Übungsblatt 10 (15.12.2015)

Dialogmodell, Zustandsbasierter Test, Komponententest, Implementierung, Code Coverage, Systemtest

In dieser Übung:

- ✓ Verfeinern Sie den UI-Entwurf des Webclients durch die Erstellung eines Dialogmodells.
- ✓ Vervollständigen Sie Ihren Entwurf für den *Movie Manager* Webclient.
- ✓ Spezifizieren Sie Komponententestfälle für die *Movie Manager* Anwendung.
- ✓ Implementieren Sie Ihren erstellten Entwurf für die *Movie Manager* Anwendung
- ✓ Implementieren Sie Komponententests als JUnit Testfälle.
- ✓ Protokollieren Sie die Ausführung Ihrer Komponententests.
- ✓ Überprüfen Sie die Codeüberdeckung Ihrer Komponententestfälle.

Hinweis zur Probeklausur
wir stellen Ihnen eine Probeklausur zur Verfügung. Die Abgabe der bearbeiteten Probeklausur ist freiwillig!

Die Datei "Probeklausur WS 15/16" (isw-ws1516-probeklausur.pdf) ist in Moodle unter "Lehrmaterialien" zu finden.

Für die Abgabe der Probeklausur haben wir in Moodle einen extra Abgabebereich eingerichtet. In Absprache mit dem Tutor können Sie die Probeklausur auch ausdrucken und ausgefüllt abgeben. Die Probeklausur kann bis **Montag, dem 18.01.2016** in Moodle oder persönlich bei den Tutoren in der Übungsgruppe abgegeben werden.

Die Tutoren werden Ihnen nach der Korrektur ab Dienstag, 19.01.2016 Feedback in Moodle geben. Die Probeklausur wird in den Übungsgruppen am Mi 20.01.2016 bzw. Do 21.01.2016 nachbesprochen. Somit können Sie das Feedback noch für die Klausurvorbereitung nutzen.

Aufgabe 10.1: Verfeinerung des UI-Entwurf – Dialogmodell für MovieManager Webclient

Präsenz: Nein	**Punkte:** 6	**Team:** Ja(4)	**Projekt:** Ja	**Testat**

Zur Verfeinerung Ihres Entwurfs erstellen Sie in dieser Aufgabe ein Dialogmodell. Ziel dieser Aufgabe ist es, dass Sie Ihre UI-Struktur des Webclients durch das Dialogmodell umsetzen.
Erstellen Sie ein Dialogomodell für Ihren *Movie Manager* Webclient. D.h. fügen Sie die Sichten, Benutzeraktionen und die Systemfunktionen in das Diagramm ein, so dass die Dialoge bei der Durchführung der neuen Funktionalität beschrieben sind. Bei der Erstellung des Dialogmodells ist es besonders wichtig, dass dieses einen Bezug zu Ihrer UI-Struktur besitzt. D.h. die Navigationslinks

zwischen den (Unter-)Arbeitsbereichen müssen im Dialogmodell deutlich erkennbar sein. Ändern Sie ggf. die UI-Struktur, wenn Sie durch das Dialogmodell neue Ideen haben. Beschreiben Sie diese Idee zusammen mit dem Dialogmodell. Verwenden Sie in Astah das Element „Note“ für die Beschreibung Ihrer Idee. Legen Sie das Dialogmodell als „File Attachement“ in dem Abschnitt „Design Document“ Ihres UNICASE Projekts ab.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.

Bitte speichern Sie eine .zip-Datei bis **Montag 21.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Erstelltes Dialogomodell als .asta und als .png Datei

Aufgabe 10.2: Fertigstellung des Entwurfsklassendiagramms des MovieManager Webclients

Präsenz: Nein	**Punkte:** 8	**Team:** Ja(4)	**Projekt:** Ja	**Testat**

Das Ziel der Aufgabe ist die Fertigstellung Ihres Entwurfsklassendiagrammes für den *Movie Manager* Webclient. Das vollständige Entwurfsklassendiagramm bildet die Grundlage für die spätere Implementierung des *Movie Manager* Webclients.

Vervollständigen Sie das Entwurfsklassendiagramm aus Aufgabe 9.6 mit den bisher nicht näher betrachteten Systemfunktionen. Begründen Sie die Zuordnung der Systemfunktionen zu den Operationen. Das Diagramm soll alle Klassen, Attribute und Operationen, die zur Umsetzung der Funktionalität notwendig sind beinhalten. Get- und Set-Operationen für Attribute können weggelassen werden. Ordnen Sie alle Klassen des Diagramms den Schichten UI und Modell zu. Färben Sie dazu die Klassen in Ihrem Diagramm entsprechend ein.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.

Bitte speichern Sie eine .zip-Datei bis **Montag 21.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Entwurfsklassendiagramm als .asta und als .png-Datei
- Beschreibung von noch fehlenden Dingen für Implementierung als .pdf-Datei

Hinweise zur Aufgabe 10.3 bis 10.5
Zur Sicherstellung der Qualität Ihrer Lösung gehen Sie bei der Implementierung der neuen Funktionalität in der *Movie Manager* Anwendung nach einem Test-First ähnlichen Prinzip vor, d.h.:

1. Sie spezifizieren Komponententests für die neue Funktionalität der Movie Manager Anwendung. (Aufgabe 10.3)
2. Auf Basis Ihres Entwurfs implementieren Sie die neue Funktionalität. (Aufgabe 10.4)
3. Sie führen die spezifizierten Komponententests aus, verbessern ggf. Ihre Implementierung, prüfen die Test-Code-Überdeckung und verbessern ggf. Ihre Tests sowie die Implementierung. (Aufgabe 10.5)

Die folgenden Aufgaben 10.3 bis 10.5 hängen daher stark zusammen. Geben Sie nach der Bearbeitung der Aufgaben Ihre Lösungen **nur bei Aufgabe 10.5** in Moodle ab.

Aufgabe 10.3: Spezifikation von Komponententestfällen für die MovieManager Anwendung

Präsenz: Ja	**Punkte:** 6	**Team:** **Ja(4)**	**Projekt:** Ja	**Testat**

Im Folgenden werden die Komponententestfälle für alle Operationen `(Un-)MarkLoanable` spezifiziert. Dazu wird zustandsbasierter Test verwendet, um die Zusammenhänge zwischen den Attributen `loanable` und `loaned` und den Operationen darzustellen.

1. Beschreiben Sie zuerst das Zustandsdiagramm für die Attribute `loanable`, `loaned` der Klasse `Movie` und die Operationen `(Un-)MarkMovieLoanable`. Es muss berücksichtigt werden, dass ausgeliehene Filme ausleihbar bleiben müssen. In einem Zustand unerlaubte Operationsaufrufe führen in den gleichen Zustand zurück. Kennzeichen Sie diese mit einem „u“.
2. Bei der Klasse `Series` (und analog `Season`) ist zu berücksichtigen, dass die Operation `mark` alle Teile (`Season` oder `Episode`) auch auf ausleihbar setzt. Überlegen Sie nun die Wirkung der Operation `unmark`. Sie muss berücksichtigen, dass ausgeliehene Teile ausleihbar bleiben müssen. `Unmark` könnte also die Ausleihbarkeit der Teile völlig unverändert lassen oder teilweise ändern. Legen Sie diese Frage und Optionen als Rationale in UNICASE an, wählen Sie eine Option und begründen Sie Ihre Wahl. Berücksichtigen Sie dabei, dass es um die Operation auf der Klasse geht. Als Systemfunktionen können Sie weitere Optionen anbieten.
3. Beschreiben Sie nun das Zustandsdiagramm für die Klasse `Series` und die Operationen `(Un-)MarkSeriesLoanable`. Führen Sie dazu noch ein Attribut `allpartsloanable: bool` ein, das genau dann wahr ist, wenn alle Staffeln und alle Episoden der Serie ausleihbar sind. Berücksichtigen Sie dieses neben `loaned` und `loanable` in den Zuständen.
4. Leiten Sie nun aus dem Zustandsdiagramm für `Series` 1-3 Operationsfolgen ab, die die wesentlichen Zustände, wenn `mark` bzw. `unmark` erlaubt bzw. unerlaubt sind, abdecken.
5. Spezifizieren Sie diese Operationsfolgen als logische Komponententestfälle in UNICASE als Element vom Typ „TestCase“ in dem Abschnitt „Component Tests“ im „Test Document“. Beschreiben Sie für jeden Testfall die benötigte „Infrastructure“, d.h. Stubs und Treiber, die Sie für die Testfälle benötigen, sowie jeweils die Pre- und Postcondition für den Testfall.

Beachten Sie, dass Sie zur Durchführung der Änderungen auf Teilen einer Serie Stubs für diese Teile brauchen. Achten Sie dabei darauf, dass Sie sowohl in den einzelnen TestSteps als auch bei den TestCases jeweils einen aussagekräftigen Beschreibungstext formulieren, der deutlich macht, was der Testfall testet. Beschreiben Sie für jeden Testschritt die Vor- und Nachbedingung unter „Description“ und Ausnahmen unter „Exception“. Orientieren Sie sich bei Ihrer Lösung an den vorgegebenen Testfällen der Klasse `Movie`.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.

Aufgabe 10.4: Implementierung der MovieManager Anwendung

Präsenz: Nein	**Punkte:** 8	**Team:** Ja(4)	**Projekt:** Ja	**Testat**

Ziel dieser Aufgabe ist die Implementierung der neuen Funktionalität in der *Movie Manager* Anwendung. Nutzen Sie für die Implementierung Ihr Wissen aus den Tutorials **07-EclipseModelingFramework.pdf** und **09-Programmierung-View**.

Implementieren Sie den von Ihnen auf Übungsblatt 6 erstellten GUI-Entwurf und das auf Übungsblatt 9 erstellte Dialogmodell und Entwurfsklassendiagramm. 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 als Rationale in UNICASE mit Hilfe eines Issues und passen Sie Ihre Entwürfe entsprechend an.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.
Bitte speichern Sie Ihre Änderungen am Quellcode auf dem Git Server.

Code Coverage Plug-In EclEmma
Zum Überprüfen der Codeüberdeckung durch Testfälle verwenden Sie das Eclipse Plug-In EclEmma. Die Analyse der Codeüberdeckung wird durch einen Rechtsklick auf die JUnit-Testklasse und anschließenden Wählen von „Coverage As“ → „JUnit Test“ ausgeführt. Die Ergebnisse der Analyse findet man sowohl in der Sicht „Coverage“ als auch farbig hervorgehoben im analysierten Quellcode.

Aufgabe 10.5: Implementierung und Ausführen der Komponententest, JUnit, Code Coverage für MovieManager Anwendung

Präsenz: Nein	**Punkte:** 8	**Team:** Ja(4)	**Projekt:** Ja	**Testat**

In dieser Aufgabe implementieren Sie Komponententests und überprüfen und dokumentieren Sie die Codeüberdeckung Ihrer implementierten Komponententests. Gehen Sie dazu wie folgt vor:

1. Testen Sie den von Ihnen erstellten Code aus Aufgabe 10.4. Implementieren Sie dafür die Testfälle, die Sie in Aufgabe 10.3.5 spezifiziert haben, als JUnit Tests. Beachten Sie für die Implementierung die Hinweise zum Einrichten von JUnit im Tutorial **10-UnitTest-MovieManager.pdf**. Passen Sie gegebenenfalls die in UNICASE spezifizierten Testfälle entsprechend der von Ihnen durchgeführten JUnit Test Implementierung an. Dokumentieren

Sie die Ergebnisse Ihrer Testdurchführung mit Testprotokollen in UNICASE. Korrigieren Sie gefundene Fehler, sodass alle Testfälle erfolgreich durchgeführt werden können.

2. Überprüfen Sie für die von Ihnen in Aufgabe 10.5.1 implementierten Komponententestfälle die Codeüberdeckung mit Hilfe des Eclipse Plug-Ins EclEmma. Dokumentieren Sie die ermittelte Überdeckung der getesteten Operationen in einer Textdatei.
3. Fügen Sie ggf. weitere Testfälle hinzu, sodass eine 100% Abdeckung der getesteten Operationen `(Un-)MarkSeriesLoanable` erzielt wird. Dokumentieren Sie nach nochmaliger Prüfung der Codeüberdeckung die resultierende Abdeckung Ihres Codes in einer Textdatei. Falls Sie keine 100% Abdeckung erzielen, begründen Sie dies in der Textdatei.
4. Aktualisieren Sie die spezifizierten Komponententestfälle in Ihrem UNICASE Projekt mit den Änderungen aus Aufgabe 10.5.3.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.
Bitte speichern Sie Ihre Änderungen am Quellcode auf dem Git Server.
Bitte speichern Sie Ihre **Ergebnisse aus Aufgabe 10.3, 10.4 und 10.5** in einer .zip-Datei bis **Montag 21.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- 3 exportierten Projekten „movies“, „movies.edit“ und „movies.ui“
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Codeüberdeckung vor und nach dem Anpassen und Begründung als .pdf

Aufgabe 10.6: Durchführen und Protokollieren von Systemtests für MovieManager Anwendung

Präsenz: Nein	**Punkte:** 4	**Team:** **Ja(4)**	**Projekt:** Ja	**Testat**

In Aufgabe 7.2 haben Sie Systemtestfälle für die neue Funktionalität der *Movie Manager* Anwendung aufgestellt. Führen Sie die von Ihnen spezifizierten Systemtestfälle aus, d.h. führen Sie die entsprechenden Aktionen auf der UI der *Movie Manager* Anwendung aus und erstellen Sie für jeden Testfall in UNICASE ein Testprotokoll. Dokumentieren Sie das Ergebnis Ihrer Testdurchführung in den Testprotokollen. Korrigieren Sie gefundene Fehler, sodass alle Testfälle erfolgreich durchgeführt werden können. Dokumentieren Sie in Trello als zukünftige Testfälle, wenn Sie Funktionalität auf dem GUI anbieten, die durch die Testfälle aus Aufgabe 7.2. nicht abgedeckt ist. Führen Sie diese Testfälle vor der Abschlusspräsentation aus.

Ergebnis:

Tragen Sie bitte Ihre Inhalte in Ihr Trello Board ein.
Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.
Bitte speichern Sie Ihre Änderungen am Quellcode auf dem Git Server.
Bitte speichern Sie eine .zip-Datei bis **Montag 21.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- 3 exportierten Projekten „movies“, „movies.edit“ und „movies.ui“
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)