CS2230 Data Structures & Algorithms

Lab 4

Learning objectives

- Implement the abstract data type Stack using a linked list
- Implement the abstract data type Queue using a linked list

Setup

Download lab4.zip from ICON and then unzip it. Inside you'll find a `src` folder and `test` folder. Create a *Java project from existing sources*. As with Lab3, on the Existing Sources window of the wizard, point the Source Package Folder to `src` and the Test Package Folder to `test`. When you click Finish, if a window pops up about problems, click Resolve Problems...

The Stack ADT

The Stack ADT models the idea of a stack of objects as you are familiar with in real life. If I stack pancakes one on top of the other, I'll eat them in the reverse order. We call this order Last-In-First-Out (LIFO).

Example:

s.push(13)	13
s.push(4)	4,13
s.push(17)	17,4,13
s.pop() returns 17	4,13
s.push(55)	55,4,13

The code in Stack.java defines the Stack ADT using a Java interface. Your job is to implement this interface using a linked list. In LinkedStack.java, you'll find the outline of the LinkedStack class, as well as the ListNode class. Fill in its constructor implementation and required methods.

You can run the tests for LinkedStack by right clicking LinkedStack.java and choosing Test File. The test code is in LinkedStackTest.java. Debug your implementation by fixing the simpler tests first (testIsEmpty, testNotEmpty, testPushPop), then progressing to the other ones.

CS2230 Data Structures & Algorithms

Lab 4

The Queue ADT

The Queue ADT models the idea of a waiting queue. If students line up at the dining hall cash register in order, the student who got in line first is served first. We call this order First-In-First-Out (FIFO).

Example:

q.add(13)	13
q.add(4)	13,4
q.add(17)	13,4,17
q.pop() returns 13	4,17
q.add(55)	4,17,55

The code in Queue.java defines the Queue ADT using a Java interface. Your job is to implement this interface using a linked list. In LinkedQueue.java, you'll find the outline of the LinkedQueue class, as well as the ListNode class. Fill in its fields, constructor implementation, and required methods.

You can run the tests for LinkedQueue by right clicking LinkedQueue.java and choosing Test File. The test code is in LinkedQueueTest.java. Debug your implementation by fixing the simpler tests first (testIsEmpty, testNotEmpty, testAddPop), then progressing to the other ones.