

Übungsblatt 8 (10.12.2019)

Entwurfsmuster, Rationale, Dialogmodell, Wiederholungsfragen zu Kapitel 4

In dieser Übung:

- ✓ Finden und Dokumentieren Sie Entwurfsmuster im Quellcode des *Movie Managers*.
- ✓ Bewerten Sie Entwurfsmuster und begründen Sie dies.
- ✓ Beschäftigen Sie sich mit dem Dialogmodell des *Movie Managers* und üben dabei den Umgang mit Zustandsdiagrammen.
- ✓ Beantworten Sie Wiederholungsfragen zum Vorlesungskapitel 4.

Aufgabe 8.1: Analyse der bestehenden Movie Manager App (Teil 11): Entwurfsmuster

Präsenz: Nein

Punkte: 12

Team: Ja

Testat

In dieser Aufgabe vertiefen Sie Ihr Verständnis des *Movie Manager* Quellcodes. Dazu identifizieren Sie existierende Entwurfsmuster im Quellcode des *Movie Managers*.

Betrachten Sie den folgenden Ausschnitt eines Klassendiagramms in der Vorlage *08-MovieManager-ClassDiagram.png* (Datei zu finden in Moodle), den Quellcode des *Movie Managers* sowie die nachfolgend genannten Entwurfsmuster.

Identifizieren Sie die an einem Entwurfsmuster beteiligten Klassen im Diagramm und im Quellcode. Notieren Sie in einem PDF-Dokument für jedes Entwurfsmuster, welche **Rolle die beteiligten Klassen im vorgegebenen Klassendiagramm** haben. Orientieren Sie sich an dem folgenden Beispiel für das Entwurfsmuster „Erbauer (Builder)“:

- `TextInputAttribute<X>` (Produkt)
- `TextInputAttribute.Builder` (Erbauer)
- `PortableDetailEditActivity.buildTextInputAttribute()` (Direktor)

Beschränken Sie sich bei Ihrer Suche auf die folgenden Entwurfsmuster:

- Einzelstück (Singleton)
- Fabrikmethode (Factory Method)
- Erbauer (Builder)
- Beobachter (Observer)
- Adapter (Wrapper)
- Schablonenmethode (Template Method)

Finden Sie im Klassendiagramm und im Quellcode **für jedes der obigen Entwurfsmuster mindestens eine Anwendung** dieses Entwurfsmusters und notieren Sie diese. Beachten Sie, dass eine Klasse in mehreren Mustern vorkommen kann (wobei sie in jedem Muster eine andere Rolle einnimmt). In den Folien der Technologievorlesung zu Entwurfsmustern finden Sie ein paar Tipps, die Ihnen bei der Suche helfen können.

Ergebnis:

Speichern Sie bitte Ihr PDF-Dokument mit den identifizierten Entwurfsmustern bis **Montag, den 16.12.2019, um 10.00 Uhr** in Moodle.

Aufgabe 8.2: Rationale für Entwurfsmuster

Präsenz: Ja	Punkte: 8	Team: Ja	
--------------------	------------------	-----------------	--

In dieser Aufgabe 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 gegebenen Vorschläge im Abschnitt „Zu bewertende Erzeugungsmuster“ auf der nächsten Seite des Übungsblattes. Verwenden Sie für Ihre Bewertung auch die Informationen zu den Entwurfsmustern (insbesondere zu den Klassendiagrammen) aus dem Foliensatz zu Entwurfsmuster der Technologievorlesung und den darin angegebenen Quellen.

Erstellen Sie zur Bewertung in einem PDF-Dokument eine Tabelle, die in den Zeilen die Muster und in den Spalten die 4 Entwurfskriterien auflistet. Bewerten Sie dann die Vorschläge gegen die 4 Entwurfskriterien anhand der nachfolgend aufgeführten Bewertungsskala.

Bewertungsskala

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

Begründen Sie für jedes Entwurfsmuster und jedes Kriterium Ihre Bewertung in Ihrem PDF-Dokument unterhalb der Bewertungstabelle. Gehen Sie dabei auch auf Ihre Skalenbewertung im Vergleich zu den anderen Vorschlägen ein.

Zu bewertende Erzeugungsmuster:**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 (also der durch die Fabrik erzeugten Objekte) arbeiten 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“ für das Aussehen der Benutzeroberfläche.

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

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.

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.

Fabrikmethode („Factory Method“)

Das Fabrikmethoden Entwurfsmuster beschreibt, wie ein Objekt durch Aufruf einer Operation 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
- 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:


Speichern Sie bitte Ihr PDF-Dokument mit der Bewertung und Begründung der Entwurfsmuster bis **Montag 16.12.2019 um 10.00 Uhr** in Moodle.

Aufgabe 8.3: Analyse der bestehenden Movie Manager App (Teil 12): Dialogmodell

Präsenz: Nein	Punkte: 8	Team: Nein	Testat
----------------------	------------------	-------------------	---------------

In dieser Aufgabe Ergänzen Sie das unvollständige Dialogmodell des *Movie Managers*, das Sie in der Datei 08-MovieManager-Dialogmodel.png, sowie in Jira finden. Gehen Sie dabei wie folgt vor:

- Betrachten Sie das unvollständige Dialogmodell in der Datei 08-MovieManager-Dialogmodel.png oder in Jira und vollziehen Sie es nach, indem Sie die Transitionen im *Movie Manager* ausführen.
- Führen Sie nun die Such-Funktion im *Movie Manager* aus und ergänzen Sie im Dialogmodell die fehlenden Zustände und Transitionen für die Suche. Sie können das Dialogmodell entweder per Hand ergänzen und dann einscannen oder Sie verwenden ein Zeichenwerkzeug ihrer Wahl, z.B. Paint, Visio, draw.io.

- Im Dialogmodell sehen Sie das Symbol . Erklären Sie, was dieses Symbol bedeutet, indem Sie ein Beispiel für eine Folge von Transitionen (Inkl. Angabe der beteiligten Zustände) im Dialogmodell angeben, bei der das Symbol eine Rolle spielt. Beschreiben Sie dieses Beispiel in einer Tabelle, bei der die Zeilen abwechselnd Zustände und Transitionen sind und geben Sie in einer zweiten Spalte konkrete Entsprechungen im *Movie Manager* an. Ein Beispiel für diese Darstellung ist Tabelle 1. Erklären Sie außerdem kurz, welcher Zustandsübergang durch das Symbol beeinflusst wird, und wie dieser aussehen würde, wenn das Symbol nicht da wäre. Halten Sie beides in einem PDF-Dokument fest.

Beispielabfolge von Zuständen und Transitionen im Dialogmodell	Konkrete Entsprechung im <i>Movie Manager</i>
Z: Movie Master View	Sicht Movie Master View wird angezeigt
T: select ,+‘	Tap auf den ,+‘-Button
Z: Movie Detail View Editmode	Sicht Movie Detail View im Editmode mit den Feldern zum Ausfüllen der Detail-Informationen für den zu erstellenden Film

Tabelle 1 – Beispiel für Darstellung von Zustand-Transition-Abfolge mit konkreter Entsprechung

Ergebnis:

Speichern Sie bitte Ihre Ergebnisse als .zip-Datei bis **Montag 16.12.2019 um 10.00 Uhr** in Moodle bestehend aus:

- 1x PDF- oder PNG-Datei mit Ihrem ergänzten Dialogmodell
- 1x PDF-Dokument mit Ihrer Erklärung.

Aufgabe 8.4: Wiederholung zu Kapitel 4**Präsenz:** Nein**Punkte:** 0**Team:** Nein

Diese Aufgabe dient der Wiederholung des Kapitels 4 der Vorlesung. Beantworten Sie in einem PDF-Dokument die folgenden Fragen anhand der Informationen aus dem Kapitel:

- Was ist ein Modell? Wodurch ist es charakterisiert?
- Was sind die Bestandteile einer Notation?
- Wozu verwendet man Modelle (im SWE)?
- Welche Arten von Diagrammen umfasst die UML?
- Was sind die wichtigsten Elemente der Klassendiagramme?
- Geben Sie 3 Beispiele für Codedetails an, die nicht im Klassendiagramm beschrieben werden können.
- Beschreiben Sie den Unterschied zwischen Klassen, abstrakten Klassen und Schnittstellenklassen.
- Wie kann Mehrfachvererbung in Java umgesetzt werden?
- Was sind die wichtigsten Elemente der Sequenzdiagramme?
- Welche Arten der Kommunikation werden unterschieden?
- Was ist der Unterschied zwischen einem Sequenzdiagramm und einem Kommunikationsdiagramm?
- Was sind die wichtigsten Elemente eines Zustandsdiagramms?
- Welche 3 Arten von internen Transitionen gibt es? Worin unterscheiden sie sich?
- Welche Konzepte erlauben die Beschreibung eines komplexen Zustands?
- Was ist ein Dialog? Was ist bei der Gestaltung von Dialogen zu beachten?
- Wie kann man Zustandsdiagramme zur Modellierung von Dialogen verwenden? Wie kann man Dialogmodelle aus Arbeitsbereich und Systemfunktionsbeschreibung ableiten?
- Was sind die Ziele des Entwurfs?

- Welche Entscheidungen sind bei der Umsetzung der Anforderungen im Entwurf zu treffen?
- Was ist Kohäsion bzw. Kopplung?
- Was ist eine komplexe Operation? Warum soll es möglichst wenig komplexe Operationen geben?
- Was versteht man unter OOAD und was sind die zwei Hauptschritte dabei?
- Erklären Sie den Unterschied zwischen Analyseklassenmodell und Entwurfsklassenmodell.
- Beschreiben Sie das Vorgehen zur Ableitung eines Analyseklassenmodells aus Systemfunktionen/ Use Cases und Arbeitsbereichen/Sichten.
- Beschreiben Sie das Vorgehen zur Verfeinerung des Analyseklassenmodells in ein Entwurfsklassenmodell.
- Was ist eine qualifizierte Assoziation? Wie wird sie implementiert?
- Was ist ein Muster? Wie wird es beschrieben? Was sind die Vorteile der Musterverwendung?
- Welche 3 Arten von Entwurfsmustern gibt es? Geben Sie ein Beispiel für jede Art an.
- Was ist Rationale?
- Welche Wissensbereiche gibt es in der Softwareentwicklung?
- Welche Konzepte und Beziehungen verwendet QOC, um Rationale zu beschreiben?
- Warum ist es wichtig, Rationale zu dokumentieren (welche Probleme vermeidet das, welche Vorteile hat das)?
- Warum ist es schwierig Rationale zu dokumentieren und wie kann man das erleichtern?
-
- Welche grundlegenden Systemkonzepte gibt es und welche Modellierungstechnik fokussiert auf welche Konzepte?
- Was sind die 3 Bereiche der aufgabenorientierten Systemmodellierung?
- Wie hängen in einem Projekt verwendete Modelle zusammen?

Ergebnis:

Speichern Sie bitte das PDF-Dokument mit den Antworten zu den Fragen bis **Montag, den 16.12.2019, um 10.00 Uhr** in Moodle.