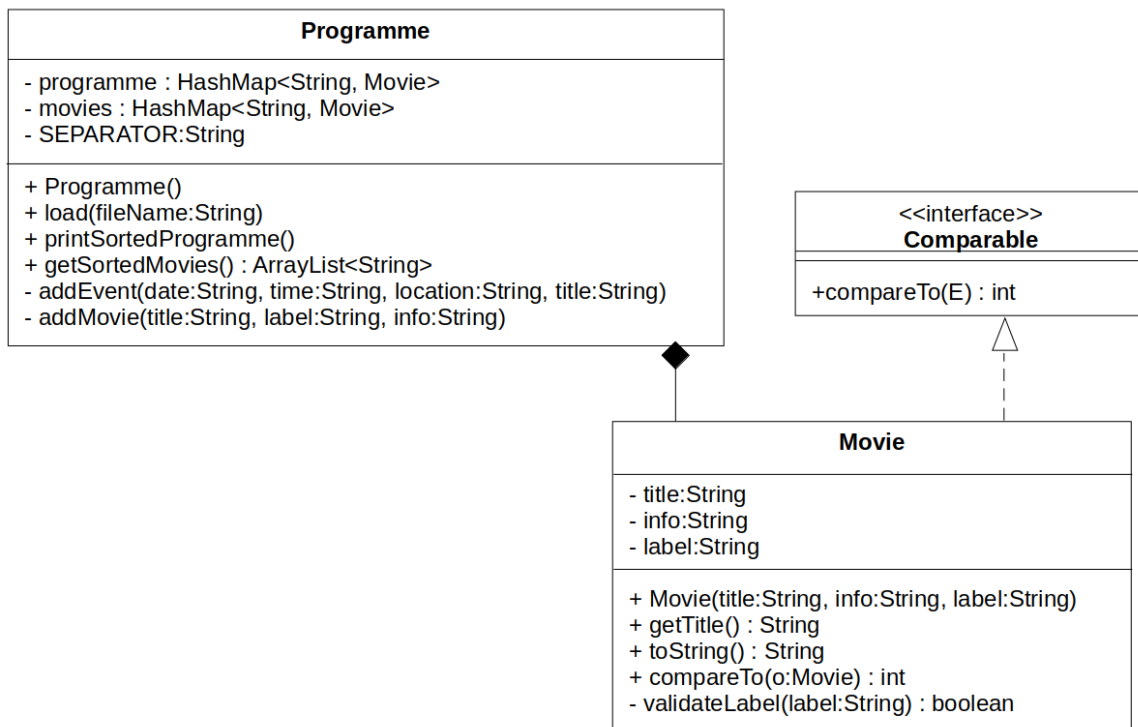


Lernstandserhebung 3

im Fach Programmieren II

Fachliche Inhalte:
Programmiersprachliche Möglichkeiten
bei der Modellierung und Programmierung von Java-Klassen,
Textdateien, Interfaces, Exceptions
Anwendung von ArrayLists und HashMaps

Das vorliegende Programmbeispiel modelliert das Filmprogramm eines Kinos. Lesen Sie den Code und verstehen Sie seine Funktion. Nutzen Sie dazu auch das vorliegende UML-Klassendiagramm.



```

package cinema;

public class Movie implements Comparable<Movie> {
    private String title;
    private String info;
    private String label;

    public Movie(String title, String info, String label) {
        this.title = title;
        if (validateLabel(label) == false)
            throw new IllegalArgumentException("Invalid label:"+label);
        this.info = info;
        this.label = label;
    }

    public String getTitle() {
        return this.title;
    }

    public String toString() {
        return this.title + " (" + this.label + "): " + this.info;
    }

    private boolean validateLabel(String label) {
        return label.matches("P[0-9]{1,2}");
    }

    public int compareTo(Movie o) {}
}
  
```

Prüfung des Labels zur Altersfreigabe mit Hilfe eines regulären Ausdrucks

```

package cinema;

import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;

public class Programme {

    private HashMap<String, Movie> programme = new HashMap<String, Movie>();
    private HashMap<String, Movie> movies = new HashMap<String, Movie>();

    private final String SEPARATOR = ";";

    public Programme() { }

    public void load(String fileName) {
        programme.clear();
        String line;
        BufferedReader reader;
        try {
            reader = new BufferedReader(new FileReader(fileName));

            while ((line = reader.readLine()) != null) {
                String[] arr = line.split(this.SEPARATOR);
                if (arr.length > 0) {
                    if (arr.length == 4)
                        addEvent(arr[0], arr[1], arr[2], arr[3]);
                    else
                        addMovie(arr[0], arr[1], arr[2]);
                }
            }
        } catch (Exception e) {
            System.out.println("Error reading programme file:" + fileName);
            e.printStackTrace();
        }
    }

    private void addEvent(String date, String time,
                          String location, String title) {
        String key = date + " " + time + ":" + location;
        Movie movie = movies.get(title);
        if (movie != null)
            programme.put(key, movie);
        else
            System.out.println("Unknown movie:"+title);
    }

    private void addMovie(String title, String label, String info) {
        Movie movie = new Movie(title, info, label);
        movies.put(title, movie);
    }

    public void printSortedProgramme() {
        ArrayList<String> allEvents = new ArrayList<String>(programme.keySet());
        Collections.sort(allEvents);
        for (String event : allEvents)
            System.out.println(event + " : " + programme.get(event));
    }

    public ArrayList<String> getSortedMovies() {
    }

    public static void main(String[] args) {
        Programme prog = new Programme();
        prog.load("Woche12.txt");
        prog.printSortedProgramme();
    }
}

```

Inhalt der Textdatei "Woche12.txt":
 Die Voegel;P16;Hitchcock-Klassiker
 Stolz und Vorurteil;P12;Literaturverfilmung
 Rubbeldiekatz;P6;Klassiker von Buck
 05/06/2025;22.30;Saal 1;Die Voegel
 05/06/2025;19.30;Saal 1;Die Voegel
 06/06/2025;20.00;Saal 2;Rubbeldiekatz
 05/06/2025;19.00;Saal 2;Stolz und Vorurteil

Aufgabe 1

Implementieren Sie das Interface Compareable der Klasse Movie, indem die Filmtitel alphabetisch verglichen werden.

Bewertung: PFHdr: / 3 Pkt. PFCmp: / 4 Pkt.

Aufgabe 2

Welche Probleme können beim Einlesen von Textdateien einen Programmabbruch auslösen? Nennen Sie die zwei grundlegenden Arten von Problemem!

Bewertung: CUFRd: / 2 Pkt.

Aufgabe 3

Schreiben Sie kurze Zeilenkommentare zu jeder Zeile der Methode addEvent der Klasse Programme!

```
private void addEvent(String date, String time,
                      String location, String title) {
    String key = date + " " + time + ":" + location;
    Movie movie = movies.get(title);
    if (movie != null)
        programme.put(key, movie);
    else
        System.out.println("Unknown movie:"+title);
}
```

Bewertung: PFComm: / 5 Pkt.

Aufgabe 4

Wie werden Daten in einer HashMap gespeichert?
Antworten Sie in wenigen aussagekräftigen Sätzen!

Bewertung: CUHaM: / 2 Pkt. CUDKl: / 2 Pkt. CUPuG: / 2 Pkt.

Aufgabe 5

In der Methode addEvent aus Aufgabe 3 wird im Fehlerfall eine Konsolenausgabe geschrieben. Welche andere Art der Fehlerdarstellung wäre möglich?
Programmieren Sie einen entsprechenden else – Code!

```
private void addEvent(String date, String time,
                      String location, String title) {
    String key = date + " " + time + ":" + location;
    Movie movie = movies.get(title);
    if (movie != null)
        programme.put(key, movie);
    else
```

Bewertung: PFErr: / 3 Pkt.

Aufgabe 10

Was macht die Methode setUpBeforeClass() generell?

Bewertung: CUStp: / 2 Pkt.

Aufgabe 11

Was macht die Implementierung von setUpBeforeClass() im oben gegebenen Quelltext?

```
package test;

import static org.junit.jupiter.api.Assertions.*;
import java.util.ArrayList;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;

import cinema.Movie;
import cinema.Programme;

class TCCinema {

    private static Programme prog = null;

    @BeforeAll
    static void setUpBeforeClass() throws Exception {
        prog = new Programme();
        prog.load("Woche12.txt");
    }

    @Test
    void testMovie() {
        String theTitle = "Terminator";
        Movie m = new Movie(theTitle, "very cool", "P16");
        String title = m.getTitle();
        assertEquals(title, theTitle);
    }
}
```

Bewertung: CUImp: / 2 Pkt.

Aufgabe 12

Implementieren Sie einen Test, der prüft dass die Sortierung der Methode getSortedMovies() der Klasse Programme für die eingeleseene Textdatei „Woche12.txt“ ein korrektes Ergebnis liefert!

Bewertung: PStest: / 6 Pkt.

Erreichte Gesamtpunktzahl: / 51 Pkt.