

Team Project 3A: Binary Tree Infix Expression Parser

Rawan Alhachami, Ollie Peel

CPT-287-R82 Intro to Data Struct. With Java

Dr. Wang

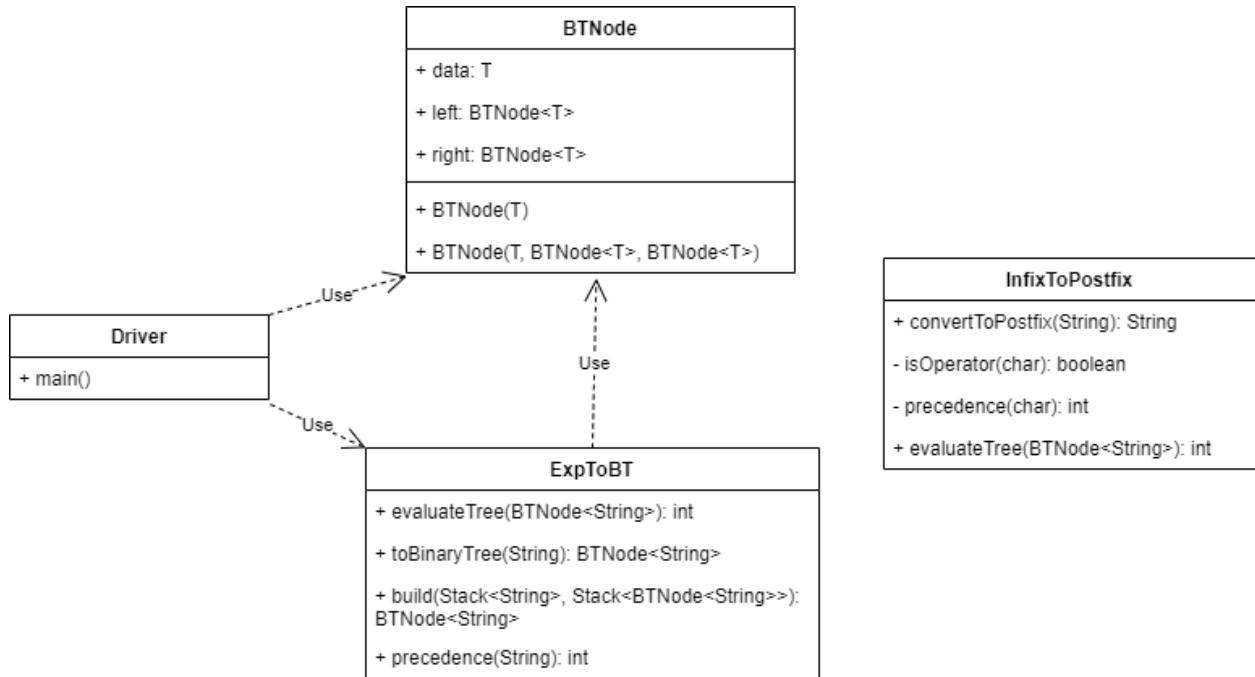
July 23, 2024

System Design

Our system starts with the main method in the Driver class which uses the ExpToBT.toBinaryTree method to convert a current expression into a binary expression tree. The ExpToBT.toBinaryTree method uses the build method in the same class to build and return binary tree nodes. The toBinaryTree method also uses the precedence method in the same class to assign and return the precedence of operators.

Our system used the stack, binary tree, and array list data structures. Stacks were used in the toBinaryTree and build methods of the ExpToBT class in order to hold the operators of the expression as well as the binary tree nodes that the expression was being converted to. Binary tree nodes were defined in the BTNode class and were used across the Driver and ExpToBT classes. As previously stated, the ExpToBT class uses the binary tree data structure to create a binary tree of an expression, and the Driver class is only provided with the root node of the resulting tree. The Driver class also uses an array list to store each expression taken from the input file.

UML Diagram



Test Cases

Case 1:

Infix Expression: $1 + 2 * 3$

Postfix Expression: $1\ 2\ 3\ *\ +$

Result: 7

Case 2:

Infix Expression: $1 + 3 > 2$

Postfix Expression: $1\ 3\ +\ 2\ >$

Result: 1

Contributions

Rawan Alhachami: Wrote code for InfixToPostfix class which reads the infix expression from the Expressions.txt sample file to convert to postfix and added helper method to check if the character is an operator and if so, check its precedence. Also added test cases including the input and output, which take the infix expression and then return the postfix result. Also included some improvements for our design in the future.

Ollie Peel: Wrote code for the Driver class which reads data from an input file and uses the ExpToBT.toBinaryTree method. Wrote the BTNode class, Evaluator class, and ExpToBT class. Also created a sample input file called Expressions.txt and set up the layout for the project report as well as worked on the system design and UML diagram sections.

Improvements

In the future, there is room for improvement in the efficiency of the evaluation. Currently, the handles single-character operands and implementation of a more sophisticated method for multi-character handling would be able to parse more complex expressions than the ones in the Expressions file.

The error handling could have also been more case-specific, like a divide-by-zero error or a mismatched parentheses error.