Übungsblatt 11 (14.01.2020)
Anforderungen, OOAD, Code-Inspektion, Metriken

In dieser Übung:
- ✓ Dokumentieren Sie weitere Anforderungen für die neue Funktionalität im *Movie Manager*.
- ✓ Erstellen Sie mit OOAD einen Entwurf für eine neue Funktionalität im *Movie Manager* (Analyseklassendiagramm).
- ✓ Führen Sie eine Code-Inspektion durch.
- ✓ Berechnen Sie Metriken für den Java Code des *Movie Managers*.

Aufgabe 11.1: Anforderungen neue Funktionalität (2)

Präsenz: Nein	**Punkte:** 10	**Team: Ja**	**Testat**

In dieser Aufgabe überlegen Sie, wie Sie die erweiterten Workspaces aus Aufgabe 10.3 durch konkrete Sichten in der GUI umsetzen und ergänzen dazu passend das Dialogmodell. Außerdem erstellen Sie logische und konkrete Systemtestfälle für Ihre neuen Systemfunktionen. Gehen Sie dabei wie folgt vor:

1) Überarbeiten Sie die Ergebnisse von 10.3. anhand der Rückmeldung Ihrer TutorIn.

2) Überlegen Sie, wie Sie die erweiterten Workspaces durch konkrete Sichten in der GUI umsetzen. Dokumentieren Sie diese Sichten als Virtual Windows. Falls Sie eine vorhandene Sicht ändern, fertigen Sie zunächst Screenshots der von Ihren Änderungen betroffenen Sichten an und ergänzen diese mit den neuen Elementen für Ihr Feature. Speichern Sie die Screenshots mit Ihren Anpassungen als PDF- oder PNG-Datei. Falls Sie neue Workspaces und damit neue Sichten verwenden, zeichnen Sie die Sichten von Hand und scannen sie dann ein oder verwenden ein Zeichenwerkzeug Ihrer Wahl, z.B. Paint, Visio, draw.io. Erklären Sie in einem PDF-Dokument, wo welche neue Systemfunktion aufgerufen wird, bzw. welche Änderungen an welcher Sicht die neuen Systemfunktionen bewirken (auch wenn sie ohne Benutzeraktion ausgeführt werden).

3) Zur Verfeinerung Ihrer UI (beschrieben durch UI-Struktur und Virtual Windows) erweitern Sie in dieser Aufgabe das Dialogmodell. Das Zustandsdiagramm in der Datei *11-MovieManager-Dialogmodel.png* (Datei zu finden in Moodle) zeigt das Dialogmodell für die Funktionalität des bisherigen Movie Managers (Arbeitsbereich oder Sicht als Zustände, Benutzeraktionen und Systemfunktionen als Transitionen).

   Ergänzen Sie das Dialogmodell um die neue Funktionalität des Movie Managers. D.h. fügen Sie Ihre Arbeitsbereiche bzw. Sichten, Benutzeraktionen und die Systemfunktionen in das Diagramm ein, sodass die Dialoge bei der Durchführung der neuen Funktionalität beschrieben sind. Fügen Sie insbesondere auch Übergänge zwischen existierenden Zuständen und Ihren neuen Zuständen ein. Bei der Erweiterung des Dialogmodells ist es besonders wichtig, dass dieses eine Verfeinerung Ihrer UI-Struktur ist. D.h. die Navigationslinks zwischen den (Unter-) Arbeitsbereichen müssen im Dialogmodell deutlich erkennbar sein.

4) Überlegen Sie, welche wichtigen Systemtestfälle Sie für Ihre neuen und geänderten Systemfunktionen aus Aufgabe 10.3 benötigen. Beschreiben Sie die logischen und konkreten

Systemtestfälle in der Tabelle *11-MovieManager-Systemtests.xlsx* (Datei zu finden in Moodle). Beachten Sie dabei, dass die logischen Testfälle möglichst keine konkreten UI-Details enthalten im Gegensatz zu den konkreten Testfällen. Orientieren Sie sich bei der Spezifikation Ihrer Systemtestfälle an den bereits vorhandenen Systemtestfällen im Jira-Projekt des Movie Managers.

Ergebnis:
Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 20.01.2020 um 10.00 Uhr** in Moodle bestehend aus:

- PNG- oder PDF-Dateien mit den Sichten als Virtual Windows
- 1x PDF-Dokument mit Ihren Erklärungen zu dem Zusammenhang von Systemfunktionen und Sichten
- 1x PNG-Datei 11-MovieManager-Dialogmodel.png mit Ihrem ergänzten Dialogmodell
- 1x Excel-Datei 11-MovieManager-Systemtests.xlsx mit Ihren Systemtestfällen

Aufgabe 11.2: Analyseklassendiagramm neue Funktionalität

Präsenz: Ja	**Punkte:** 8	**Team: Ja**	**Testat**

In dieser Aufgabe fertigen Sie einen Entwurf für die neue Funktionalität an**.** Dazu erstellen Sie mit OOAD ein **Analyseklassendiagramm**. Sie können das Analyseklassendiagramm entweder per Hand erstellen und dann einscannen oder Sie verwenden ein Zeichenwerkzeug ihrer Wahl, z.B. Paint, Visio, draw.io. Da die Icons für das erste Analyseklassendiagramm in den Werkzeugen oft nicht verfügbar sind, können Sie auch Klassensymbole mit Stereotypen verwenden. Gehen Sie dabei nach den Schritten und Regeln aus der Vorlesung vor (siehe Folien 31ff. im Foliensatz 7 der Vorlesung). Beachten Sie dabei auch das Beispiel aus der Vorlesung und das Arbeitsblatt aus der Technologievorlesung.

1) Gehen Sie von den in Aufgabe 10.3. und 11.1. beschriebenen Anforderungen aus und erstellen Sie zunächst ein erstes Analyseklassendiagramm ohne Attribute und Operationen mit den Entitäts-, Steuerungs-, und Dialogklassen der neuen Funktionalität, so dass alle **neuen** Systemfunktionen, **die Daten ändern**, abgedeckt sind. Begründen Sie, warum Sie welche Klassen und Beziehungen im Diagramm darstellen.
2) Erstellen Sie nun auf Basis des ersten Analyseklassendiagramms ein verfeinertes Analyseklassendiagramm, indem Sie alle für die neuen Funktionen relevanten Attribute und Operationen ergänzen und kurz begründen. Ergänzen Sie ggf. auch neue Attribute oder Operationen, die für die Anpassung bereits bestehender Funktionen nötig sind. Die Operationen entsprechen Interaktionen auf dem UI (z.B. SelectMovie) oder Systemfunktionen. Überlegen Sie dabei eine sinnvolle Verteilung der Kontrollklassen auf die anderen Klassen und begründen Sie die Verteilung und daraus folgende Änderungen bei den Assoziationen kurz in einem PDF-Dokument.

Ergebnis:
Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 20.01.2020 um 10.00 Uhr** in Moodle bestehend aus:

- 1x PNG-Datei des ersten Analyseklassendiagramms mit Entitäts-, Steuerungs-, und Dialogklassen
- 1x PNG-Datei des verfeinerten Analyseklassendiagramms mit Entitäts-, Steuerungs-, und Dialogklassen sowie Attributen und Operationen
- 1x PDF-Dokument mit Begründung zur Darstellung der Klassen im ersten Analyseklassendiagramm und zu der Verteilung der Kontrollklassen im verfeinerten Analyseklassendiagramm

Vorbereitung für Aufgabe 11.3 Code-Inspektion
In der nachfolgenden Aufgabe 11.3 prüfen Sie die „Movie Manager“ App durch Inspektion auf Fehler. Dazu benötigen Sie Quelltext, der als Inspektionsobjekt dient. Laden Sie sich das Projekt **11-MovieManager.zip** aus Confluence herunter und öffnen Sie das Projekt in Android Studio. Schauen Sie sich die Klassen `Movie` im Package `de.moviemanager.data`, sowie die Klasse `Watched` im Package `de.moviemanager.ui.masterlist.categorizer` im Ordner `main/java` an.

Aufgabe 11.3: Analyse der bestehenden Movie Manager App (Teil 17): Code-Inspektion

Präsenz: Ja	**Punkte:** 4	**Team: Ja**	

1) Inspizieren Sie (ad-hoc) den Code der Klassen `Movie` und `Watched` (siehe Vorbereitung) auf Fehler. Dokumentieren Sie **alle gefundenen Fehler** in einem PDF-Dokument. Ordnen Sie jedem gefundenen Fehler eine Fehlerklasse zu und begründen Sie kurz Ihre Zuordnung. Verwenden Sie zur Fehlerdokumentation die folgende Tabelle und orientieren Sie sich bei Ihrer Lösung an dem gegebenen Beispiel.

Ent-deckerIn	**Datum**	**betroffenes Produkt**	**ausgeführte QS-Aktivität**	**Beschreibung**	**Fehlerklasse**	**Status**
Astrid Rohmann	14.01. 2020	Movie Manager	Code-Inspektion	Die Operation getName() in der Klasse Performer gibt immer null zurück, da nicht auf das Attribut zum Speichern des Namens zugegriffen wird.	2, wesentliche Funktion ist falsch umgesetzt. Nur mit großen Einschrän-kungen einsetzbar.	Beho-ben

Verwenden Sie zur Kategorisierung der Fehlerklasse das folgende Schema:

Klasse	**Bedeutung**
1	Systemabsturz mit ggf. Datenverlust; der Prüfling ist in dieser Form nicht einsetzbar
2	Wesentliche Funktion ist fehlerhaft; Anforderung nicht beachtet oder falsch umgesetzt; der Prüfling ist nur mit großen Einschränkungen einsetzbar
3	Funktionale Abweichung bzw. Einschränkung (»normale« Fehler); Anforderung fehlerhaft oder nur teilweise umgesetzt; Prüfling kann mit Einschränkungen genutzt werden
4	Geringfügige Abweichung; Prüfling kann ohne Einschränkung genutzt werden
5	Schönheitsfehler (z.B. Mangel im Maskenlayout); Prüfling kann ohne Einschränkung genutzt werden

2) Beheben Sie nun die gefundenen Fehler.

Hinweis: *Insgesamt sind 4 Fehler im Quellcode versteckt.*

Ergebnis:

Speichern Sie bitte das Ergebnis als .zip-Datei bis **Montag 20.01.2020 um 10.00 Uhr** in Moodle bestehend aus:

- 1x PDF-Datei mit den gefundenen Fehlern
- 1x Android Studio-Projekt **11-MovieManager.zip** mit der entsprechenden Fehlerbehebung

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

SonarQube
Die Plattform **SonarQube** ist ein Tool für die statische Analyse von Quellcode. Dabei berechnet es u.a. Metriken und weist auf potentielle Probleme im Code hin. Eine Anleitung für dieses Tool finden Sie in Confluence unter: https://confluence-se.ifi.uni-heidelberg.de/x/BID5Bw

Aufgabe 11.4: Analyse der bestehenden Movie Manager App (Teil 18): Metriken

Präsenz: Nein	**Punkte:** 6	**Team:** Nein	

1) Installieren und starten Sie SonarQube gemäß obiger Anleitung. Analysieren Sie damit das bereitgestellte Projekt der Movie Manager App (ohne die Erweiterung von Blatt 7).

2) Betrachten Sie unter „Measures“ die folgenden Metriken für die Dateien in Pfad *mmapp/app/src/main/java/de/moviemanager*.
   - Unter *Size -> Functions* die Anzahl der Funktionen pro Datei
   - Unter *Complexity -> Cyclomatic Complexity* die zyklomatische Komplexität einer Datei
   - Unter *Maintainability -> Code Smells* die möglichen Probleme im Code einer Datei

   Dokumentieren Sie die Metriken in einem PDF-Dokument. Das Dokument soll **für jeweils die obersten 10 Dateien im oben genannten Pfad** (in der Listenansicht, über *View as -> List* einzustellen, absteigend sortiert) die oben genannten Metriken beinhalten. Betrachten Sie zu jeder dieser Metriken jeweils eine Datei und begründen Sie den angezeigten Wert für die jeweilige Metrik.

3) Wozu hilft Ihrer Meinung nach die Erhebung von Metriken von Quellcode? Sammeln Sie in einem PDF-Dokument jeweils mindestens einen Vor- und Nachteil der Erhebung von Metriken für die Entwicklung.

Ergebnis:
Speichern Sie bitte Ihre PDF-Dokumente mit den erhobenen Metriken und Vor- und Nachteil(en) bis **Montag 20.01.2020 um 10.00 Uhr** in Moodle.