

Programmieren II – Programmrealisierung mit Java

1. Gegenstand der Prüfung

Es soll ein kleines Kino-Verwaltungssystem mit Hilfe der Standard-Java-Plattform (Java SE 8) entwickelt werden unter Berücksichtigung von Konventionen, Code-Qualität, Fehlertoleranz und mit Hinblick auf zukünftige Wartung.

Die genauere Aufgabenbeschreibung befindet sich auf Seite 2+3, bitte lesen Sie sich auch die detaillierteren Angaben zur Spezifikation gründlich durch. In der wesentlichen Implementierungsgestaltung sind Sie frei, genauere Vorgaben existieren aber für einige Elemente, die es zu beachten gilt!

1. Hilfsmittel

Während der Bearbeitung sind die Notebooks von LAN und WLAN und anderen Netzverbindungen zu trennen. Sie dürfen im Prinzip alle Ressourcen nutzen, die Sie offline verfügbar haben, insbesondere könnte die Java Dokumentation oder das Skript zur Vorlesung hilfreich sein. Sie können eine IDE Ihrer Wahl benutzen.

Die Abgabe per MOODLE: zur Abgabe des Resultats mit Hilfe des MOODLE-Systems dürfen Sie das Notebook kurz mit dem DHBW-Netz verbinden.

Technische Vorgaben: mindestens Java 8.

2. Bewertungskriterien

Zur Bewertung werden verschiedene Kriterien herangezogen:

- Geforderte Features umgesetzt?
- Korrektheit
- Umgang mit Fehlersituationen
- Code-Qualität, Konventionen, Redundanzfreiheit, best practices
- Funktionierendes Endresultat
- Alle notwendigen Dateien im abgegebenen Resultat enthalten?

3. Ergebnis

Das gesamte Endresultat inklusive notwendiger Eingabedateien packen Sie bitte in eine ausführbare JAR-Datei. **Geben Sie der JAR-Datei als Dateinamen Ihre Matrikelnummer, da das MOODLE-System anonym arbeitet. Ohne die Matrikelnummer ist das Ergebnis nicht zuordenbar.** Symbolisch wird im weiteren Verlauf die Datei hier aber `kino.jar` genannt.

Wichtig: Überprüfen Sie vor Abgabe, ob sich die Sourcen im erzeugten JAR befinden! Achten Sie darauf, wirklich die korrekte JAR-Datei hochzuladen. Achten Sie auf den Dateinamen.

4. Bearbeitungszeit

Für die Testklausur: 1 Stunde.

Aufgabenstellung: Filme im Kino

Es soll ein kleines Verwaltungssystem für Filme in einem Kino geschrieben werden. Als erste Aufgabe sollen dazu aktuellen Filme aus einer Datei eingelesen und nach Spieldauer sortiert wieder in eine andere Datei geschrieben werden. Weiterhin werden während des Programmlaufs noch Informationen ausgegeben: Film mit kürzester Spieldauer, Film mit längster Spieldauer, durchschnittliche Spieldauer aller Filme.

Hinweis: da zu diesem Zeitpunkt das Handling von Dateien noch nicht Gegenstand der Vorlesung war, beschränken wir uns darauf, die einzulesenden Daten in einem String-Array zu simulieren und die auszugebenden Daten auf dem Bildschirm darzustellen.

Die Bedienung des Programmes soll auf der Kommandozeile so erfolgen:

```
java -jar <ihrProgramm.jar>
```

Wird das Programm so gestartet, soll es mit Testdaten/Dummydaten laufen und sein Resultat auf dem Bildschirm ausgeben.

Beispiel (!) für den Aufruf:

```
C:\java>java -jar kino.jar
```

```
Die tönende Wochenschau :   15 Minuten
           Charade       :   99 Minuten
           Toy Story     :   99 Minuten
           Ghostbusters  :  104 Minuten
           Jenseits von Afrika : 238 Minuten
           Dr. Schiwago   :  300 Minuten
```

Information

Längste Spieldauer : Dr. Schiwago (300 Minuten)

Kürzeste Spieldauer: Die tönende Wochenschau (15 Minuten)

Durchschnittliche Spieldauer: 142,50 Minuten

Die "Eingabedatei" (das String-Array) enthält zeilenweise jeweils Titel und Spieldauer eines Filmes mit einem Trennzeichen getrennt. Das Format für die einzelne Zeile können Sie selbst frei wählen. Ein Beispiel:

```
String[] DATEN = { "Dr. Schiwago---300", "Ghostbusters---104", ... }
```

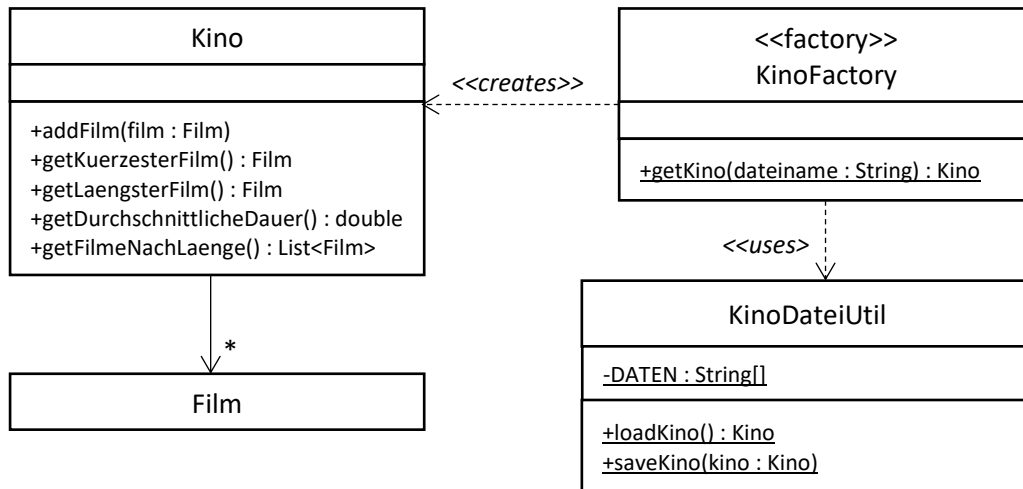
Die Ausgabe auf dem Bildschirm (s.o.) enthält ebenfalls Titel und Spieldauer, ist aber in aufsteigender Reihenfolge nach der Spieldauer der Filme sortiert. Bei gleicher Spieldauer wird nach Titel sortiert.

Der Sourcecode sollte nach gängigen Java-Konventionen - insbesondere für Groß- und Kleinschreibung und allgemeine Namensvergabe - geschrieben sein.

Sinnvoller Einsatz von Exception-Handling wird erwartet. Nutzen Sie weiterhin Annotation und Generics wo es sinnvoll erscheint. Achten Sie auf saubere Datenkapselung.

UML-Diagramm

Das folgende UML-Diagramm gilt als Rohentwurf bzw. grundlegende Vorgabe. Es ist nicht vollständig. Sie dürfen/sollten ggf. zusätzliche Methoden schreiben und hinsichtlich Attribute oder Konstruktoren werden gar keine Vorgaben gegeben.



Eine Klasse Start (nicht im Diagramm gezeigt) verbindet alles und steuert das Hauptprogramm.

Implementierungshinweise

Die Projektsprache ist Deutsch. Programmieridiome (wie get/set, add etc.) bleiben natürlich in Englisch.

Die Klasse `KinoFactory` benützt die Klasse `KinoDateiUtil` zum Einlesen der Datei. Da wir keine echten Dateien verwenden, soll `loadKino()` die Daten aus dem dargestellten String-Array "DATEN" empfangen (das simuliert das Einlesen einer Datei) und `saveKino()` den bereits angedeuteten Bildschirm-Output durchführen.

Verwenden Sie zum Einlesen der Daten die angebotenen Tools aus dem Paket `java.util.regex`.

Viel Erfolg und hoffentlich auch ein bisschen Spaß!