

CSE 2102: Introduction to Software Engineering

Inheritance & Polymorphism

Assigned: March 2, 2023, Due: March 23, 2023

Consider the class `Movie` in Homework #4.

1. Define a class `Documentary` that is derived from the `Movie` class. `Documentary` inherits the name, MIPAA rating from the `Movie` class. In addition, `Documentary` contains TV rating of the type `String`. For the `Documentary` class, implement:

- A constructor that creates an instance of a documentary by accepting inputs for all the instance variables: -- name, MIPAA rating, TV rating.
- Accessor and mutator methods for all the instance variables. Bear in mind that some instance variables and methods could be inherited from the class `Movie`.
- `writeOutput()` method that prints the values of all the instance variables for a given documentary.

2. Extend the class `Documentary` to `DocumentarySeries`. `DocumentarySeries` inherits the name, MIPAA rating from the `Movie` class, and TV rating from the `Documentary` class. In addition, `documentarySeries` adds two variables -- number of seasons, and number of episodes of type `int`. For the `DocumentarySeries` class, implement:

- A constructor that creates an instance of a documentary Series by accepting inputs for all the instance variables: -- name, MIPAA rating, TV rating, number of seasons, and number of episodes.
- Accessor and mutator methods for all the instance variables. Bear in mind that some instance variables and methods could be inherited from the classes `Movie` and `Documentary`.
- `writeOutput()` method that prints the values of all the instance variables for a given documentary series.

Write a test client `TestDocumetaries.java` that tests the implementation of the `Documentary` and `DocumentarySeries` classes. This test client prompts for a data file which contains the details of each documentary or a documentary series; one per line. The attributes on each line are separated by commas. There is no special indicator to distinguish between a documentary and a documentary series, except that the documentary series will have two extra attributes. For example, considering the following two lines, `Alone` is a documentary series due to the presence of two extra attributes – it has 9 seasons and 107 episodes, whereas `The Tindler Swindler` is a documentary due to the absence of these two attributes.

`Alone, R, TV-14, 9, 107`

`The Tindler Swindler, R, TV-MA`

The test client prints an aggregate report summarizing the contents of the data file. This aggregate report will print the total number of documentaries which are not documentary series,

the total number of documentary series, the total numbers for each MPAA and TV-Ratings, documentary series with the greatest number of seasons, and documentary series with the greatest number of episodes. The set of MPAA and TV ratings are not known in advance, and will be discovered on the fly from the contents of the data file. For example, the data file above results in the following output:

Number of Documentaries: 1

Number of Documentary Series: 1

Number of R Documentaries: 2

Number of TV-14 Documentaries: 1

Number of TV-MA Documentaries: 1

Alone is the documentary series with highest number of seasons – 9

Alone is the documentary series with highest number of episodes – 107

OUTPUT:

```
Output - Movie (run)
run:
Number of Documentaries: 1
Number of Documentary Series: 1
Number of R Documentaries: 2
Number of TV-14 Documentaries: 1
Number of TV-MA Documentaries: 1
Alone is the documentary series with highest number of seasons - 9
Alone is the documentary series with highest number of seasons - 107
```

Submission

The following deliverables must be submitted on HuskyCT by midnight on March 23, 2023.

- Java and class files, Java files should be well-documented.
- At least 2 test cases that you used to test the code. Each test case will be a txt file. The output from each test case should be submitted as a screenshot as the one shown above.
- Please make sure that your code compiles, we will test your code offline with specific test cases (common to all).
- Late submissions (without any legitimate excuse) will incur a penalty of 10% per day.