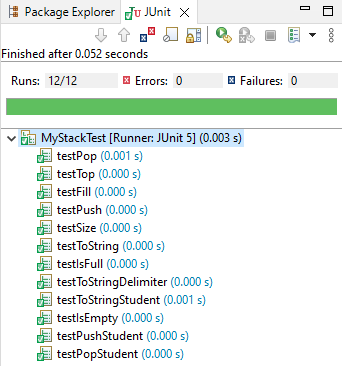
**Approach, design & algorithm** –

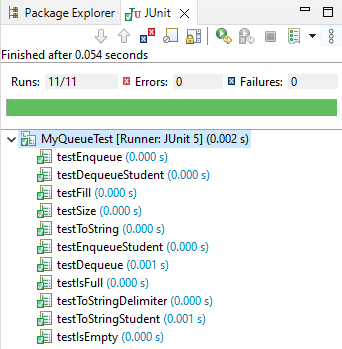
* Read Project-2C Write-up and the .html doc for Notation.java
* Created custom Exception classes from the above materials
* Coded MyQueue class based on QueueInterface.java
* Coded MyStack class based on StackInterface.java
  + Had more trouble on this than MyQueue, partly because we already made our own Queue class in Lab 3.
* Ran the MyQueueTest and MyStackTest, made sure each test case passed, although I had to add throws Exception clause for most of them.
* Coded the Notation class from the .html doc. Struggled the most on this one, since we had recently learned postfix notation.
  + I noticed that NotationGui.java and NotationTest.java would only catch InvalidNotationFormatException. Not wanting to change the code on the Gui, I basically spam coded **throw new** InvalidNotationFormatException() everywhere in the Notation class.
* Ran NotataionTest.java. Like the other Junit tests, I had to manually add **throws** InvalidNotationFormatException to each test case.
* Ran NotationGui.java

**Test Cases**

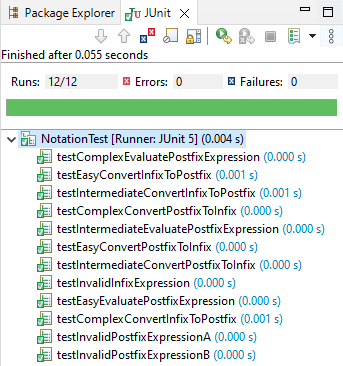
MyStackTest



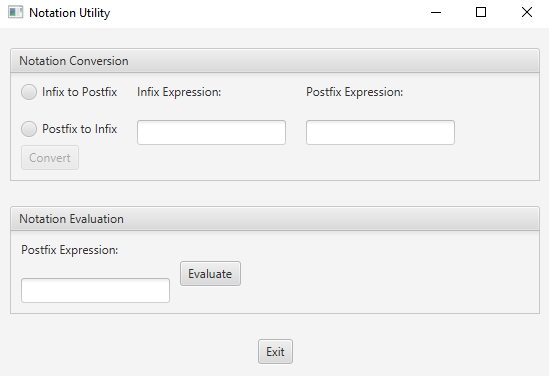
MyQueueTest



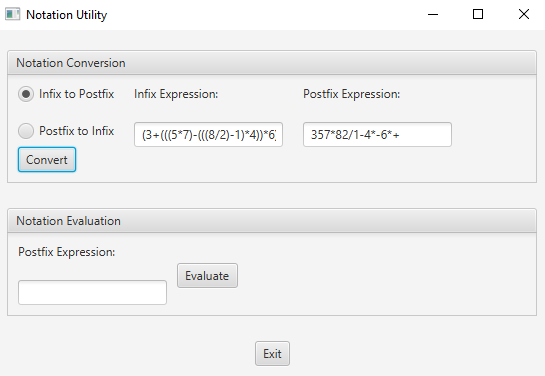
NotationTest



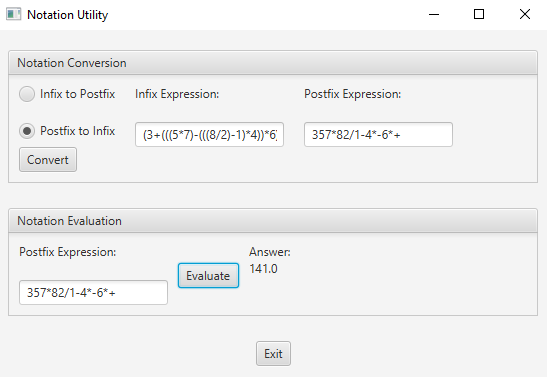
**NotationGui**



**Infix to Postfix**



**Evaluate Postfix Expression**



**Learning Experience**

* Became more practiced in Junit tests
  + Discovered Double.valueOf() method
* Learned how to catch an Exception to throw another Exception
* Gained more understanding about stack and queues

**Assumptions**

* User will not jump straight into coding like I did.
* Expression will only consist of numbers, parentheses and basic arithmetic operations