Project 2

Design

The main objective for this project was determining infix for postfix expression and vis vera. I homed in on converting this expression back and forth until I was comfortable. I started by looking at the slides to understand the topic. After, I watched videos on YouTube consisting of other converting the problems also. I then practiced working on the problems. I studied all these expressions to a point of clear visualization. This helped me implements the conversion methods smoothly. For notation implementation I wrote down pseudo code and worked off the draft.

Test Cases

All with evaluations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prompt | Input | Output | Expected Output | Screenshots |
| In to post | 5+4\*3 | 543\*+  17 | 543\*+  17 | Graphical user interface, application  Description automatically generated |
| In to post | 5/4 | 54/  1 | 54/  1 | Graphical user interface, application  Description automatically generated |
| post to in | 2 3 1 \* + 9 - | (2+(3\*1))-9  -4 | (2+(3\*1))-9  -4 | Graphical user interface, application  Description automatically generated |
| Post to in | 562\*+ | 5+(6\*2)  17 | 5+(6\*2)  17 | Graphical user interface, application  Description automatically generated |

Junit

In top to bottom order are the screenshots of my junit test cases. Stack, queue, and Notation classes.

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Learning Experience

This was the first time I had to use infix and postfix data. I learned that concept and how to convert them from each other. I ran into a few troubles while working on this project. The first one was making sure each exception class was correctly implemented for either the stack or queue. I used logic like if I keep adding then one of these tries will create Overflow. This works with underflow for both stack and queue. Junit testing was killing me on this project But I enjoyed the journey. I had problems with the toString with the parameter (delimiter) in both stack and queue at first. The delimiter is displayed right after the user input, the final input does not have a following delimiter. It took some time but after some trouble shooting I used the endsWith() command with an if statement to return the final string without the delimiter. Another issue stemmed from adding parentheses when converting post to infix. I added parentheses when there was an operator was added following two or more inputs. I was stuck for a but determining how to change to correspond with pop. I modified the toString method again to get the correct infix.