Assignment Prefix: Lab06

Due Date: Friday, Feb. 24th @ 11:59pm

Points: 100

This is an individual assignment.

Restrictions:

You cannot use any predefined Java classes in writing this lab.

Create a NetBeans project named Lab05 and save it to a location like the desktop or your flash drive. In the project you will do the following:

In this assignment, you are to create several Stack classes.

Begin this assignment by creating a **Stack** ADT interface as described in Code Fragment 6.1. This should be a generic interface. Use this interface in creating the following Stack classes (each class should implement the Stack interface):

- **ArrayStack** as described in Code Fragment 6.2.
- **LinkedStack** as describe in Code Fragment 6.4.
- You should use your SinglyLinkedList class from Lab05 (Code Fragments 3.14 and 3.15).
- **ArrayStackBad**
    - Similar to the ArrayStack (above) except:
    - Have the push always add a new element at array index 0, shifting any existing stack entries to higher indexes to make room for the new element.
    - Have pop return the element at array index 0, shifting any existing stack entries to lower indexes to fill in the space created by the pop operation.
- **LinkedStackBad**
    - Similar to the LinkedStack (above) except:
        - Use the addLast method for the push
        - Use the removeLast method for the pop
- **ArrayListStack**
    - Create an adaptor class that uses the ArrayList for storage.
    - Use the ArrayList implementation from Code Fragments 7.3, 7.4 & 7.5.
    - The ArrayList implementation requires the List interface from Code Fragment 7.1.

In your Client Class:

- Create a stack of each type where the element is an Integer.
    - Use random integer values from 0 to 99.
- For each stack type, time how long it takes to do push 100,000 elements onto the stack and then pop the 100,000 elements off the stack.
- Your output should look something like the following:

    push/pop ArrayStack: N = 100,000 time = 12 (msec)
    push/pop LinkedStack: N = 100,000 time = 9 (msec)
    push/pop ArrayListStack: N = 100,000 time = 6,830 (msec)
    …
- Use printf to get your values to "line up" correctly.
- Depending on your computer and the amount of memory you have, it may be necessary to adjust the number of elements (N) to keep the runtimes reasonable. If any given test takes much more than one minute (60,000 msec) you might want to consider decreasing N for that test.

Things to turn in:

- Open a Microsoft Word document
- Copy and Paste the source code of **the various Stack classes** (make sure to use *Ctrl + A* to select all the source code of the program, *Ctrl + C* to copy, and Ctrl + V to paste.).
- Copy and Paste the source code of the C**lient** Class
- Copy and paste the output of the client program
- Next, zip the Project folder.
- Finally on blackboard, submit both your Word document and project zipped file.