Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 3 REPORT

ELİF AKGÜN
1801042251

Course Assistant: ÖZGÜ GÖKSU

1 INTRODUCTION

1.1 Problem Definition

In this assignment, the InfixToPostfix class was created to calculate the value of postfix expressions. This class evaluates postfix expressions, given infix expressions must be convert postfix expressions. This assignment based on stack data structures. A file is given and program reads that file and use as input.

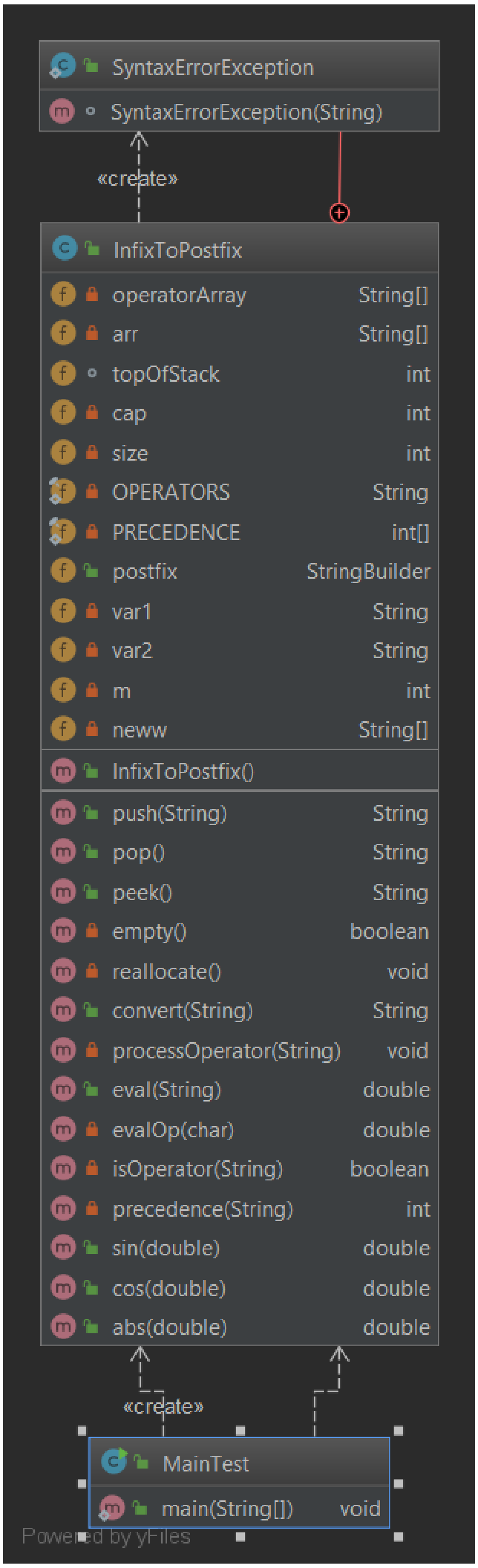
.

1.2 System Requirements

My solution also does not require a specific piece of hardware This program work with 128KB of memory, it does not keep memory so much and it needs the interface to work on the smart phone. It can work if it is provided.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams

The only thing that a user needs to do is write the expression in the file that to be converded infix to postfix than evaulate. User can run this program to click the run button.

2.3 Problem Solution Approach

During the project development phase, the objects that should be used first were considered and class was created for those objects.

InfixToPostfix class method requirements that based on stack:

- **String push(String obj):** Insert a new item on top of the stack.
- **String pop()** : Remove and return the top item on the stack
- **String peek()**: Return the top item on the stack
- **boolean empty()**: Return true if the stack is empty

This methods based on stack data structure and perform like that.

Other class method requirements:

- **void reallocate() :** Reallocates the array containing the stack array data
- **String convert(String infix) :** Converts a string from infix to postfix
- **void processOperator(String op) :** This method processes the operators and all operators controlled different way.
- **double eval(String expression) :** This method evaulate the postfix expression that converted from infix.
- **double evalOp(char op) :** Evaluates the current operation and according to operation, it implements different things.
- **boolean isOperator(String ch) :** Determine whether a String is an operator
- **int precedence(String op) :** Determine the precedence of an operatör
- **double sin(double value) :** Determine the sinus value of the given number from taylor theorem
- **double cos(double value) :** Determine the cosinus value of the given number from taylor theorem
- **double abs(double value) :** Determine the absolute value of the given number

The ExperimentList class inner class requirements:

- **public static class SyntaxErrorException:** This class extends the Exception and reports a syntax error.

Time Complexity of Methods:

Method	Explanation
String push(String obj)	This method does not include any loop so its complexity is O(1).
String pop()	This method does not include any loop so its complexity is O(1).
String peek()	This method does not include any loop so its complexity is O(1).
void reallocate()	This method does not include any loop so its complexity is O(1).
boolean empty()	This method does not include any loop so its complexity is O(1).
String convert(String infix)	This method includes a while loop so its complexity is O(n).
void processOperator(String op)	This method includes a while loop so its complexity is O(n).
double eval(String expression)	This method includes a while loop so its complexity is O(n).
double evalOp(char op)	This method includes only switch case statement so its complexity is O(1).
boolean isOperator(String ch)	This method does not include any loop so its complexity is O(1).
precedence(String op)	This method does not include any loop so its complexity is O(1).
double sin(double value)	This method does not include any loop so its complexity is O(1).
double cosinus(double value)	This method does not include any loop so its complexity is O(1).
double abs(double value)	This method does not include any loop so its complexity is O(1).

Space complexity is O(n) because one dimensional arrays used.

3 RESULT

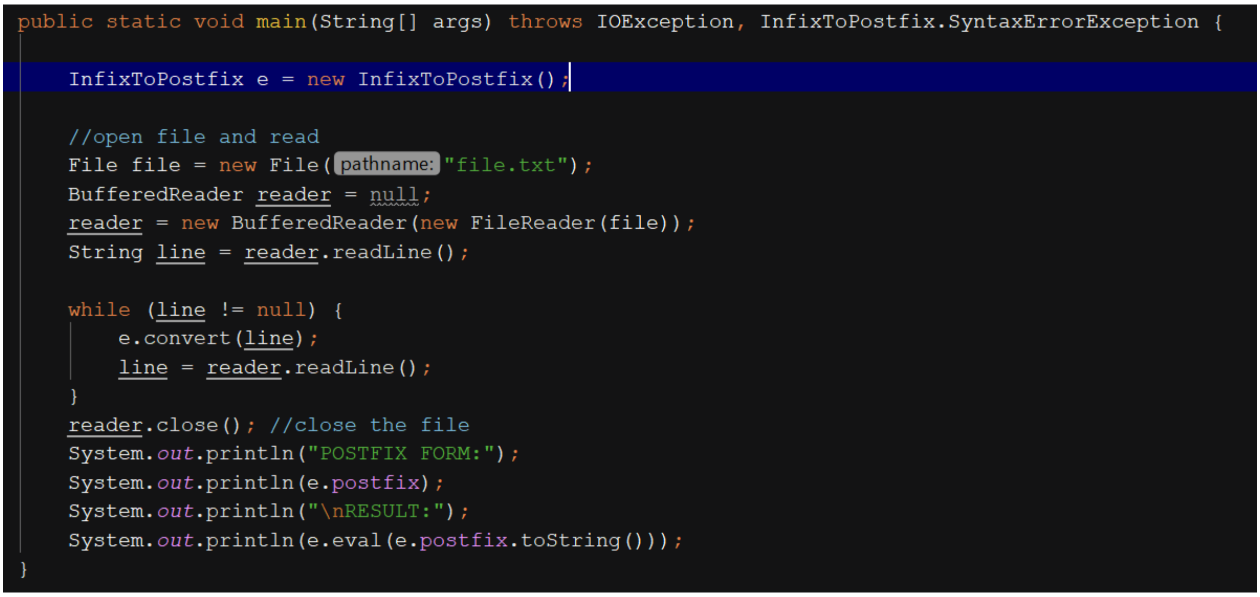
3.1 Test Cases

Input file:

```
y=3
z=16

( y + sin( y * z ) ) + ( z * abs( -10.3 ) )
```

Main test file:



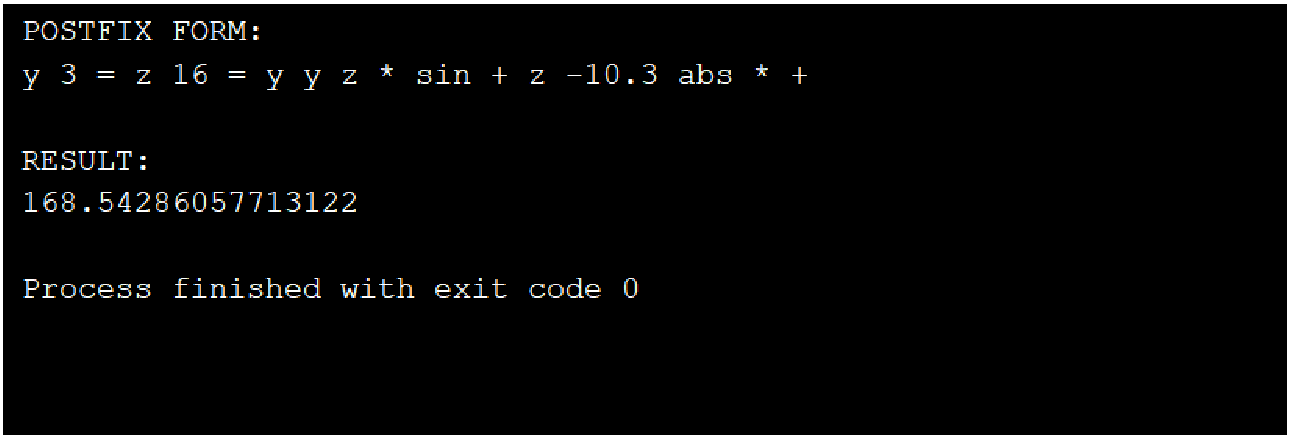
```
public static void main(String[] args) throws IOException, InfixToPostfix.SyntaxErrorException {

    InfixToPostfix e = new InfixToPostfix();

    //open file and read
    File file = new File( pathname: "file.txt");
    BufferedReader reader = null;
    reader = new BufferedReader(new FileReader(file));
    String line = reader.readLine();

    while (line != null) {
        e.convert(line);
        line = reader.readLine();
    }
    reader.close(); //close the file
    System.out.println("POSTFIX FORM:");
    System.out.println(e.postfix);
    System.out.println("\nRESULT:");
    System.out.println(e.eval(e.postfix.toString()));
}
```

3.2 Running Results



```
POSTFIX FORM:
y 3 = z 16 = y y z * sin + z -10.3 abs * +

RESULT:
168.54286057713122

Process finished with exit code 0
```