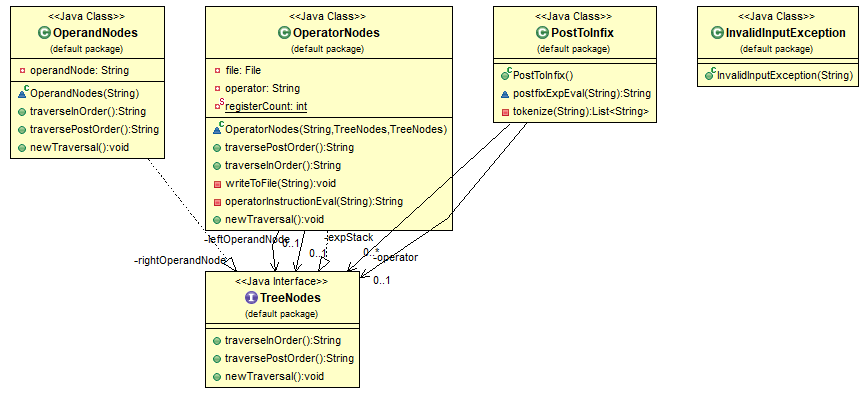
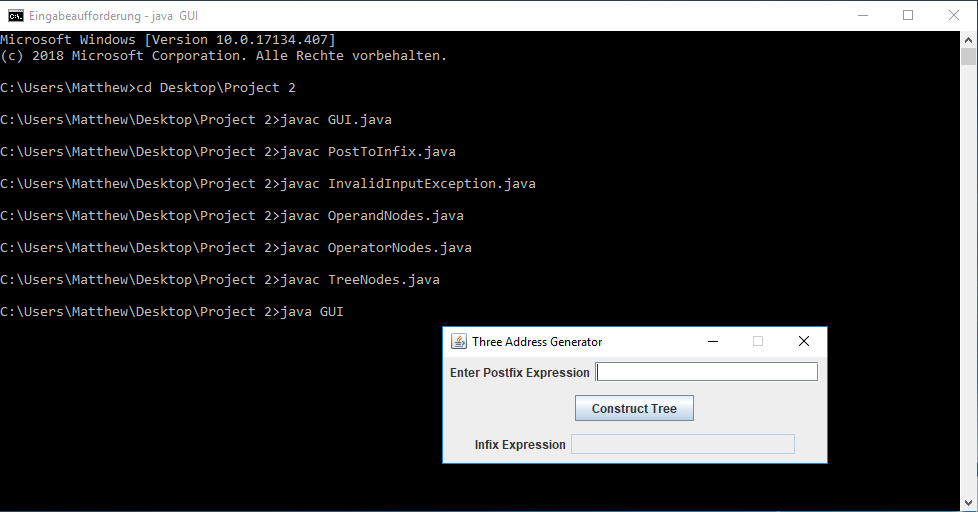
|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** | **Pass?** |
| **Test Case from Project Prompt:**  **3 5 9 +- 2 3 \* /** | **( ( 3 - ( 5 + 9 ) ) / ( 2 \* 3 )**  **Three Address File:**  **Add R0 5 9**  **Sub R1 3 R0**  **Mul R2 2 3**  **Div R3 R1 R2** | **( ( 3 - ( 5 + 9 ) ) / ( 2 \* 3 )**  **Three Address File:**  **Add R0 5 9**  **Sub R1 3 R0**  **Mul R2 2 3**  **Div R3 R1 R2** | **Yes** |
| **Test Case for expression w/o spaces:**  **(2+3\*5)-8/5\*(5-2)**  **NOTE: I had to uses spaces between digits because code treats combined digits as single digit** | **( ( 3 - ( 5 + 9 ) ) / ( 2 \* 3 ) )**  **Three Address File: (New Segment)**  **Add R0 5 9**  **Sub R1 3 R0**  **Mul R2 2 3**  **Div R3 R1 R2** | **( ( 3 - ( 5 + 9 ) ) / ( 2 \* 3 ) )**  **Three Address File: (New Segment)**  **Add R0 5 9**  **Sub R1 3 R0**  **Mul R2 2 3**  **Div R3 R1 R2** | **Yes** |
| **Test Case for invalid input:**  **2 3 &** | **Throws Exception:**  **“Invalid input.**  **The Postfix Expression can only contain integer operands (0-9) and the following arithmetic operators:**  **+ - \* /.”**  **Three Address File: Nothing Added** | **Throws Exception:**  **“Invalid input.**  **The Postfix Expression can only contain integer operands (0-9) and the following arithmetic operators:**  **+ - \* /.”**  **Three Address File: Nothing Added** | **Yes** |
| **Test Case for invalid operator:**  **10 2 8 \* % 3 -** | **Throws Exception:**  **“Invalid input.**  **The Postfix Expression can only contain integer operands (0-9) and the following arithmetic operators:**  **+ - \* /.”**  **Three Address File: Nothing Added** | **Throws Exception:**  **“Invalid input.**  **The Postfix Expression can only contain integer operands (0-9) and the following arithmetic operators:**  **+ - \* /.”**  **Three Address File: Nothing Added** | **Yes** |
| **Additional Test Case:**  **9 6 + 8 2 / - 3 \*** | **( ( ( 9 + 6 ) - ( 8 / 2 ) ) \* 3 )**  **Three Address File: (New Segment)**  **Add R0 9 6**  **Div R1 8 2**  **Sub R2 R0 R1**  **Mul R3 R2 3** | **( ( ( 9 + 6 ) - ( 8 / 2 ) ) \* 3 )**  **Three Address File: (New Segment)**  **Add R0 9 6**  **Div R1 8 2**  **Sub R2 R0 R1**  **Mul R3 R2 3** | **Yes** |

**UML Class Diagram:**

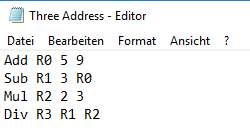
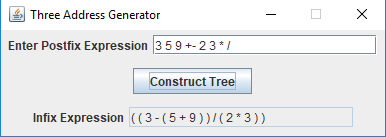


**Test Case Table:**

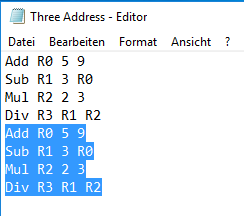
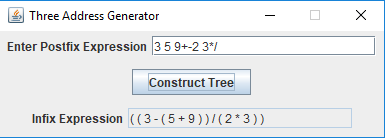
Screen Capture of me successfully compiling and executing my Java program:



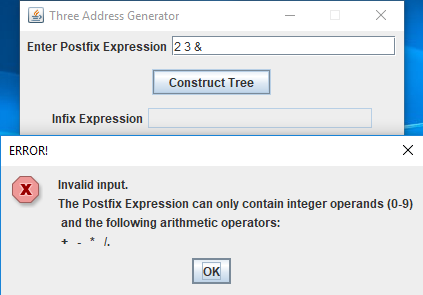
**Screen Captures of Test Case #1:**

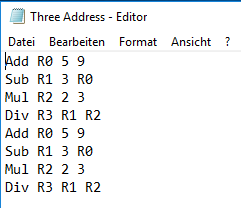


**Screen Captures of Test Case #2:**

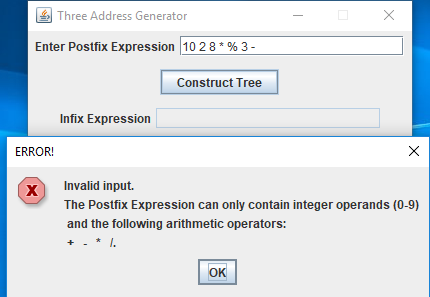


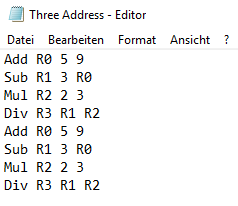
**Screen Captures of Test Case #3:**



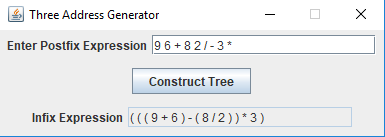


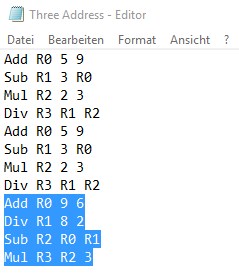
**Screen Captures of Test Case #4:**





**Screen Captures of Test Case #5:**





**Lessons Learned:**

This was an admittedly challenging project for me. Fortunately, I was able to learn some new things to help me as a program designer. Most importantly, I learned how to implement an arithmetic tree data structure in Java. This was used to pair operand nodes to an operator node in order to create an infix expression. It was interesting to do a project that is essentially the reverse of what we did last time around.