

Übungsblatt 1 (15.10.2013)

Java, Qualitätssicherung

In dieser Übung:

- ✓ Unternehmen Sie erste Schritte in Java.
- ✓ Stellen Sie Überlegungen zur Qualitätssicherung hinsichtlich Software Fehlern an.
- ✓ Organisieren Sie Ihr Team für die Aufgabenbearbeitung im Verlauf des Semesters.

Speichern von Eclipse-Projekten in Moodle:

Um uns in die Lage zu versetzen, Ihren Quellcode schnell zu korrigieren, bitten wir Sie, für alle Eigenentwicklungen im Rahmen der zukünftigen Übungen folgende Konventionen für Ihre Eclipse-Projekte einzuhalten:

- **Projekt benennen:** nachname-aufgabennummer. Beispielsweise **mueller-a12**.
- **Bibliotheken verwenden:** Alle verwendeten Bibliotheken (für dieses Arbeitsblatt noch nicht notwendig) müssen in einem Unterverzeichnis *lib* enthalten sein und in den Java *Build Path* des Projekts eingebunden werden.
- **Projekt exportieren** (durch Eclipse unterstützt): Zur Abgabe komprimieren Sie das gesamte Projekt als archive file im zip-Format und speichern diese Datei in Moodle. Die Dateien *.classpath* und *.project*, sowie verwendete Bibliotheken (im *lib*-Verzeichnis) müssen enthalten sein.

Ein MUSS für Ihre zukünftigen Programmertätigkeiten:

Kommentieren Sie stets Ihren Quellcode und wenden Sie die Java Programmierrichtlinien konsequent an! Geben Sie Quellcode immer in dem Zustand ab, in dem Sie diesen selbst nach ein paar Monaten noch verstehen können.

Abgabe von Programmieraufgaben:

Die Programmieraufgaben auf den nachfolgenden Übungsblättern müssen von jedem Teammitglied separat bearbeitet und abgegeben werden. Eine Ausnahme hiervon bilden nur Aufgaben welche explizit zur Bearbeitung durch ein Team gekennzeichnet sind.

Aufgabe 1.1: Java – Klassen und Objekte

Punkte: 12

Team: Nein

Erstellen Sie die Klassen `Movie` und `Performer` sowie die Enumeration `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 nachfolgende 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`-Operation 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 zuzuordnen. Erweitern Sie die Klasse `Movie` um eine Klassenvariable `nextNumber`, die die jeweils nächste zu vergebende Folgenummer erhält. Setzen Sie nächste `number` 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.

Ergebnis:

Speichern Sie bitte Ihr Eclipse-Projekt bis **Montag 21.10.2013 um 10.00 Uhr** in Moodle. Das Eclipse-Projekt soll die von Ihnen programmierten, kommentierten, kompilierfähigen und lauffähigen Klassen `Performer.java`, `Movie.java` und `Gender.java` enthalten.

Aufgabe 1.2: Java – Anwendung (1)

Punkte: 10

Team: Nein

Erstellen Sie eine Anwendung „**MovieManager**“, welche die Zuordnung von `Performer` zu `Movies` durchführt. Schreiben Sie dazu eine neue Klasse `MovieManager`. 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`

Zur 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 `MovieManager`. Aus den Ausgaben der laufenden Anwendung muss sofort der Ablauf des Anwendungs-Szenarios erkennbar sein.

Eine **Benutzungsschnittstelle**, welche eine Interaktion ermöglicht, beispielsweise per `commandline` oder per GUI-Bibliothek `Swing`, **muss nicht erstellt werden**.

Ergebnis: Speichern Sie bitte Ihr exportiertes Eclipse-Projekt als `.zip`-Datei bis **Montag 21.10.2013 um 10.00 Uhr** in Moodle.

Aufgabe 1.3: Vorbereitung Qualitätssicherung

Punkte: 4	Team: Nein	
-----------	------------	--

Beschreiben Sie **2 dynamisch zu entdeckende Softwarefehler** (durch echte oder gedankliche Ausführung des Codes mit Testfällen) und **2 statisch zu entdeckende Softwarefehler** (manuelle oder automatische Überprüfung des Codes, ohne ihn (gedanklich/abstrakt) auszuführen) entsprechend des nachfolgenden Beispiels. Bitte beschreiben Sie maximal 1 statischen Softwarefehler, der durch den Compiler entdeckt werden kann.

Beispiel für Fehlerbeschreibung:

Softwarefehler	Erklärung	Statisch/Dynamisch + Begründung
Toter Code	Code, der nie ausgeführt wird	Statisch, kann Compiler entdecken

Ergebnis:

Speichern Sie bitte Ihre Ergebnisse als PDF bis **Montag 21.10.2013 um 10.00 Uhr** in Moodle.

Teameinteilung:

Im Laufe des kommenden Semesters werden Sie einige der Übungsaufgaben als 2er Team bearbeiten. Hierfür ist zunächst eine Bildung der 2er Teams notwendig. Die Mitglieder eines Teams müssen sich jeweils in derselben Übungsgruppe befinden. Sollten Sie keinen TeampartnerIn finden, bekommen Sie in der ersten Gruppenübung einen zugeteilt.

Zusätzlich zu den normalen Übungsaufgaben werden Sie ein umfangreiches Softwareprojekt bearbeiten, in welchem Sie die Anpassungswünsche eines Kunden an eine bestehenden Filmdatenverwaltungssoftware („Movie Manager“) umsetzen. Die Bearbeitung des Projektes erfolgt in 4er Teams welche jeweils aus zwei 2er Teams gebildet werden. Die Einteilung der 4er Teams erfolgt jeweils durch den/die Übungsgruppenleiter/in.

Aufgabe 1.4: Einschreibung in die Übungsgruppe

Punkte: Keine	Team: Ja(2)	Projekt
----------------------	--------------------	----------------

Wie oben unter der **Teameinteilung** beschrieben müssen Sie sich einerseits für eine Übungsgruppe entscheiden (Gruppe A und B Termin Dienstag 14.15 Uhr, Gruppe C Termin Donnerstag 14.00 Uhr) und andererseits in 2er Teams aufteilen. Entscheiden Sie sich zunächst für eine der drei Übungsgruppen (A/B/C) und bilden Sie dann entsprechend Teams, in dem Sie sich unter:

<http://elearning2.uni-heidelberg.de/mod/questionnaire/view.php?id=90009>

für die Bildung eines 2er Teams eintragen (Link **Teambildung und Wahl der Übungsgruppen** im Moodle Kurs). Sollten Sie keine Teampartner/in für ein 2er Team finden können, wählen Sie lediglich eine Übungsgruppe und kommen dann zum ersten Übungsgruppen Termin am 15.10. bzw. 17.10. Dort wird dann für alle verbleibenden Gruppenmitglieder die Zuteilung zu einem Team durchgeführt.

Die Zusammenstellung der 4er Teams können Sie dann ab 21.10.2013 unter:

<http://elearning2.uni-heidelberg.de/mod/page/view.php?id=90010>

einsehen (Link **Übersicht Teams und Übungsgruppen** im Moodle Kurs). Gehen Sie am Dienstag, 22.10.2013 bzw. am Donnerstag, 24.10.2013 in die Übung, um die anderen Mitglieder Ihres Teams kennenzulernen.

Ergebnis:

Füllen Sie das Formular zur Teambildung und Übungsgruppenanmeldung bis **(Achtung!) Freitag 17.10.2013 um 18.00 Uhr** in Moodle aus.