Übungen zu „Einführung in Software Engineering“ (WS 18/19)
Prof. Dr. Barbara Paech, Astrid Rohmann, Marcus Seiler
http://se.ifi.uni-heidelberg.de

Übungsblatt 12 (15.01.2019)

Sequenzdiagramm und OOAD, Implementierung und Qualitätssicherung der neuen Funktionalität, Wiederholungsfragen zu Kapitel 5

In dieser Übung:

- ✓ Erstellen Sie mit OOAD einen Entwurf für eine neue Funktionalität im *Movie Manager* (Entwurfsklassendiagramm und Sequenzdiagramm).
- ✓ Beginnen Sie mit der Implementierung und Qualitätssicherung der neuen Funktionalität des *Movie Managers*.
- ✓ Beantworten Sie Wiederholungsfragen zum Vorlesungskapitel 5.

Aufgabe 12.1: Sequenz- und Entwurfsklassendiagramm für die neue Funktionalität

Präsenz: Nein	**Punkte:** 16	**Team: Ja**	**Testat**

In dieser Aufgabe fertigen Sie mit Hilfe ihres Analyseklassendiagramms aus Aufgabe 11.2 und der bestehenden Implementierung des *Movie Managers* einen Entwurf zur Umsetzung und Integration der Ausleih-Funktionalität an. Hierfür ist eine Integration der User Interface (UI) Klassen in die bestehende UI (Packages ***moviemananger.ui****), die Erweiterung der Steuerungsschicht (Package ***moviemananger.util)*** und die Erweiterung des Datenmodells (Package ***moviemananger.data***) zu modellieren. Nutzen Sie dabei Ihr Wissen aus den Tutorials zur SWT-, Dialog- und Context Menu-Programmierung, sowie Ihre Erfahrungen von Übungsblatt 7. Die Erstellung des Sequenzdiagramms und des Entwurfsklassendiagramms sind im nachfolgenden hintereinander beschrieben, können aber auch parallel durchgeführt werden.

1) Die Vorlage *12-MovieManager-SequenceDiagram.png* (Datei zu finden in Moodle) enthält das noch unvollständige Sequenzdiagramm der Systemfunktion *lendMovie()*, das den Ablauf der Systemfunktion modelliert. Überlegen Sie sich den weiteren Ablauf der Systemfunktion und ergänzen Sie das Sequenzdiagramm. Wenn die Vorlage nicht zu Ihrem bisherigen Entwurf passt, erstellen Sie ein komplett neues Sequenzdiagramm. Sie können das Sequenzdiagramm entweder per Hand ergänzen/erstellen und dann einscannen oder Sie verwenden ein Zeichenwerkzeug ihrer Wahl, z.B. Paint, Visio o.ä. Orientieren Sie sich bei der Erstellung des Sequenzdiagramms an den vorhandenen Sequenzdiagrammen im Jira-Projekt.
2) Ihr erstelltes Sequenzdiagramm wird sowohl Klassen der Modellschicht als auch der Oberflächenschicht enthalten. Ordnen Sie die Klassen des Sequenzdiagramms den jeweiligen Schichten zu. Färben Sie hierzu die Klassen innerhalb des Sequenzdiagramms **grün** für Modellklassen, **rot** für Oberflächenklassen und **blau** für Klassen des SWT-Frameworks. Welches sind Aufrufe von Oberflächen-Klassen auf Modellschicht-Klassen? Was bewirken diese Aufrufe? Listen Sie die Aufrufe in einem PDF-Dokument auf und beschreiben Sie diese kurz.
3) Erstellen Sie passend zu dem in Aufgabe 12.1.1 zu modellierenden Ablauf im Sequenzdiagramm sowie auf Basis des Analyseklassendiagramms aus Aufgabe 11.2 ein

Entwurfsklassendiagramm, das die Umsetzung der Systemfunktion *lendMovie()* modelliert. Verwenden Sie dazu entweder die Vorlage *12-MovieManager-ClassDiagram.png* (Datei zu finden in Moodle) oder Ihr ergänztes Klassendiagramm des *Movie Managers* aus Aufgabe 7.1. Überlegen Sie, durch welche (vorhandenen oder neuen) Klassen und Operationen die Analyseklassen und ihre Operationen und Assoziationen verfeinert werden. Beschreiben Sie Ihre Überlegungen in einem PDF-Dokument und begründen Sie diese kurz.

4) Überlegen Sie sich analog zu der Systemfunktion *lendMovie()* die Abläufe für die verbleibenden Systemfunktionen *receiveReturnedMovie()*, *changeDueDate()*, *showOverdueMovies()* und *showLentMovies()*. Vervollständigen Sie auf Basis dieser Abläufe das Entwurfsklassendiagramm aus Aufgabe 12.1.3 und beschreiben und begründen Sie wieder Ihre Überlegungen.

Ergebnis:
Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 21.01.2019 um 10.00 Uhr** in Moodle bestehend aus:

- 1x PNG-Datei des eingefärbten Sequenzdiagramms
- 1x PDF-Dokument mit der Beschreibung der Schicht-übergreifenden Kommunikation
- 1x PNG-Datei des Entwurfsklassendiagramms
- 1x PDF-Dokument mit der Beschreibung und Begründung zur Verfeinerung der Klassen, Operationen und Assoziationen

Aufgabe 12.2: Implementierung und Qualitätssicherung der neuen Funktionalität (1)

Präsenz: Nein	**Punkte:** 13	**Team: Ja**	**Testat**

Ziel dieser Aufgabe und der Aufgabe 13.1 auf dem nächsten Übungsblatt ist die Implementierung der neuen Funktionalität im *Movie Manager* einschließlich der Qualitätssicherung mittels Komponenten- und Systemtests sowie statischer Analyse. Nutzen Sie für die Implementierung und Qualitätssicherung Ihr Wissen aus den Tutorials und den bisherigen Übungsblättern. **Für dieses Übungsblatt genügt die Implementierung und Qualitätssicherung der Systemfunktionen *Lend Movie*, *Change Due Date* und *Receive Returned Movie*.** Gehen Sie wie folgt vor:

1) Implementieren Sie die neue Funktionalität auf Basis des von Ihnen erstellten Entwurfs, d.h. des in Aufgabe 11.1 erstellten Dialogmodells und des in Aufgabe 12.1 erstellten Entwurfsklassendiagramms und Sequenzdiagramms. Wichtig ist, dass Ihre Entwürfe konsistent mit der Implementierung sind. Sollten Sie während der Implementierung von Ihren ursprünglichen Entwürfen abweichen, so dokumentieren Sie diese Abweichung in einem PDF-Dokument und passen Ihre Entwürfe entsprechend an.

   Hinweis: Für die Umsetzung der Funktionen Lend Movie, Change Due Date und Receive Returned Movie bietet es sich an, ein Widget analog zum `WatchDateWidget` *zu implementieren.*

2) Entwerfen und Implementieren Sie für die neuen Operationen **in den Modellklassen** entsprechende JUnit-Tests, wie auf Übungsblatt 9, die alle wichtigen Fälle (insbesondere auch Ausnahmen) abdecken. Dokumentieren Sie Ihre logischen und konkreten Komponententestfälle in der Tabelle *12-MovieManager-ComponentTestcases.xlsx* (Datei zu finden in Moodle). Orientieren Sie sich bei der Implementierung an den vorhandenen JUnit-Testfällen im Ordner `test`. Führen Sie die JUnit-Tests unter Verwendung des Werkzeugs EclEmma aus. Überlegen Sie auf Basis der Ausgabe von EclEmma, welche weiteren Komponententestfälle für eine 100%ige Anweisungsüberdeckung der neuen Operationen in den Modellklassen nötig sind. Implementieren und dokumentieren Sie die fehlenden

Komponententests. Führen Sie alle Tests nochmal aus und dokumentieren Sie alle gefundenen Fehler in einem PDF-Dokument. Verwenden Sie zur Fehlerdokumentation eine Tabelle wie in Aufgabe 11.3. Beheben Sie alle gefundenen Fehler.

3) Führen Sie Ihre in Aufgabe 11.1 spezifizierten Systemtestfälle zu den implementierten Systemfunktionen aus. Dokumentieren Sie alle gefundenen Fehler in einem PDF-Dokument. Verwenden Sie zur Fehlerdokumentation eine Tabelle wie in Aufgabe 11.3. Beheben Sie alle gefundenen Fehler.

4) Führen Sie eine statische Analyse durch, indem Sie die Werkzeuge Checkstyle und SpotBugs auf Ihren Quellcode anwenden. Formatieren Sie den Quellcode so, dass er den Java-Konventionen entspricht und ändern Sie dann Ihren Quellcode so, dass er keine SpotBugs Regeln verletzt.

Ergebnis:
Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 21.01.2019 um 10.00 Uhr** in Moodle bestehend aus:

- 1x Eclipse-Projekt 12-MovieManager.zip mit Implementierung der geforderten Systemfunktionen der neuen Funktionalität, JUnit-Testfällen sowie entsprechenden Fehlerbehebungen
- 1x PDF-Datei mit den gefundenen Fehlern aus Komponenten- und Systemtests sowie der statischen Analyse
- 1x Excel-Datei 12-MovieManager-ComponentTestcases.xlsx mit logischen und konkreten Komponententestfällen
- Ggf. 1x PDF-Datei mit der Beschreibung der Abweichungen vom Entwurf

Aufgabe 12.3: Wiederholung zu Kapitel 5

Präsenz: Nein	**Punkte:** 0	**Team:** Nein	

Diese Aufgabe dient der Wiederholung des Kapitels 5 der Vorlesung. Beantworten Sie in einem PDF-Dokument die folgenden Fragen anhand der Informationen aus dem Kapitel:

- Was ist Qualität?
- Was ist ein Fehler, was ein Mangel?
- Unterscheiden Sie die 3 Arten von Fehlern und geben Sie jeweils eine passende Fehleraufdeckungstechnik an.
- Was ist Qualitätsmanagement, was Qualitätssicherung?
- Unterscheiden Sie die 3 Arten von Qualitätssicherung. Wie ergänzen sich diese? Geben Sie jeweils Beispiele an.
- Erklären Sie wichtige Testbegriffe
- Was sind die Vor- und Nachteile des Testens? Warum ist vollständiger Test nicht möglich?
- Was beinhaltet eine Testspezifikation?
- Was ist ein logischer bzw. konkreter Testfall?
- Was ist der Unterschied zwischen Black-Box- und White-Box-Test?
- Welche typischen Fehler hat eine Komponente?
- Wie werden Black-Box-Testfälle systematisch erstellt?
- Was ist Äquivalenzklassentest? Was ist hier abzudecken? Wie kann man das vereinfachen?
- Was ist ein Kontrollflussgraph?
- Welche Überdeckungsmöglichkeiten gibt es beim White-Box-Test? Welche Probleme gibt es dabei?

- Was ist Grey-Box-Test?
- Was ist zustandsbezogener Test?
- Was ist beim Klassentest zu beachten?
- Wie hängen Komponenten bzw. Klassen voneinander ab?
- Was ist Integrationstest? Welche Strategien gibt es dafür? Welche typischen Probleme deckt er auf?
- Was ist Systemtest? Wie leitet man Systemtestfälle ab?
- Vergleichen Sie Komponenten-,Integrations-, und Systemtest bzgl. zu entdeckender Fehler, Testrahmen, Teststrategie und typischer Fehler
- Erklären Sie wichtige Begriffe der Testdurchführung, z.B. Testrahmen und Testendekriterium.
- Was ist statische Analyse?
- Welche Fehlerzustände können dadurch entdeckt werden? Geben Sie Beispiele an.
- Welche Vorteile hat die statische Analyse?
- Welche Metriken werden im Software Engineering verwendet? Welche grundlegenden Arten gibt es? Geben Sie Beispiele an.
- Welche Vorteile hat die Verwendung von Metriken?
- Was ist der Unterschied zwischen statischer Analyse und Review/Inspektion?
- Welche Arten der Inspektion gibt es?
- Wie läuft der technische Review ab? Welche Rollen sind beteiligt? Welche wichtigen Regeln gibt es? Was sind typische Probleme und Abhilfen?
- Was ist eine Lesetechnik? Welche Arten gibt es?
- Was ist perspektiven-basierte Inspektion und warum ist sie wichtig?
- Wozu sollten Reviewergebnisse verwendet werden?
- Was umfasst die analytische Qualitätssicherung bei SWE im Großen?
- Welche Arten von Kosten bei der Softwareentwicklung, insbesondere Fehlerkosten gibt es?
- Geben Sie 2 wichtige Grundsätze bei der analytischen QS an.
- Welche Arten von Prüfergebnissen gibt es? Welche sollten vermieden werden?
- Was ist Fehlermanagement?
- Wie sieht ein Schema für Fehlermeldungen aus?
- Was ist der Unterschied zwischen Fehlerklassen und Fehlerpriorität? Geben Sie jeweils Beispiele an.
- Was ist ein Issue Tracker?
- Was ist Debugging?
- Was ist Konfigurationsmanagement? Welche Begriffe und Techniken sind wichtig?
- Welchen Beitrag leistet es zu den Kernfragen des SWE?

Ergebnis:

Speichern Sie bitte das PDF-Dokument mit den Antworten zu den Fragen bis **Montag 21.01.2019 um 10.00 Uhr** in Moodle