

Übungsblatt 7 (24.11.2015)

Systemtestfälle, EMF-Modellerweiterung, Klassendiagramme, Sequenzdiagramme

In dieser Übung:

- ✓ Überarbeiten Sie die Anforderungen für das neue Feature der Webausleihe.
- ✓ Aktualisieren Sie Ihre Sprintplanung.
- ✓ Spezifizieren Sie Systemtestfälle für das neue Feature der Webausleihe.
- ✓ Erweitern Sie das Modell der *Movie Manager* Anwendung und generieren Quelltext.
- ✓ Erstellen Sie Klassendiagramme für den Quellcode der *Movie Manager* Anwendung.
- ✓ Erstellen Sie ein Sequenzdiagramm für eine Operation der *Movie Manager* Anwendung.

Auftrag zur Realisierung der Serienverwaltung

Mit Ihrer Präsentation zu den Anforderungen haben Sie die Firma *MovieExperience* davon überzeugt, Ihr Team mit der Realisierung der Webausleihe zu beauftragen. Um Ihre Anforderungen für das neue Feature als Grundlage für den Entwurf und die Implementierung verwenden zu können, sind die letzten Änderungswünsche des Auftraggebers hinsichtlich der Anforderungen zu berücksichtigen.

Aufgabe 7.1: Überarbeitung der Anforderungen und Sprintplanung

Präsenz: Nein

Punkte: 4

Team: Ja(4)

Projekt: Ja

Testat

Um die nächsten Wochen an Ihrem Projekt gut weiterarbeiten zu können, ist es notwendig, dass Sie Teile Ihrer Lösung zum 6. Übungsblatt überarbeiten. Überprüfen und überarbeiten Sie alle Anforderungen (User Tasks, Personae- und Rollenbeschreibungen, Systemfunktionen, Daten, UI-Struktur, GUI-Mock-Ups) Ihres Projekts anhand der Anmerkungen aus der Testatabnahme und der Präsentation. Berücksichtigen Sie insbesondere die Änderungswünsche der Auftraggeberin zu Ihren Anforderungen. Pflegen Sie alle Änderungen in Ihr UNICASE Projekt ein. Überarbeiten Sie Ihre Sprintplanung in Trello. Aktualisieren Sie auch die Zuweisungen Ihrer Teammitglieder zu den projektrelevanten Aktivitäten dieses Aufgabenblattes, d.h. beschreiben Sie, wer an der Erstellung welchen Artefaktes beteiligt ist.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server und tragen Sie Ihre Inhalte in das Trello Board ein. Bitte speichern Sie Ihre Ergebnisse in einer .zip-Datei bis **Montag 30.11.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)

Hinweis zur Qualitätssicherung

Ihr Auftraggeber, die Firma *MovieExperience*, legt sehr viel Wert auf die Qualität des *Movie Manager* Produkts. Daher möchte Ihr Auftraggeber, dass Ihr Team bereits vor dem Entwurf und der Implementierung des neuen Features die Systemtestfälle konkretisiert.

Aufgabe 7.2: Systemtests

Präsenz: Nein	Punkte: 5	Team: Ja(4)	Projekt: Ja	Testat
---------------	-----------	-------------	-------------	--------

In dieser Aufgabe erstellen Sie erste logische und konkrete Systemtestfälle für Ihre Webausleihe. Orientieren Sie sich bei der Spezifikation Ihrer Testfälle an dem Beispiel-Systemtest in Ihrem UNICASE Projekt. Gehen Sie wie folgt vor:

Überlegen Sie sich für Ihre Systemfunktionen aus Aufgabe 6.3 logische und konkrete Systemtestfälle. Achten Sie bei der Wahl Ihrer Testfälle darauf, dass **jede neue Systemfunktion** und **jeder Ausnahmefall**, der zu einer Fehlermeldung führen würde, abgedeckt ist. Achten Sie insbesondere darauf, dass Ihre Testfälle auch die **Vorbedingungen** der Systemfunktionen überprüfen. Für das Sortieren ist es ausreichend, wenn Sie einen Testfall exemplarisch für ein Sortierkriterium betrachten. Sie brauchen für Serien, Staffeln und Episoden nicht jede Systemfunktion einzeln durch Testfälle abdecken. Wählen Sie für Serien, Staffeln und Episoden nur repräsentative Testfälle aus und begründen Sie diese Auswahl. Tragen Sie die Testfälle in Ihr UNICASE Projekt in den Abschnitt „Test Document“ unter „System Test“ ein. Verwenden Sie dafür das Systemtest-Modellelement.

Simulieren Sie die Abläufe anhand Ihrer GUI-Mock-Ups. Erweitern Sie Ihre GUI-Mock-Ups so, dass auch mögliche Fehlermeldungen angezeigt werden.

Ergebnis:

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

Bitte speichern Sie Ihr Ergebnis in einer **.zip-Datei** bis **Montag 30.11.2015 um 10.00 Uhr** in Moodle bestehend aus:

- PDF mit Begründung der Testfallauswahl
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)

Quellcode EMF Movie Manager

Zur späteren Implementierung Ihrer Webausleihe stellt Ihnen die Firma *MovieExperience* den Quellcode des *Movie Manager* Produkts zur Verfügung. Die Implementierung der *Movie Manager* Anwendung basiert auf dem Framework EMF und **wird in der Zentralübung in dieser Woche** näher vorgestellt. Der *Movie Manager* Webclient ist mit GWT implementiert.

Auf diesem und den kommenden Übungsblättern werden Sie mit dem Quellcode des *Movie Manager* Produkts arbeiten. Damit Sie die zugehörigen Übungsaufgaben möglichst gut bearbeiten können, ist es besonders wichtig, dass Sie sich ausführlich in die bestehende Implementierung einarbeiten. Die nachfolgenden Aufgaben 7.3 bis 7.6 unterstützen Sie bei der Einarbeitung.

Laden Sie sich zunächst den Quellcode des *Movie Manager* Produkts aus Ihrem Git Repository in Ihre Eclipse Entwicklungsumgebung. Eine Anleitung zum Klonen und Importieren des Quellcodes finden Sie in Moodle (07-Quellcode-MovieManager.pdf).

Aufgabe 7.3: Erweiterung der Movie Manager Anwendung mit EMF

Präsenz: Ja	Punkte: 5	Team: Ja(2)	Projekt: Nein	Testat
-------------	-----------	-------------	---------------	--------

In dieser Aufgabe üben Sie die Modellerweiterung und Codegenerierung mit EMF in der *Movie Manager* Anwendung. Die Programmierung mit EMF wird **in der Zentralübung in dieser Woche** vorgestellt. Ein zusätzliches Tutorial für die EMF Programmierung finden Sie in Moodle. Zum Einüben der Modellerweiterung soll das Modell der *Movie Manager* Anwendung um einen Regisseur erweitert werden. Diese Funktionalität kennen Sie bereits aus Aufgabe 2 des 2. Übungsblattes. Führen Sie folgende Schritte durch:

1. Legen Sie die Klasse `Director` in dem Ecore-Modell der *Movie Manager* Anwendung an. Lassen Sie die Klasse `Director` von der Klasse `Performer` erben.
2. Fügen Sie der `Director` Klasse ein `numDirectedMovies` Attribut hinzu und setzen Sie den Typ des Attributes auf `int`.
3. Erstellen Sie in der `Director` Klasse eine Assoziation `moviesDirected` vom Typ `Movie`. Setzen Sie die obere Grenze der Assoziation auf `-1`.
4. Generieren Sie den Quellcode, indem Sie in der Datei `movies.genmodel` „Generate Model Code“ und „Generate Edit Code“ wählen.

Ergebnis:

Bitte speichern Sie Ihr Ergebnis in einer **.zip-Datei** bis **Montag 30.11.2015 um 10.00 Uhr** in Moodle bestehend aus:

- 3 Exportierte Eclipse-Projekte mit `movies`, `movies.edit` und `movies.ui` Projekten

Hinweis zu Aufgabe 7.4

Das in der nachfolgenden Aufgabe 7.4 zu erstellende Klassendiagramm bildet die Grundlage für den späteren Entwurf und die Implementierung Ihrer Webausleihe und soll im weiteren Projektverlauf gepflegt werden. Es ist daher **besonders wichtig**, dass Sie bei der Bearbeitung der Aufgabe 7.4 sehr sorgfältig vorgehen.

Aufgabe 7.4: Klassendiagramm für die Movie Manager Anwendung**Präsenz:** Ja**Punkte:** 8**Team:** Ja(4)**Projekt:** Ja**Testat**

Erstellen Sie in dem Werkzeug Astah (Download in Moodle) ein **Klassendiagramm** für die Modellklassen der *Movie Manager* Anwendung. Schauen Sie sich dafür den Quelltext und die Ecore-Modelldatei des Projekts *movies* der *Movie Manager* Anwendung an.

1. Zeichnen Sie alle Klassen und Enumerationen des Projekts *movies* in das Klassendiagramm ein (ohne Attribute und Operationen).
2. Ergänzen Sie im Klassendiagramm für alle Klassen und Enumerationen jeweils alle Attribute und Operationen. Zeichnen Sie für Attribute alle Modifier, den Typ, und ggf. Default-Werte ein. Zeichnen Sie für Operationen alle Modifier, den Namen und den Rückgabewert ein.
3. Erkennen Sie aus dem Quellcode die Assoziationen zwischen den Klassen und zeichnen Sie diese in das Klassendiagramm ein. Benennen Sie die Assoziationen und ergänzen Sie die Multiplizitäten.
4. Beschreiben Sie in 1-2 Sätzen pro Assoziation im Klassendiagramm wie die Klassen zusammenarbeiten. Speichern Sie die Beschreibungen in einer Textdatei.
5. Legen Sie das erstellte Klassendiagramm in Ihrem UNICASE Projekt als asta-Datei in dem Abschnitt „Design Document → Architecture“ als „File Attachment“ ab. Achten Sie auf einen ausdrucksstarken Namen, sodass man sofort erkennen kann, um welche Datei es sich handelt.

Ergebnis:

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

Bitte speichern Sie Ihr Ergebnis in einer **.zip-Datei** bis **Montag 30.11.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Klassendiagramm als .asta-Datei
- Beschreibungen als .pdf-Datei
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)

Hinweis zu Aufgabe 7.5 und Aufgabe 7.6

Die nachfolgenden Aufgaben 7.5 und 7.6 dienen der weiteren Einarbeitung in den Quellcode der *Movie Manager* Anwendung. Die Aufgaben vertiefen die Grundlagen zu Klassen- und Sequenzdiagrammen und dienen als Vorbereitung auf die Aufgaben des nächsten Übungsblattes.

Aufgabe 7.5: Klassendiagramm für die UI der Movie Manager Anwendung

Präsenz: Ja	Punkte: 10	Team: Ja(2)	Projekt: Nein	
--------------------	-------------------	--------------------	----------------------	--

Erstellen Sie in dem Werkzeug Astah (Download in Moodle) ein **Klassendiagramm** für den UI Code der *Movie Manager* Anwendung. Schauen Sie sich dafür den Quelltext des Projekts `movies.ui` der *Movie Manager* Anwendung an.

- Zeichnen Sie folgenden 5 Klassen der Benutzeroberfläche vom Projekt `movies.ui` in das Klassendiagramm ein (ohne Attribute und Operationen):
 - `movies.ui.handler.MovieViewHandler`
 - `movies.ui.views.MoviesView`
 - `movies.ui.views.contentprovider.MoviesContentProvider`
 - `movies.ui.views.labelprovider.CustomColumnLabelProvider`
 - `movies.ui.views.sort.TreeViewerColumnSorter`
- Ergänzen Sie im Klassendiagramm für die Klasse `TreeViewerColumnSorter` alle Attribute und Operationen. Zeichnen Sie für Attribute alle Modifier, den Typ, und ggf. Default-Werte ein. Zeichnen Sie für Operationen alle Modifier, den Namen und den Rückgabewert ein.
- Erkennen Sie aus dem Quellcode die Assoziationen zwischen den Klassen und zeichnen Sie diese in das Klassendiagramm ein. Benennen Sie die Assoziationen und ergänzen Sie die Multiplizitäten.
- Weiterhin ist es wichtig, dass alle Vererbungsstrukturen der Tiefe 1 und Benutzungen von Schnittstellenklassen mit Hilfe dieses Klassendiagramms sichtbar gemacht werden. Beachten Sie dazu die Vererbungen, die von außerhalb der Klassen kommen sowie die benutzten Schnittstellenklassen.
- Beschreiben Sie in 2-3 Sätzen pro Assoziation im Klassendiagramm wie die Klassen zusammenarbeiten. Speichern Sie die Beschreibungen in eine Textdatei.
- Legen Sie das erstellte Klassendiagramm in Ihrem UNICASE Projekt als `asta`-Datei in dem Abschnitt „Design Document → Architecture“ als „File Attachment“ ab. Achten Sie auf einen ausdrucksstarken Namen, sodass man sofort erkennen kann, um welche Datei es sich handelt.

Ergebnis:

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

Bitte speichern Sie Ihre Ergebnisse in einer **.zip-Datei** bis **Montag 30.11.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Klassendiagramm als `.asta`-Datei
- Beschreibungen als `.pdf`-Datei
- Exportiertes UNICASE-Projekt als `.ecp` (damit Stand des Projekts bewertet werden kann)

Aufgabe 7.6: Sequenzdiagramm für eine Operation der Movie Manager**Anwendung**

Präsenz: Nein	Punkte: 8	Team: Ja(2)	Projekt: Nein	Testat
----------------------	------------------	--------------------	----------------------	---------------

Benutzen Sie die Datei 07-Sequenzdiagramm.asta (zu finden in Moodle) für das Zeichnen des **Sequenzdiagramms** der `execute`-Operation aus dem UI Code der *Movie Manager* Anwendung. Beachten Sie dabei, dass es notwendig ist, alle in der Sequenzdiagrammvorlage eingezeichneten Objekte auch zu verwenden, d.h. dass diese Objekte Nachrichten senden und/oder empfangen.

1. Zeichnen Sie in dem Werkzeug Astah ein **Sequenzdiagramm** für die Operation `execute()` in der Klasse `MovieViewHandler` im Package `movies.ui.handlers`. Benutzen Sie den Quellcode der *Movie Manager* Anwendung und verwenden Sie den Debug-Modus von Eclipse.

*Setzen Sie einen **Breakpoint** in der ersten Zeile der Methode `execute()` in der Klasse `MovieViewHandler`. Starten Sie dann den Debug-Modus und lösen Sie dann die implementierte Funktion über das Kontext-Menü **Show Movies** im Navigator aus. Durchlaufen Sie die Operationen bis zum Ende. Setzen Sie ggf. weitere Breakpoints. Tragen Sie alle Aufrufe zwischen den Klassen in das Sequenzdiagramm ein.*

2. Ihr erstelltes Sequenzdiagramm wird sowohl Klassen der Modellschicht als auch der Oberflächenschicht enthalten. Ordnen Sie die Klassen des Sequenzdiagramms den jeweiligen Schichten zu. Färben Sie hierzu die Klassen innerhalb des Sequenzdiagramms **grün** für Oberflächen-Klassen und **rot** für Modellklassen.
3. Welches sind Aufrufe von Oberflächen-Klassen auf Modellschicht-Klassen? Was bewirken diese Aufrufe? Listen Sie die Aufrufe auf und beschreiben Sie diese in einer Textdatei.
4. Legen Sie das erstellte Sequenzdiagramm in Ihrem UNICASE Projekt als .asta-Datei in dem Abschnitt „Design Document → Architecture“ als „File Attachment“ ab. Achten Sie auf einen ausdrucksstarken Namen, sodass man sofort erkennen kann, um welche Datei es sich handelt.

Ergebnis:

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

Bitte speichern Sie Ihre Ergebnisse in einer **.zip-Datei** bis **Montag 30.11.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Sequenzdiagramm als .asta-Datei
- Schicht-übergreifende Kommunikation als .pdf-Datei
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)