

Übungsblatt 12 (21.01.2020)

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: 10

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 **datenändernden Ausleih-Funktionen** an. Nutzen Sie dabei Ihr Wissen aus den Tutorials zur Android-, Dialog- und Menu-Programmierung, sowie Ihre Erfahrungen von Übungsblatt 7 und Arbeitsblatt 12. Die Anpassung des Sequenzdiagramms und des Entwurfsklassendiagramms sind im nachfolgenden hintereinander beschrieben, können aber auch parallel durchgeführt werden.

Hinweise:

- Für die Umsetzung der Funktionen *Lend Movie*, *Change Due Date* und *Receive Returned Movie* bietet es sich an, sich an der Implementierung von *Watch Movie* zu orientieren und ein Attribut für das Rückgabedatum in der Klasse *Movie* einzuführen, dessen getter und setter der Umsetzung der oben genannten Systemfunktionen dienen.
 - Die folgenden Unteraufgaben beschreiben den Weg für diese Umsetzung. Sie können auch eine andere Umsetzung wählen. Beschreiben Sie dann, wie das Sequenzdiagramm für diese Umsetzung aussehen müsste oder modellieren Sie dieses. Modellieren Sie außerdem das dazu passende Entwurfsklassendiagramm.
 - Eine mögliche Lösung für das erweiterte Analyseklassendiagramm aus Aufgabe 11.2. finden Sie in der Datei *12-MovieManager-AnalysisDiagLend.png* in Moodle. Sie können im Folgenden diese Lösung als Ausgangsbasis verwenden oder Ihre eigene Lösung.
- 1) Die Vorlage *12-MovieManager-SequenceDiagram.png* (Datei zu finden in Moodle) enthält das Sequenzdiagramm der Systemfunktion *Watch Movie* (siehe auch Probeklausur). Betrachten Sie den dazu passenden Code für das Setzen des *WatchDate*-Attributes. Das Sequenzdiagramm zeigt die relevanten Klassen und Objekte. Überlegen Sie, welche Anpassungen des Ablaufs oder der Klassen für die Systemfunktion *Lend Movie* nötig sind und beschreiben Sie diese in

einem PDF-Dokument. Die Klassen für die Umsetzung der Speicherung sind nicht modelliert und sind auch nicht zu ändern, bis auf die Klassen `MovieTransformations` und `MovieFromJsonObject`.

- 2) Erstellen Sie passend zu dem in Aufgabe 12.1.1 gegebenen Sequenzdiagramm und Ihren Überlegungen ein **Entwurfsklassendiagramm**, das an der Umsetzung der Systemfunktion *Lend Movie* beteiligten Klassen. Ergänzen Sie die benötigten Beziehungen. Färben Sie die Klassen wie im Sequenzdiagramm ein: **grün** für Datenklassen, **rot** für Oberflächenklassen, **blau** für Klassen des Android-Frameworks und der Java Standardbibliothek und **gelb** für Hilfsklassen des Movie-Manager-Frameworks. Sie können das Entwurfsklassendiagramm entweder per Hand erstellen und dann einscannen oder Sie verwenden ein Zeichenwerkzeug ihrer Wahl, z.B. Paint, Visio, draw.io. Überlegen Sie, durch welche Klassen und Operationen die Analyseklassen aus Aufgabe 11.2. und deren Operationen und Assoziationen verfeinert werden. Beschreiben Sie Ihre Überlegungen in einem PDF-Dokument und begründen Sie diese kurz.
- 3) Überlegen Sie sich analog zu der Systemfunktion *Lend Movie* die Abläufe für die verbleibenden Systemfunktionen *Receive Returned Movie* und *Change Due Date*. Beschreiben Sie, was sich beim Ablauf dieser Funktionen im Vergleich zu *Lend Movie* ändert.

Ergebnis:

Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 27.01.2020 um 10.00 Uhr** in Moodle bestehend aus:

- 1x PDF-Dokument mit den beschriebenen Anpassungen des Ablaufs und der Klassen
- 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: 18	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 sind die Systemfunktionen *Lend Movie*, *Change Due Date* und *Receive Returned Movie* zu implementieren. 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.
- 2) Entwerfen und Implementieren Sie für die neuen Operationen **in der Movie-Klasse** 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 aus. Beheben Sie alle gefundenen Fehler.

- 3) Führen Sie Ihre in Aufgabe 11.1 spezifizierten Systemtestfälle zu den von Ihnen 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 das Werkzeug SonarLint 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 SonarLint Regeln verletzt.

Ergebnis:

Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 27.01.2020 um 10.00 Uhr** in Moodle bestehend aus:

- 1x Android Studio-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

*Hinweis: In der Abgabe Ihres Codes können Sie zur Reduzierung der Speichergröße den Ordner **build** (Unterordner von app) löschen, da dieser aus den Dateien in src erneut erzeugt werden kann.*

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 27.01.2020 um 10.00 Uhr** in Moodle