COMPX201/Y05335
Linked List

Due: Friday 23rd May 11:59pm

Part One:
Define Java classes to implement a dynamic linked list of Strings following the specification given below.

1. **The Linked List:** Define a class called `StrLinkedList` in a file called `StrLinkedList.java`. This class is to implement a dynamic linked list of Strings (`String`) that supports the following public methods:
   - `isEmpty()` - returns boolean true if the linked list is empty.
   - `getLength()` - returns (as an int) a count of the number of values (i.e. nodes) in the linked list; zero if the list is empty.
   - `hasValue(String s)` - returns a boolean value that is true if the linked list contains a node whose value is the same as `s`; false otherwise.
   - `getValueAt(int i)` - returns the String value at position `i`. Note, it is possible that the user will enter a number greater than the length of the list.
   - `add(String s)` - adds **to the head of the list** a new node with the String value `s`; then returns void.
   - `remove(String s)` - finds the first node whose String value is `s` and removes that node without affecting the order of the remaining nodes in the list. If the list has no node with that value, the method should leave the list unchanged. The method should return void.
   - `print()` - for each node in the list, print to the screen (i.e. `System.out`) its String value, followed by a pointer to the next note (`->`), such that all values in the list end up being printed in the order they appear in the list.
     For example `E->D->C->B->A->`
2. **The Node:** Define an inner class called `Node` for nodes in your `StrLinkedList`. It should have the following:
   - a member variable called "`value`" to hold the String value.
   - a member variable called "`next`" to hold a link to another `Node.`
   - a constructor that takes a String argument and copies that value into the node's "`value`" member variable.
   - Ideally, these nodes and their data should be protected from being manipulated by anything other than the list class itself, by making the inner class and/or its members private .... but this is not essential for this assignment.
3. **Debugging:** Write a program class that creates one or more of your `StrLinkedList` objects and tests that all your methods work as per the specification. Example output from one such file will be provided to you. Your test program will not be marked, but is for your own solution development process.

Part Two:

Define a new class called `LottoDraw`. This class should have a `main` method and a collection of supporting methods. It should use multiple instances of your `StrLinkedList` to solve the following problem description. You are setting up a fundraiser lottery draw. Before opening the lottery draw to the public, you must determine whether it is a feasible fundraising activity. To do this, you are going to develop a program that simulates a lottery draw. Lottery tickets contain six numbers, drawn from a list of 39 numbers total. You are required to write a program that will:

1. hold a list of all possible lottery numbers (numbers “1” to “40”)
2. randomly select six of these numbers as the “winning numbers”
3. generate 100 lottery tickets, which each contain six randomly selected numbers
4. for each lottery ticket, calculate the prize that would be won (prizes are listed below)
5. calculate the total prize money that would be given
6. calculate the sales from the lottery tickets (each ticket costs $10.00)
7. calculate the fundraising profit that would be made

Prize pool:

- Zero matching numbers: $0.00
- One matching numbers: $0.00
- Two matching number: $10.00
- Three matching numbers: $100.00
- Four matching numbers: $1,000.00
- Five matching numbers: $10,000.00
- Six matching numbers: $100,000.00

As an indication of Solution Quality – how easy would it be to change (a) the number of possible lottery numbers (e.g. 1-20 instead of 1-40), (b) the number of tickets sold, (c) the cost of a ticket, (d) the amount of numbers each ticket holds (e.g. 4 instead of 6), (e) the prize pool (e.g. you do not win any money unless you have four or more matches).

Assessment:

Your solution will be marked on the basis of how well it satisfies the specification, how well you have approached the problem (Solution Quality), how well-formatted and easy to read your code is (Code Quality), and whether each class and public method has at least some comment explaining what it does, what it’s for, and what any of its arguments are (i.e. documentation). Your code should compile and run as a console program from the command-line (i.e. no GUI or IDE).

Submission:

Create an empty directory (i.e. folder) using your student ID number as the directory name. Place copies of your **source code (.java files) in this directory**. If you wish to communicate with the marker any additional information then you may include a plain text README file, **but nothing else (e.g. no compiled code (.class files) or IDE folders or files)**. Compress and upload this directory through the Canvas submission page for this assignment.