CIS351-LinkedList Lab

Submission Instructions

1. Submit the completed **SimpleLinkedList.java** file in Blackboard.

Download Materials

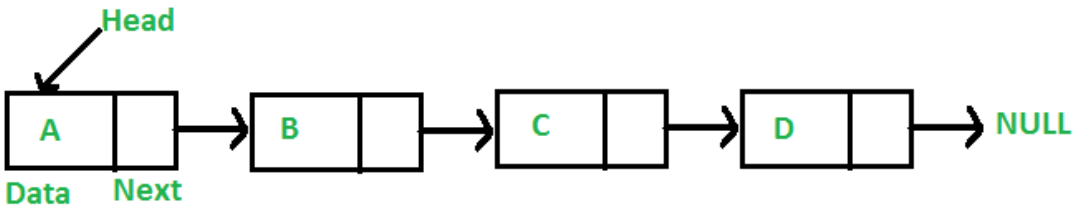
Download the following 2 files from Blackboard:

- SimpleLinkedList.java - UNFINISHED
- Node.java - DO NOT MODIFY.
- LinkListDriver.java - DO NOT MODIFY.

Background

You will be implementing a simple linked list today. Every linked list stores data in the form of Node. You are already provided with a `Node.java` class that will be stored in the `SimpleLinkedList` class.

Linked list is a linear data structure. Unlike arrays, linked list elements are not stored at the contiguous location, the elements are linked using pointers/references as shown below.



You have learned in class that we can insert Node at the start of the linked list or at the end of the linked list as well. Since, this today's SimpleLinkedList.java is very simple, it can only store Node with String. And it has very imited functionality, like - items may only be added at the start or end of the list, however, items may only be removed from the beginning of the list.

Few of the methods of SimpleLinkedList are already implemented for you so you can get an understanding of manipulation in LinkedList is done.

Instructions

1. Your **first task** of this lab is to read Node.java and SimpleLinkedList.java so you understand what these both classes are doing. You should NOT proceed without reading these two classes first.
2. For today's lab, complete the two unfinished methods in SimpleLinkedList.java: **contains** and **addAtEnd**.
3. In both of these methods, we left instruction in the form of method documentation which will help you to implement the method bodies. **NOTE: MAKE SURE to read the comments first before trying to implement the code.**

Testing program correctness

You can use the LinkListDriver.java file to test for program correcteness. This makes calls to SimpleLinkedList class to test the methods you have just written to check for their correctness. Open this LinkListDriver.java file and spend a few min on understanding how it is instantiating list, inserting data at various positions, retrieving them and also finding out the size of the list.

Now, make sure all THREE java files are in the same folder. Finally, run LinkListDriver.java. If it successfully runs, it should display the output of your program.

If you have correctly implemented your methods, you should see the following output:

```
Inserted A, B and C in the list

Expected output: List contains A at start is true
Your program output: List contains A at start is true

Expected output: List contains B in middle is true
Your program output: List contains B in middle is true

Expected output: List contains C in end is true
Your program output: List contains C in end is true

Expected output: List contains Q is false
Your program output: List contains Q is false

Expected output: list size 1
Your program output: list size 1

Expected output: list size 2
Your program output: list size 2

Current state of the list - X Y Z
Your program retrieves, current state of the list X Y Z
```

Grading Criteria

Total points: 100 points (Each complete method is worth 50 point)

Farzana Rahman / frahman@syr.edu