina Richthammer, Michael Glaß

Übungsblatt 7: Objekt-Orientierte Programmierung

Abgabetermin: 10.01.2021, 23:59 Uhr

Geben Sie **Programmieraufgaben als Java Code (*.java Dateien)** ab. Alle anderen Aufgaben, die Text oder Grafiken erfordern, geben Sie **als PDF Dateien** ab. PDFs können Sie beispielsweise mit dem kostenlosen Programm *LibreOffice* erstellen. Alternativ können Sie etwas mehr Zeit investieren und LaTeX lernen, was Sie im späteren Studium immer wieder brauchen werden.

Wenn Sie **mehrere Dateien** abgeben wollen, dann fassen Sie diese zu **einem ZIP File** zusammen.

Präsenzaufgaben werden direkt im Tutorium bearbeitet, werden nicht bepunktet und müssen nicht abgegeben werden.

Präsenzaufgabe

Sie möchten ein Programm schreiben, mit dem Sie tracken können, welche Fernsehserien Sie aktuell schauen und welche Folgen Sie bereits gesehen haben. Überlegen Sie sich dazu, wie Sie in Java eine entsprechende Datenbank realisieren können. Beschreiben Sie dazu zwei Klassen: eine Klasse `TVShowsDB`, die eine Liste von Serien speichert und eine Klasse `TVShow`, die die Informationen zu einer bestimmten Serie beinhaltet.

Die Klasse `TVShow` soll den Titel der Serie, die Anzahl der verfügbaren Folgen und die Nummer der zuletzt gesehenen Folge enthalten. Die Attribute sollen alle `private` deklariert werden und über entsprechende sinnvoll benannte Methoden angepasst werden können. Sie brauchen die Methoden nicht auszuprogrammieren. Es genügt die Aufgaben der Methoden als Text oder Pseudocode zu beschreiben.

Die Klasse `TVShowsDB` soll eine Liste von Serien vom Typ `TVShow` enthalten. Es soll die Möglichkeit geben Serien hinzuzufügen und zu löschen. Außerdem soll man die Liste mit allen verfügbaren Informationen ausgeben können.

Wie könnte eine `main` Methode aussehen, die eine Datenbank erstellt, mehrere Serien hinzufügt und ggf. die gesehenen Folgen anpasst oder Serien wieder löscht?

Aufgabe 1: Fahrrad fahren

(6+2)

a) Kopieren Sie die Klasse `Bicycle2` aus der Vorlesung in Kapitel 7 in eine entsprechende Java Datei. Schreiben Sie dann eine `main` Methode, die folgendes tut:

1. Erzeugen Sie eine Instanz eines Fahrrads vom Typ `Bicycle2` ohne konkrete Parameter.
2. Geben Sie den Status des Objekts durch Aufruf von `printState()` aus. Welche Ausgabe wird erzeugt? Sie können Ihre Antwort als Kommentar direkt im Java-Code angeben.
3. Erzeugen Sie ein zweites Fahrrad vom Typ `Bicycle2`. Diesmal soll bei der Konstruktion festgelegt werden, dass das Fahrrad im ersten Gang ist. Die Geschwindigkeit und Trittfrequenz soll 0 sein. Geben Sie auch den Status des zweiten Fahrrads aus. Welche Ausgabe wird erzeugt?
4. Legen Sie beim ersten Fahrrad den zweiten Gang ein. Die Trittfrequenz soll auf 10 geändert werden und die Geschwindigkeit erhöht sich um 11.5.
5. Geben Sie den Status beider Fahrräder nochmals aus. Welche Ausgabe ist zu beobachten?
6. Synchronisieren Sie das zweite Fahrrad mit dem ersten. D.h., Fahrrad 2 soll durch Aufruf von `syncBikes` die selben Parameter erhalten, wie das erste Fahrrad. Geben Sie nochmals den Status beider Räder aus um die Korrektheit zu prüfen.

b) Schreiben Sie eine verbesserte Klasse `Bicycle3`, die sicherstellt, dass keine ungültigen Gänge eingelegt werden können und dass die Trittfrequenz nicht negativ sein kann. Die Gänge 1 bis 9 sind gültig. Wenn ein Gang kleiner 1 angegeben wird, dann soll der erste Gang eingelegt werden und bei Gängen größer 9 soll die 9 gewählt werden. Genauso soll bei negativer Trittfrequenz die 0 gespeichert werden.

Führen Sie nochmal alle Schritte aus Aufgabenteil a) aus. Welche Ausgaben unterscheiden sich nun und wie?

Aufgabe 2: Studierende und Mitarbeiter

(8)

Gegeben sei folgende Person-Klasse.

```
1 public class Person {
2     String name = ""; // Name einer Person
3     int alter = 0; // Alter einer Person
4
5     // Kontruktor der Klasse Person - initialisiert ein Person-Objekt
6     public Person(String name, int alter) {
7         this.name = name;
8         this.alter = alter;
9     }
10
11     // Allgemeine Ausgabe von Personendaten
12     public void ausgabe() {
13         System.out.println(name + " ist " + alter + " alt.");
14     }
15
16     public static void main(String[] args) throws Exception {
17
18         Person p1 = new Person("Anton", 23);
19         Person p2 = new Person("Berta", 23);
20         Person p3 = new Person("Christa", 21);
21
22         p1.ausgabe();
23         p2.ausgabe();
24         p3.ausgabe();
25     }
26 }
```

Die Person-Klasse eignet sich noch nicht, um Studierende und Mitarbeiter einer Uni zu verwalten. Im Folgenden sollen Klassen und Objekte erstellt werden, um eine entsprechende Verwaltung zu ermöglichen.

- Implementieren Sie eine Klasse `Student`, die alle Eigenschaften einer `Person` erbt und zusätzlich noch eine Matrikelnummer speichern kann. Überschreiben Sie die bisherige Implementierung der Methode `ausgabe()` mit einer Version, die die Daten eines Studierenden ausgibt (Name, Alter, Matrikelnummer).
- Implementieren Sie eine Klasse `Mitarbeiter`, die alle Eigenschaften einer `Person` erbt und zusätzlich noch einen Abteilungsnamen speichern kann. Überschreiben Sie die bisherige Implementierung der Methode `ausgabe()` mit einer Version, die die Daten eines Mitarbeiters ausgibt (Name, Alter, Abteilung).
- Implementieren Sie ein Programm, welches mehrere `Student`- und `Mitarbeiter`-Objekte einliest und in einem `Person`-Array speichert (`Person []`). Anschließend sollen alle Objekte, die in dem Array gespeichert sind, mit der Methode `ausgabe()` auf der Konsole ausgegeben werden.
- Diskutieren Sie: Was versteht man unter dynamischer Bindung von Methoden, und wie wird dies in Teilaufgabe c) eingesetzt?

Anmerkung: Im Programm wird davon ausgegangen, dass der Nutzer stets sinnvolle Daten eingibt - die möglichen Exceptions bei Fehleingabe werden lediglich nach außen weitergegeben.