

Übungsblatt 2 (22.10.2013)

Einführung Software Projekt, Java, Programmierrichtlinien, Source Code Analyse, Metriken, Pair Programming

In dieser Übung:

- ✓ Beginnen sie die Arbeit am Software Projekt
- ✓ Vertiefen Sie Ihre Kenntnisse über die Programmiersprache Java.
- ✓ Lernen Sie die Java-Programmierrichtlinien kennen.
- ✓ Analysieren Sie Ihren programmierten Java Code.
- ✓ Erheben Sie Metriken auf Ihren programmierten Java Code.
- ✓ Lernen Sie die Prinzipien des Pair-Programming kennen

Hinweise zur Projekt Aufgabe

Die Projektaufgabe dauert das ganze Semester über und ist aufgeteilt in Teilaufgaben, welche Sie als Aufgaben auf den wöchentlichen Übungsblättern im Team bearbeiten. Für die Abgabe ist es ausreichend, wenn eines der Teammitglieder die Bearbeitung abgibt.

Im Rahmen des Projekts unterstützen Sie die fiktive Firma *MoviExperience* als externes Entwicklerteam, um deren Produkt *Movie Manager* zu verbessern. Der *Movie Manager* ist eine Desktop Anwendung zur Verwaltung von Filmdaten. Die Firma *MoviExperience* fungiert dabei als Ihr *Auftraggeber*, welcher die *Anforderungen* für die von Ihnen neu zu entwickelnden Features aufstellt. Diese orientieren sich einerseits an den Wünschen von bestehenden Kunden, andererseits wird damit versucht neue Kunden für den *Movie Manager* zu gewinnen.

Nachfolgende finden sie die wichtigen Meilensteine des ersten Teils des Projektes.

Datum	Übungsblatt	Meilenstein	Beschreibung
29.10.2013	1, 2	Präsentation des Teams	Kennenlernen Movie Manager
12.11.2013	3,4	Präsentation der Anforderungen	Anforderungen für neues Feature
10.12.2013	5,6,7	Präsentation des Entwurfs	Entwurf für neues Feature

Aufgabe 2.1: Einführung Software Projekt

Punkte: 8	Team: Ja(4)	Projekt
-----------	-------------	---------

Zum genauen Verständnis des *Movie Managers* ist es erforderlich, dass Sie die bisherige Funktionalität der Anwendung genau kennen und dokumentieren. Laden Sie sich dafür zunächst die für Ihr System passende Version des Movie Managers unter:

<http://svn.ifi.uni-heidelberg.de/unicase/mm/index.html>

oder direkt über den Link *Download Movie Manager* im Moodle Kurs herunter und installieren Sie diese, indem Sie das ZIP Archiv an einen geeigneten Ort entpacken und über das Ausführen der Datei *MovieManger.exe* (Windows) bzw. *MovieManger* (Linux/Mac) starten.

Erkunden und dokumentieren Sie nun die Funktionen der Movie Manager Anwendung in geeigneter Form. Orientieren Sie sich dabei an nachfolgendem Beispiel.

Bezeichnung	Funktion in Anwendung	Beschreibung
Neues Projekt Anlegen	In der <i>Navigator</i> Ansicht das Kontext Menu öffnen, dann die Option <i>Other >> New Project</i> wählen.	Als Gliederungselement zur Organisation von Filmdaten können in der Navigator Ansicht neue Projekt angelegt werden

Insgesamt sollten Sie mindestens zehn unterschiedliche Funktionen dokumentieren. Überlegen Sie sich zusätzlich zur bestehenden Funktionalität ein weiteres **neues Feature** (Systemverantwortlichkeit, d.h. ggf. mehrere Funktionen) für den Movie Manager. Bei diesem kann es sich entweder um die Erweiterung der bestehenden Funktionalität oder um ein komplett neues Feature handeln. Bei dem neuen Feature sollte es sich um ein Alleinstellungsmerkmal für den Movie Manager gegenüber anderen Filmverwaltungsprogrammen handeln, welches diesen für neue Kunden besonders attraktiv macht. Dokumentieren Sie auch die Funktionen des neuen Features in der oben aufgeführten Form.

Hinweis: Beachten Sie dass es notwendig ist die Scroll Funktionalität in der Detailansicht des Movie Managers zu verwenden um auch die in dieser enthaltenen Buttons auf der rechten Seite zu sehen.

Ergebnis: Speichern Sie bitte eine PDF-Datei mit den dokumentierten (alten und neuen) Movie Manager Funktionen bis **Montag 28.10.2013 um 10.00 Uhr** in Moodle.

Organisation Teampräsentation

In der nachfolgenden Aufgabe erstellen Sie eine Präsentation, in welcher Sie die Mitglieder Ihres Teams und die Movie Manager Anwendung mit Ihrem neuen Feature vorstellen. Die Präsentation zeigen Sie am kommenden Dienstag 29.10.2013 in der Zentralübung.

Bestimmen Sie eine Person, die die Präsentation zeigt. Jedes Teammitglied muss im Verlauf der Vorlesung mindestens einmal präsentieren für das 4er-Team(siehe Meilensteine 2.1.) oder in der 2.Projektphase für das 2er-Team. *Die beste Präsentation wird prämiert!*



Aufgabe 2.2: Vorbereitung Teampräsentation

Punkte: 4	Team: Ja(4)	Projekt
-----------	-------------	---------

Verwenden Sie als Grundlage für die Präsentation Ihre Beschreibung der Movie Manager Funktionalität aus der vorherigen Aufgabe 2.1. Erstellen Sie nun eine Präsentation mit 3-4 Folien, in welcher Sie:

- ihr Team mit einem von Ihnen gewählten Namen sowie die Mitglieder vorstellen,
- die bisherige Funktionalität des Movie Managers zusammenfassend erklären, und
- Ihr neues Feature als besonderes Alleinstellungsmerkmal vorstellen.

Die Dauer der Präsentation sollte bei ca. 3 Minuten liegen. Die beste Präsentation wird prämiert!

Ergebnis: Speichern Sie bitte die Folien Ihrer Präsentation als PDF-Datei bis **Montag 28.10.2013 um 10.00 Uhr** in Moodle und zeigen Sie Ihre Präsentation **im Rahmen der Zentralübung am 29.10.2013 (Pflicht!)**.

Checkstyle:

Das Eclipse Plug-in „Checkstyle“ hilft bei der Programmierung von Java Code, der die Java Code Conventions einhält. Eine Anleitung für dieses Plugin finden Sie unter „Tutorials“ in Moodle.

Aufgabe 2.3: Java-Programmierrichtlinien**Punkte: 4****Team: Nein**

1. Machen Sie sich kurz mit den „Java Code Conventions“ vertraut, die Sie in Moodle finden. Eine Übersicht über die wichtigsten Java Code Conventions mit Beispielen finden Sie hier: <http://www.cs.umbc.edu/courses/undergraduate/202/spring11/projects/coding-standards.shtml>
2. Benutzen Sie das Eclipse-Plugin „Checkstyle“ und wenden Sie es auf Ihren Quellcode aus Aufgabe 1.2 „Java – Anwendung (1)“ an, d.h. formatieren Sie den Quellcode so, dass er den Konventionen entspricht.

Ergebnis:

Bitte speichern Sie den formatierten Quellcode als vollständig exportiertes Eclipse-Projekt als .zip-Datei bis **Montag 28.10.2013 um 10.00 Uhr** in Moodle.

Google CodePro AnalytiX:

Das Eclipse Plug-in „Google CodePro AnalytiX“ analysiert Java Code und hilft potentielle Probleme zu finden, z.B. mögliche Fehler oder „Bad Smells“. Eine Anleitung für dieses Plugin finden Sie unter „Tutorials“ in Moodle.

Aufgabe 2.4: Source Code Analyse & Metriken**Punkte: 9****Team: Nein****Teilaufgabe a) Source Code Analyse:**

Benutzen Sie die Funktion „Audit Code Using ...“ mit dem Regelsatz „Potential Errors and Refactorings“ des Eclipse-Plugin „Google CodePro AnalytiX“ und wenden Sie sie auf Ihren Quellcode aus Aufgabe 1.2 „Java – Anwendung (1)“ an, d.h. ändern Sie den Quellcode so, dass er keine Audit-Regeln verletzt.

Teilaufgabe b) Metriken:

1. Benutzen Sie die Funktion „Compute Metrics“ des Eclipse-Plugin „Google CodePro AnalytiX“ und erheben Sie folgenden Metriken für Ihre Ergebnisse aus Aufgabe 1.2 „Java – Anwendung (1)“:
 - Anzahl der Quelltextzeilen in allen Operationen einer Klasse
 - Durchschnittliche Anzahl Quelltextzeilen pro Operation einer Klasse
 - Durchschnittliche Anzahl Parameter einer Klasse
 - Anzahl Attribute in einer Klasse
 - Anzahl Operationen in einer Klasse

Dokumentieren Sie die Metriken in einer Textdatei. Diese Textdatei soll **für jede von Ihnen neu erstellte Klasse** die oben genannten Metriken beinhalten. **Vergleichen** Sie anschließend

die Metriken Ihrer erstellten Klassen miteinander, d.h. dokumentieren Sie die Gemeinsamkeiten und erklären Sie die Unterschiede in der gleichen Textdatei.

2. Was bewirkt Ihrer Meinung nach die Erhebung von Metriken von Quellcode? Sammeln Sie einige Vor- und Nachteile, die die Erhebung von Metriken hervorbringt.

Ergebnis: Speichern Sie bitte eine PDF-Datei mit den erhobenen Metriken und den Vor- und Nachteilen bis **Montag 28.10.2013 um 10.00 Uhr** in Moodle.

Aufgabe 2.5: Java – Anwendung (2)

Punkte: 4

Team: Nein

Ihre Anwendung „Movie Manager“ aus Blatt 1 soll zusätzlich das Anlegen eines spezialisierten Performers, nämlich eines **Directors**, anbieten. Ein **Director** kann sowohl in einem Film mitspielen als auch die Regie führen.

1. Der **Director** soll als neue Klasse realisiert werden. Der **Director** erbt alle Attribute, Operationen und Assoziationen vom normalen **Performer**, unterscheidet sich aber durch das neue Attribut der Anzahl der **Movies**, bei der er Regie geführt hat.
2. Bei der Erstellung eines **Movies** sollte man gleich einen **Director** zuweisen können.

Ergebnis:

Bitte speichern Sie Ihr exportiertes Eclipse-Projekt in gepackter Form als .zip-Datei bis **Montag 28.10.2013 um 10.00 Uhr** in Moodle.

Aufgabe 2.6: Pair Programming

Punkte: 8

Team: Ja (2)

1. Erweitern Sie den Movie Manager durch **Pair Programming** (2 Personen programmieren gleichzeitig an einem Computer). Protokollieren Sie mit, wann Sie die Tastatur gewechselt haben und wer von Ihnen was getan hat. Der **Observer** fasst kurz zusammen, was der **Driver** in seiner Zeit gemacht hat.
2. Implementieren Sie ein Feature zum Ausleihen von Filmen wie folgt: :
 - a. Legen Sie eine **neue Klasse Customer** an, welche Informationen zu einem Kunden speichert. Ein **Customer** soll einen Namen (Typ **String**) und eine Liste ausgeliehener Filme haben.
 - b. Die Klasse **Movie** soll ein **zusätzliches Attribut customer** (Typ **Customer**) bekommen, welches speichert, von welchem Kunden der Film ausgeliehen wurde.
 - c. Erweitern Sie die Klasse **Customer** um eine Operation **loanMovie()**, welche als Übergabeparameter einen auszuleihenden Film erhält. In der Implementierung der Operation muss sichergestellt werden, dass der Film nicht schon von einem anderen Kunden ausgeliehen ist. Nur dann kann der Film auch tatsächlich vom Kunden ausgeliehen werden, sonst soll eine entsprechende Fehlermeldung erzeugt werden.
 - d. Erweitern sie die Operation **loanMovie()** so, dass ein Kunde **nur 5 Filme gleichzeitig** ausleihen darf. Implementieren Sie eine entsprechende Meldung, die warnt, wenn versucht wird mehr als 5 Filme auszuleihen.
3. Schreiben sie ein Hauptprogramm mit **main()** –Operation, in welchem Sie das neue Feature ausführlich testen. Hierbei müssen die beiden Fehlermeldungen („Film ist schon an Kundin <Name der Kundin> verliehen“, „Kunde <Name des Kunden> hat schon 5 Filme ausgeliehen“) mindestens einmal erzeugt werden.
4. Beschreiben Sie Ihre Erfahrungen, die Sie während des Pair Programming gemacht haben.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am Quellcode auf dem SVN Server, in dem Sie das Eclipse Projekt per Share Project Funktion und anschließendem Commit zum SVN Server übertragen! Speichern Sie das Projekt im **trunk** ihres SVN Repositories.

Speichern Sie das Ergebnis als.zip-Datei bis Montag 28.10.2013 um 10.00 Uhr in Moodle. Bestehend aus:

- Exportiertes Eclipse-Projekt mit Customer Implementierung
- 1x PDF mit Beschreibung des Ablaufes des Pair Programming einschließlich Erfahrungsbericht

Es ist ausreichend, wenn eines der beiden Teammitglieder die Ergebnisse in Moodle abgibt.

Aufgabe 2.7: Vorbereitung: Kommunikation mit den Kunden		
Punkte: 4	Team: Ja(4)	Projekt

Formulieren Sie Anforderungen für die *Movie Manger* Anwendung aus der Sicht der NutzerInnen im ersten Kundengespräch. Beschreiben Sie hierfür allgemein, wie Anforderungen formuliert sein müssen, damit sie auch für die NutzerInnen verständlich sind und geben Sie als Beispiel an, wie die Kundenanforderungen für das neue Feature aus Aufgabe 2.1 ausgesehen hätten (bevor dieses zu den einzelnen Funktionen konkretisiert wurde).

Ergebnis: Bitte speichern Sie bitte eine PDF-Datei mit der Anforderungsformulierung aus Sicht der NutzerInnen und der Beschreibung des neuen Features bis **Montag 28.10.2013 um 10.00 Uhr** in Moodle.