Anthony Borza
Project 1 Write up:

UML Class Diagram:

Main GUI Class extends JFrame

Class: public class MainGUI extends JFrame

- Creates
  - Frame width to 380
  - Frame height to 140
  - 2 JLabels
  - 2 JTextField
  - 1 JButton
  - Infix Expression
  - String

Constructor: public MainGUI ()

- 3 JPanel's
  - 1 for Enter Infix Expression text field
  - 2 for evaluate Button
  - 3 for results text filed

Action Listener Method: public Class Press Implements Action Listener

- Public void actionPerformed(ActionEvent event)
  - Try and catch used for exceptions
    - Arithmetic Exception
    - Number Format Exception
  - If statement for evaluate Button
    - Gets the infix text
    - calls Infix Expression class
    - calls expression method in Infix Expression class
    - performs calculation
    - returns results in the result text box

Main Method: public static void main (String [] args)

- MainGUI frame = new MainGUI new frame object
  - Sets title of frame
  - Sets size of frame
  - Sets location
  - Sets frame visibility to true
  - Sets it so it closes on exit

Infix Expression Class

Class: public class InfixExpression

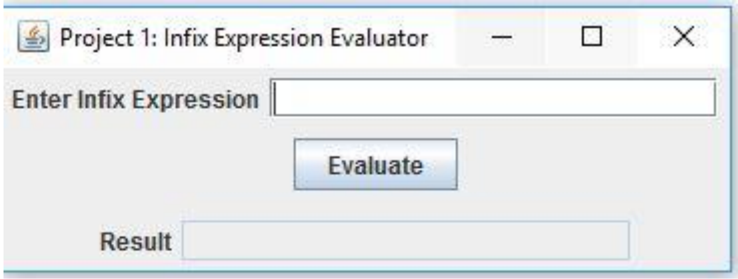
- 3 static integer variables

Method: public static int expression (String infix)

- 2 declared Strings and 3 Integers

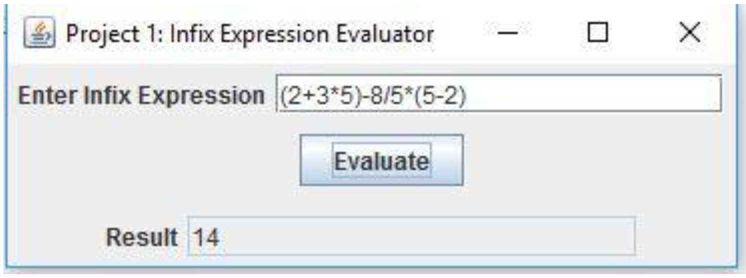
Test Plan and Screenshots:

GUI Display:



Infix Expression:

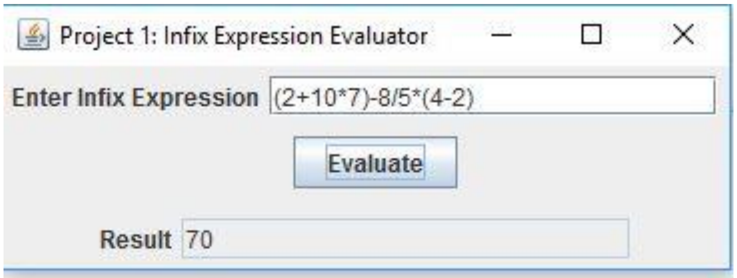
Case 1: Enter in Infix Expression (2+3*5)-8/5*(5-2)



Enter Infix Expression (2+3*5)-8/5*(5-2)

Result 14

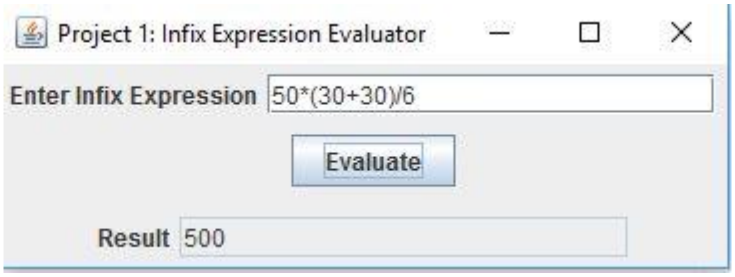
Case 2: Enter in Infix Expression (2+10*7)-8/5*(4-2)



Enter Infix Expression (2+10*7)-8/5*(4-2)

Result 70

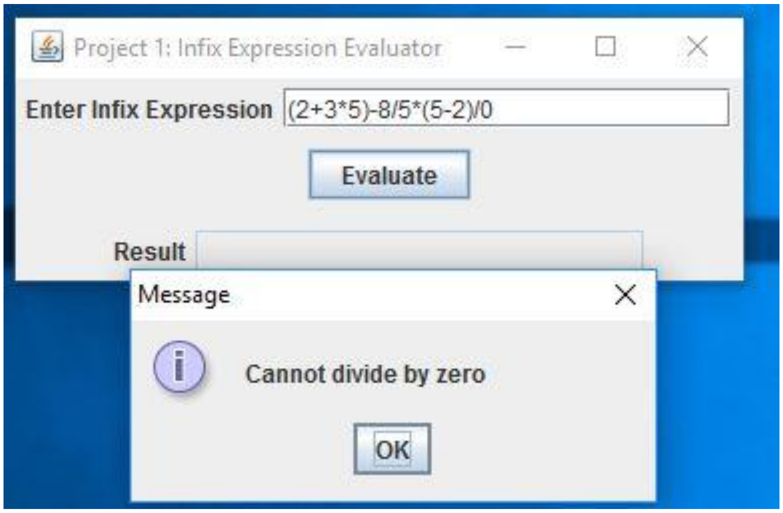
Case 3: Enter in Infix Expression 50*(30+30)/6



Enter Infix Expression 50*(30+30)/6

Result 500

Case 4: Cannot divide by zero (2+3*5)-8/5*(5-2)

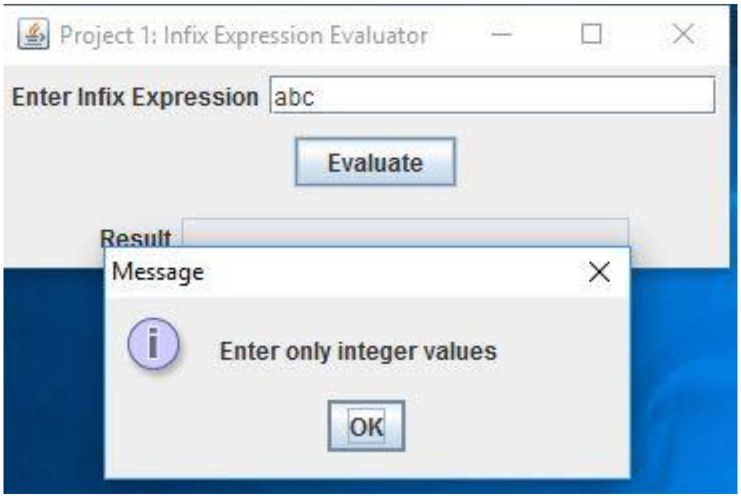


Enter Infix Expression (2+3*5)-8/5*(5-2)/0

Result

Cannot divide by zero

Case 5: Cannot enter in strings abc



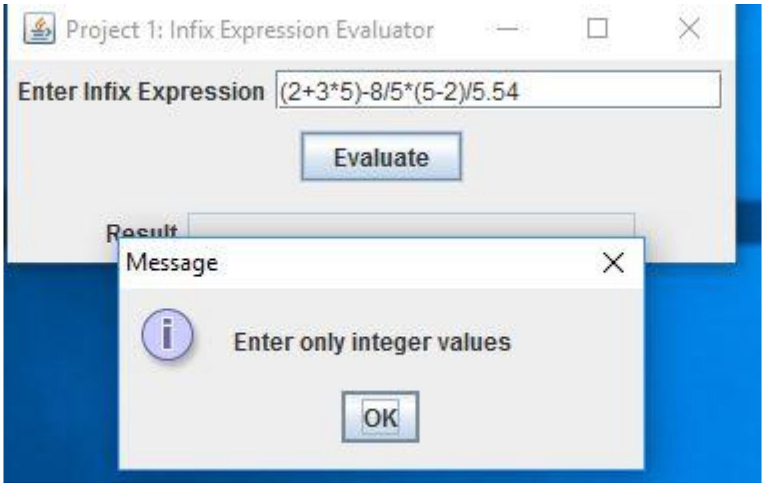
Project 1: Infix Expression Evaluator

Enter Infix Expression: abc

Message

Enter only integer values

Case 6: Cannot enter in doubles (2+3*5)-8/5*(5-2)/5.54



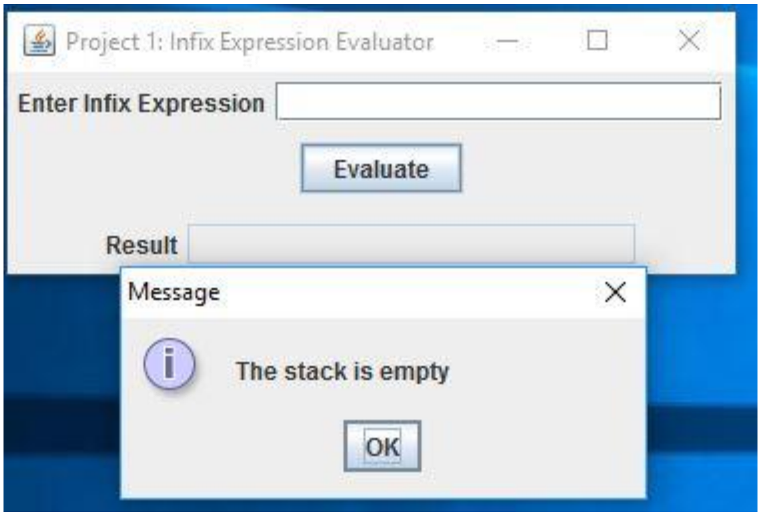
Project 1: Infix Expression Evaluator

Enter Infix Expression: (2+3*5)-8/5*(5-2)/5.54

Message

Enter only integer values

Case7: The stack is empty



The stack is empty

Lessons Learned:

After completing this project there are many things that were learned. I realize how important stacks are to data structures, and how elements are inserted, retrieved, and removed. Like in previous programming classes, understanding what the problem is asking, and how to implement a solution to solve the problem only comes through trial and error. The hardest part of this project was developing and algorithm that would evaluate and infix expression. Once I figured out how to do that developing the GUI, and calling the class with the infix expression code was easy. All in all, to complete this project, I spent a total of 30 hours.