

## Aufgabe 5

Lösen Sie folgende Aufgabe in einer objektorientierten Programmiersprache Ihrer Wahl. Ein Regal habe 5 Fächer, in die je 30 Disks passen. Das Regal beinhaltet DVDs, CDs und BDs der Genres *Musik*, *Action*, *Komödie*, *Thriller* und *Fantasy*. Jede Disk ist mit 1 bis 10 bewertet, wobei 10 für sehr gut steht.

- Deklarieren Sie einen Aufzählungsdatentyp für den `Typ` der Disk. Deklarieren Sie einen weiteren Aufzählungsdatentyp für das `Genre` der Disk.

```
3 public enum Typ {
4     DVD, CD, BD
5 }

Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Typ.java

3 public enum Genre {
4     MUSIK, ACTION, KOMOEDIE, THRILLER, FANTASY
5 }
```

Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/examen/examen\_66116/jahr\_2014/herbst/regal/Genre.java

- Deklarieren Sie eine Klasse `Disk`, die den `Typ`, das `Genre`, die `Bewertung`, sowie den `Titel` der Disk speichert.

```
4 public class Disk {
5     private Typ typ;
6     private Genre genre;
7     private int bewertung;
8     private String titel;
9
10    /**
11     * Erzeuge eine neue Disk. Die Bewertung wird automatisch mit Hilfe
12     ↪ des Genres
13     * berechnet.
14     *
15     * @param typ Typ der Disk
16     * @param genre Genre der Disk
17     * @param titel Titel der Disk
18     */
19    public Disk(Typ typ, Genre genre, String titel) {
20        this.typ = typ;
21        this.genre = genre;
22        this.titel = titel;
23        this.bewertung = (int) erstelleStdBewertung(this.genre);
24    }
25 }
```

Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/examen/examen\_66116/jahr\_2014/herbst/regal/Disk.java

- Deklarieren Sie ein Array `Regal`, mit den Abmessungen des oben genannten Regals, das Disks enthält. Initialisieren Sie das Array als leeres Regal.

```
3 public class Regal {
4     private Disk[] [] regal = new Disk[5][30];
5
6     /**
7     * Erzeuge eine neues (leeres) Regal.
```

```

8      */
9      public Regal() {
10         for (int i = 0; i < regal.length; i++) {
11             for (int j = 0; j < regal[i].length; j++) {
12                 regal[i][j] = null;
13             }
14         }
15     }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2014/herbst/regal/Regal.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Regal.java)

- Initialisieren Sie die Bewertung einer Disk. Schreiben sie dazu eine rekursive Methode `erstelleStdBewertung`, der ein Genre einer Disk übergeben wird und die die Bewertungen für alle Disks nach folgenden Regeln bezüglich des Genres vergibt:
  - Eine Disk mit Genre Musik wird mit einer 3 bewertet.
  - Komödien werden mit 2 bewertet.
  - Thriller werden mit dem zweifachen einer Komödie bewertet
  - Ein Fantasy-Film wird mit dem 1,5 fachen eines Thrillers bewertet.
  - Ein Actionfilm wird wie ein Thriller bewertet.

```

41     public double erstelleStdBewertung(Genre genre) {
42         if (genre.equals(Genre.MUSIK)) {
43             return 3;
44         } else if (genre.equals(Genre.KOMOEDIE)) {
45             return 2;
46         } else if (genre.equals(Genre.THRILLER)) {
47             return 2 * erstelleStdBewertung(Genre.KOMOEDIE);
48         } else if (genre.equals(Genre.FANTASY)) {
49             return 1.5 * erstelleStdBewertung(Genre.THRILLER);
50         } else if (genre.equals(Genre.ACTION)) {
51             return erstelleStdBewertung(Genre.THRILLER);
52         } else {
53             return 0;
54         }
55     }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2014/herbst/regal/Disk.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Disk.java)

- Schreiben Sie eine Methode `mittlereBewertung`, die die mittlere Bewertung der Disks im Regal berechnet.

```

22     public double mittlereBewertung() {
23         int anzahl = 0;
24         int bewertungspunkteGesamt = 0;
25         for (int i = 0; i < regal.length; i++) {
26             for (int j = 0; j < regal[i].length; j++) {
27                 if (regal[i][j] != null) {
28                     anzahl++;
29                     bewertungspunkteGesamt += regal[i][j].gibBewertung();
30                 }
31             }
32         }
33         if (anzahl == 0) {
34             // Falls das Regal komplett leer ist.

```

```

35         return 0;
36     }
37     return bewertungspunkteGesamt / anzahl;
38 }
39 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2014/herbst/regal/Regal.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Regal.java)

### Additum Der komplette Code

```

3 public enum Typ {
4     DVD, CD, BD
5 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2014/herbst/regal/Typ.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Typ.java)

```

3 public enum Genre {
4     MUSIK, ACTION, KOMOEDIE, THRILLER, FANTASY
5 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2014/herbst/regal/Genre.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Genre.java)

```

3 @SuppressWarnings("unused")
4 public class Disk {
5     private Typ typ;
6     private Genre genre;
7     private int bewertung;
8     private String titel;
9
10    /**
11     * Erzeuge eine neue Disk. Die Bewertung wird automatisch mit Hilfe des
12     ↪ Genres
13     * berechnet.
14     *
15     * @param typ Typ der Disk
16     * @param genre Genre der Disk
17     * @param titel Titel der Disk
18     */
19    public Disk(Typ typ, Genre genre, String titel) {
20        this.typ = typ;
21        this.genre = genre;
22        this.titel = titel;
23        this.bewertung = (int) erstelleStdBewertung(this.genre);
24    }
25
26    /**
27     * Gib die Bewertung der Disk zurück.
28     *
29     * @return Bewertung
30     */
31    public int gibBewertung() {
32        return this.bewertung;
33    }
34
35    /**
36     * Berechne rekursiv die Bewertung einer Disk.

```

```

37     * @param genre Genre der Disk
38     *
39     * @return Bewertung
40     */
41     public double erstelleStdBewertung(Genre genre) {
42         if (genre.equals(Genre.MUSIK)) {
43             return 3;
44         } else if (genre.equals(Genre.KOMOEDIE)) {
45             return 2;
46         } else if (genre.equals(Genre.THRILLER)) {
47             return 2 * erstelleStdBewertung(Genre.KOMOEDIE);
48         } else if (genre.equals(Genre.FANTASY)) {
49             return 1.5 * erstelleStdBewertung(Genre.THRILLER);
50         } else if (genre.equals(Genre.ACTION)) {
51             return erstelleStdBewertung(Genre.THRILLER);
52         } else {
53             return 0;
54         }
55     }
56 }
57

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2014/herbst/regal/Disk.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Disk.java)

```

3     public class Regal {
4         private Disk[][] regal = new Disk[5][30];
5
6         /**
7          * Erzeuge eine neues (leeres) Regal.
8          */
9         public Regal() {
10             for (int i = 0; i < regal.length; i++) {
11                 for (int j = 0; j < regal[i].length; j++) {
12                     regal[i][j] = null;
13                 }
14             }
15         }
16
17         /**
18          * Berechne die mittlere Bewertung für ein Regal.
19          *
20          * @return mittlere Bewertung
21          */
22         public double mittlereBewertung() {
23             int anzahl = 0;
24             int bewertungspunkteGesamt = 0;
25             for (int i = 0; i < regal.length; i++) {
26                 for (int j = 0; j < regal[i].length; j++) {
27                     if (regal[i][j] != null) {
28                         anzahl++;
29                         bewertungspunkteGesamt += regal[i][j].gibBewertung();
30                     }
31                 }
32             }
33             if (anzahl == 0) {
34                 // Falls das Regal komplett leer ist.
35                 return 0;
36             }
37             return bewertungspunkteGesamt / anzahl;
38         }
39     }

```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66116/jahr_2014/herbst/regal/Regal.java`