

Übungsblatt 9 (08.12.2015)

SWT View Programmierung, Entwurfsmuster, Rationale, Zustandsübergangsdiagramm, Dialogmodell, OOAD

In dieser Übung:

- ✓ Üben Sie die Programmierung einer View mit SWT.
- ✓ Finden und Dokumentieren Sie Entwurfsmuster im Quellcode der *Movie Manager* Anwendung.
- ✓ Bewerten Sie die Verwendung von Entwurfsmustern in der *Movie Manager* Anwendung.
- ✓ Verfeinern Sie Ihren UI-Entwurf für die *Movie Manager* Anwendung durch die Erstellung eines Dialogmodells.
- ✓ Vervollständigen Sie Ihren Entwurf für die *Movie Manager* Anwendung.
- ✓ Erstellen Sie mit OOAD einen Entwurf für Ihren *Movie Manager* Webclient.

Aufgabe 9.1: Programmierung: View

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

Im Tutorial **09-Programmierung-View.pdf** ist ausführlich beschrieben, wie Sie eine View für die *Movie Manager* Anwendung programmieren. Führen Sie zunächst dieses Tutorial durch. Beginnen Sie erst danach mit dem Bearbeiten dieser Aufgabe.

Implementieren Sie wie im Tutorial beschrieben eine **neue View**, die alle Movies eines Projekts anzeigt, deren Gesamtbewertung mindestens 3 beträgt. Orientieren Sie sich für Ihre Lösung an den bereits vorhandenen Views (`PerformersView`, `MoviesView` und `SeriesView`). Gehen Sie dabei wie folgt vor:

1. Implementieren Sie den `ContentProvider` `RatedMoviesContentProvider` im package `movies.ui.views.contentprovider`, der alle Movies eines Projekts bereitstellt deren Gesamtbewertung mindestens 3 beträgt.
2. Implementieren Sie die View `RatedMoviesView`, die eine Tabelle mit allen Movies darstellt. In der ersten Spalte soll der Name des Movies und in der zweiten Spalte die Gesamtbewertung ausgegeben werden.
3. Binden Sie die View in die Benutzeroberfläche entsprechend der Beschreibung im Tutorial ein (siehe `plugin.xml`).

Ergebnis:

Bitte speichern Sie eine .zip-Datei bis **Montag 14.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- 3 exportierten Projekte „movies“, „movies.edit“ und „movies.ui“

Aufgabe 9.2: Entwurfsmuster

Präsenz: Nein

Punkte: 6

Team: Ja(2)

Projekt: Nein

Testat

Finden Sie existierende Entwurfsmuster in einem gegebenen Klassendiagramm. Das zu betrachtende Klassendiagramm *09-Entwurfsmuster.asta/.png* finden Sie in Moodle. Der Quellcode der *Movie Manager* Anwendung bildet die Basis des Klassendiagramms. Verwenden Sie für die Bezeichnung der Klassen die Klassennamen aus dem gegebenen Klassendiagramm. Notieren Sie für jedes Muster welche **Rolle** die **beteiligten Klassen im vorgegebenen Klassendiagramm** ausführen. Beschränken Sie sich bei der Suche auf die folgenden Entwurfsmuster:

- Abstrakte Oberklasse
- Einzelstück
- Fabrikmethode
- Kompositum
- Stellvertreter (Proxy)
- Beobachter

Finden Sie im Klassendiagramm **mindestens 3** verschiedene Anwendungen dieser Entwurfsmuster, und notieren Sie diese. Ziehen Sie ggf. auch den Quellcode der *Movie Manager* Anwendung bei Ihrer Suche zu Rate. Beachten Sie, dass eine Klasse in mehreren Mustern vorkommen kann (wobei sie in jedem Muster eine andere Rolle einnimmt).

Ergebnis:

Speichern Sie bitte Ihre Notizen als .pdf bis **Montag 14.12.2015 um 10.00 Uhr** in Moodle.

Aufgabe 9.3: Rationale für Entwurfsmuster

Präsenz: Nein

Punkte: 6

Team: Ja(2)

Projekt: Nein

Bewerten Sie Entwurfsmuster zum Erzeugen von Objektinstanzen („Erzeugungsmuster“). Verwenden Sie die nachfolgenden **Entwurfskriterien** für die Bewertung:

- **Komplexität**
 1. geringe Anzahl der Klassen
 2. geringe Anzahl der Beziehungen zwischen den Klassen
- **Flexibilität, Erweiterbarkeit**
 3. Leichtes Hinzufügen neuer Klassen
 4. Leichtes Hinzufügen neuer Attribute

Bewerten Sie die Vorschläge am Ende dieser Aufgabe. Verwenden Sie die **Rationale-Funktionalität von UNICASE** um die Vorschläge gegen die Entwurfskriterien zu bewerten: Erstellen Sie ein Element vom Typ „Issue“, in welchem Sie die Vorschläge (durch Elemente vom Typ „Proposal“) anhand der Entwurfskriterien 1 bis 5 (zu erstellen als Element vom Typ „Criteria“) bewerten und sich für eine Lösung (durch Element vom Typ „Solution“) entscheiden. Speichern Sie die Rationale-Elemente unter „Design Document“ -> „Packages“.

Begründen Sie für jeden Vorschlag und jedes Kriterium Ihre Bewertung im Beschreibungstext des dafür von Ihnen erstellten Issues, insbesondere die Einschätzung auf der Skala im Vergleich zu den anderen Vorschlägen. Verwenden Sie die Assessment Matrix des erstellten Issues und benutzen Sie folgendes Schema für Ihre Bewertung:

Skala	Bedeutung
2	Eigenschaft wird optimal erfüllt
1	Eigenschaft wird gut erfüllt
0	Eigenschaft wird ausreichend erfüllt
-1	Eigenschaft wird nicht ausreichend erfüllt
-2	Eigenschaft kann gar nicht unterstützt werden

Vorschläge für Erzeugungsmuster:**Erzeugungsmuster Vorschlag 1: Abstrakte Fabrik („Abstract Factory“)**

Das Entwurfsmuster Abstrakte Fabrik definiert eine Schnittstelle zur Erzeugung einer Familie von Objekten, wobei die konkreten Klassen der zu instanziiierenden Objekte nicht näher festgelegt werden. Es findet Anwendung, wenn

- ein System unabhängig von der Art der Erzeugung seiner Produkte arbeiten soll,
- ein System mit einer oder mehreren Produktfamilien konfiguriert werden soll,
- eine Gruppe von Produkten erzeugt und gemeinsam genutzt werden soll oder
- wenn in einer Klassenbibliothek die Schnittstellen von Produkten ohne deren Implementierung bereitgestellt werden sollen.

Eine typische Anwendung ist die Erstellung einer grafischen Benutzeroberfläche mit unterschiedlichen Themes.

Eine Abstrakte Fabrik vereinigt die Verantwortlichkeiten „Zusammenfassung der Objektgenerierung an einer Stelle“ und „Möglichkeit zu abstrakten Konstruktoren“.

Erzeugungsmuster Vorschlag 2: Einzelstück („Singleton“)

Das Singleton Entwurfsmuster stellt sicher, dass von einer Klasse genau ein Objekt existiert. Das Singleton ist darüber hinaus üblicherweise global verfügbar. Es findet Verwendung, wenn

- nur ein Objekt zu einer Klasse existieren darf und ein einfacher Zugriff auf dieses Objekt benötigt wird oder
- das einzige Objekt durch Unterklassenbildung spezialisiert werden soll.

Anwendungsbeispiele sind

- ein zentrales Protokoll-Objekt, das Ausgaben in eine Datei schreibt.
- Druckaufträge, die zu einem Drucker gesendet werden, sollen nur in einen einzigen Puffer geschrieben werden.

Erzeugungsmuster Vorschlag 3: Erbauer („Builder“)

Das Erbauer Entwurfsmuster trennt die Konstruktion komplexer Objekte von deren Repräsentationen, wodurch dieselben Konstruktionsprozesse wiederverwendet werden können. Es wird verwendet, wenn

- zu einem komplexen Objekt unterschiedliche Repräsentationen existieren sollen,
- die Konstruktion eines komplexen Objekts unabhängig von der Erzeugung der Bestandteile sein soll oder
- der Konstruktionsablauf einen internen Zustand erfordert, der vor einem Klienten verborgen werden soll.

Typische Anwendungen sind z. B. Applikationen zur Konvertierung.

Erzeugungsmuster Vorschlag 4: Fabrikmethode („Factory Method“)

Das Fabrikmethoden Entwurfsmuster beschreibt, wie ein Objekt durch Aufruf einer Methode anstatt durch direkten Aufruf eines Konstruktors erzeugt wird.

Die Fabrikmethode findet Anwendung, wenn eine Klasse die von ihr zu erzeugenden Objekte nicht kennen kann bzw. soll, oder wenn Unterklassen bestimmen sollen, welche Objekte erzeugt werden.

Typische Anwendungsfälle sind Frameworks und Klassenbibliotheken. Die Fabrikmethode stellt u.a. eine Lösung dar, um sich den Zielen der geringen Kopplung und der hohen Kohäsion anzunähern.

Ergebnis:

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

Bitte speichern Sie Ihre Ergebnisse bis **Montag 14.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

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

Implementierung:

Die Implementierung ist erst für das nächste Übungsblatt vorgesehen. Sie können aber bereits jetzt parallel zum Entwurf versuchen, Ihren logischen Entwurf prototypisch in der *Movie Manager* Anwendung zu implementieren. Dadurch fällt es Ihnen vielleicht leichter, die Zusammenhänge besser zu verstehen. Dazu können Sie beispielsweise das Modell der *Movie Manager* Anwendung erweitern oder bereits neu hinzugefügte Sichten und Dialoge umsetzen.

Aufgabe 9.4: Verfeinerung des UI – Entwurfs: Dialogmodell für die MovieManager Anwendung

Präsenz: Nein

Punkte: 4

Team: Ja(4)

Projekt: Ja

Testat

Zur Verfeinerung Ihres UI (beschrieben durch UI-Struktur und Sichten) erstellen Sie in dieser Aufgabe ein Dialogmodell. Ziel dieser Aufgabe ist es, dass Sie einen Teil der von Ihnen bestimmten UI-Struktur durch das Dialogmodell umsetzen. Verwenden Sie zum Erstellen des Dialogmodells die Vorlage *09-DialogModel.asta* in Ihrem UNICASE Projekt in dem Abschnitt „Design Document“. Das Zustandsdiagramm in dieser Vorlage zeigt das Dialogmodell für die Funktionalität der bisherigen *Movie Manager* Anwendung (Sichten, Benutzeraktionen und Systemfunktionen).

Ergänzen Sie das Dialogmodell um die neue Funktionalität der *Movie Manager* Anwendung. 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. Fügen Sie insbesondere auch Übergänge zwischen existierenden Sichten der *Movie Manager* Anwendung und Ihren neuen Sichten 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. 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 Attachment“ 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 14.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Erstelltes Dialogmodell als .asta und als .png Datei

Aufgabe 9.5: Fertigstellung des Entwurfs der MovieManager Anwendung

Präsenz: Ja	Punkte: 8	Team: Ja(4)	Projekt: Ja	Testat
-------------	-----------	-------------	-------------	--------

Das Ziel der Aufgabe ist die Fertigstellung Ihres Entwurfsklassendiagrammes für die *Movie Manager* Anwendung. Das vollständige Entwurfsklassendiagramm bildet die Grundlage für die spätere Implementierung der neuen Funktionalität in der *Movie Manager* Anwendung.

1. Vervollständigen Sie das Entwurfsklassendiagramm aus Aufgabe 8.4 mit den bisher nicht näher betrachteten Systemfunktionen. Begründen Sie als Rationale in UNICASE (Modellelement „Issue“) die Zuordnung der Systemfunktionen als Operationen zu den Klassen. Das Diagramm soll alle Klassen, Attribute und Operationen, die zur Umsetzung der neuen 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. Legen Sie das Entwurfsklassendiagramm als „File Attachment“ in dem Abschnitt „Design Document“ Ihres UNICASE Projekts ab.
2. Beschreiben Sie, was in Ihrem gesamten Entwurf (Entwurfsklassendiagramm, Sequenzdiagramm, GUI-MockUp und Dialogmodell) der *Movie Manager* Anwendung noch fehlt, damit dieser von einem beliebigen Entwicklungsteam als Grundlage für eine Implementierung ausreichend ist.

Ergebnis:

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

Bitte speichern Sie eine .zip-Datei bis **Montag 14.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Entwurfsklassendiagramm als .asta und als .png-Datei
- Beschreibung von noch fehlenden Dingen für Implementierung als .pdf-Datei

Aufgabe 9.6: Entwurf des Movie Manager Webclients nach OOAD

Präsenz: Ja	Punkte: 8	Team: Ja(4)	Projekt: Ja	Testat
-------------	-----------	-------------	-------------	--------

In Aufgabe 8.3 und 8.4 haben Sie bereits die neue Funktionalität für die *Movie Manager* Anwendung nach OOAD entworfen. Ziel dieser Aufgabe ist der Entwurf der neuen Funktionalität für den *Movie Manager Webclient*.

Erstellen Sie für Ihren *Movie Manager* Webclient einen Entwurf. Gehen Sie beim Entwurf nach der in Übungsblatt 8 vorgestellten Methode vor, d.h.

- Erstellen Sie analog zu Aufgabe 8.3.1 ein **Analyseklassendiagramm** und analog zu Aufgabe 8.3.2 ein **Analyseklassendiagramm mit Klassenmodellierungselementen** für **alle semantischen Systemfunktionen** des *Movie Manager* Webclients. Begründen Sie die Zuordnung der Systemfunktionen im Analyseklassendiagramm, indem Sie das UNICASE-Modellelement „Issue“ verwenden. Füllen Sie alle relevanten Felder des „Issues“ aus. Speichern Sie die „Issues“ im Abschnitt „Design Document“ Ihres UNICASE Projekts. Sollten Sie auf Grund der Entwurfsdiagramme Ihre UI-Struktur oder GUI-MockUps ändern, begründen Sie Ihre Entscheidung in einem „Issue“.
- Setzen Sie anschließend das Analyseklassendiagramm um als Erweiterung des **Entwurfsklassendiagramms** für den Webclient aus Aufgabe 8.1. Beschränken Sie sich dabei auf die Umsetzung der Funktionalität für Movie, also **Loan- und ReturnMovie**. Ordnen Sie alle Klassen des Diagramms den Schichten UI und Modell zu, indem Sie die Klassen im Diagramm entsprechend einfärben.

- Geben Sie allen Diagrammen eine geeignete Benennung und legen Sie diese im „Design Document“ als Element vom Typ „File Attachment“ in UNICASE ab.

Ergebnis:

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

Bitte speichern Sie Ihre Ergebnisse in einer .zip-Datei bis **Montag 14.12.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Diagramme als .asta und .png (Analyseklassendiagramme und Entwurfsklassendiagramm)