Rajahow Parajuli

* Approach, Design & Algorithm:
  + This was straight up assignment for me. You had kindly give use the infix and postfix algorithms so after making the stacks and queue it was a piece of cake.
  + I made a gui that acted as the driver for the notation class.
  + The notation class used the self-made stack and queues for the methods and algorithms.
  + The stack and queues implemented the provide interfaces.
  + I had methods to check every check to ease of use for evaluating is the current char as you parse thought the expression. And it also helps the readability of the code.
  + My initials plan was close to the execution the only problem I had with process was to use the overflow and underflow exception as invalid expression. I over complicated it and when trying to code it worked with much thinking.
* Test Plan & Test Cases:

Pictures for cases are below the table.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Actual output | expected | passed |
| (3+(((5\*7)-(((8/2)-1)\*4))\*6)) as infix | 357\*82/1-4\*-6\*+ | 357\*82/1-4\*-6\*+ | yes |
| 357\*82/1-4\*-6\*+  As postfix | (3+(((5\*7)-(((8/2)-1)\*4))\*6)) | (3+(((5\*7)-(((8/2)-1)\*4))\*6)) | yes |
| (3+(((5\*7)-(((8/2)-1)\*4))\*6)) as post | Error | Error | yes |
| 357\*82/1-4\*-6\*+  As infix | Error | Error | yes |
| No setting selected | Error yelling at you to select a thing | Error | yes |
| 357\*82/1-4\*-6\*+  To eval | 141.0 | 141.0 | yes |
| 357\*82/1-4\*-6\*+  To eval | Error | Error | yes |

Highlights and assumptions:

The assumptions I made for this program are:

* The users kinda know what do.
* They do need to know where their mistake is when testing the code
* The input will only have digits, operators (+-/\*) and parentheses only

The highlights:

I learned how to use stacks and queues in junctions. I also learned how to chain exceptions.