CIS2168 006 Assignment 3
Movie Collections as Linked Lists

1. Objectives

- Learn how to program using the linked list data structure
- Help you understand how linked lists work
- Improve your skills in writing Java applications that consist of more than one class

2. Overview

You write a Java application simulating collections of favorite movies. You will use singly linked lists to add the movies to the collections, search for movies in the collections, change the movies in the collections, and display the information about the collections.

You will implement all the functionalities in two parts.

Part A). Basic Movie Collections

In this first part, you will implement very simple basic movie collections. Your application must be able to do the following:

- Represent each movie using the title, the year it was first released, the genre, and the user rating score.
    - Genre: main genres based on http://www.filmsite.org/genres.html
        - Action, adventure, comedy, crime, historical, horror, musical, science fiction, war, western
    - User rating score: how much audiences like a movie with scales from 1.0 to 10.
- Create a movie collection of at least 6 movies.
- Add a movie to a movie collection
- Display the detailed information about each movie in a movie collection
- Search a collection for a movie with a title given by the user. If the search is successful, print the details about the found movie on the screen. Otherwise, let the user know the movie is not in the collection.

Part B) Advanced Movie Collections

In this second part, you will add more advanced functionalities to basic movie collections. In addition to what is done in Part A), your application must be able to do the following:

- Tell the user how many movies are currently in the collection at the user's request
- Print all the details about a movie at a specific location in the collection. The location is given by the user as 1 for the first, 2 for the second, etc.

- Replace an existing movie at a specific location in the collection by a new movie. The location (1 for the first, 2 for the second, etc) and the new movie information are given by the user. Display the information about the replaced movie on the screen.
- Add a new movie to a specific location in the collection. The new movie information and the location (1 for the first, 2 for the second, etc) are given by the user. Print the new collection after the addition.
- Remove the existing movie at a specific location from the movie collection. The location (1 for the first, 2 for the second, etc) is given by the user. Print the information about the removed movie on the screen. If no movie was removed, informing the user.
- Remove an existing movie with a given title from the collection. It is assume that there is only one copy per title. The user enters the title.
- Search the movie collection for all movies in a specific genre given by the user. If the search is successful, print the details of the found movies on the screen. Otherwise, let the user know that no movies in the collection is in that genre.
- Display a menu of operations that the user can perform, including quitting the entire operation. Keep asking the user until the user decides to quit.

3. Implementation Requirements

Part A.

- You must use a singly linked list to represent a movie collection.
- You must code a Java class named **Movie** to represent all individual movies. Make sure that you use correct Java data types to represent the attributes of movies like title, year, user rating score, etc. This class is similar to the class Magazine I covered in the lecture.
- You must code a Java class named **MovieNode** to represent all data nodes in the linked list. Each data node is used to store 1 movie in the listed list. This class is similar to the class MagazineNode I covered in the lecture.
- You must write a Java class named **MovieCollectionLinkedList** to represent all movie collections. This class is similar to the class MagazineCollectionSingleLinked I covered in the lecture.
- You must write a driver class as a Java Main class that tests the implementation of MovieCollectionLinkedList. Suppose that this driver class is named **MovieCollectionLinkedListTest**. This class must do the following
    - Create a movie collection
    - Add into the collection at least 6 movies in at least 2 genres. Each movie is added to the end of the linked list.
    - Display the information about each movie in the collection on the screen.
    - Ask the user for a movie title
    - Search the collection for the movie and print the search result

  This class is similar to the class MagazineCollectionSingleLinked I covered in the lecture.

Part B.

- You must revise your MovieCollectionLinkedList class to include all the additional functionalities. For example, you need to add a data field size to handle the number of movies in the collection.
- You are not allowed to use the LinkedList provided by Java API or the SingleLinkedList<E> I covered in class. In another word, you MUST implement your own linked list.

Bonus points:

You will get bonus points if you make your application more practical, such as handling more attributes about a movie or making the user interface friendlier and better looking.
But you will NOT get bonus if your programs do not complete all minimum required functionalities and requirements in the first place.

4. Submission Requirements & Grading

Submit your entire application that includes all requirements in both Part A) and Part B).
It is due by 11:50pm, Thursday, September 24.

Part A) of Assignment 3 is the basis of Part B) to be given next week. Part A) will be revised in Part B) with additional features. Part A) is not required to submit. The new version Part B) will be required to submit.
Note:
For your own benefit, please complete Part A) by the time I gave Part B) the following Tuesday. Failure to do so will hurt your grade in Assignment 3.

5. Major Steps

5.1 Complete Part A)

a. Understand the four classes Magazine, MagazineNode, MagazineListSingleLinked, and MagazineListSingleLinkedTest I gave in the lecture and also attached to this assignment.
b. Code the class Movie. Write a driver class for the class Movie and test the implementation of Movie.
c. Write the class MovieNode and the enclosing class MovieCollectionLinkedList.
d. Write the driver class MovieCollectionLinkedListTest.

5.2 Complete Part B)

a. Understand the programs SingleLinkedList.java and SingleLinkedListTest.java I covered in class.
b. Revise MovieCollectionLinkedList to add new methods
c. Revise MovieCollectionLinkedListTest to include new tests of new functionalities.
d. Add one method at a time into MovieCollectionLinkedList, then test it rightway in the MovieCollectionLinkedListTest.

6. Detailed Hints

For Part B)

- Many new methods in Part B are similar to the methods in SingleLinkedList<E>.

- The methods for removing movies from the collection and search by genre are new methods that you must write yourself.
- Make sure that you have some accessor methods in Movie to access information about a movie.

For Part A)

Implement the class Movie and its driver class

- Make sure that you have the following in the Movie class:
  - Data fields describing each movie attribute using correct data types: title, year, genre, user rating score
  - A constructor to create a movie object with a given title
  - A toString() method to return the description of a movie in a string. The description must include every attribute.
- Make sure that you have the following in the driver class for Movie class:
  - A main(..) method that does the following
    - Create some objects of class Movie.
    - Display all the details of each Movie object

Implement the classes MovieNode and MovieCollectionLinkedList

- Make sure that you have the following in the **MovieNode** class:
  - Data field referencing a Movie object
  - Data field referencing the immediate following neighbor MovieNode
  - A constructor for creating a MovieNode for a given movie
- Make sure that you have the following in the **MovieCollectionLinkedList** class:
  - A data field for the head pointer, which will link to the first MovieNode later
  - A constructor for creating an empty list
  - A method add(..) for adding a Movie object to the collection
  - A method toString() to return the description of each movie in a string
  - A method find(…) for searching the collection with a given movie title

Implement the class MovieCollectionLinkedListTest

Make sure that you have the following in the class:

a. A main(…) method
   i. Create at least one object of **MovieCollectionLinkedList** class
   ii. Call the method add in **MovieCollectionLinkedList** class to add at least 6 movies
   iii. Call method find in **MovieCollectionLinkedList** class to find a title given by the user and display the details on the screen