

CSD 3464 – ASSIGNMENT 05 (Question 3)

Overview:

The following question is an **extension** to *Absolute Java* (6th Ed.)'s Chapter 08 Programming Project Q3 (pg. 518-519). Please follow the instructions included in **this** document and implement the following Java files:

- ⇒ RentalTester.java (*Contains a main () method*)
- ⇒ Rental.java
- ⇒ Movie.java
- ⇒ Action.java
- ⇒ Comedy.java
- ⇒ Drama.java

The above classes/files should both be inside a package called *q3*.

Instructions

Step 01 – Movie.java:

Start by reviewing the partially completed `Movie` class provided. This is the class should act as the starting point for this assignment.

As you will notice `Movie` has a public static enum field called `RatingSystem` that is used to store the Motion Picture Association of America (MPAA) ratings (i.e. G, PG-13, etc.). The movie class has three additional protected fields to store the `Movie`'s `idNumber`, `title`, and `rating`. Furthermore, `Movie` has a three-argument and copy constructor provided to you.

Inside the `Movie` class you will also notice a set of method stubs that you have been tasked to complete. **You must implement** the class' `toString()`, `equals()`, and `calculateLateFees()` methods as defined below.

- `toString()` should return a `String` in the format:
"Movie Title: " + `title` + "\nID Number: " + `idNumber` + "\nRating: " + `rating`
- `equals()` takes any `Object` and returns `true` if and only if this movie and the provided movie `Object` have the same `idNumber`
 - **HINT:** See `equals()` method on page 512 of textbook
- `calculateLateFees()` takes the number of days late the `Movie` is as an `int` and returns the total fees incurred based on the **default** rate of \$2.00 a day.

Lastly, you are required to add the getters and setters for all three protected fields of the `Movie` class.

- `getIdNumber()` should return `idNumber`
- `setIdNumber()` should update the value of `idNumber`
- `getTitle()` should return `title`
- `setTitle()` should update the value of `title`
- `getRating()` should return the value stored in the `rating` field.
- `setRating()` which takes a parameter of type `RatingSystem` and assigns the value to the `rating` field.

Step 02 – `Action.java`, `Comedy.java`, `Drama.java`:

Once you have completed the `Movie` class in full you are now ready to implement three classes which **extend** it: 1) `Action`, 2) `Comedy`, and 3) `Drama`. All three classes should have single three-argument constructor that takes an `int idNumber`, `String title`, and `RatingSystem rating` and makes a call to its superclass constructor via `super()`.

Additionally, the three classes should override the `Movie` class' `calculateLateFees()` method as defined below:

- `Action` movies have a late fee of \$3.00 a day
- `Comedy` movies have a late fee of \$2.50 a day
- `Drama` movies have a late fee of \$2.00 a day

Step 03 – `Rental.java`:

Next you are required to create a public `Rental` class. The rental class will contain protected `Movie movie` field that represents the movie rented. Additionally, the class will contain a protected `int` field to store the `customerID` of the individual who rented the movie and another protected `int` field to store a positive integer representing the `numDaysLate` the movie is past due its rental date.

The `Rental` class should contain a three-argument constructor that takes a `Movie movie`, `int customerID`, and an `int numDaysLate` and assigns them to the appropriate field.

Additionally, `Rental` class should have a copy constructor that takes a `Rental rental` object and creates a **deep copy** of it.

- Ensure you use the `Rental` class' copy constructor within the `Movie`'s copy constructor

Furthermore, the `Rental` class should have a method called `calcLateFees` that returns a `double` value indicating the late fees owed for the movie based on the `numDaysLate` and by calling `movie's calculateLateFees()` method.

Lastly, include the getters and setters for all three protected fields in the `Movie` class.

- `getMovie()` should return a copy of the `movie` field
- `setMovie()` can directly assign the reference of the passed in `Movie` object to the `movie` field
- `getCustomerID()` should return `customerID`
- `setCustomerID()` should update the value of `customerID`
- `getNumDaysLate()` should return `numDaysLate`
- `setNumDaysLate()` should update the value of `numDaysLate`

Step 04 – RentalTest.java()

The final step in this assignment is to create a class `RentalTester` that contains a `main` method as well as a static `double lateFeesOwed` method.

- At the beginning of the `main` method create:
 - `Action` movie object with the following properties:
 - `idNumber: 12345`
 - `title: John Wick`
 - `rating: R`
 - `Comedy` movie object with the following properties:
 - `idNumber: 3333`
 - `title: 21 Jump Street`
 - `rating: PG-13`
 - `Drama` movie object with the following properties:
 - `idNumber: 6789`
 - `title: Spotlight`
 - `rating: R`
- Next, create the following three `Rental` objects:
 - A `Rental` object named **actionRental** with the following properties:
 - `customerID: 111`
 - `movie: The Action movie you created earlier in the main`
 - `numDaysLate: 2`
 - A `Rental` object named **comedyRental** with the following properties:
 - `customerID: 111`
 - `movie: The Action movie you created earlier in the main`
 - `numDaysLate: 3`
 - A `Rental` object named **dramaRental** with the following properties:
 - `customerID: 111`
 - `movie: The Action movie you created earlier in the main`
 - `numDaysLate: 1`
- Thirdly, add the above `Rental` objects to an array object of the appropriate size.
- Lastly, call the static `double lateFeesOwed` method in this class, passing the array of `Rental` objects, to calculate the total late fees owed. Display this amount to the console.