CENG 202 Data Structures

Spring 2015

HOMEWORK-1

Due: 25 March 2015, 23:59

RULES

1. Late submission is not allowed.

2. **There is no teaming up.** The homework has to be done individually. In case of cheating, **all involved will get zero. You cannot make any objections to a grade 0. Your emails will not be responded in this case and your grade will never change for both sides**. Be very careful and honest about this rule!

3. Be careful about input/output specifications (do not print any unnecessary characters, whitespaces.) Your homeworks will be read automatically.

4. Codes not compiled will directly get 0.

5. Submit your homework electronically through MOODLE. No excuse for late submission even for 1 minute. In such a case, dont try to send emails to teacher/asistants. Emails will not be accepted and responded.

In this homework you will create a movie database application. You should implement the requested methods and use the given interface.

Problem Definition

1. The database keeps records of users, movies, and ratings of movies. The information kept for each of them is given in Table below.

User	Movie	Rating
username(unique) age gender	name(unique) genre year director actors	moviename username score

Table 1. Database

2. The actions provided by the program will be:
    a. add a new user
    b. delete a user

c. add a movie
d. delete a movie
e. add actors to a movie
f. rate a movie (using movie name and user name)
g. list movies (with their additional information, actors and average score)
h. list ratings of a movie

3. Methods will throw the following exceptions (***Throw exceptions by writing the following exception messages.).*** The exceptions that the methods can throw are provided in Table 2.
    a. Movie Already Exists
    b. User Already Exists
    c. Movie Does Not Exist
    d. User Does Not Exist
    e. File Not Exists

METHOD	POSSIBLE EXCEPTIONS
addMovie	Movie Already Exists
deleteMovie	Movie Does Not Exist
rateMovie	Movie Does Not Exist
listMovies	-
addActor	-
addUser	User Already Exists
deleteUser	User Does Not Exist
listRatings	-
getMovieDataFromFile	File Not Exists

Table 2. Exceptions that the methods can throw

4. You should define one class for each database item, namely, movie, rating and user.
5. You should have a class named **HW1.java** which implements the interface MovieAppInt, whose definition is given below. Directly copy MovieAppInt in your project file.
6. HW1.java should keep the database, specifically, an ArrayList of all movies, ratings and users.
7. **You shouldn't change the definition of the MovieAppInt.**
8. **HW1.java should implement MovieAppInt. You can add extra methods to HW1.java.**
9. **Your HW will be tested automatically by calling the methods in HW1.java that also exist in MovieAppInt.**
10. You are expected to implement all the methods defined in MovieAppInt. Every method will have a distinct credit.
11. You can define extra classes.
12. You should strictly obey input/output specifications, otherwise you cannot get credit.
13. listMovies(), listRatings() are the two methods that will generate output and they should write the output to the file taken as the argument in the format given below.
14. getMovieDataFromFile() will take a file name which contains movie information in the following format:

a. At each line one movie data will exist
b. For each movie <name, genre, year, director, actors> information will be given in order. Each of these items will be seperated by a semicolon(;) and each actor will be seperated by comma(,).
c. Example input file:

input.txt

Wolverine;Action;2013;James Mangold;Hugh Jackman, Tao Okamoto, Brian Tee, Rila Fukushima
The Usual Suspects;Crime;1995;Bryan Singer;Kevin Spacey, Gabriel Byrne, Chazz Palminteri

15. For questions about the homework, private emails **will not be responded by teacher/asistants**! Ask questions to the mailgroup.
16. Good luck!

listMovies(String fileName)

1. for each movie, prints the following data in order, each one in a seperate line:
   - movie name
   - director
   - year
   - genre
   - average rating
   - actors (separated from each other by a comma)
2. prints an extra new line character between two movies
3. Example output file for the call listMovies("c:/temp/output.txt"):

C:/temp/output.txt

Wolverine
James Mangold
2013
Action
8.3
Hugh Jackman, Tao Okamoto, Brian Tee, Rila Fukushima

The Usual Suspects
Bryan Singer
1995
Crime
8.7
Kevin Spacey, Gabriel Byrne, Chazz Palminteri

listRatings()

1. prints ratings of a movie.
2. on each line there will be user name and score given by this user separated by a comma

3. Example output file for the call listMovies("Saw 2","c:/temp/output.txt"):

C:/temp/output.txt

Ali 8.2
Veli 6.1
Elif 2.6

```
public interface MovieAppInt {
  //movie functions
  void addMovie(String movieName, int year, String genre, String directorName);
  void deleteMovie(String movieName);
  void rateMovie(String movieName, String userName, int rate);
  void listMovies(String fileName);
  void addActor(String movieName, String actorName);
  void getMovieDataFromFile(String fileName);

  //user functions
  void addUser(String username, int age, String gender);
  void deleteUser(String name);

  //rating functions
  void listRatings (String movieName, String fileName);
}
```