Team Project 2A

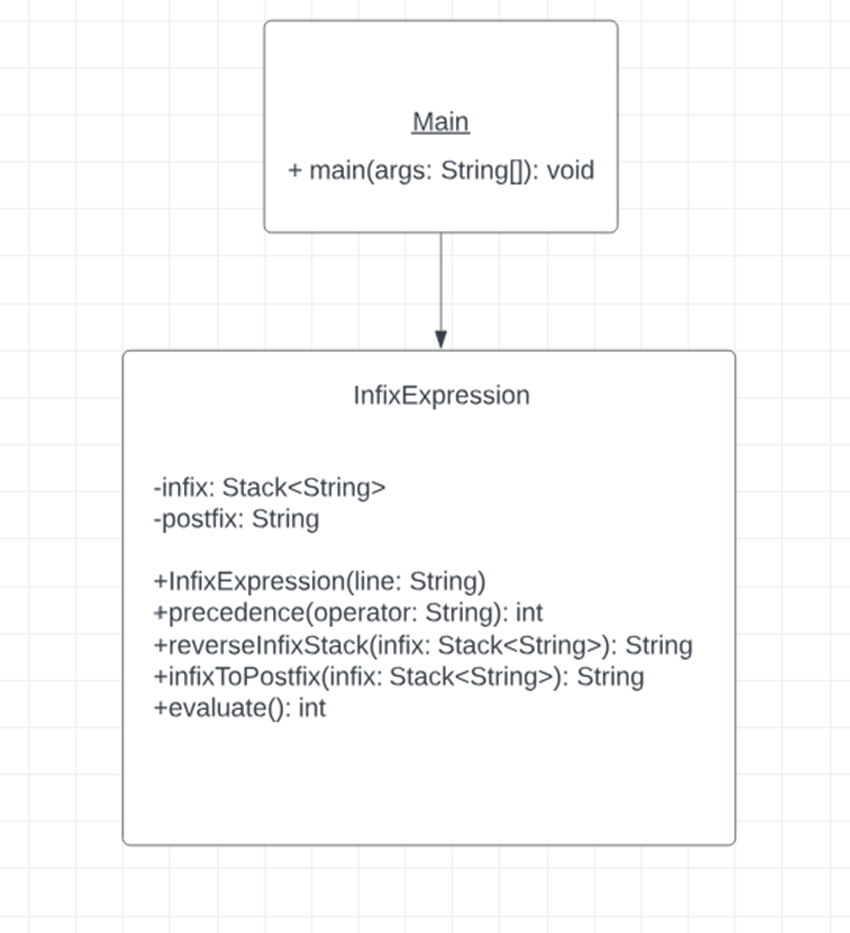
Infix Expression Parser

Andrew Lange, Mason Johnson, Matt Hays, Steven Valet

**Design Explanation**

The project is broken into the main method and the “InfixExpression” class. The main method loops through each line of the input file, upon each iteration all white space is corrected to have a leading and following space for each operator, parenthese, and logical operator. In the “InfixExpression” class the constructor creates a stack object with a scanner and logical branch. Once an “InfixExpression” object is created the constructor prints the contents of the stack to the console. Then the stack is reversed and converted to postfix via helper methods. Once that is completed the evaluate method is called to obtain the result of the expression and said expression is printed to the console.

**UML Chart Diagram**



**Test cases**

Test case 1:

**12 + ( 6 \* ( 2 \* 1 ))**

Expected result:

**24**

Actual result:

**24**

Test case 2:

**5 > 3**

Expected result:

**1**

Actual result:

**1**

**Team Member Contributions**

**Andrew Lange -** UML diagram, finished report

**Mason Johnson -** UML diagram and design outline

**Matt Hays -** Started repository, code logic and methods

**Steven Valet -** Started report, code error correcting

**Future Improvements**

Improvements could be made by moving the white space correction to the constructor. The need for a postfix conversion could also be removed in the future so that the program could read the expression as is and evaluate from the original infix formatting.