Tutorial 5
Single Linked Lists

Overview

- The goal of this tutorial is to familiarise yourself with the functionality of the abstract data type: Single Linked Lists.
- Understand how to break down larger problems into sub-problems
- We will develop an application that implements a single linked list and we will test the functionality of the following operations performed on the list: add, remove and print nodes.

Task 1

Create a NetBeans application (SLL) that implements a single linked list. Perform the following tasks:

1. **Add to your application the following classes as described in the lecture notes:**
    - LinearList interface,
    - Node,
    - SLList
    - Tester – the class the has main() method.
   Some code is also provided on the Web.

2. **Implement the main() method in the SLList that performs:**
    - Create a SLList object and add a number of nodes.
    - The information from the node may be a string
    - Check if the list is empty
    - Display the current size of the list
    - Print all the elements from the list
    - Remove an element from a given position
    - List again the content of the list
    - Get and print the information stored in the element on position 3

3. **Test the application and make sure it works properly.**

Task 2

In this week's lab you will solve a train problem using single linked list data structure. The train's wagons are added at the end of the train. Each wagon has information regarding the type of product it has inside. For example: iron signifies that the wagon has iron type products.
Develop a NetBeans application that has a GUI interface that allows you to:
- Add a new wagon, with a certain type of products inside, to the end of the train.
- To list all products carried by the wagons currently attached to your train.

- Check the total number of wagons attached to the train. If the total number is higher than 5, the application displays a warning message asking you to remove (detached) the extra wagons from the train.
- To remove a wagon from the train giving the position of the wagon.

NOTE:

- First design your GUI and decide what graphical elements you need, and then add functionality to the buttons.
- You must include in the project the 3 files (Node.java, SLList.java, LinearList.java) that implement the Single Linked List structure in order to store details about the wagons attached to the train.
- The printList() method from SLList class prints the elements from the list directly in the output window. Write a new method (**printToString()**) that creates a String object with all the elements from the list and then return this string object.

| Liquid | Wood | Iron | Paper | Engin |