

**Gebze Technical University  
Computer Engineering**

**CSE 222 - 2019 Spring**

**HOMEWORK 3 REPORT**

**Rumeysa KARAKAVAK  
141044063**

Course Assistant : Özgü Göksu

# 1 INTRODUCTION

## 1.1 Problem Definition

This homework contains 2 part. In the first part is a little image processing. There is a ASCII binary file that contains only 1 and 0s as input. If there are neighbor 1's (they must be adjacent from top, bottom, left or right) on this file, there is a path between them. They are clusters of 1's. We print as output is the count of this clusters that we say white component of image.

In the second part we must to work on infix notation expression. This expression must convert to prefix notation. Then that conversion, program will must calculate the result of the expresion.

## 1.2 System Requirements

This program was coded and can running on IntelliJIDEA(it can runs another platforms probably). Used latest version of JDK 11.0.2. .

Operating System (i.e. Windows 10, Linux or Mac OS X 10.3.8)

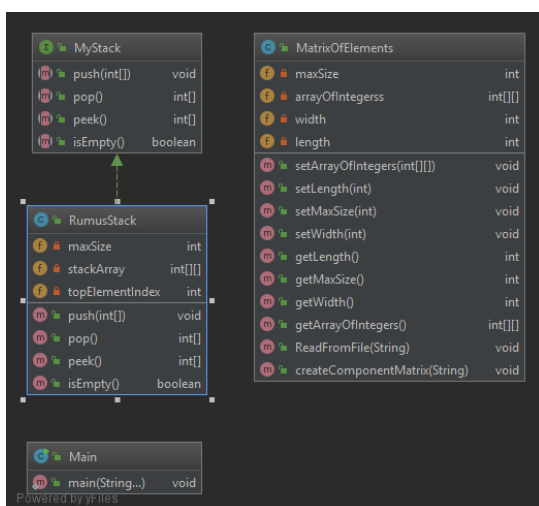
Processor Speed (i.e. Pentium 4, 3.2 GHz or Power PC G5, 2.0 GHz)

Memory, a.k.a. RAM (i.e. 512 MB)

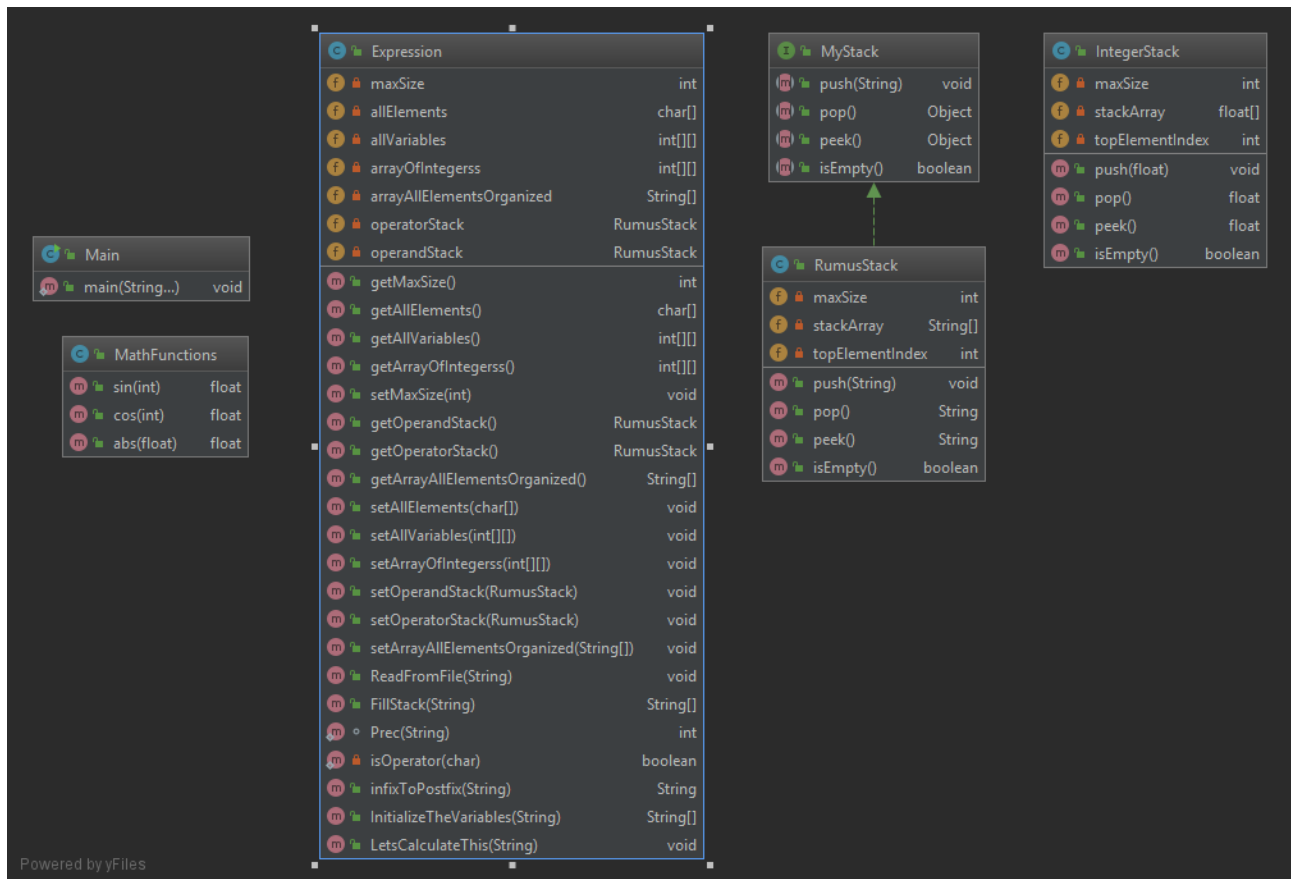
Hard Disk Space (i.e. 80 GB available)

# 2 METHOD

## 2.1 Class Diagrams



## Second Part



## 2.2 Use Case Diagrams

In the first part we need a ASCII text file that contains 0 and 1s. The program must get the file name as a command line argument. When program starts to run, `readFromFile` method reads the ones and zeros and fills in an integer array and starts to search the 1's in the file. Then found the ones, it cluster them and rename the character as a alphabetic character. After this process, counts the different alphabetic characters and prints this count on screen. This is the output.

In the second part we need a text file that contains variables and values of them and an expression that format is prefix. When program starts to run, the variables evaluate and expression read by a method. Then the prefix notation converts the postfix notation by another method. When converted the expression to postfix notation, calculates the result of this expression. This is the output.

## 2.3 Problem Solution Approach

This homework's goal was using the stack. The stack is my own implementation. Implementing this stack, used primitive two dimensional array for the first part and primitive one dimensional array for the second part. Using the stack really big deal and helpful. For the first part i made clustering just looked at their neighbors without caring for other components. And for the second part i calculated organized the expression owing to stack.

### 3 RESULT

### 3.1 Test Cases

## Part 1

## Test files

```
0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 0 0
0 1 1 1 1 0 0 1 0 1 0
0 1 1 0 1 0 0 0 0 1 1
0 0 0 1 1 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0
```

[illegible]

## Part 2

## Test files

$w=5$

x=60

$$(w + 4) * (\cos(x + 120) - 4.5)$$

and

 $y=30$ 
$$z=3$$
$$(y + \sin(y * z)) + (z * \text{abs}(-10.3))$$

### 3.2 Running Results

## PART 1

```
000000000000
00A000B00000
0AAAA00C0D0
0AA0A0000DD
000AA000DD0
000000000000
```

```
Number of white components : 4
```

```
00AAAAA00000000000000000000000000000000000000BBB0
0000AAA0000000000000000CCCC00000000000000000000BBB0
0000AAA00000000000000000C000000000000000000000BBB0
0000AAA0DD00000000000000C000000000000000000000BBB00
00000AAA0DD0000000000000C00000000000000000000000000
000000AAA000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000EEEEEE00000000000000000000000000
00000000000000000000EEEE0000000000000000000000000000
00000FFF000000000000EEEE0000000000000000GG000000000
000000FF000000000000EEEE000000000000GG0000000000000
000000FF000000000000EEEE000000000000GG0000000000000
00000000000000000000EEEE000000000000GGG000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
0HHH0000000000000000000000000000000000000000000000
00HHH0000000000000000000000000000000000000000000III
00000000000000000000000000000000000000000000000III
```

```
Number of white components : 9
```

## PART 2

y=30, z=3

```
Postfix notation = y sin y z * + z abs -10.3 * +  
Result is = 61.9
```

w=5, x=60

```
Postfix notation = w 4 + cos x 120 + 4.5 - *  
Result is = -49.5
```

- Main titles -> 16pt , 2 line break
- Subtitles -> 14pt, 1.5 line break
- Paragraph -> 12pt, 1.5 line break