JOBSHEET IX LINKED LIST

# Learning Objective

After learning this practicum, students will be able to:

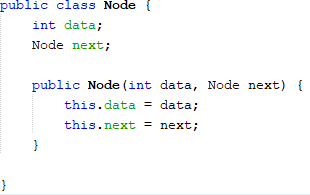
1. Create a linked list data structure
2. Create a program that implements linked list
3. Differentiate the problems that can be solved with linked list

# 1st Lab Activities

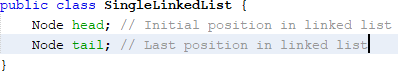
In this practicum, we will implement how to create single linked list with **nodes** data representation, accessing the linked list, and adding the data.

# Steps

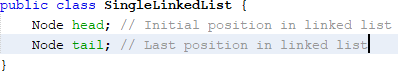
* + - 1. Create a new package named **week11**
      2. Add these following classes:
         1. Node.java
         2. SingleLinkedList.java
         3. SLLMain.java
      3. Create Node class



* + - 1. Add these following attributes in class **SingleLinkedList**



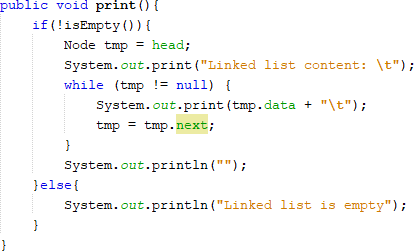
* + - 1. For the next step, we will implement methods that are exist in **SingleLinkedList**



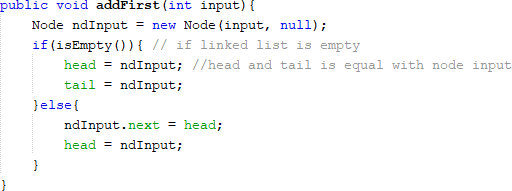
* + - 1. Add method isEmpty()



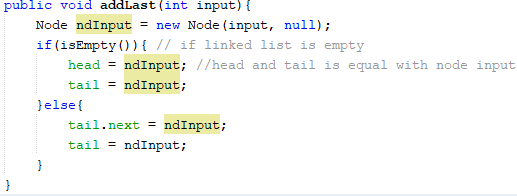
* + - 1. Implement this method to display the data with traverse process



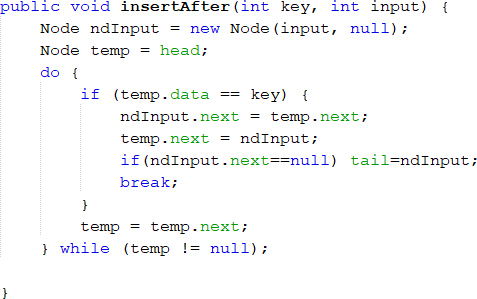
* + - 1. Implement method **addFirst**()



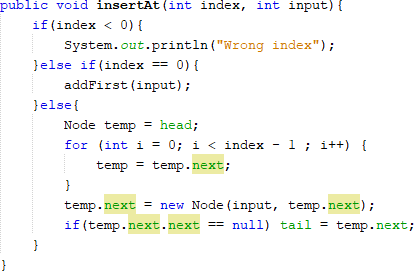
* + - 1. Implement method **addLast**()



* + - 1. Implement method **insertAfter** (), to insert a node that stores data that were inputted by the user after data ***key***



* + - 1. Add these following codes to add a node based on defined index

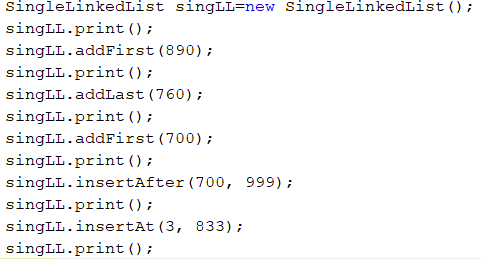


* + - 1. In class **SLLMain,** create main function and instantiate a new object from

**SingleLinkedList** class

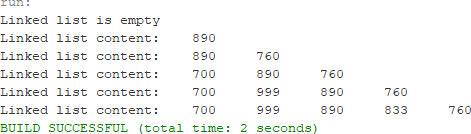


* + - 1. Add methods for inserting data, as well as displaying the data for each insert process so that we can track the changes



# Result

Check if the result match with following image:



# Questions

* + - 1. Why the output of the program in first line is “Linked list is empty”?
      2. Please explain the usage of these following codes in:



* + - 1. In **SingleLinkedList,** what is the usage of this following code in **insertAt**?

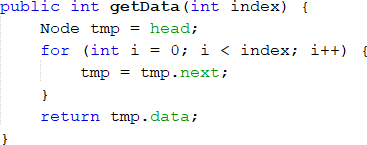


# 2nd Lab Activities

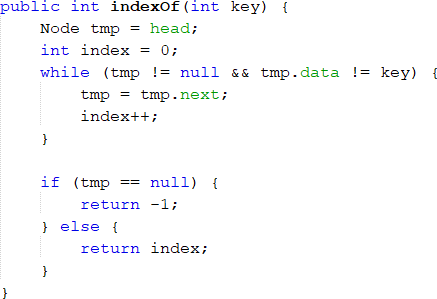
In this practicum, we will try to learn and implement how to access node elements, get index, and node removal in a Single Linked List

# Steps

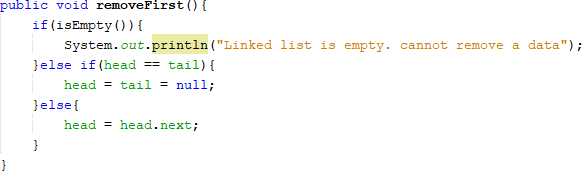
* + - 1. Implement methods to access data and index in linked list
      2. Add methods to get data based on certain index from class **SingleLinkedList**



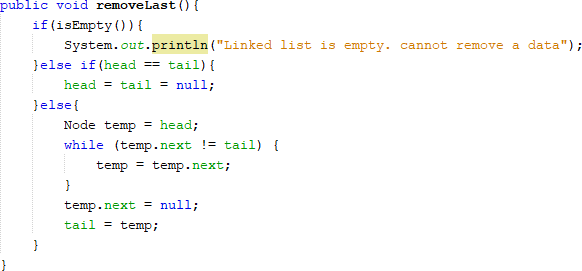
* + - 1. Implement method **indexOf**



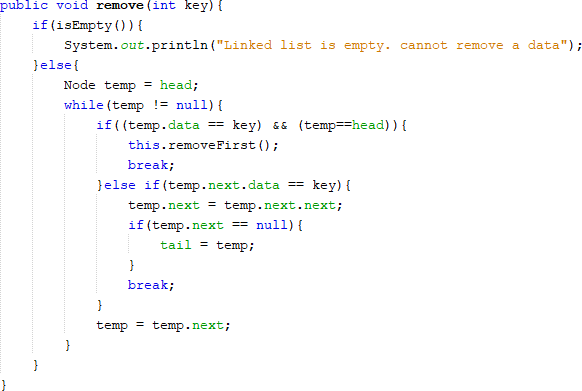
* + - 1. Add method **removeFirst()** in class **SingleLinkedList**



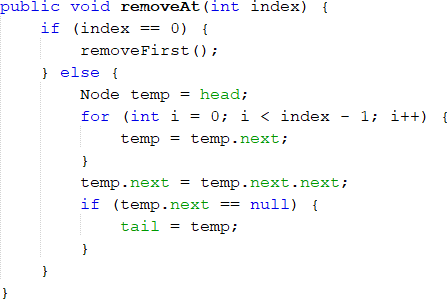
* + - 1. Add this method to remove data that is in the last of the list from class **SingleLinkedList**



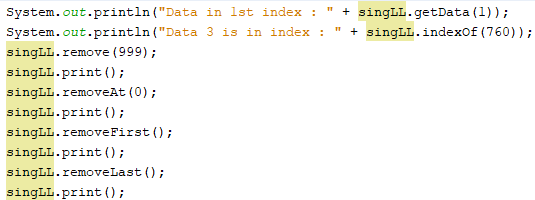
* + - 1. Next, we will implement method **remove()**



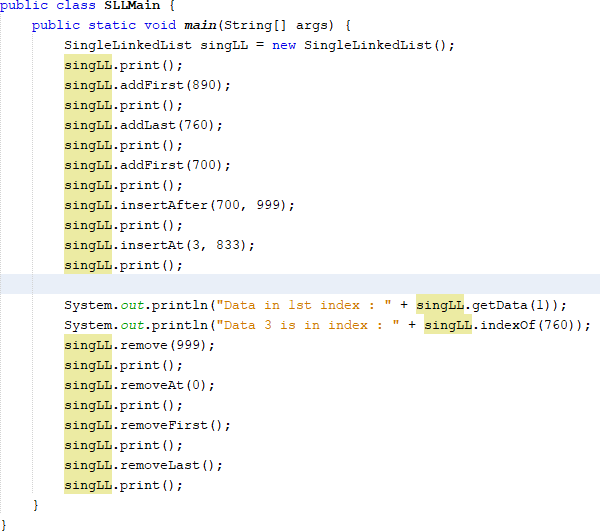
* + - 1. Create a method to remove a node based on defined index



* + - 1. Next, we will try to access and remove data in main method in class **SLLMain** by adding these codes



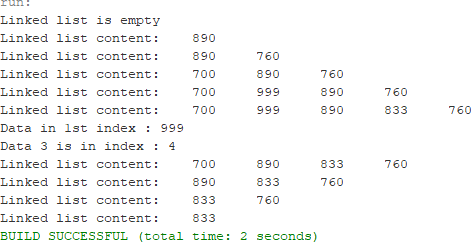
* + - 1. Method **SLLMain** becomes like this:



* + - 1. Execute the class **SLLMain**

# Result

Check if the result match with following image:



# Questions

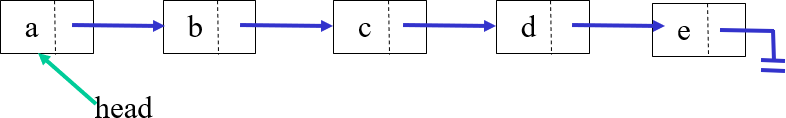
* + - 1. Why we use **break** keyword in remove function? Please explain
      2. Please explain why we implement these following codes in method remove



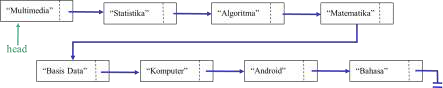
* + - 1. What are the outputs of method indexOf? Please explain each of the output!

# Assignments

1. Create a method **insertBefore()** to add node before the desired keyword
2. Implement the linked list from this following image. You may use 4 method of adding data we’ve learnt



1. Create this following **Stack** implementation using Linked List implementation



1. Create a program that helps bank customer using linked list with data are as follows: Name,address, and customerAccountNumber
2. Implement **Queue** in previous number with **linked list** concept