LAB 4: LINKED LISTS

F27SG – SOFTWARE DEVELOPMENT 3 (5 MARKS)

The topic of this lab is **linked lists**. Download Lab4.zip from Vision and import the project into Eclipse. Your task in the lab is to complete the implementation and Junit tests for this project. Read the source code and make sure you understand what is going on and what is missing. Before you start, you will need to download this and import it into Eclipse:

File -> Import -> Existing Projects into Workspace

Then select the project you downloaded. The project is organized as follows:

- The src directory contains all the source files
  - **LinkedList.java** is the implementation of a linked-list class (recursive version) from the lecture, with the difference that it stores integers and not objects.
  - **Stack.java** is the skeleton implementation of Stacks using linked lists. You should complete this.
- The test directory contains the unit tests for the project.

1. UNIT TESTS FOR ITERATING LINKED LIST (1 POINT)

In part 2 you will implement two methods to iterate over the list. These are

- `int size()` which should return the number of nodes in the linked list
- `int total()` which should sum up the values in the linked list

Empty skeleton implementations of the methods can be found in **LinkedList.java.** Your task in this part is to implement suitable Junit tests for these operations, covering

- empty lists
- list with more than one element

The file **LinkedListTest.java** contains empty test methods for each of these cases. Your task is to complete them.

2. ITERATING OVER THE LIST (2 POINTS)

Your task here is to complete the implementation of the `int size()` and `int total()` methods described in part 1. You can find empty skeleton implementation for these methods (with dummy return values) in **LinkedList.java**. Implement them and ensure that the tests from part 1 succeed.

3. IMPLEMENTING AND TESTING A STACK (2 POINTS)

The **Stack.java** class contains a skeleton implementation of a Stack using a Linked List. Your task is to complete this implementation by implement the following methods:

- top()
- pop()
- push(int i)

You should also develop suitable test cases for these methods in **StackTest.java**. Empty methods are here provided for your testing. You can reuse much of the test methods you implemented in lab 2 for this task (replacing strings with integers). Make sure that the tests also includes test that getSize() returns the correct size.

REVERSING THE LIST (OPTIONAL)

Implement a method that reverses the list. Write suitable test methods.