

Kino - Teil 1

Frank Hedecke, Ian List

1 Beschreibung

Es soll ein Kino mit mehreren Sälen simuliert werden.

Der Einlass für eine Voführung ist 15 Minuten vor Filmstart. Nach Filmende sind weitere 15 Minuten für die Säuberung eingeplant. Filmvorführungen in einem dürfen sich nicht überschneiden.

Das Kino erstellt ein Tagesprogramm mit allen Vorführungen, geordnet nach Kinosälen.

2 Aufgaben

1. Lies dir alle Hinweise durch.
2. Implementiere das Kino, sodass Vorstellungen hinzugefügt werden können, die sich nicht überschneiden. Außerhalb der Öffnungszeiten, finden keine Vorstellungen statt.
3. Das Kino soll ein Tagesprogramm erstellen, entsprechend der Vorlage.
4. Erfülle die zur Verfügung gestellten Tests.

3 Hinweise

- Die Klassen *Cinema* und *Movie* sind vorgegeben, müssen aber erweitert werden.
- Du benötigst Klassen für Zeitpunkte und -intervalle.
- Ein Film der von 14:30 bis 16:00 Uhr läuft und auch so im Programm angegeben ist, blockiert den Saal von 14:15 bis 16:15 Uhr.
- Die Vorlagen und die Testklassen gibt es auch im [Cloudstore](#).

4 Testklasse

```
1 import static org.junit.Assert.*;
2
3 import org.junit.After;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 public class CinemaTest {
8
9     //
10    // Tests fuer Aufgabenstellung Teil 1
11    //
12
13    Cinema cinema;
14
15    @Before
16    public void setUp(){
17        cinema = new Cinema(8, 0, 22, 0);
18    }
19
20    @After
21    public void tearDown(){
22        cinema = null;
23    }
24
25    @Test
26    public void testDayPlan() {
27        cinema.addHall();
28        cinema.addHall();
29        Movie m1 = new Movie("Movie1", 45);
30        Movie m2 = new Movie("Movie2", 90);
31        assertFalse(cinema.addScreening(m2, 1, 21, 0));
32        assertTrue(cinema.addScreening(m2, 1, 13, 0));
33        assertFalse(cinema.addScreening(m1, 1, 14, 30));
34        assertTrue(cinema.addScreening(m1, 1, 15, 0));
35        assertTrue(cinema.addScreening(m2, 2, 15, 0));
36        assertFalse(cinema.addScreening(m2, 3, 13, 0));
37        assertFalse(cinema.addScreening(null, 1, 10, 0));
38    }
39
40    @Test
41    public void testHalls(){
42        assertTrue(cinema.getNumberOfHalls() == 0);
43        cinema.addHall();
44        cinema.addHall();
45        cinema.addHall();
46        assertTrue(cinema.getNumberOfHalls() == 3);
47    }
48
49    @Test
50    public void timeTesting(){
```

```

51     assertEquals(cinema.getOpenTime(), "08:00");
52     assertEquals(cinema.getClosingTime(), "22:00");
53     cinema.addHall();
54     Movie m1 = new Movie("Movie1", 45);
55     cinema.addScreening(m1, 1, 13, 0);
56     String[] dayPlanHallOne = cinema.getProgram(1);
57     boolean correctness = false;
58     for(String s : dayPlanHallOne){
59         if(s.contains("13:00 ... 13:45 Movie1")){
60             correctness = true;
61         }
62     }
63     assertTrue(correctness);
64
65 }
66
67 }
68

```

5 Vorlage Klassen

```

1  public class Cinema {
2
3      public Cinema(int startH, int startM, int endH, int endM) {
4
5          /* Öffnungszeiten des Kinos
6           * von startH:startM Uhr bis endH:endM Uhr
7           */
8      }
9
10     public String getOpenTime(){
11         // z.B. "09:30"
12         return null;
13     }
14     public String getClosingTime(){
15         // z.B. "23:05"
16         return null;
17     }
18
19     public void addHall(){}
20
21     public int getNumberOfHalls() {
22         return 0;
23     }
24
25     public boolean addScreening(Movie m ,int hall, int startH, int
startM) {
26         // Startzeit des Films: startH:startM
27         return true;
28     }
29
30     public String[] getProgram() {

```

```

31         return null;
32     }
33
34     public String[] getProgram(int Hall) {
35         return null;
36     }
37
38 }
39
1  public class Movie {
2
3     private int runningTime;
4     private String title;
5
6     public Movie (String title, int runningTime) {
7         this.title = title;
8         this.runningTime = runningTime;
9     }
10
11     public int getRunningTime() {
12         return this.runningTime;
13     }
14
15     public String getTitle() {
16         return this.title;
17     }
18 }
19

```

6 Vorlage Tagesprogramm

```

1  --- Saal 1 ---
2
3  13:00 ... 14:30 Avengers
4  15:00 ... 17:30 Lord of the Rings
5
6  --- Saal 2 ---
7
8  09:30 ... 12:12 Harry Potter
9  13:00 ... 15:00 Wag the Dog
10 16:00 ... 18:30 Lord of the Rings
11
12
13 --- Saal 3 ---
14
15 no program
16

```