S21 Challenge: Movies

Description: Write a Java application in NetBeans that utilizes a class written for the application called **Movie**. The application creates instances of **Movie** objects and utilizes its members such as properties, methods, and fields.

Purpose: To gain experience in creating classes and writing a program that creates object instances of the class.

Requirements:
Project Name: <Pawprint>MoviesS21

For the project name, follow the same naming scheme used in the first challenge. The project name is to be comprised of your pawprint, with the first letter capitalized, followed by MoviesS21. For example, if the pawprint is abcxyz9 the project is to be named Abcxyz9MoviesS21.

Create a Java Enumeration called **Genre** with the following values:

- ACTION, ANIMATION, COMEDY, DRAMA, FANTASY, HORROR, ROMANCE, SCI_FI (i.e., Science Fiction), SUSPENSE, WESTERN, UNKNOWN

Create a class called **Movie** that contains the following:

- Nine private fields: A String called **name**, a String called **director**, a String called **summary**, a float called **rating** with an initial value of 0.0f, a double called **revenue** with an initial value of 0.0, a Genre called **genre**, a Date called **releaseDate** with a default value of October 14, 1888; a String called **runtime**, and an int called **version**.
- You should have created a Java Enumeration in the previous step, so the type Genre should work for the variable **genre.**
- A public class field called **numOfMovies,** which is an int, and has a default value of 0.
    - Go on Oracle docs and read the differences between class fields and instance fields
- Four constructors that chain each other:
    1. The first constructor is the no-arg constructor which sets name and director to empty strings, sets the version to 0, and increments numOfMovies by 1.
    2. The second constructor receives parameters name, director, and runtime.
    3. The third constructor receives name, director, summary, genre, releaseDate, and runtime.
    4. The fourth constructor receives name, director, summary, rating, revenue, genre, releaseDate, and runtime.

- Constructor 1, the no-arg constructor, will set the version to 0, sets the name and director to empty strings, and increments the numOfMovies by 1.
- Constructor 2 receives the name, director, and runtime.
- Constructor 3 receives name, director, summary, genre, releaseDate, runtime, and sets the version to 1.
- Constructor 4 receives name, director, summary, rating, revenue, genre, releaseDate, runtime, and sets the version to 1.

- The constructors are to take the data they receive as parameters and use them to set the values of the corresponding fields in the class.
- You must use the keyword "this" to differentiate between the class fields and parameters with the same name.
- Do not write the same lines of code twice in the constructors (or anywhere else in the program), make sure you chain the constructors and only set the fields once, which is explained more in the next step.
- Instead of setting the same values for constructors 2, 3, and 4 (having the same lines of code), you should chain all four constructors. Therefore, constructor 4 will take eight parameters and chain with constructor 3. The first line in constructor 4 will call constructor 3 passing name, director, summary, genre, releaseDate, and runtime, then constructor 3 will set name, director, summary, genre, releaseDate, and runtime fields with the parameters sent, and constructor 4 will set the remaining fields rating and revenue. Constructor 3 will take six parameters and chain with constructor 2. The first line in constructor 3 will call constructor 2 passing name, director, and runtime, then constructor 2 will set name, director, and runtime class fields with the parameters sent, and constructor 3 will set the remaining fields summary, genre, releaseDate, and version to 1. Constructor 2 will take three parameters and chain with constructor 1. The first line in constructor 2 will call constructor 1, then constructor 2 will set the name, director, and runtime fields with the parameters sent. The name, director, and runtime will be set to empty strings, the numOfMovies will be incremented by 1, and the version field will be set with a default value of 0 in the no-arg constructor. The default values are stated above.
- For help with chaining, you can look at the class notes, which includes a similar example as the follow examples below:
    - https://www.geeksforgeeks.org/constructor-chaining-java-examples/
    - https://beginnersbook.com/2013/12/java-constructor-chaining-with-example/
- The following methods:
    - **setName**
        - Receives name information and sets the name field using it.
        - Increments the version field value by 1.
        - Does not return anything.
    - **setDirector**
        - Receives director information and sets the director field using it.
        - DOES NOT increment the version field value.
        - Does not return anything.
    - **setRating**
        - Receives the rating information and sets the rating field using it.
        - Increments the version field value by 1.
        - Does not return anything.
    - **setRevenue**
        - Receives the revenue information and sets the revenue field using it.
        - Increments the version field value by 1.
        - Does not return anything.

- **setReleaseDate**
  - Receives the releaseDate information and sets the releaseDate field using it.
  - Increments the version field value by 1.
  - Does not return anything.
- **setGenre**
  - Receives the genre and sets the genre field using it.
  - DOES NOT increment the version field value.
  - Does not return anything.
- **setSummary**
  - Receives the summary and sets the summary field using it.
  - Increments the version field value by 1.
  - Does not return anything.
- **setRuntime**
  - Receives the runtime and sets the runtime field using it.
  - DOES NOT increment the version field value.
  - Does not return anything.
- **getName** - Returns the name field value.
- **getDirector** - Returns the director field value.
- **getRating** - Returns the rating field value
- **getRevenue** - Returns the revenue field value
- **getReleaseDate** - Returns the releaseDate field value
- **getGenre** - Returns the genre field value.
- **getSummary** - Returns the summary field value.
- **getVersion** - Returns the version field value.
- **getRuntime** – Returns the runtime field value.
- **playMovie**
  - Print movie's runtime to the standard output (i.e., The Console)
  - Print the end time of movie to the standard output (i.e., The Console). The end time of movie should be current time added with the runtime of the movie. For example, runtime is "2h10m" for this movie, current time is "01-21-2021 5:00 PM", then the end time of this movie should be "01-21-2021 7:10 PM".
  - The output should look like the following where the <> information is the field data obtained through the get methods:

    The runtime of <name> is <runtime>
    <Name> will end at 01-21-2021 7:10 PM
  - Does not return anything.

- **NOTE**: You must use the keyword "this" in order to distinguish between the parameters sent to these methods and the class fields in the Movie class. By doing this, you can use the same name for the parameters as the class fields.
- **NOTE**: Try to avoid writing the same lines of code twice. For example, you could increment the version field by 1 in each of the corresponding setter methods, however, if you have to modify that program later, you will need to change the code in each place. It would be better to create an incrementVersion() method that increments the version field, then invoke the method in the correct places in the corresponding setter methods.

S21 Challenge: Movies

In the main() method of the main class do the following:

- Create an instance of Movie, assign the instance to a variable named movie1, and supply a name of “Soul”, a director of “Pete Docter and Kemp Powers”, and a runtime of “1h40m” during the creation in the constructor’s parameters.
- Use the movie1 instance and the setGenre() method to set the genre to ANIMATION, the setSummary() method to set the summary to “After landing the gig of a lifetime, a New York jazz pianist suddenly finds himself trapped in a strange land between Earth and the afterlife.”, setRating() method to set the rating to 8.1, setRevenue() method to set the revenue field to 82,700,000 which was the movie’s revenue worldwide ($82.70 million), and setReleaseDate() method to set the releaseDate to Dec 25, 2020.
- Create an instance of Movie, assign the instance to a variable called movie2, and supply a name of “Transformers: The Last Knight”, a director of “Michael Bay”, a summary of “A deadly threat from Earth's history reappears and a hunt for a lost artifact takes place between Autobots and Decepticons, while Optimus Prime encounters his creator in space.”, a rating of 5.2, a revenue of 602,800,000 which as the movie’s revenue worldwide ($602.8 million), a genre of ACTION, a releaseDate of June 21, 2017; and a runtime of “2h34m” during the creation in the constructor’s parameters.
- Create an instance of Movie, assign the instance to a variable called movie3, and supply a name of “Forrest Gump”, a director of “Robert Zemeckis”, a summary of “The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate and other historical events unfold through the perspective of an Alabama man with an IQ of 75, whose only desire is to be reunited with his childhood sweetheart. ”, a genre of DRAMA, a releaseDate of July 6, 1994; and a runtime of ”2h22m” during the creation in the constructor’s parameters.
- Use the movie3 object and the setRating() method to set the rating of 8.8 and the setRevenue() method to set the revenue to 679,800,000 which was the movie’s worldwide revenue ($679.8 million).
- Last, create an instance of Movie, assign the instance to a variable called movie4, and do not supply any parameters to the constructor by calling the no-arg constructor.
    - Use the movie4 instance and the setName() method to set the name to “The Godfather”
    - Use the movie4 instance and the setDirector() method to set the director to “Francis Ford Coppola”
    - Use the movie4 instance and the setSummary() method to set the summary to " An organized crime dynasty's aging patriarch transfers control of his clandestine empire to his reluctant son. "
    - Use the movie4 object and the setGenre() method to set the genre to DRAMA
    - Use the movie4 object and the setRating() method to set the rating to 9.2
    - Use the movie4 instance and the setRevenue() method to set the revenue to 287,258,196 which was the movie’s worldwide revenue ($287.2 million).
    - Use the movie4 instance and the setReleaseDate() method to set the releaseDate to Mar 24, 1972.
    - Use the movie4 instance and the setRuntime() method to set the runtime to “2h55m”.

S21 Challenge: Movies

- For movie1, movie2, movie3, and movie4 print the name, director, summary, genre, rating, revenue, releaseDate, runtime, and version to the standard output (i.e., The Console), call playMovie() method in the Movie class to print out the runtime and the end time of each movie.
- For movie1 and movie2, use the get methods to get the name, director, summary, genre rating, revenue, releaseDate, runtime, and version to print.
- For movie3 and movie4, create a print() method in the Movie class in order to print to the standard output with all the fields in the correct format as described below. Then in the main, call the print method for these two instances.
- Precede the info that is printed with a line that says “MOVIE 1:” before the Movie1 information, “MOVIE 2:” before the Movie2 information, and so on.
- Place an empty line between the Movie1, Movie2, Movie3, and Movie4 information so there is at least one space in-between each of them.
- Place “Name: ” before the name, “Director: “ before the director, “Summary: ” before the summary, “Genre: “ before the genre, “Rating: “ before the rating, “Revenue: “ before the revenue, “Release Date: “ before the releaseDate, ”Runtime: ” before the runtime, and “Version: “ before the version.
    - Note: There is a space “ “ between the “:” and the next character.
- Place “Number of Movies: ” before the numOfMovies. You will print this at the end to show how many instances of Movie were created. Since numOfMovies is static, there is a particular way to call this field. Make sure you do it correctly.
- Place an empty line between the last Movie information and the number of Movies information, so there is at least one space in-between.
- The output should look like the following where the <> information is the field data obtained through the get methods or printed through the print method.

MOVIE 1:
Name: <name>
Director: <director>
Summary: <summary>
Genre: <genre>
Rating: <rating>
Revenue: <revenue>
Release Date: <releaseDate>
Runtime: <runtime>
Version: <version>
The runtime of <name> is <runtime>
<Name> will end at <calculatedDate>

MOVIE 2:
Name: <name>
Director: <director>

Summary: <summary>
Genre: <genre>
Rating: <rating>
Revenue: <revenue>
Release Date: <releaseDate>
Runtime: <runtime>
Version: <version>
The runtime of <name> is <runtime>
< Name> will end at <calculatedDate>

MOVIE 3:
Name: <name>
Director: <director>
Summary: <summary>
Genre: <genre>
Rating: <rating>
Revenue: <revenue>
Release Date: <releaseDate>
Runtime: <runtime>
Version: <version>
The runtime of <name> is <runtime>
<Name> will end at <calculatedDate>

MOVIE 4:
Name: <name>
Director: <director>
Summary: <summary>
Genre: <genre>
Rating: <rating>
Revenue: <revenue>
Release Date: <releaseDate>
Runtime: <runtime>
Version: <version>
The runtime of <name> is <runtime>
<Name> will end at <calculatedDate>

Number of Movies: <numOfMovies>

- The output of the **rating** field will be a string with 1 decimal place like “8.8”.
- The output of the **revenue** field should be a string with commas and a dollar sign at the front like “$460,000,000”.

- The output of the **releaseDate field** should be a format like “Jan 21, 2021”. Where the month is only 3 characters, followed by the day of the month which is 1-31 and not 01-31, followed by a comma, then followed by the year which is a 4-digit integer.
- The output of **version** will be a string with no decimal places like “2”.
- **Note:** You may need to convert an int to a string or a double to a string in order to accomplish this, there are built in methods for this, do not write them from scratch (i.e., never “re-create the wheel”). In addition, take a look at string formatters to help you print revenue or date formatters to print the dates correctly.
- **Note:** If you can print revenue by taking the leading 4 digits and putting “thousand”, “million”, “billion”, or “trillion” at the end of it, you may be able to receive bonus points. For example, 160,010 would print “$160.0 thousand”, then 460,050,150 would print “$460.0 million”, then 1,450,000,100 would print “$1.450 billon”, and 500,451,450,000,000 would print “500.4 trillion”. You only need to handle 1 to 4 commas for printing.

Run your application and make sure everything works as expected. ZIP the project directory using NetBeans (i.e., Export the project) and submit it on Canvas.

NOTE: If you would like to create additional Movies, then do so, these requirements are the minimum and creating additional states, behaviors, and objects are usually rewarded when going above and beyond the rest of the class. Have fun and experiment so you learn object-oriented programming and Java beyond the class requirements. Some ideas would be to create leading actors, have some review information, maybe put ratings from different sources, using great OOP practices, making sure to write lines of code only once, having a fancy output, there is a lot you can do with this project, just use proper OOP techniques and going above and beyond the rest of the class is usually rewarded. Make sure the TAs know about your extra work by posting a comment on canvas and/or sending an email in the course email. On the other hand, if you weren’t able to complete a task, letting the TAs know will usually be rewarded as well. Honesty will save the grader time and will usually be rewarded as well.

Things to submit on Canvas:

1. The zip file created after you export your project from NetBeans.
2. Submit screenshots of your application running and ALL of code for proof with the system clock and pawprint. The system clock must have the date and time and be submitted before the due date to be valid.
    - **Note:** Just take a screenshot and directly submit them to canvas. You can do a file upload, then submit the NetBeans zip file and image files, created after taking screenshots, together. If you have multiple screenshots (which you will), submit all of them to canvas. This will be easier for grading instead of making a separate zip file or placing them in your project’s zip file, which will be hard to open and find.