

Übungsblatt 1 (13.10.2015)

Teamorganisation, Java, Konfigurationsmanagement

In dieser Übung:

- ✓ Organisieren Sie Ihr Team für die Aufgabenbearbeitung im Verlauf des Semesters.
- ✓ Unternehmen Sie erste Schritte in der Programmiersprache Java.
- ✓ Bereiten Sie das Thema Konfigurationsmanagement vor.

Aufgabe 1.1: Einschreibung in die Übungsgruppe

Präsenz: Nein	Punkte: keine	Team: Ja(2)	Projekt: Nein	
---------------	---------------	-------------	---------------	--

Wie im Hinweisblatt zu den Übungen (**Hinweisblatt** in Moodle) unter der **Teameinteilung** beschrieben müssen Sie sich einerseits für eine Übungsgruppe entscheiden und andererseits in 2er Teams aufteilen. Entscheiden Sie sich zunächst für eine der drei Übungsgruppen, in dem Sie sich unter:

<https://www.mathi.uni-heidelberg.de/muesli/lecture/view/537>

für eine Übungsgruppe eintragen. Bilden Sie dann die entsprechenden 2er Teams, in dem Sie sich unter:

<https://elearning2.uni-heidelberg.de/mod/questionnaire/view.php?id=260225>

eintragen (Link **Teambildung** im Moodle Kurs). Sollten Sie keineN TeampartnerIn für ein 2er Team finden können, wählen sie lediglich eine Übungsgruppe und kommen dann zum ersten Übungsgruppentermin am Mittwoch, den 21.10. bzw. Donnerstag, den 22.10. Dort wird dann für alle verbleibenden Gruppenmitglieder die Zuteilung zu einem Team durchgeführt.

Ergebnis:

Tragen Sie sich in MÜSLI in einer der Übungsgruppen ein und füllen Sie das Formular zur Teambildung bis **Dienstag 20.10.2015 um 23.55 Uhr** in Moodle aus.

Aufgabe 1.2: Java – Klassen und Objekte

Präsenz: Nein	Punkte: 12	Team: Nein	Projekt: Nein	Testat
---------------	------------	------------	---------------	--------

Erstellen Sie die Klassen `Movie` und `Performer` sowie den Aufzählungstyp `Gender` mit den angegebenen Attributen:

<code>public class Movie</code>	<code>public class Performer</code>	<code>public enum Gender</code>
<code>private String title;</code> <code>private int time;</code>	<code>private String firstname;</code> <code>private String lastname;</code> <code>private Gender gender;</code>	<code>MALE, FEMALE;</code>

Beantworten Sie die nachfolgenden Fragen, in dem Sie die Klasse `Movie` programmieren.

1. **Lernziel Konstruktor:** Wie würde ein geeigneter Konstruktor für die Klasse `Movie` aussehen, der die Attribute `title` und `time` bei Objekterzeugung initialisiert? Programmieren Sie diesen Konstruktor.
2. **Lernziel main, Objektdeklaration und Objekterzeugung:** Erzeugen Sie ein Objekt der Klasse `Movie` in der `main`-Methode der Klasse. Deklarieren Sie das Objekt in einer Zeile und erzeugen Sie das Objekt in einer zweiten Zeile, um sich den Unterschied bewusst zu machen.
3. **Lernziel Getter/Setter Operationen:** Erzeugen Sie für die beiden Attribute `title` und `time` jeweils eine `get` und eine `set`-Operation (z.B. `setTitle`). Benutzen Sie dafür die von Eclipse bereitgestellte Hilfsfunktion „Generate Getter/Setter Methods“.
4. **Lernziel this:** Erläutern Sie die Bedeutung der `this`-Referenz anhand der Operation `setTitle`, die den Titel zuweist. Schreiben Sie Ihre Erklärung als Kommentar an die Operation.
5. **Lernziel statische Attribute:** Erweitern Sie die Klasse `Movie` um ein Attribut `number`, welches es ermöglicht, allen Filmen eine eindeutige, ganze fortlaufende Nummer zuordnet. Erweitern Sie die Klasse `Movie` um eine Klassenvariable `nextNumber`, die die jeweils nächste zu vergebende Folgenummer enthält. Setzen Sie die nächste Nummer sobald ein `Movie`-Objekt erzeugt wird.
6. **Lernziel Instanz-Operation:** Erweitern Sie die Klasse `Movie` um eine Operation `showInformation`, die alle Attribute des Objekts ausgibt.

Programmieren Sie die Klasse `Performer` mit den angegebenen Attributen `firstname`, `lastname` und `gender` und erstellen Sie einen geeigneten Konstruktor sowie Getter/Setter-Operationen.

7. **Lernziel Objektreferenz:** Erweitern Sie die Klasse `Performer` um ein Attribut `movie`, welches eine Referenz auf einen `Movie` darstellt. Das Attribut soll auch über entsprechende Operationen gespeichert und abgerufen werden können.

Ergebnis:

Speichern Sie bitte Ihr Eclipse-Projekt bis **Montag 19.10.2015 um 10.00 Uhr** in Moodle. Das Eclipse-Projekt soll die von Ihnen programmierten, kommentierten, kompilier- und lauffähigen Klassen `Performer.java`, `Movie.java` und `Gender.java` enthalten.

Aufgabe 1.3: Java – Anwendung (1)

Präsenz: Nein	Punkte: 10	Team: Nein	Projekt: Nein	Testat
---------------	------------	------------	---------------	--------

Erstellen Sie eine Anwendung „**MyFirstMovieManager**“, welche die Zuordnung von `Performer` zu `Movies` durchführt. Schreiben Sie dazu eine neue Klasse `MyFirstMovieManager`. In einem `Movie` können 1 oder mehrere `Performer` (1..*) mitspielen. Ein `Performer` kann in beliebig vielen `Movies` mitspielen.

1. **Lernziel Collections:** Unterstützen Sie durch entsprechende Attribute und Operationen an der Klasse `Movie` und `Performer` folgende Funktionalität:
 - **Funktionalität 1: Movie Management**
 - Neue `Performer` zu einem `Movie` hinzufügen
 - Abfrage von einzelnen `Performern` im `Movie`
 - Abfrage aller `Performer` aus dem `Movie`
 - Abfrage aller `Performer` mit dem Attribut `outstanding = true`;

- Löschen von Performern aus Movie
- **Funktionalität 2: Performer Management**
 - Eigenschaften ändern
 - Eigenschaften abfragen
 - Kennzeichnen eines/einer ausgezeichneten Performer als outstanding

Für die Zuordnung von Performern zu Movies können Sie entweder eine Collection vom Typ `java.util.ArrayList`, `java.util.HashMap` oder `java.util.HashSet` verwenden.

- Dokumentieren Sie im Quellcode die Antwort auf die Frage: „**Für welche Collection habe ich mich entschieden?**“ Die Frage kann durch die drei Optionen „ArrayList“, „HashMap“ oder „HashSet“ beantwortet werden.
 - Nennen Sie Vor- und Nachteile aller drei Collections als Kommentar im Quellcode.
 - Begründen Sie Ihre Entscheidung dadurch, dass Sie die wichtigsten Argumente kennzeichnen.
- 2. Lernziel Anwendung:** Überlegen Sie sich ein Anwendungsszenario, in dem alle Operationen benötigt werden. Realisieren Sie dieses in der main-Operation einer neuen Klasse `MyFirstMovieManager`. Aus den Ausgaben der laufenden Anwendung muss sofort der Ablauf des Anwendungsszenarios erkennbar sein. Eine **Benutzungsschnittstelle**, welche eine Interaktion ermöglicht, beispielsweise per Kommandozeile oder per GUI-Bibliothek Swing, **muss nicht erstellt werden**.

Ergebnis:

Speichern Sie bitte Ihr exportiertes Eclipse-Projekt als .zip-Datei bis **Montag 19.10.2015 um 10.00 Uhr** in Moodle. Das Eclipse-Projekt soll die von Ihnen programmierten, kommentierten, kompilier- und lauffähigen Klassen `Performer.java`, `Movie.java`, `Gender.java` und `MyFirstMovieManager.java` enthalten.

Aufgabe 1.4: Vorbereitung Konfigurationsmanagement

Präsenz: Nein

Punkte: 6

Team: Nein

Projekt: Nein

Lesen Sie zur Vorbereitung des Themas Konfigurationsmanagement Kapitel 21 aus dem Buch von Ludewig und Lichter (Link **Literatur/Buchkapitel Ludewig und Lichter** im Moodle Kurs).

Tragen Sie anschließend die Inhalte zu den folgenden 5 Punkten in das Trello Board „Vorbereitung Konfigurationsmanagement“ des Teams ISW ein:

- Wichtige Begriffe (und ihre Definitionen)
- Wichtige Techniken des Konfigurationsmanagements
- Die Tätigkeiten im Software Engineering, die dadurch unterstützt werden
- Den Beitrag der Techniken zu den in der Vorlesung genannten Kernfragen
- Vor- und Nachteile des Einsatzes der Techniken

Für jeden der 5 Punkte ist eine eigene Liste vorhanden, in die Sie die Inhalte eintragen können. Orientieren Sie sich für Ihre Einträge an dem vorgegebenen Beispiel einer Software Einheit.

Diese Aufgabe ist also eine Crowd-Sourcing-Aufgabe, d.h. Sie erstellen mit allen VorlesungsteilnehmerInnen gemeinsam diese Übersicht und können vorhandene Einträge bearbeiten und verbessern. Sie lernen am meisten, wenn Sie sich aktiv beteiligen. Die Inhalte des Boards „Konfigurationsmanagement“ werden in der Vorlesung besprochen (wenn nichts drin ist, wird auch nichts besprochen).

Ergebnis:

Tragen Sie bitte Ihre Inhalte bis **Montag 19.10.2015 um 10.00 Uhr** in das Trello Board ein.