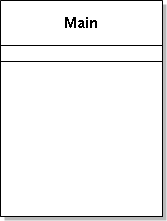
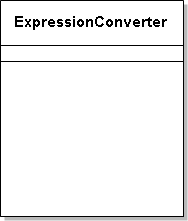
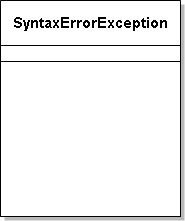
Sairam Soundararajan

CMSC350: Data Structures and Analysis

University of Maryland Global Campus

Professor Specasio

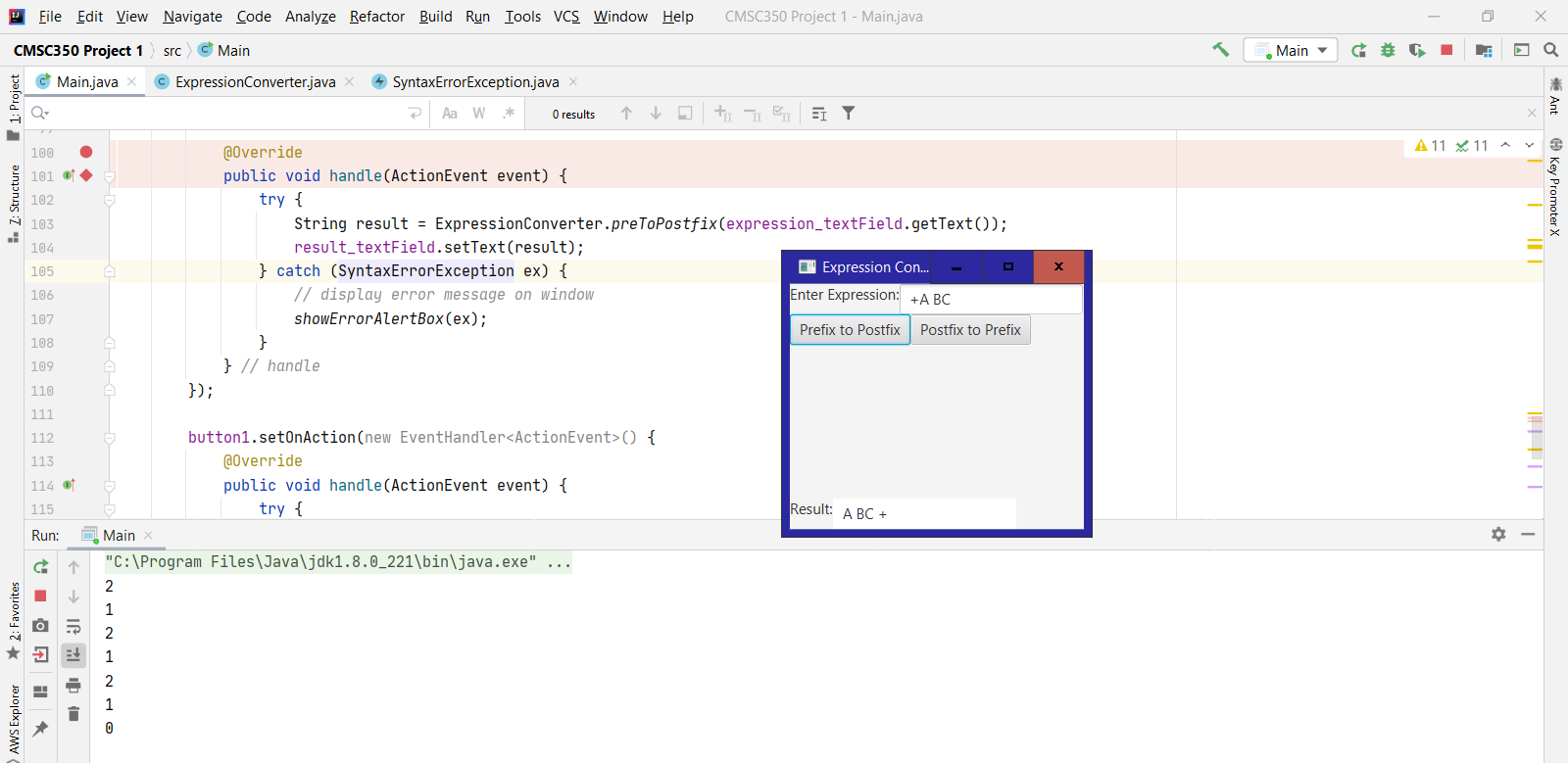
**UML Diagram:**

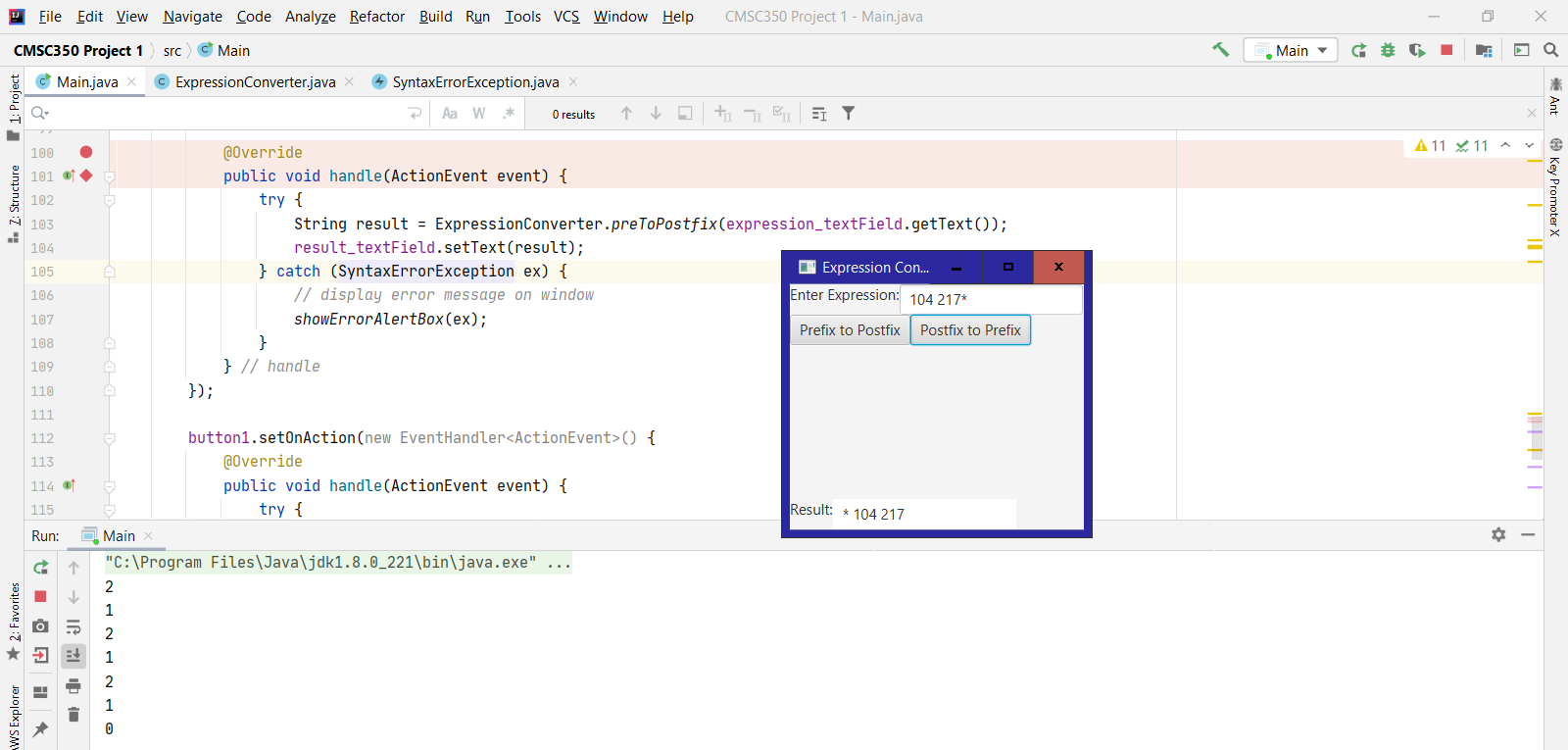
  

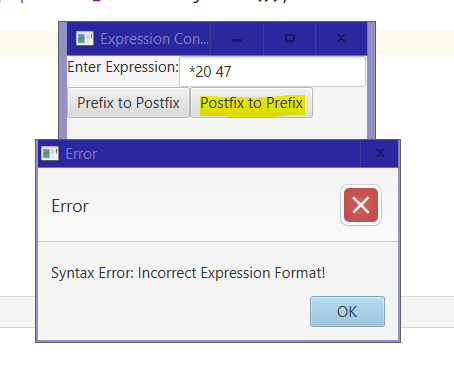
**Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Output | Actual Output | Pass? |
| 1 | Enter Expression: 10 14\*  Postfix to Prefix | Result: \* 10 14 | \* 10 14 | YES |
| 2 | Enter Expression: + A BC  Prefix to Postfix | Result: + A BC | Result: + A BC | Yes |
| 3 | Enter Expression: \*20 47  Postfix to Prefix | Error  Syntax Error: Incorrect Expression Format! | Error  Syntax Error: Incorrect Expression Format! | Yes |

**Snapshots:**







**Lessons Learned:**

For this first project, I used JavaFX because I have more experience with that than I do with Swing. I did not use any third-party applications or special tools to help me in making the GUI or making the code work. I was a bit rusty on working on Java as it has officially been a year since I have worked on Java. I was able to make a GUI with some guidance from my tutor. Again, I hand coded the GUI. My tutor also guided me in the right direction when it came to the expression converter class. When I was at student at MC (Montgomery College), I used to work a lot with JavaFX and learned about stacks, exceptions, and tokenizers. At first, I had some struggle with getting the conversions to work when it came to inputting a prefix to convert to a postfix and vice versa, and converting a postfix expression to prefix while there was no space between the operand and operator. I was able to overcome those obstacles, however. Making the exception class was not too difficult. I did not use JOptionPane to display the error message because I used JavaFX. I did, however, use border panes in the project and an alert box to display the error message. One of the things that I learned was that I needed to decrement a counter variable in the tokenizer method in order for the program to be able to convert a postfix expression to prefix even with there being no space between the operand and operator. I used NClass for constructing my UML diagrams.