CS 445 Lab 5: Stacks and Queues

Introduction

In this lab, you will practice using the Stack and Queue ADTs to accomplish solve two problems: reversing the order of a queue, and keeping track of the minimum value in a stack. You will also get practice with inheritance and the process of extending a class.

The primary goal of this lab is to practice and gain confidence with the Stack and Queue ADTs, learning their respective operations and how they interact. Another goal is to gain practice using the data structures we learn in class to solve problems.

In lecture, we learned about the Stack ADT, a LIFO data structure, and the Queue ADT, a FIFO data structure.

Your TA will give a lesson reviewing these ADTs and covering the problems your code will be solving in this lab.

Exercise

After the TA's lesson, complete the following steps:

1. Download the provided code from the course website. The following Java files are provided in package `cs445.lab5`.

- `StackInterface.java` is a Java interface representing the ADT Stack
- `QueueInterface.java` is a Java interface representing the ADT Queue
- `LinkedQueue.java` is a linked-chain implementation of ADT Queue
- `EmptyQueueException.java` is an exception that LinkedQueue will throw when you attempt to dequeue from a queue that is empty. LinkedStack just uses Java's built-in EmptyStackException for this purpose

- `LinkedStack.java` is a linked-chain implementation of ADT Stack
- `QueueReverser.java` is a class that has the stub method you will implement for reversing a queue.
- `MinStack.java` is a class extending LinkedStack that has stub methods you will need to implement to create the MinStack data structure
- `Lab5Tester.java` is an example test client for testing the QueueReverser and MinStack classes. You do not need to write code in here, but may add additional tests if you like.

2. Implement the algorithm discussed on the slides for reversing a Queue by using a Stack in `QueueReverser`. Be sure to read the comments and complete all TODOs
3. Implement the `MinStack` class as discussed on the slides. Be sure to complete all TODOs carefully and think about potential errors that might occur.
4. Test your work by using the `Lab5Tester` class. Run it using command-line arguments `reverse` or `minstack` to test the queue-reversal and `MinStack` class, respectively.

Conclusion

In this lab, you wrote implementation code for a queue reversal algorithm, and the `MinStack` data structure. You also practiced using the methods of queues and stacks. As the course continues, it will be helpful to have these skills when implementing other algorithms.