

Software-Qualitätssicherung Workshop

Inhaltsverzeichnis

0	Vor	bereitung und Ablauf	2
	0.1	Angabe	2
	0.2	Abgabe	2
		0.2.1 Benennung der Abgaben	2
		0.2.2 Git	2
	0.3	Aufgabenstellung	3
	0.0		
		0.3.1 Ausgangssituation	3
		0.3.2 Anforderungen	3
		0.3.3 Anwendungsfallbeschreibung	4
_	_		
1	Test	ten und Testisolation	5
	1.1	Äquivalenzklassen und Testfallableitung	5
	1.2	Implementierung	5
		1.2.1 Bugsuche	5
		1.2.2 Unit Tests	5
	1.3	Testisolation	5
	1.4	Codecoverage	5
2	Cod	leanalyse und Refactoring	6
	2.1	Testreview	6
	2.2	Statische Code Analyse	6
	2.3		
	2.0	Refactoring	6
		2.3.1 Refactoring Patterns	7
		2.3.2 Vorgehensweise	7
		2.3.3 Aufgabenstellungen	7
	2.4	Theoriefragen	7

Sollten Sie während des Workshops Fragen haben, stehen Ihnen die Tutoren jederzeit zur Verfügung. Bitte machen Sie keine Notizen auf der Angabe und lassen Sie die Angabe nach dem Workshop bei Ihrem Laborrechner liegen.



0 Vorbereitung und Ablauf

Der Workshop besteht aus einer Reihe aufbauender Beispiele, die Sie in Kleingruppen von jeweils zwei bis drei Studierenden bearbeiten. Es stehen Ihnen Arbeitsplätze im Labor (mit IntelliJ, Git, Textverarbeitungsprogrammen, usw.) sowie Accounts zur Verfügung. Verwenden Sie zum Erstellen Ihrer Lösungen ausschließlich die Laborrechner.

Sie dürfen bei der Bearbeitung der Beispiele alle Unterlagen (Vorlesungsfolien, JavaDoc, usw.) sowie das Internet zur Hilfe nehmen. Sollten während der Bearbeitung der Beispiele Fragen auftauchen, melden Sie sich einfach bei einem Tutor.

Ziel dieser Übung ist es, dass Sie gemeinsam eine möglichst komplette Lösung der Beispiele erarbeiten und ihr Vorgehen dabei dokumentieren.

0.1 Angabe

Die Angabe besteht aus diesem Dokument, einer Protokollvorlage (siehe TUWEL) sowie einer Teilimplementierung der Aufgabenstellung:

"QS Workshop Project - Movierental.zip"

Die Teilimplementierung ist ein bereits lauffähiges Java Programm basierend auf Java8 und JavaFX8. Das bestehende Projekt können Sie mit IntelliJ öffnen. Alle Bibliotheken die Sie zum Lösen der Aufgabenstellung benötigen (JUnit, Mockito, ...) sind bereits importiert und können verwendet werden. Hier finden Sie noch einige Links zur Dokumentation der verwendeten Bibliotheken.

- Mockito: http://mockito.github.io/mockito/docs/current/org/mockito/Mockito.html
- JUnit: https://github.com/junit-team/junit/wiki/Getting-started
- Hamcrest: https://code.google.com/p/hamcrest/wiki/Tutorial

0.2 Abgabe

Verwenden Sie für das Lösen der Beispiele die mitgelieferte Vorlage aus dem TUWEL. Laden Sie am Ende des Workshops die Lösung als zip-File hoch (siehe Instruktionen im TUWEL). Neben der Lösung zum jeweiligen Beispiel erstellen Sie ein Protokoll in dem Sie Ihre Schritte kurz zusammengefasst festhalten.

0.2.1 Benennung der Abgaben

Auf Ihrem Laborrechner finden Sie Ihre Gruppennummer, tragen Sie diese Gruppennummer gemeinsam mit ihren Namen und Ihren Matrikelnummern in die dafür vorgesehenen Feldern ein. Erstellen Sie Ihr Protokoll als PDF und benennen Sie das Protokoll nach folgendem Namensschema: QSWorkshop-WS15-Protokoll-GruppeXX.pdf ersetzen Sie XX dabei durch ihre Gruppennummer.

0.2.2 Git

Einige Beispiele enthalten ein git-Repository. Halten Sie jeden Ihrer Schritte im git-Repository fest. Sie benötigen dazu lediglich folgende Befehle:

```
# Add all untracked files
git add -A

# Commit all changes with the given message
git commit -am "<message>"

# Edit last commit message
git commit --amend -m "<new_message>"
```

Denken Sie immer daran aussagekräftige Commit Messages zu wählen!



0.3 Aufgabenstellung

Sie wurden als Mitarbeiter eines mittelständischen Softwareunternehmens mit der Umsetzung einer Applikation für einen kleinen Filmverleih betraut.

0.3.1 Ausgangssituation

Der Filmverleih "Quality Blockbusters" erlebt gerade eine kleine Renaissance und darf sich über regen Kundenzuwachs freuen. Da das Unternehmen bisher sehr klein war, wurden Kundendaten sowie die zum Verleih stehenden Filme mittels analoger Karteikarten verwaltet. Außerdem war bisher ein Mitarbeiter für alle Tätigkeiten verantwortlich. Durch die rasche Expansion des Filmverleihs wird die weitere händische Abarbeitung der Aufgaben als zu aufwendig angesehen. Aus diesem Grund hat sich die Geschäftsführung entschieden, das Kundenverwaltungs- sowie das Filmverwaltungs- und das Verleihsystem mit einer Applikation zu managen. Zudem sollen mehrere neue Mitarbeiter eingestellt werden, die wahrscheinlich jeweils unterschiedliche Berechtigungen in der neuen Applikation benötigen.

Ihre Kollegen haben aus dem Projektauftrag bereits den folgenden Anforderungskatalog erstellt und wichtige Anwendungsfälle genauer beschrieben.

Anmerkung: Für eine bessere Übersicht sind hier nur die für diese Übung relevanten Anforderungen und Anwendungsfälle aufgelistet.

0.3.2 Anforderungen

1. Anforderung: Stammdaten Verwaltung

Es sollen Kunden sowie Filme im System angelegt und bearbeitet werden können.

2. Anforderung: Ausleihvorgang

Es soll möglich sein, im Zuge einer Buchung mehrere Filme auszuleihen und zurückzugeben.

3. Anforderung: Buchungsbeleg

Es soll bei jedem Verleihvorgang ein Buchungsbeleg (Zeitpunkt, Kundenname, Kundennummer, Gesamtkostenabrechnung, Videopoints, Rabatte, Ausleihen und Rückgaben) erstellt werden.

4. Anforderung: Stammkunden anlegen

Kunden sollen von Mitarbeitern zu Stammkunden gemacht werden können. Stammkunden erhalten einen Rabatt von 1.5% auf jede Buchung.

5. Anforderung: Filmbewertungen

Das System muss auf Wunsch Bewertungen für Filme von 0.0 (Sehr schlecht) bis 5.0 (Sehr gut) über ein Onlineservice laden können.

6. Anforderung: Kostenberechnung nach Tagen

Die Kosten für einen Film werden nach Tagen berechnet, nach der Formel: Basispreis * Tage Danach werden noch etwaige Rabatte und Preismodifikationen durchgeführt.

7. Anforderung: Preismodifikation nach Genre

Je nach Genre werden die Filme um einen bestimmten Faktor teurer oder billiger. Klassiker (x0.9), Kinderfilme(x0.75), Horror (x1.1), SciFi (x1) und Fantasy (x1.25).

8. Anforderung: Rabatte

Stammkunden erhalten 1.5% Rabatt, für 10 Videopoints erhalten Kunden 2% Rabatt, für 20 Videopoints erhalten Kunden 6% Rabatt. Die Videopoints müssen danach entsprechend reduziert werden. Die Rabatte für Stammkunden und Videopoints sind kumulativ.

9. Anforderung: Videopoints

Pro $2 \in$ Rechnungssumme wird dem Kunden ein Videopoint gutgeschrieben. Stammkunden wird weiters ein Videopoint pro Buchung gutgeschrieben.

10. Anforderung: Prüfung der Altersfreigabe

Beim Verleih von Filmen muss die Freigabe (FSK 0, 6, 12, 16 18 und RATED 21) geprüft werden.



0.3.3 Anwendungsfallbeschreibung

1. Film verleihen			
Primäre Aktoren:	Kassier		
Vorbedingungen:	Der Kassier ist im System angemeldet. Es existiert ein Kunde. Es existiert ein Film.		
Nachbedingungen:	Der Film wurde der Ausleihe des gewählten Kunden hinzugefügt. Ein Buchungsbeleg wurde erstellt.		
Szenario:	Ein Kunde kommt mit dem Film, den er ausleihen will und seinem Entleihausweis zum Schalter. Der Mitarbeiter startet den Entleihvorgang und gibt die Kundennummer und die Filmnummer im System ein (um den Vorgang zu beschleunigen, kann er dazu auch den Barcodescanner nutzen). Während eines Entleihvorgangs können immer auch Filme zurückgegeben werden (siehe dazu: 4. Film zurückgeben). Sobald der Entleihvorgang abgeschlossen ist, wird die Buchungsbestätigung finalisiert.		

2. Film zurückgeben			
Primäre Aktoren:	Kassier		
Vorbedingungen:	Der Kassier ist im System angemeldet. Es existiert ein Kunde. Es existiert ein Film. Ein Kunde hat einen einen Film in der Ausleihe.		
Nachbedingungen:	Der Film wurde der Ausleihe des gewählten Kunden entfernt. Ein Buchungsbeleg wurde erstellt. Die Videopoints des Kunden wurden aktualisiert.		
Szenario:	Ein Kunde kommt mit dem Film, den er zurückgeben will und seinem Entleihausweis zum Schalter. Der Mitarbeiter startet den Rückgabevorgang und gibt die Kundennummer im System ein (um den Vorgang zu beschleunigen kann er dazu auch den Barcodescanner nutzen). Danach wählt er den Film aus, der zurückgegeben werden soll. Während eines Rückgabevorgangs können immer auch Filme entliehen werden (siehe dazu: 3. Film verleihen). Sobald der Rückgabevorgang abgeschlossen ist, wird die Buchungsbestätigung finalisiert.		

Software-Qualitätssicherung - WS15 - Workshop - Angabe



2.3.1 Refactoring Patterns

Verwenden Sie unter anderem folgende Refactoring Patterns:

- Encapsulate Field
- Extract Method
- · Move Method
- Extract Class

Analysieren Sie jedes Refactoring Pattern und entscheiden Sie ob und in welcher Weise deren Anwendung sinnvoll ist, um den bestehenden Code zu verbessern. Wenn möglich schreiben Sie einen Test, der die Funktionalität Ihres Refactorings überprüft und wenden Sie danach das jeweilige Pattern an.

2.3.2 Vorgehensweise

- Refactoring Pattern identifizieren (z.B. "Move Method")
- Tests laufen lassen (es sollten keine Tests fehlschlagen)
- Wenn möglich einen Test für das Refactoring implementieren
- Refactoring Pattern anwenden
- Tests laufen lassen (es sollten keine Tests fehlschlagen)
- Dokumentieren mit Begründung der Auswahl

Sollten Sie der Meinung sein, ein Refactoring anwenden zu können, zu dem Sie keinen Pattern Namen kennen, erstellen Sie eine kurze Beschreibung dazu.

2.3.3 Aufgabenstellungen

Modellklassen

- Erhöhung der Datenkapselung.
- Im aktuellen Code müssen, sollte ein Genre oder eine Altersfreigabe hinzukommen, viele Klassen manuell geändert werden. Versuchen Sie das Genre und die Altersfreigabe als eigene Klasse oder Enumeration zu extrahieren.
- Extrahieren Sie die Berechnung des Gesamtpreises in ein eigenes Service.

UI

Zukünftig soll das Projekt auch durch eine Web-UI, bzw. eine Konsolenschnittstelle, erweitert werden. In der derzeitigen Implementierung steckt die komplette Logik der Eingabevalidierung in der UI.

• Die Validierung der Eingaben soll in eine eigene Validierungsklasse ausgelagert werden.

Sonstiges

Versuchen Sie auch die Lesbarkeit und damit auch die Wartbarkeit des Codes zu erhöhen.

- Sinnvolle Benennung von Variablen und Methoden
- Einhalten von Coding Guidelines
- Wahl der richtigen Sichtbarkeit (private, public, protected, etc.)
- · Verwendung von Packages
- Denken Sie neben Interfaces auch an andere Sprachkonstrukte wie Enumerations, Vererbung, usw.

2.4 Theoriefragen

Beantworten Sie in Ihrem Protokoll noch folgende Theoriefragen:

- 1) Wozu dient statische Code Analyse? Welche Vorteile/Nachteile entstehen, wenn der Quellcode mittels statischer Code Analyse untersucht wird?
- 2) Ab welchem Zeitpunkt im Softwareentwicklungszyklus ist der Einsatz von Code Analyse sinnvoll? Begründen Sie Ihre Antwort.