

### 5. Übungsaufgabe: Softwaretechnik

Übung Nr. 5 Architekturmuster in der Softwarekonstruktion (MVC)

#### Vorbemerkung

Die folgende Übung dient der praktischen Umsetzung und Anwendung der vorgestellten Anforderungstechniken und Methoden aus der Vorlesung. Es dient dem Versuch das Architekturmuster MVC an einer Anwendung zu veranschaulichen.

Lesen Sie die Übungsaufgabe sorgfältig und erarbeiten Sie eine Lösung.

Abgabe in 2er Teams.

Abgabeform: 1 Java-Projekt in Archive-Form (Zip/Rar etc.)

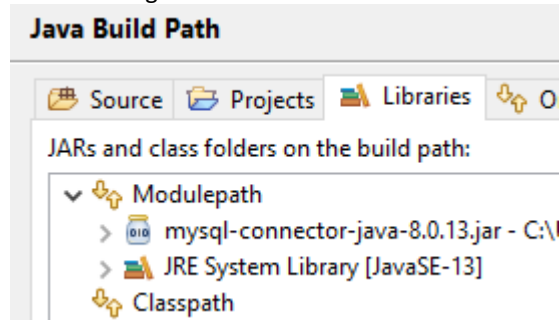
#### Vorbereitungen

Programme bedienen sich meist einer Datenbank für die Verwaltung größerer Datenmengen, diese Dienste werden oft mit dem jeweiligen Programm direkt ausgeliefert und im Hintergrund gestartet. Für die Durchführung der Übungsaufgabe benötigen Sie einen Datenbank-Dienst, laden Sie sich dazu die bekannte [Toolsammlung Xampp herunter](#)<sup>1</sup>.

Für die Versuchsvorbereitung benötigen Sie weiterhin folgende Tools die Ihnen u.a. im Aufgabenordner zur Verfügung gestellt wurden:

- MySQL J/Connector (jar Datei) Java Lib für die jdbc Schnittstelle

**Hinweis:** vergessen Sie nicht den JDBC-Treiber in Ihrem Build Path zur Verfügung zu stellen.



- Schemabild der Datenbank
- SQL-Datei zum Erstellen der Datenbankstruktur
- SQL-Datei zum befüllen der Datenbank
- Das Java Rahmenprogramm welches als Ausgangslage werden kann.

---

<sup>1</sup> Alternativ können Sie auch MariaDB-Server oder MySQL Server native herunterladen. Für die reine Entwicklung auf nicht produktiven Umgebungen wird jedoch im allg. das Xampp-Paket verwendet.

**Aufgabenstellung: Sakila Viewer (MVC)**

Sakila ist eine Datenbank die eine Zusammenstellung von Filmen, Schauspielern sowie möglichen Kategorien in einer Datenbank in Relation zueinander zur Verfügung stellt.

Entwerfen Sie nun im Folgenden ein grafisches Programm basierend auf AWT, welches die Inhalte der Datenbank nach den spezifizierten Anforderungen ausliest. Verfolgen Sie bei der Implementierung dabei den Ansatz nach MVC. Für die Kommunikation zwischen View und Controller kann das Beobachter Muster verwendet werden.

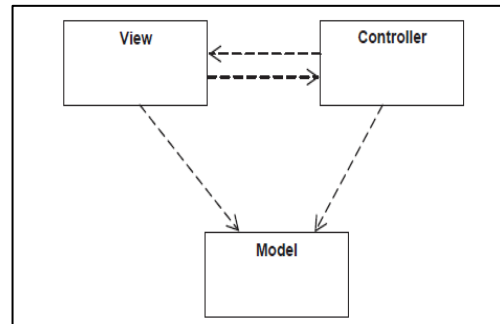


Abbildung 1: Model View Controller, Abhängigkeiten

Entwerfen Sie zunächst ein Datenbank-Model als Singleton-Klasse, welches alle erforderlichen Business-Operationen<sup>2</sup> für Ihr Programm bereitstellt. Weiterhin sind zusätzliche Model-Klassen erforderlich, orientieren Sie sich an folgenden Modelklassen<sup>3</sup> und erarbeiten Sie eine geeignete Klassenstruktur:

- Movie
- Actor
- Category

Folgende Operationen könnte Ihre Singleton-Model Klassen bspw. umsetzen:

- `Movie[] getCinemaMovies();`
- `Category[] getCinemaCategories();`
- `Movie[] getMoviesByCategory(Category catObject);`
- `Movie[] getMoviesByName(String pattern);`

**Hinweis:** Das Datenbank-Schema ist viel detaillierter und enthält mehr Information als für die Übungsaufgabe tatsächlich notwendig. Es wird daher von sogenannten OR-Mappern abgeraten. Kern der Aufgabe ist es Teilgebiete der Softwaretechnik umzusetzen:

- Erarbeitung und Identifizierung von Java Klassen auf Basis eines anderen Schemas (hier das DB-Schema).
- Umsetzung von Entwurfsmustern im Großen und im Kleinen.

<sup>2</sup> In diesem Zusammenhang ist die Geschäftslogik gemeint die von Eingaben oder Ausgaben des Nutzers losgelöst sind.

<sup>3</sup> Modelklassen sollten ggf. mit dem Iterator Muster ausgestattet werden.

 <b>Hochschule Harz</b>		
<b>Übung Nr.: 5</b>	Architekturmuster in der Softwarekonstruktion (MVC)	

Eine Datenbankverbindung wird wie folgt aufgebaut:

```
import java.sql.*;
...
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/sakila", "root", "");
```

Abfragen können u.a. wie folgt durchgeführt werden:

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM film");
```

#### **Folgende Anforderungen werden an die View gestellt:**

- eine zentrale Liste mit dem aktuellen aktiven Model Daten (die Filme).
- die aktuellen Modelldaten sollen über eine Combobox auswählbar sein können
  - d.h. die Kategorien die möglich sind (0: Alle, 1: Action, 2: Animation... etc.)
  - die Bestätigung der ComboBox<sup>4</sup>-Änderung soll mit einem Button realisiert werden.
- Ein Textfeld was bei Texteingabe zur Laufzeit ein Textmatching durchführt und nur Filme anzeigt die dem Textmuster entsprechen: (Aktivierung: ab 4 Zeichen)
  - ein leeres Textmuster zeigt entsprechend alle Filme an.
- Sollte ein Film angewählt werden so werden in einem extra Textbereich (TextArea), nur lesbar, die Beschreibung des Filmes geliefert.
- Weiterhin sind auf der Oberfläche 2 CheckBoxen die je nach Zustandsänderung die Listenelemente anpassen:
  - Zeige Film Veröffentlichungsjahr: Filmname (Jahr)
  - Zeige Filmlänge: Filmname [92min]
    - ➔ Bei Aktivierung aller Checkboxes: Filmname (1990) [82min]

**Hinweis:** Sie sind freigestellt die Oberfläche so zu gestalten wie Sie es für die Aufgabe als richtig erachten. Das betrifft das verwendete Layout und die Positionierung der UI-Elemente.

#### **Abschlussbemerkung:**

Das ist die letzte Aufgabe für das Modul Softwaretechnik. Bei Schwierigkeiten und Problemen wird in der nächsten Übungssitzung, wenn erforderlich auf diese eingegangen und diskutiert. Weiterhin wird in der Vorlesung das MVC Muster aus theoretischer Sicht ausführlich behandelt. Sie sollten damit in der Lage sein, eine Anwendung auf Basis von Entwurfsmuster wie MVC zu entwerfen.

**Viel Erfolg.!**

---

<sup>4</sup> In AWT werden ComboBox Elemente als Choice geführt.