

Matrikelnummer

--

Hinweise

- Bearbeitungszeit: 60 Minuten
- Maximale Punktzahl: 50
- Hilfsmittel: keine
- Benötigte Klassen- oder Schnittstellen-Imports müssen von Ihnen nicht explizit angegeben werden

Punkte

Aufgabe	Maximale Punktzahl	Erreichte Punktzahl
1	15	
2	20	
3	15	
Gesamt	50	

Viel Erfolg

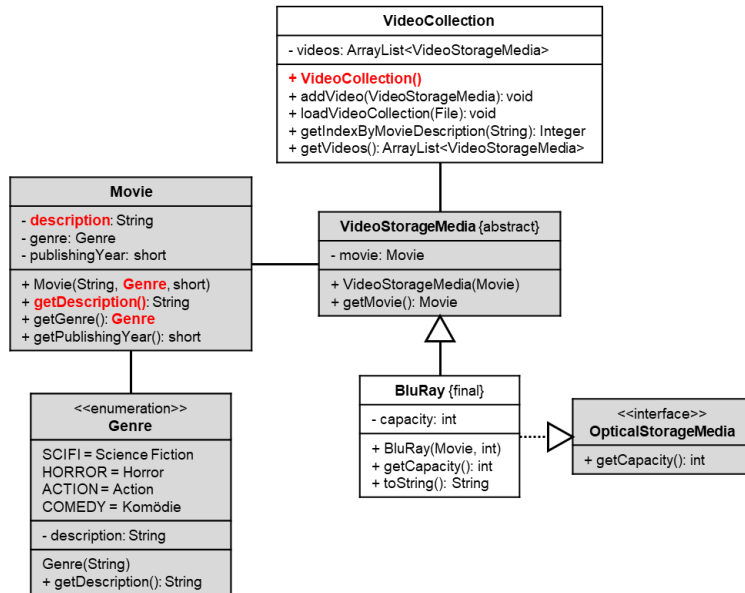
Aufgabe 1

- a) Erläutern Sie, was man in der Programmierung unter den Begriffen Upcast, Polymorphie und Downcast versteht (3 Punkte)
- b) Erläutern Sie kurz, ob eine abstrakte Methode in einer nicht-abstrakten Klasse definiert werden kann oder nicht (2 Punkte)
- c) Erläutern Sie kurz, was man unter einer Schnittstelle bzw. einem Interface versteht (2 Punkte)
- d) Skizzieren und erläutern Sie die Ereignisbehandlung in Java (3 Punkte) und benennen Sie das dazugehörige Entwurfsmuster (1 Punkt)
- e) Erläutern Sie kurz den wesentlichen Unterschied zwischen den Klassen *FileOutputStream* und *FileWriter* (2 Punkte)
- f) Erläutern Sie, was die Besonderheit der Schnittstelle *Serializable* darstellt (2 Punkte)

Aufgabe 2

- Erstellen Sie die Klasse *BluRay* anhand des abgebildeten Klassendiagramms (7 Punkte)
- Erstellen Sie die Klasse *VideoCollection* anhand des abgebildeten Klassendiagramms (13 Punkte)

Klassendiagramm



Hinweise zur Klasse *BluRay*

- der Konstruktor soll alle Attribute initialisieren bzw. mit den eingehenden Werten belegen
- die Methode *toString()* soll die Attribute der Klasse gemäß der nachfolgenden Formel zurückgeben
Formel: **<movie.description>**, **<movie.genre.description>**, **<movie.publishingYear>**, **<capacity>GB**
Beispiel: Avengers: Endgame, Action, 2019, 100GB

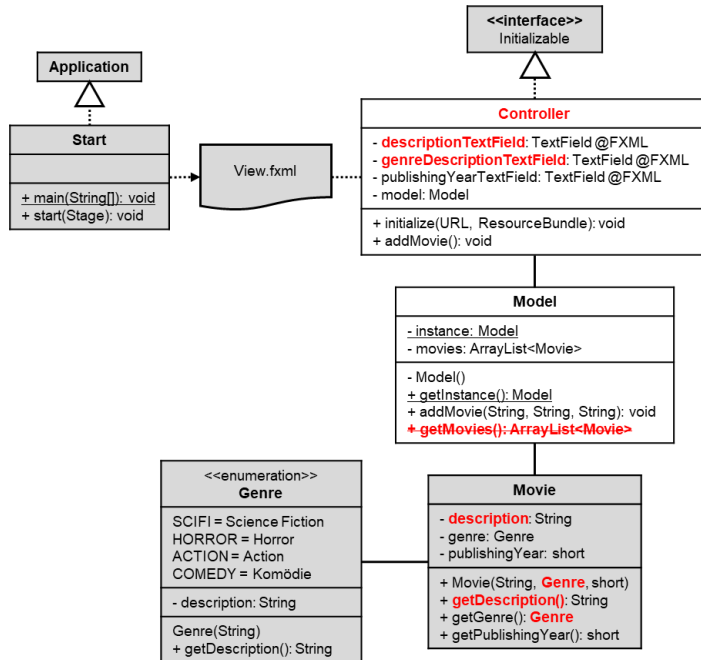
Hinweise zur Klasse *VideoCollection*

- der Konstruktor soll alle Attribute initialisieren bzw. mit den eingehenden Werten belegen
- ~~die Methode *addVideo(VideoStorageMedia)* soll der Videoliste das eingehende Video hinzufügen~~
- die Methode *loadVideoCollection(File)* soll **die Videoliste der eingehenden Datei auslesen und der Videoliste (Attribut *videos*) zuweisen**. Fangen Sie die Ausnahmen *IOException* und *ClassNotFoundException* ab und lösen Sie als Fehlerbehandlung die Ausnahme *VideoLoadException* aus. Hinweis: **Die Videoliste wurde** mit Hilfe der Standardserialisierung in der Datei gespeichert.
- die Methode *getIndexByMovieDescription(String)* soll den Index des Videos zur eingehenden Film-Beschreibung zurückgeben. **Im Fehlerfall soll der Wert *null* zurückgegeben werden**

Aufgabe 3

- Erstellen Sie die Klasse **Controller** anhand des abgebildeten Klassendiagramms (6 Punkte)
- Erstellen Sie die Klasse **Model** anhand des abgebildeten Klassendiagramms (9 Punkte)

Klassendiagramm



Grafische Darstellung

The screenshot shows a window titled "Film anlegen" with the following fields and controls:

- Beschreibung**: A text field containing "The Avengers".
- Genre**: A text field containing "Action".
- Erscheinungsjahr**: A text field containing "2012".
- Film anlegen**: A button at the bottom.

Hinweise zur Klasse **Model**

- der Konstruktor soll alle Instanzattribute initialisieren bzw. mit den eingehenden Werten belegen
- die Methode `addMovie(String, String, String)` soll aus den eingehenden Zeichenketten einen Film erzeugen und diesen der Filmliste hinzufügen
- das **Model** soll als Singleton implementiert werden

Hinweise zur Klasse **Controller**

- die Methode `initialize(URL, ResourceBundle)` soll das **Model** initialisieren
- die Methode `addMovie()` soll die Methode `addMovie(String, String, String)` der Klasse **Model** aufrufen und dieser die eingegebenen Werte der Eingabefelder übergeben