Lokesh Sankar Ramesh

Farnaz Eivazi

CMSC 204-36708

02/22/2022

**UML**

**Diagram

Description automatically generated**

Junit Tests Screenshots

MyQueueTest

Graphical user interface, text

Description automatically generated

MyStackTest

Graphical user interface, text

Description automatically generated

NotationTest

Graphical user interface, text, application

Description automatically generated

**GUI Running**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Lessons Learned:**

**Throughout the course of Assignment 2, I have garnered a deep fundamental understanding of the overall infix to postfix and postfix to infix conversions, and I have also learned more about generic array based stack and queue implementations. Learning from Project 1, I also used the Eclipse debug feature quite a bit for this project and it may have saved me some time. Before this assignment, I had minimal experience with stack and queue implementations in Java but this assignment has allowed me to develop a good understanding of how they work.**

**There are still some improvements I can make in the future. For one, I should’ve probably planned out my code better before actually doing it. Given it had been quite some time from when I did my UML diagram, I should’ve planned out the logic for many of my Stack and queue methods better beforehand. I found that I made lots of really small mistakes which wasted a lot of debugging time, such as forgetting to initialize doubleS in MyStackTest and wondering why I’m getting so many errors. I could have definitely have prevented spending significant time for this.**

**In the future, I hope to spend more time planning out the logic behind each of the methods in my utility class as well as any other classes I have to ensure I spend less debug time.**

**Github Screenshot**