HW 04: Movie Comparator

Release date: Friday, September 16th, 2016
Due date: Friday, September 23th, 2016 – 11:59 pm

Goals

- Model a real world problem using Java classes and methods.
- Use selection to guide the control flow of a program.

Description

The staff of the Goodrich Wabash Landing Movie Theater is trying to decide on which movie to show on a special event next Friday. They hired you to write a program that helps them compare each pair of candidate movies.

To perform this task, you should develop a class called "`Movie`". Each movie entity is modelled as an Object of this class. A movie has a number of attributes that are represented as data members of the Object that represents the movie. These include the movie's rating as set by the professionals, the average of the ratings given by the users and the number of these ratings.

Given two movies, you are required to supply a Class method that compares two Movie Objects. Details on the `Movie` data members, Class methods and Object methods are given in the Requirements Section.

Requirements

You are required to develop a public Class called "`Movie`". The data members of this Class are shown in Table 1. At the time of the creation of a new `Movie`, the values of all its data members should be initialized using the constructor arguments. The prototype of the constructor is, thus:

```
public Movie(String name, double criticRating, double usersRating, int
                                numUsersRatings)
```

The constructor checks that the values supplied for `criticRating` and average `usersRating` are in the proper range from 1 (inclusive) to 5 (inclusive). If any of the values does not lie in the correct range, the constructor sets the corresponding variable to a default value equal to 5.

Since all data members have private access modifiers, you should supply accessor methods for each data member. The names of these methods are shown in Table 2.

A user can supply a rating for the movie. In this case, the value of numUsersRatings should be incremented and usersRating should be updated accordingly. A public Object method named "addUserRating" should be implemented in the Movie class. This method takes an integer value as an argument. The method checks if this value lies between 1 (inclusive) and 5 (inclusive). If this is the case, the usersRating and the numUsersRatings of the Object are updated accordingly and a true value is returned to the caller. Otherwise, the method does not change the values of the data members and returns false. The signature of the addUserRating method is:

```
public boolean addUserRating(int newRating)
```

A Class method should be implemented in the class Movie that compares two movies. The prototype of this method is:

```
public static int compareMovies(Movie movie1, Movie movie2)
```

Applying a specific comparison criteria, the method decides on which movie is better than the other. It returns the value 1, if movie1 is better than movie2. It returns the value 2, if the movie2 is better than movie1. Otherwise, it returns a zero.

The decision of compareMovies on which movie is better is based on the values of the criticRating and the usersRating of the two movies. The following decision rules are followed.

1. A movie that has higher criticRating, and higher or equal usersRating should be considered better than the other movie.

2. In case a movie has a higher criticRating and a lower usersRating than the other movie, a score is computed for each of the two movies as follows:

```
smartScore = 0.5 * criticRating + 0.3 * usersRating + 0.1 * reviewRange
```

   This score is called the smartScore. The value of reviewRange of a movie is determined based on the value of numReviews of the movie as shown in Table 3. The movie with higher smartScore is considered better than the other movie. If the smart scores of both movies are equal, the two movies are considered equally good.

3. If the two movies have equal values of criticRating, the movie that has higher usersRating is considered better than the other. If the values of usersRating of the two movies are equal, the movies are considered equally good.

Data Member Definition	Contents
`private String name`	Contains the name of the movie
`private double criticRating`	Represents a rating given to the movie by professionals. The range of the value of this variable is [1, 5]
`private double usersRating`	Represents the average of the ratings of the movie given by the users who watched the movie. The range of the value of this variable is [1, 5]
`private int numUsersRatings`	Contains the number of users who supplied a rating for this movie. This is an integer number greater than or equal to 0.

Table 1: Movie Object data members

Accessor Method Prototype	Description
`public String getName()`	Returns the name of the movie
`public double getCriticRating()`	Returns the rating of the movie as set by the professionals
`public double getUsersRating()`	Returns the average users rating
`public int getNumUsersRatings()`	Returns the number of ratings given by the users

Table 2: Accessors in the Movie Class

numUsersRatings	[0, 1000]	[1001, 5000]	[5001, 10000]	[10001, 15000]	[15001, 20000]
reviewRange	1	2	3	4	5
numUsersRatings	[20001, 25000]	[25001, 30000]	[30001, 50000]	[50001, 100000]	> 100000
reviewRange	6	7	8	9	10

Table 3: The value of `reviewRange` corresponding to `numUsersRatings` of a movie

Rubric

- Correct implementation of the constructor: 20 point
- Correct implementation of the accessor methods: 10 points
- Correct implementation of the addUserRating method: 30 points
- Correct implementation of the compareMovies method: 40 points

Submission Instructions

Submit your files to Vocareum by following these instructions. Keep in mind that only your last submission will be considered.

Good Luck