

CMSC204 Assignment 1

Class: CMSC204

Program: Assignment #2

Instructor: Professor Kuijt

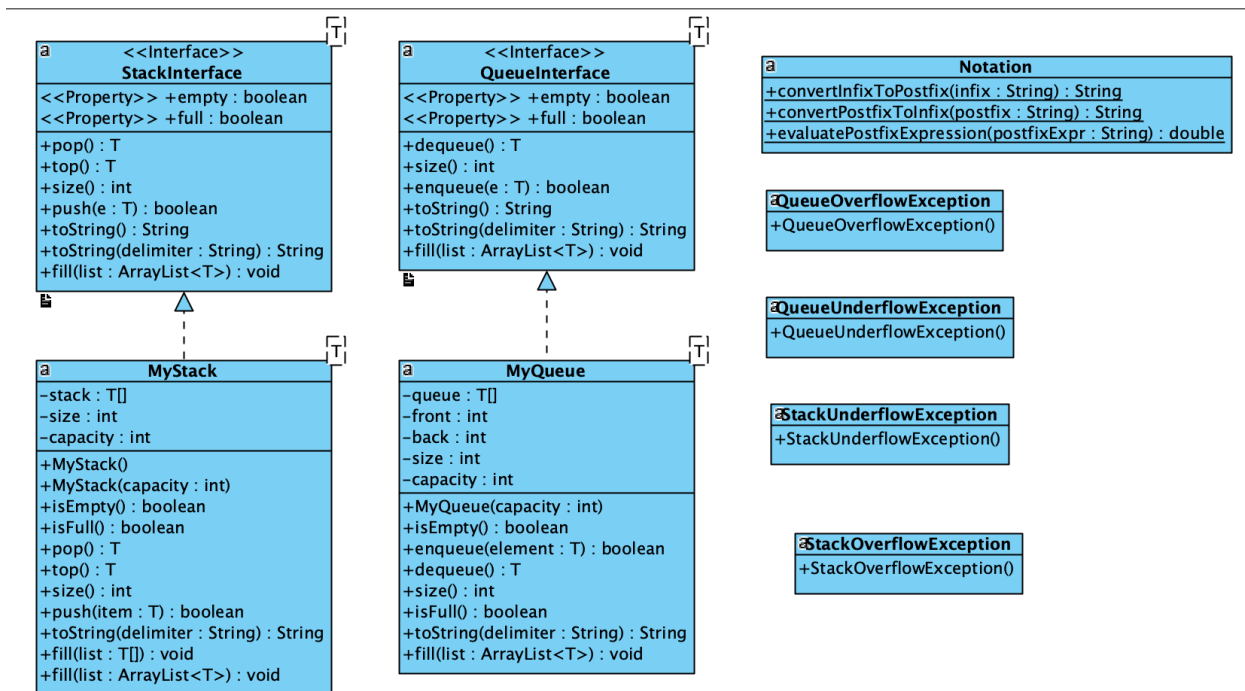
Summary of Description:

Postfix prefix converter/evaluator

Due Date: February 19, 2023

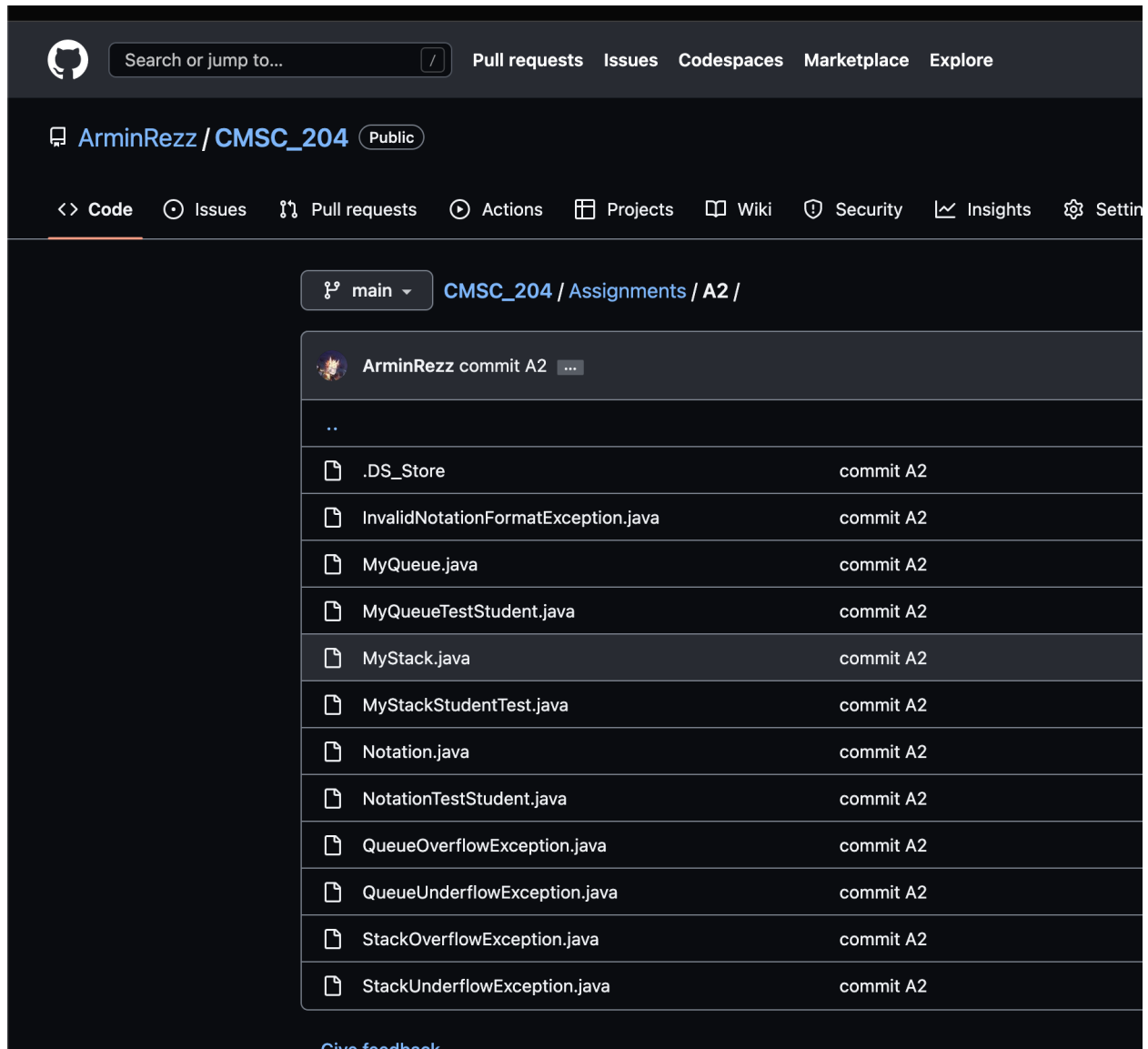
Integrity Pledge: I pledge that I have completed the programming assignment independently.
I have not copied the code from a student or any source.

UML:



Screenshots:

```
30 *MgmCompanyGui, Class representing the GUI for company information
6
7 import java.text.NumberFormat;
28
29 public class MgmCompanyGui {
30
31     private JTextField infixtxt;
32     private JTextField postfixlbl;
33     private JTextField postfixtxt;
34     private JButton convert;
35     private JButton evaluate;
36     private JButton exit;
37     private JButton evalanswer;
38
39     // Constructor
40     public MgmCompanyGui() {
41         // Initialize GUI components
42         @SuppressWarnings("unchecked")
43         JFrame frame = new JFrame("MgmCompanyGui");
44         frame.setSize(400, 300);
45         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46
47         JPanel panel = new JPanel();
48         panel.setLayout(new BorderLayout());
49
50         // Infix to Postfix Conversion Section
51         JSectionHeader sectionHeader = new JSectionHeader("Notation Conversion");
52         sectionHeader.setBorder(BorderFactory.createTitledBorder("Notation Conversion"));
53         panel.add(sectionHeader, BorderLayout.NORTH);
54
55         // Infix to Postfix Conversion Controls
56         JRadioButton infixToPostfix = new JRadioButton("Infix to Postfix");
57         JRadioButton postfixToInfix = new JRadioButton("Postfix to Infix");
58         JTextfield infixExpression = new JTextfield("3-4+6");
59         JTextfield postfixExpression = new JTextfield("34-6+");
60         JButton convert = new JButton("Convert");
61
62         infixToPostfix.setSelected(true);
63
64         panel.add(infixToPostfix, BorderLayout.WEST);
65         panel.add(postfixToInfix, BorderLayout.WEST);
66         panel.add(infixExpression, BorderLayout.EAST);
67         panel.add(postfixExpression, BorderLayout.EAST);
68         panel.add(convert, BorderLayout.SOUTH);
69
70         // Postfix Evaluation Section
71         JSectionHeader sectionHeader2 = new JSectionHeader("Notation Evaluation");
72         sectionHeader2.setBorder(BorderFactory.createTitledBorder("Notation Evaluation"));
73         panel.add(sectionHeader2, BorderLayout.NORTH);
74
75         // Postfix Evaluation Controls
76         JTextfield postfixExpression2 = new JTextfield("34-6+");
77         JButton evaluate = new JButton("Evaluate");
78         JLabel answerLabel = new JLabel("Answer:");
79         JLabel answerValue = new JLabel("5.0");
80
81         panel.add(postfixExpression2, BorderLayout.EAST);
82         panel.add(evaluate, BorderLayout.EAST);
83         panel.add(answerLabel, BorderLayout.EAST);
84         panel.add(answerValue, BorderLayout.EAST);
85
86         // Exit Button
87         JButton exit = new JButton("Exit");
88         panel.add(exit, BorderLayout.SOUTH);
89
90         // Event Listeners
91         infixToPostfix.addActionListener(new ActionListener() {
92             public void actionPerformed(ActionEvent e) {
93                 // Convert Infix to Postfix
94                 String infix = infixtxt.getText();
95                 result = Notation.convertInfixToPostfix(infix);
96                 postfixlbl.setVisible(true);
97                 postfixtxt.setText(result);
98             }
99         });
100
101         postfixToInfix.addActionListener(new ActionListener() {
102             public void actionPerformed(ActionEvent e) {
103                 // Convert Postfix to Infix
104                 String postfix = postfixtxt.getText();
105                 result = Notation.convertPostfixToInfix(postfix);
106                 infixlbl.setVisible(true);
107                 infixtxt.setText(result);
108             }
109         });
110
111         evaluate.addActionListener(new ActionListener() {
112             public void actionPerformed(ActionEvent e) {
113                 // Evaluate Postfix Expression
114                 String postfix = postfixtxt.getText();
115                 result = Notation.evaluatePostfix(postfix);
116                 answerValue.setText(result.toString());
117             }
118         });
119
120         exit.addActionListener(new ActionListener() {
121             public void actionPerformed(ActionEvent e) {
122                 System.exit(0);
123             }
124         });
125
126         frame.add(panel);
127         frame.setVisible(true);
128     }
129 }
```



Lessons Learned:

What have you learned?

Learned Stacks Queues and how to employ them.

What did you struggle with?

I struggled with getting the evaluation to work properly, also had to add many throws because there were exceptions that weren't checked for.

What would you do differently on your next project?

Nothing really probably wouldve spent more time optimizing notation function for evaluation

What parts of this assignment were you successful with, and what parts (if any) were you not successful with?

Successfull with all parts.

Note:

I did have to add try catches and throws to the junit tests because they were giving me errors otherwise.