CS 2401 – Elementary data structures and algorithms
Fall 2022: Lab 5

Due Date: Friday, October 7, 2022 – end of the day.

Objective: The goal of this assignment is to practice linked list.

Background: An astronomy organization maintains a list of planets in the solar system and some relevant information about each planet. You are hired to develop software that can read a text file with information on the planets and make a program that allows astronomers to modify the planets' list.

Assignment: We will be given a text file with planet information. Three lines in the file are reserved for the record of a planet. These three lines respectively represent the name of the planet, a long integer diameter of a planet, and the number of the moons that planet has.

A sample input file is provided below, which gives a list of planets between two planets: Mercury and Saturn. The planets in the list are: Mercury → Venus → Earth → Mars → Saturn. Note that the list contains five planets and the record of a planet is composed of three lines (name, diameter, and the number of moons).

```
Mercury
4879
0
Venus
12104
0
Earth
12756
1
Mars
6805
2
Saturn
120536
62
```

Your task is to read the file content, construct a linked list to store the records in the sequence they appear in the file, and allow flexibility to the astronomers so that she/he can perform certain operations (e.g., add, remove).

CS 2401 Assignment #5
(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

The program will contain three java files: Planet.java, PlanetLinkedList.java, and Lab_5.java. Each class is explained below.

Planet.java:

This file is provided with the assignment and must not be changed. The content of the file is also given below. Each planet's information (name, diameter, and moons) will be stored in one Planet object.

```java
class Planet{
  private String name;
  private long diameter;
  private int moon;

  Planet next;

  Planet(String n, long d, int m){
    name = n;
    diameter = d;
    moon = m;
  }


  String getName(){
    return name;
  }

  long getDiameter(){
    return diameter;
  }

  int getMoon(){
    return moon;
  }

  public String toString(){
    return "Name: "+name+
           "\nDiameter: "+diameter+
           "\nMoon: "+moon;
  }
}
```

PlanetLinkedList.java:

You will create a linked list of Planet objects. The linked list of Planet objects must be constructed as a PlanetLinkedList object. The private field variable `head` is the head of the linked list. **You are only allowed to write in the incomplete methods of PlanetLinkedList.java.** The incomplete PlanetLinkedList.java file is provided with this assignment. Here is the incomplete code of the PlanetLinkedList.java file.

```java
import java.io.FileWriter;

class PlanetLinkedList{
  private Planet head;
```

```
  PlanetLinkedList(){
  }

  PlanetLinkedList(Planet initial){
    head=initial;
  }

  PlanetLinkedList(String n, long d, int m){
    head = new Planet(n, d, m);
  }

  void insert(Planet insertee, int pos){
      // Write your code here.
  }

  void remove(int pos){
      // Write your code here.
  }

  int countNodes(){
      int count=0;
      // Write your code here.
    return count;
  }

  void printLL(){
      // Write your code here.
  }

  void swap(int pos1, int pos2){
      // Write your code here.
  }

  void search (String n){
      // Write your code here.
  }

  void search(long d){
      // Write your code here.
  }

  void writeLinkedListToFile(String filename){
      // Write your code here.
  }

}
```

You are allowed to write or modify the body of the methods that are marked with the text
// Write your code here. **You are NOT allowed to change any of the headers. You are NOT allowed to write any other method.**

The PlanetLinkedList class will act as a linked list. The methods are explained further below.

CS 2401 Assignment #5

- **insert:** Insert a new planet (Planet object) in a particular location index. Index of the first item (that is, head) in the linked list is 0.
- **remove:** Remove a record (that is, a Planet object) from a specific location in the linked list mentioned in the input 'pos' parameter. Head of the linked list is considered to be in position 0.
- **countNodes:** Return the number of nodes (that is, planets) in the linked list.
- **printLL:** Print the linked list content on the terminal (name, diameter, and the number of moons of the planet in a sequence as they appear in the linked list.)
- **swap:** Swap two planets in two specific position indices, 'pos1' and 'pos2'.
- **search (with a string):** Search a planet's name in the linked list. Name should not be case sensitive. (Please look at the output of the program carefully.)
- **search (with a long integer):** Search a planet's diameter in the linked list.
- **writeLinkedListToFile:** Write the content of the linked list in an output file. The format of the output file should be the same as the input. The planet's information content should be written to the file as they appear in the linked list.

Lab5.java:

This is the only file that has the main method. You will create the linked list (PlanetLinkedList) in this file. You must complete the method **constructLLFromFile** in this file. **You are not allowed to change any of the headers. You are not allowed to write any other method.**

You are NOT allowed to modify the main method. The main method is already designed to validate correctness of your code.

The incomplete **Lab5.java** file is provided with this assignment. Here is the content of that file.

```java
import java.io.File;
import java.util.Scanner;

class Lab_5{
  public static void main(String[] args){
    // You are not allowed to change the main method.

    String inputfilename = "input.txt";
    String outputfilename= "output.txt";
    PlanetLinkedList solarSystem = constructLLFromFile(inputfilename);

    solarSystem.printLL();

    System.out.println();

    System.out.println(
            "Number of nodes in the linked list: "
            +solarSystem.countNodes());
```

```java
    System.out.println();

    System.out.println("I will find diameter 4879 in the linked
list.");
    solarSystem.search(4879);
    System.out.println();

    System.out.println("I will find diameter 12756 in the linked
list.");
    solarSystem.search(12756);
    System.out.println();

    System.out.println("I will find Neptune in the linked list.");
    solarSystem.search("Neptune");
    System.out.println();

    System.out.println("I will find Pluto in the linked list.");
    solarSystem.search("Pluto");
    System.out.println();

    System.out.println("I will find Jupiter in the linked list.");
    solarSystem.search("Jupiter");
    System.out.println();

    System.out.println("I am going to remove the current head.");
    solarSystem.remove(0);

    System.out.println();

    solarSystem.printLL();

    System.out.println();

    System.out.println("I am going to insert a new record in position
4.");

    Planet aNewPlanet = new Planet("Jupiter", 142984, 67);
    solarSystem.insert(aNewPlanet, 4);

    System.out.println();

    solarSystem.printLL();

    System.out.println("I am going to insert a new planet in position
5.");

    aNewPlanet = new Planet("Uranus", 51118, 27);
    solarSystem.insert(aNewPlanet, 5);

    System.out.println();

    solarSystem.printLL();

    System.out.println("I am going to insert a new planet in position
7.");
```

```
    aNewPlanet = new Planet("Neptune", 49528, 14);
    solarSystem.insert(aNewPlanet, 7);

    System.out.println();

    solarSystem.printLL();


    System.out.println();
    System.out.println("I am going to swap position 0 and 3.");
    solarSystem.swap(0, 3);

    System.out.println();
    solarSystem.printLL();

    System.out.println();
    System.out.println("I am going to swap position 1 and 7.");
    solarSystem.swap(1, 7);

    System.out.println();
    System.out.println("I am going to swap position 1 and 4.");
    solarSystem.swap(1, 4);

    System.out.println();
    solarSystem.printLL();

    System.out.println();
    System.out.println("I am going to write the linked list to file:
"+outputfilename);
    solarSystem.writeLinkedListToFile(outputfilename);

  }

static PlanetLinkedList constructLLFromFile(String theInputFile){
    System.out.println("Constructing the linked list from
"+theInputFile);
    // Change the body of this method to construct the required linked
list.

    return null;
  }

}
```

For the sample input file (given earlier in this document), in a file named input.txt, the terminal output is below (marked in blue).

Constructing the linked list from input.txt
-------- Content of the linked list --------
****

Name: Mercury
Diameter: 4879
Moon: 0

CS 2401 Assignment #5
(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

****
Name: Venus
Diameter: 12104
Moon: 0
****
Name: Earth
Diameter: 12756
Moon: 1
****
Name: Mars
Diameter: 6805
Moon: 2
****
Name: Saturn
Diameter: 120536
Moon: 62
-------- End of content of the linked list --------

Number of nodes in the linked list: 5

I will find diameter 4879 in the linked list.
The record with diameter 4879 is found in position 0

I will find diameter 12756 in the linked list.
The record with diameter 12756 is found in position 2

I will find Neptune in the linked list.
The name Neptune is not found in the linked list.

I will find Pluto in the linked list.
The name Pluto is not found in the linked list.

I will find Jupiter in the linked list.
The name Jupiter is not found in the linked list.

I am going to remove the current head.

-------- Content of the linked list --------
****
Name: Venus
Diameter: 12104
Moon: 0
****
Name: Earth
Diameter: 12756
Moon: 1
****

Name: Mars
Diameter: 6805
Moon: 2
****
Name: Saturn
Diameter: 120536
Moon: 62
-------- End of content of the linked list --------

I am going to insert a new record in position 4.

-------- Content of the linked list --------
****
Name: Venus
Diameter: 12104
Moon: 0
****
Name: Earth
Diameter: 12756
Moon: 1
****
Name: Mars
Diameter: 6805
Moon: 2
****
Name: Saturn
Diameter: 120536
Moon: 62
****
Name: Jupiter
Diameter: 142984
Moon: 67
-------- End of content of the linked list --------
I am going to insert a new planet in position 5.

-------- Content of the linked list --------
****
Name: Venus
Diameter: 12104
Moon: 0
****
Name: Earth
Diameter: 12756
Moon: 1
****
Name: Mars
Diameter: 6805

Moon: 2
****
Name: Saturn
Diameter: 120536
Moon: 62
****
Name: Jupiter
Diameter: 142984
Moon: 67
****
Name: Uranus
Diameter: 51118
Moon: 27
-------- End of content of the linked list --------
I am going to insert a new planet in position 7.
Position 7 is not valid. Nothing changed.

-------- Content of the linked list --------
****
Name: Venus
Diameter: 12104
Moon: 0
****
Name: Earth
Diameter: 12756
Moon: 1
****
Name: Mars
Diameter: 6805
Moon: 2
****
Name: Saturn
Diameter: 120536
Moon: 62
****
Name: Jupiter
Diameter: 142984
Moon: 67
****
Name: Uranus
Diameter: 51118
Moon: 27
-------- End of content of the linked list --------

I am going to swap position 0 and 3.

-------- Content of the linked list --------

****

Name: Saturn
Diameter: 120536
Moon: 62
****

Name: Earth
Diameter: 12756
Moon: 1
****

Name: Mars
Diameter: 6805
Moon: 2
****

Name: Venus
Diameter: 12104
Moon: 0
****

Name: Jupiter
Diameter: 142984
Moon: 67
****

Name: Uranus
Diameter: 51118
Moon: 27
-------- End of content of the linked list --------

I am going to swap position 1 and 7.
The position 7 is invalid.
Returning without any change.

I am going to swap position 1 and 4.

-------- Content of the linked list --------
****

Name: Saturn
Diameter: 120536
Moon: 62
****

Name: Jupiter
Diameter: 142984
Moon: 67
****

Name: Mars
Diameter: 6805
Moon: 2
****

Name: Venus

CS 2401 Assignment #5

Diameter: 12104
Moon: 0
****
Name: Earth
Diameter: 12756
Moon: 1
****
Name: Uranus
Diameter: 51118
Moon: 27
-------- End of content of the linked list --------

I am going to write the linked list to file: output.txt
Wrote the LL to output file: output.txt

The **output.txt** file will have the following content.

```
Saturn
120536
62
Jupiter
142984
67
Mars
6805
2
Venus
12104
0
Earth
12756
1
Uranus
51118
27
```

Deliverables:

You must submit two files (`PlanetLinkedList.java`, and `Lab_5.java`) after you write your codes in the designated areas (incomplete methods). You must not submit any other file. Your TA will instruct you with further details, if any.

CS 2401 Assignment #5
(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

Grading Criteria:

- o [5 points] The Program **compiles and runs**.
- o [5 points] The program is **indented** correctly.
- o [5 points] The program is **documented** properly.
- o [5 points] The program uses the correct **variable types**.
- o [5 points] The program uses meaningful **variable names**.
- o [75 points] The program has all the correct methods with correct logic and generates correct output.

- ▪ Late submission: [**-10**] points for every 24 hours after the deadline.

If you need any clarification, please ask your TA for further details.