Clare DuVal and Jonathan Ayala

September 21, 2018

Lab 4

Lab Report

Requirements Analysis

Functional Requirements

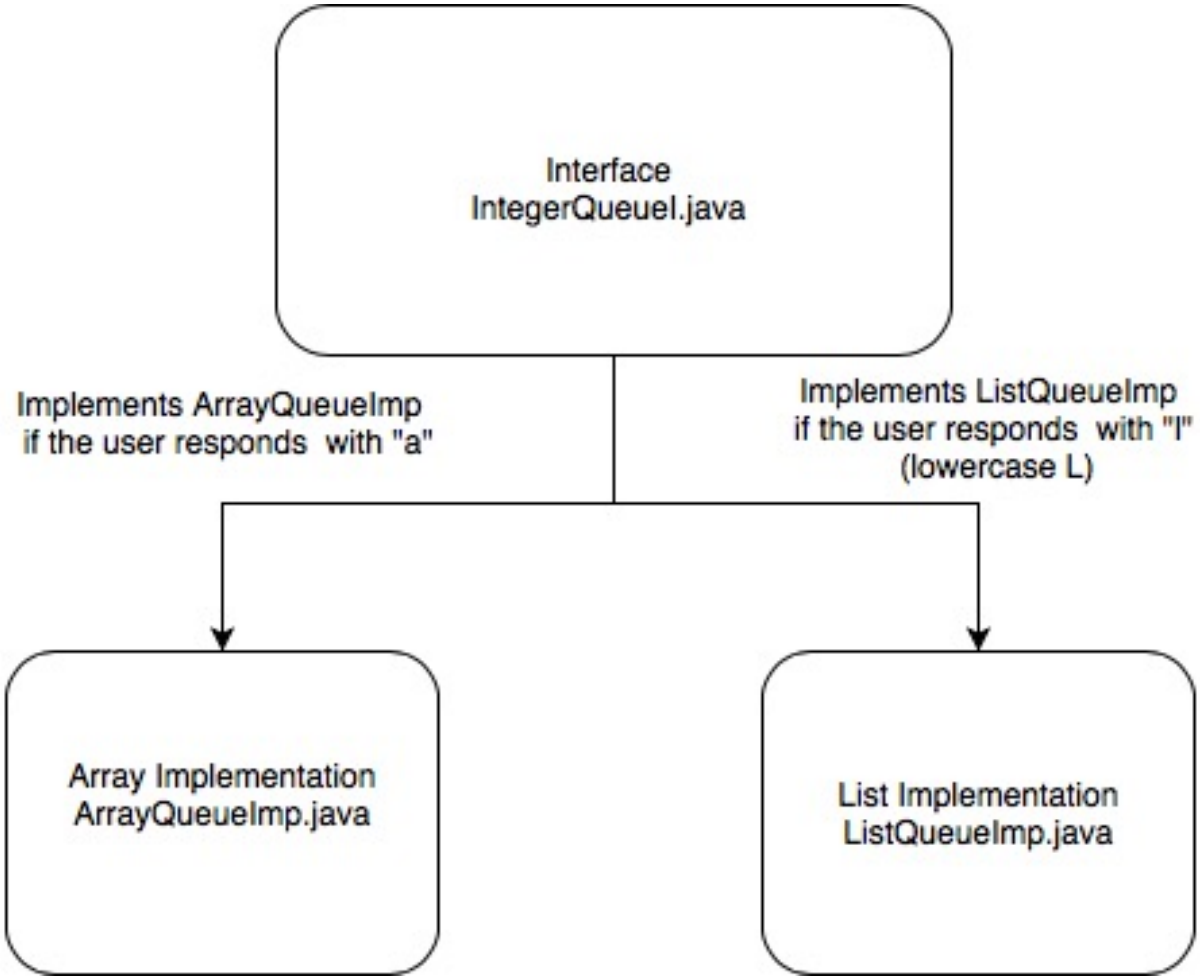
- As a user I can choose the option to initiate the program to call array or list implementation by responding with “array” or “list”.
- As a user I can view the queue to see the integers added into the queue in first in, first out order.

Non-Functional Requirements

- The system must be written in Java for it to be run on Unix.
- The system must be able to compile on Clemson University’s computers for the user to be able to run it.
- The system initializes the IntegerQueueI interface from the main function and has a array implementation as well as a list implementation.

Design

IntegerQueueI Interface and Implementations



Testing

To test the success of the program, we first tried responding with “array” to request an Array Implementation. This is the resulting outcome:

“ 42

17

37 “

Next, we responded with anything other than “list” to request a List Implementation. This is the resulting outcome:

“ 42

17

37 “

We then tried requesting for implementations with “List”, “please”, and “no” and all produced the same output for a list array. We tried doing a while loop to test for bad outcome, but the program had issues overwriting the string.

Deployment

The user must navigate to the directory before the one stated in the package. From the command line when the user types “ls” the terminal displays the cpsc2150 directory and the makefile. This makefile compiles when “make” is written on the command line. It runs the program when the user types “run make”. The program commences and will ask the user for “array” if they want to run with array implementation. When they input anything other than “array” it commences with list implementation. The program displays the outputs mentioned in Testing of this lab report. And when the user types “make clean” it removes all .class files in cpsc2150/MyQueue/ directory.