CS 2410 – Fall 2017
Assignment #3 – Super Lame Movie Directory

Introduction

The purpose of this assignment is to get experience:

- Designing and using basic classes
- Using Scanner for Commandline I/O
- Use Dialogs for GUI I/O
- Use FileReader and PrintWriter for file I/O
- Formatting output

Tasks

Create one project for this assignment. There are two tasks that you must complete in this project. Place each task in an appropriately named package as stated in each task. Note that there will be overlap in the two tasks. It will be explained.

In order to receive full credit your code must:

Follow appropriate class conventions

Task 1 – Super Lame Command Line Directory

Your job is to create a movie directory that can be managed from the command line. It will be very simple. A user will be presented with a menu that gives the following options:

1. List directory contents
2. Add movie to directory
3. Display average rating
4. Quit program

After selections 1-3 are completed, the program should give this menu prompt again.

Directory File

The directory should be stored in a file called cs2410-directory.data inside the data folder (you should know what this means if you watched the class setup videos). Each individual movie record will be stored as a set of two lines. You do not need to sort the records. Each record is of the following format:

<Movie Title>
<Rating> <Year Released> <Review (number of stars)> <Month> <Day> <Year>
//Last three are for last day watched. Each item separated by space

You can assume that the file is formatted properly (that means you do not have to worry about checking that each item is formatted properly when reading in).

Listing Directory

When listing the directory, it should be formatted as follows:

Star Wars PG 1977 5 stars 1/5/2018

Each column should be aligned properly. The title, rating, and year should be left aligned; the number of stars and date right aligned. For information about formatting you can see pages 146-148 in the Liang book, or read on your own about printf formatting.

Adding to Directory
To add a movie to the directory, complete individual prompts asking for title, rating, release year, and date last watched (can be done with month, day, and year separately). You do not need to check for errors (although you would in a real program).

You should immediately save the new movie info to the directory file. Then print a confirmations message such as the following.

> The following movie has been added to the directory:
> Harry Potter and the Goblet of Fire (2005) PG-13
> Stars: 5
> Last Watched: 1/22/2018

Display Average Stars
Simply calculate the average age of all movies in the directory, and print a message such as:

> Your average movie ratings is 3.7 stars

There is a purpose in asking you to calculate this average. You may run into trouble. There is a very simple solution, but the issue often goes unnoticed until you get an error. Try to figure it out (it'll be simple if you come to class!)

Quit
End the program.

Organization
This main method should start the constructor of a class called CommandDirectory. This class should be contained in a package called "command". Of course, you know that what that really means is the entire package is cs2410.assn3.command, so I didn't even really have to mention that.

The main method can be in that same class, or in a class all by itself. If it is in a class by itself, please call that class Main, so that it is easy for the TA to know how to run your program.

Task 2 – Super Lame GUI Directory

For this task you are going to do the exact same thing, but with a GUI interface. All input and output from your program will be done through Dialogs. The behavior will be exactly the same. There should be a main method that starts this program. Same instructions as above.

Organization
This main method should start the constructor of a class called GUIDirectory. This class should be contained in a package called "gui".

The main method can be in that same class, or in a class all by itself. If it is in a class by itself, please call that class Main, so that it is easy for the TA to know how to run your program.

STOP – Collaborate and Listen!

The naïve student will jump in and write two completely different programs. The novice student will notice that the two tasks are similar and will do a bunch of copy and paste. The clever student will realize that this is just a single program, but gives the user two different options to view and interact with data.

So, what does that mean? Well…I'm going to let you figure that out. Use your experience from CS1/CS2 to help you. You actually already know what to do. Remember, one of the key advantages we get from

Object-Oriented Programming is code reuse.

I will give you a hint though. You should add one more package called “directory”.

Grading

Your score will be based on the following:

1. Proper conventions followed – 5 pts
2. Proper Javadoc conventions – 5 pts
3. Appropriate package structure – 5pts
4. Programs start properly – 5pts
5. Read file properly – 10 pts
6. Add to file properly – 10 pts
7. Calculate star average properly – 10 pts
8. Command Line
    a. Menu behaves appropriately – 5pts
    b. Properly formatted output – 10 pts
    c. Input behaves appropriately – 10pts
9. GUI
    a. Menu behaves appropriately – 5pts
    b. Properly formatted output – 10pts
    c. Input behaves appropriately – 10pts

What/How To Turn In

Submit your files on Canvas according to class conventions.

Due: February 2, 2018