

## Übungsblatt 13 (28.01.2020)

### Implementierung und Qualitätssicherung der neuen Funktionalität, Wiederverwendung und Weiterentwicklung

In dieser Übung:

- ✓ Implementieren und sichern Sie die Qualität der neuen Funktionalität des *Movie Manager*.
- ✓ Beurteilen Sie Wiederverwendung durch Frameworks.
- ✓ Bewerten Sie den Aufwand für eine Weiterentwicklung des *Movie Managers*.

#### Aufgabe 13.1: Implementierung und Qualitätssicherung der neuen Funktionalität (2)

Präsenz: Ja	Punkte: 16	Team: Ja	Testat
-------------	------------	----------	--------

Ziel dieser Aufgabe ist, aufbauend auf Blatt 11 und 12 die Implementierung der neuen Funktionalität im *Movie Manager* einschließlich der Qualitätssicherung mittels Komponenten- und Systemtests sowie statischer Analyse fertig zu stellen. Auf Blatt 12 haben Sie die Anforderungen 1-3 von Blatt 10 implementiert. Implementieren Sie nun die verbleibenden Anforderungen 4-7. Nutzen Sie für die Implementierung und Qualitätssicherung Ihr Wissen aus den Tutorials und den bisherigen Übungsblättern, insbesondere das Wissen von Übungs- und Arbeitsblatt 7 für die Sortierung der Filme. Gehen Sie dabei wie folgt vor:

- 1) Implementieren Sie die neue Funktionalität auf Basis der von Ihnen erstellten Anforderungsdokumentation der Aufgaben 10.3 und 11.1. Beschreiben Sie textuell die nötigen Änderungen für Ihre Implementierung. Fassen Sie die neuen oder geänderten Klassen anhand eines Entwurfsklassendiagramms zusammen und zeichnen Sie die wichtigsten Beziehungen ein. Ergänzen Sie ggf. für die Beziehungen wichtige existierende Klassen.
- 2) Entwerfen und Implementieren Sie für neue 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 *13-MovieManager-ComponentTestcases.xlsx* (Datei zu finden in Moodle). Orientieren Sie sich dabei 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 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 03.02.2020 um 10.00 Uhr** in Moodle bestehend aus:

- 1x Android Studio-Projekt 13-MovieManager.zip mit Implementierung der Ausleihfunktionalität, JUnit-Testfällen sowie entsprechenden Fehlerbehebungen
- 1x PDF-Datei mit den gefundenen Fehlern aus Komponenten- und Systemtests sowie der statischen Analyse
- 1x Excel-Datei 13-MovieManager-ComponentTestcases.xlsx mit logischen und konkreten Komponententestfällen
- 1x PDF oder PNG-Datei mit der Beschreibung Ihrer Implementierung anhand eines Entwurfsklassendiagramms

*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 13.2: Erhebung und Analyse von Metriken nach der Implementierung der neuen Funktionalität

Präsenz: Ja	Punkte: 3	Team: Ja	Testat
-------------	-----------	----------	--------

Die nachfolgenden Tabelle zeigt Metriken zu einem Teil des ausgegebenen *Movie Manager* Codes *MovieManager.zip* getrennt nach Paketen. Verwenden Sie das Werkzeug SonarQube und erheben Sie analog zu den gegebenen Werten die Metriken für Ihren *Movie Manager* Code nach der Implementierung der neuen Funktionalität. Vergleichen Sie anschließend Ihre Werte mit denen des ausgegebenen Codes. Dokumentieren Sie Gemeinsamkeiten und erklären Sie Unterschiede, indem Sie Auffälligkeiten beschreiben (also z.B. warum sind bei Ihnen im Paket X viel mehr Zeilen/Operationen/Klassen hinzugekommen als im Paket Y).

Metrik	de.moviemanager	de.moviemanager.core	de.moviemanager.data	de.moviemanager.ui	de.moviemanager.ui.adapter	de.moviemanager.ui.detail	de.moviemanager.ui.masterfragments	de.moviemanager.ui.masterlist	de.moviemanager.ui.view
Lines of Code	11373	1340	765	8337	1343	2480	613	1027	531
Functions	1344	181	123	947	149	301	53	136	69
Classes	157	16	12	109	21	26	8	23	6
Cyclomatic Complexity	1930	240	132	1381	208	421	95	174	101
Code Smells	27	0	0	24	0	10	0	0	0

**Ergebnis:**

Speichern Sie bitte Ihr PDF-Dokument mit den erhobenen Metriken sowie der Beschreibung von Gemeinsamkeiten und Unterschieden bis **Montag 03.02.2020 um 10.00 Uhr** in Moodle.

### Aufgabe 13.3: Wiederverwendung – Frameworks der Movie Manager Anwendung

Präsenz: Ja	Punkte: 4	Team: Ja	
-------------	-----------	----------	--

Beim Umsetzen und Testen von neuen Funktionalitäten im *Movie Manager* haben Sie für die Programmierung des GUI mit *Android* sowie für die Programmierung von Komponententests mit *JUnit* gearbeitet.

- 1) Beschreiben Sie in einem PDF-Dokument für jedes der beiden Frameworks (Android und JUnit), wie es bei der Umsetzung des *Movie Managers* genutzt wurde.
- 2) Ordnen Sie die Frameworks jeweils in die in der Vorlesung (siehe Folie 11 und 12 im Foliensatz 13) vorgestellte *Klassifikation nach Einsatzart* und *Klassifikation nach Architektur* ein. Begründen Sie jeweils Ihre Zuordnung zu einer bestimmten Klassifikation.
- 3) Bewerten Sie die Vor- und Nachteile des Einsatzes der Frameworks in Bezug auf die folgenden Kriterien und begründen Sie kurz Ihre Bewertung.
  - Einsatz des Frameworks in kleinen Projekten (Teamgröße max. 4 EntwicklerInnen)
  - Einsatz des Frameworks auf unterschiedlichen Plattformen
  - Einsatz des Frameworks zur Trennung von GUI und Daten

#### Ergebnis:

Speichern Sie bitte Ihr PDF-Dokument bis **Montag, den 03.02.2020, um 10.00 Uhr** in Moodle.

### Aufgabe 13.4: Weiterentwicklung Movie Manager – Diskussion

Präsenz: Ja	Punkte: 5	Team: Ja	
-------------	-----------	----------	--

Der *Movie Manager* soll um die Funktionalität zur **Verwaltung von Auszeichnungen (Awards) für Filme und SchauspielerInnen** erweitert werden. Zu einem Award werden jeweils der Name der Auszeichnung z.B. „Oscar bester Film/beste SchauspielerIn“ und das Jahr, in dem die Auszeichnung vergeben wurde, gespeichert. Die Verwaltung von Auszeichnungen umfasst jeweils das Anlegen, Ändern und Löschen von Awards für Filme und SchauspielerInnen sowie das Anzeigen welche Filme und SchauspielerInnen welche Awards verliehen bekommen haben.

Im Nachfolgenden sollen nun die Vor- und Nachteile für unterschiedliche Umsetzungsmöglichkeiten dieser neuen Funktionalität auf GUI- und Datenebene diskutiert werden.

- Auf GUI-Ebene kann die Funktionalität entweder über **ein Attribut mit zusätzlichem Dialog-Fenster** oder über **einen neuen Tab** ergänzend zu den beiden vorhandenen Tabs Movies und Performers umgesetzt werden. Sowohl der Dialog als auch der Tab ermöglichen dabei das Anlegen, Ändern, Löschen und Anzeigen von Awards.
- Auf Datenebene kann die Funktionalität als **Erweiterung der bestehenden Datenklassen** oder als **neue Datenklasse** umgesetzt werden.

Bewerten Sie die zuvor beschriebenen Lösungsvorschläge für die GUI- und Datenebene, indem Sie die Vorschläge jeweils anhand der Kriterien der nachfolgenden Tabellen abschätzen. Begründen Sie Ihre Abschätzungen für die Kriterien in einem PDF-Dokument. Entscheiden Sie sich jeweils für eine Lösung auf der GUI- bzw. Datenebene und begründen Sie Ihre Wahl anhand der Kriterien.

**Kriterien zur Bewertung der Lösungsvorschläge auf GUI-Ebene:**

	Anzahl neu zu erstellender UI-Klassen	Anzahl zu ändernder UI-Klassen	Gebrauchstauglichkeit in Bezug auf den Aufwand für die NutzerInnen, um die Funktion auszuführen	Wartbarkeit in Bezug auf die nachträgliche Änderung der Award-Funktionalität
Dialog				
Tab				

**Kriterien zur Bewertung der Lösungsvorschläge auf Datenebene:**

	Anzahl neuer Datenklassen	Anzahl zu ändernder Datenklassen	Kosten in Bezug auf den Aufwand zur Integration der Daten in die GUI	Wartbarkeit in Bezug auf die nachträgliche Änderung der Award-Funktionalität
Erweiterung bestehender Datenklassen				
Verwendung neuer Datenklassen				

**Ergebnis:**

Speichern Sie bitte Ihr PDF-Dokument bis **Montag, den 03.02.2020, um 10.00 Uhr** in Moodle.