Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 3 REPORT

ÖMER ÇEVİK
161044004

Course Assistant: Özgü Göksu

1 INTRODUCTION

1.1 Problem Definition

Part 1

In that project, the problem is in image that represented as a text file using zeros and ones, we want to know how many sets of ones we have in that text using stack data structure.

Part 2

In that project, the problem is in text file there is a polynomial expression given in infix notation. The wanted is that convert infix to postfix notation and calculate the expression given variables in that text using stack data structure.

1.2 System Requirements

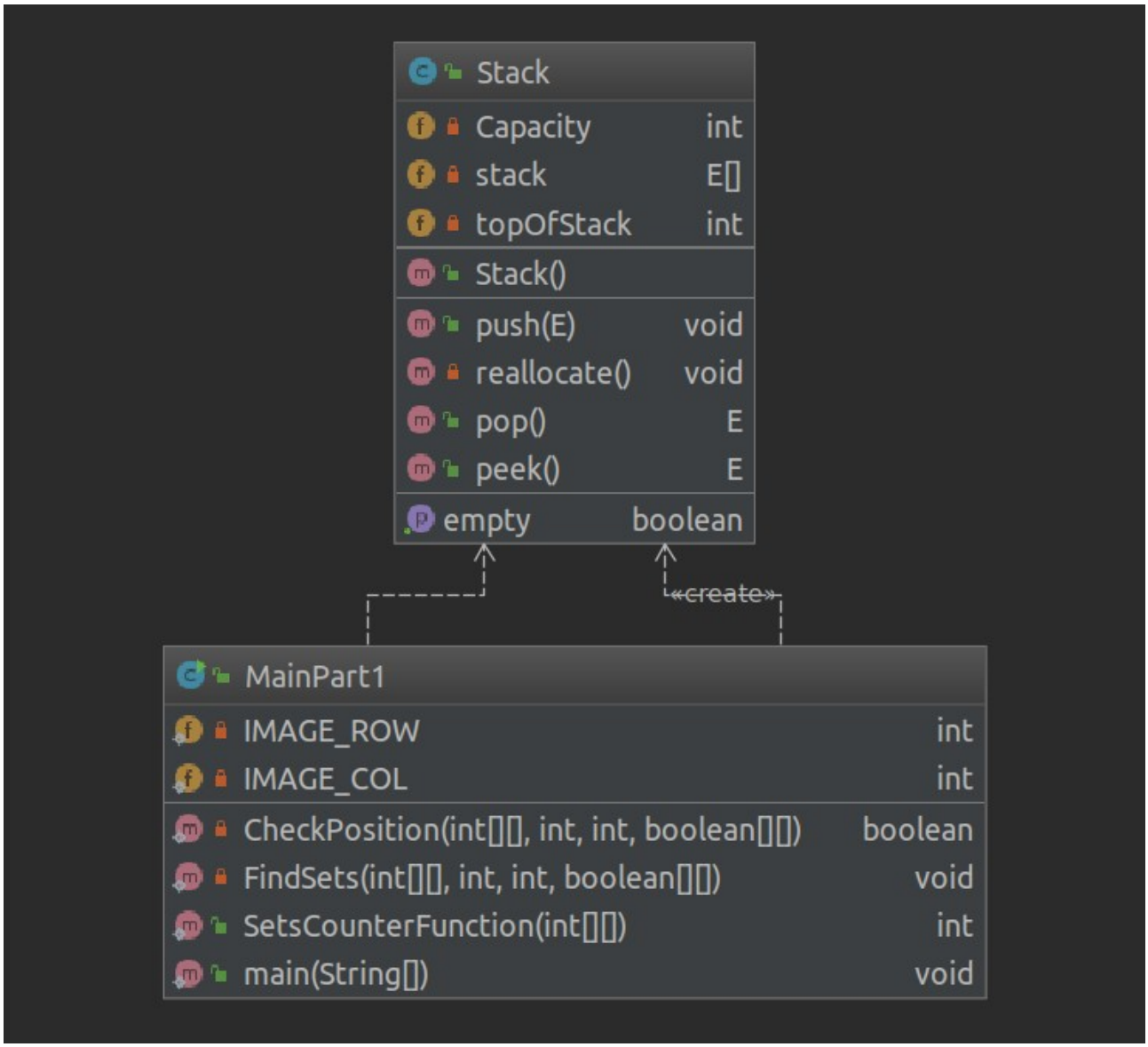
Part 1 & Part 2

My solution doesn’t require a specific piece of hardware, or maybe a certain minimal amount of memory, or a certain operating system, or a software library to be installed in order to run properly. It built by IntelliJ IDEA and used java programming language. It might run correctly if you use Java Virtual Machine and JDK is at least 8. Will my program work with 128KB of memory? I don’t know, I didn’t work with 128KB of memory but maybe it can work. If your nephew’s smartphone’s operating system is Android then it must run on it.

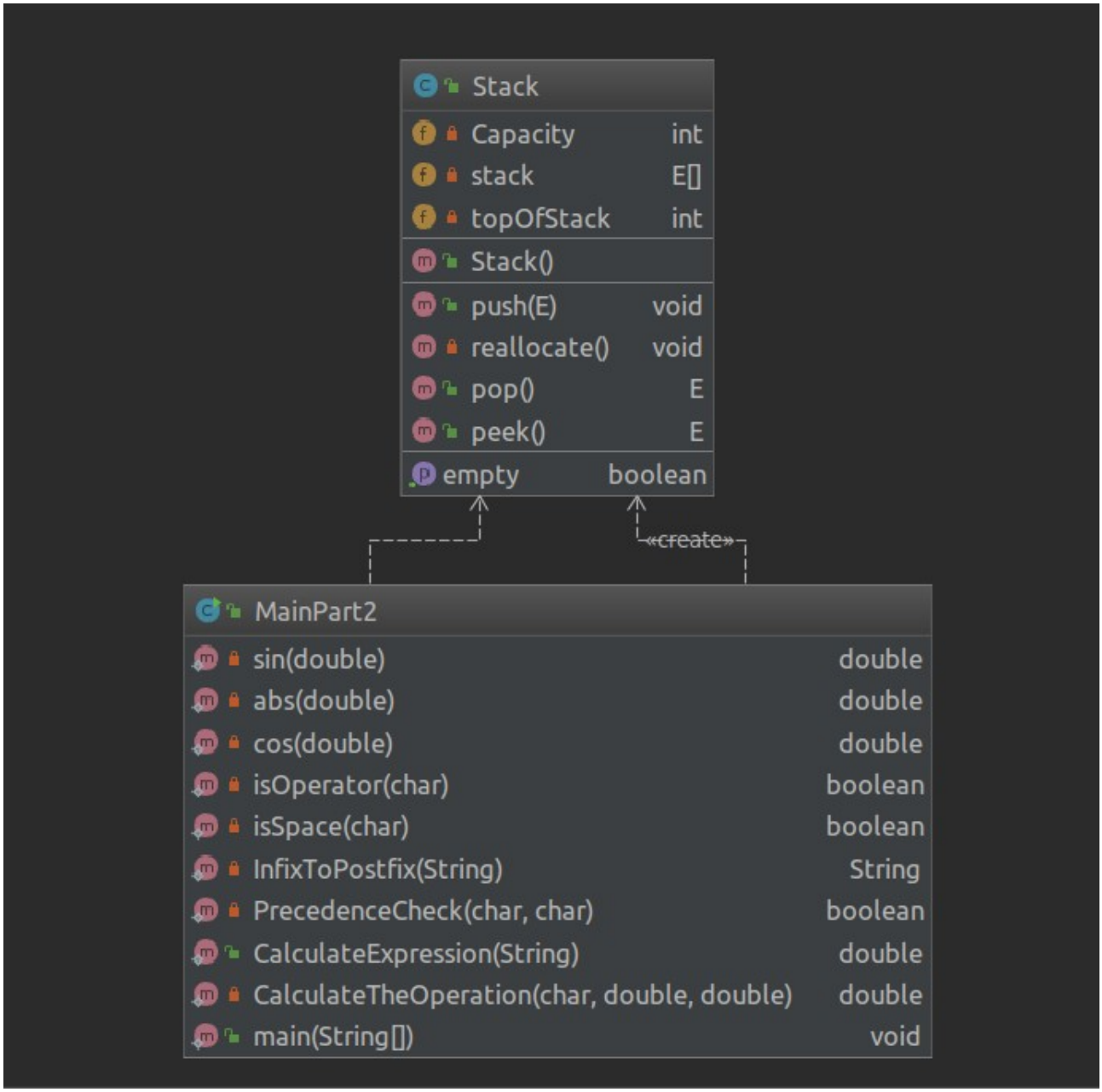
2 METHOD

2.1 Class Diagrams

Part 1

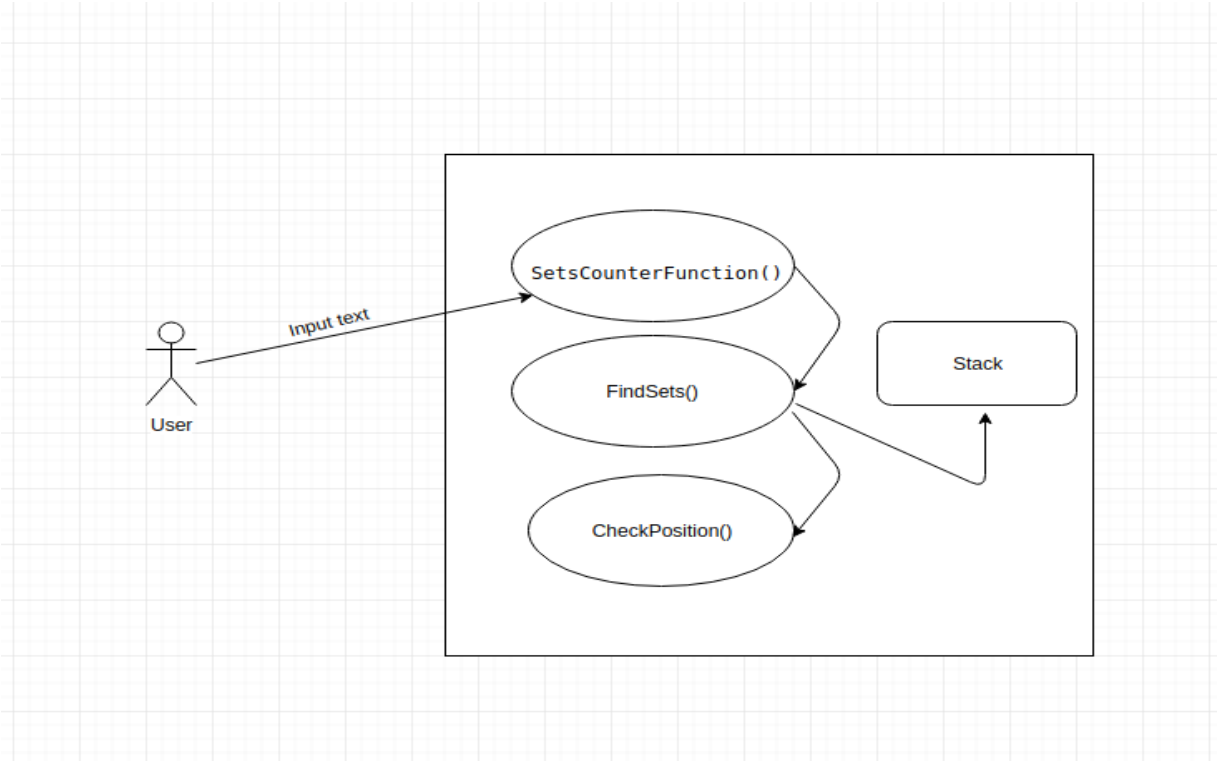


Part 2

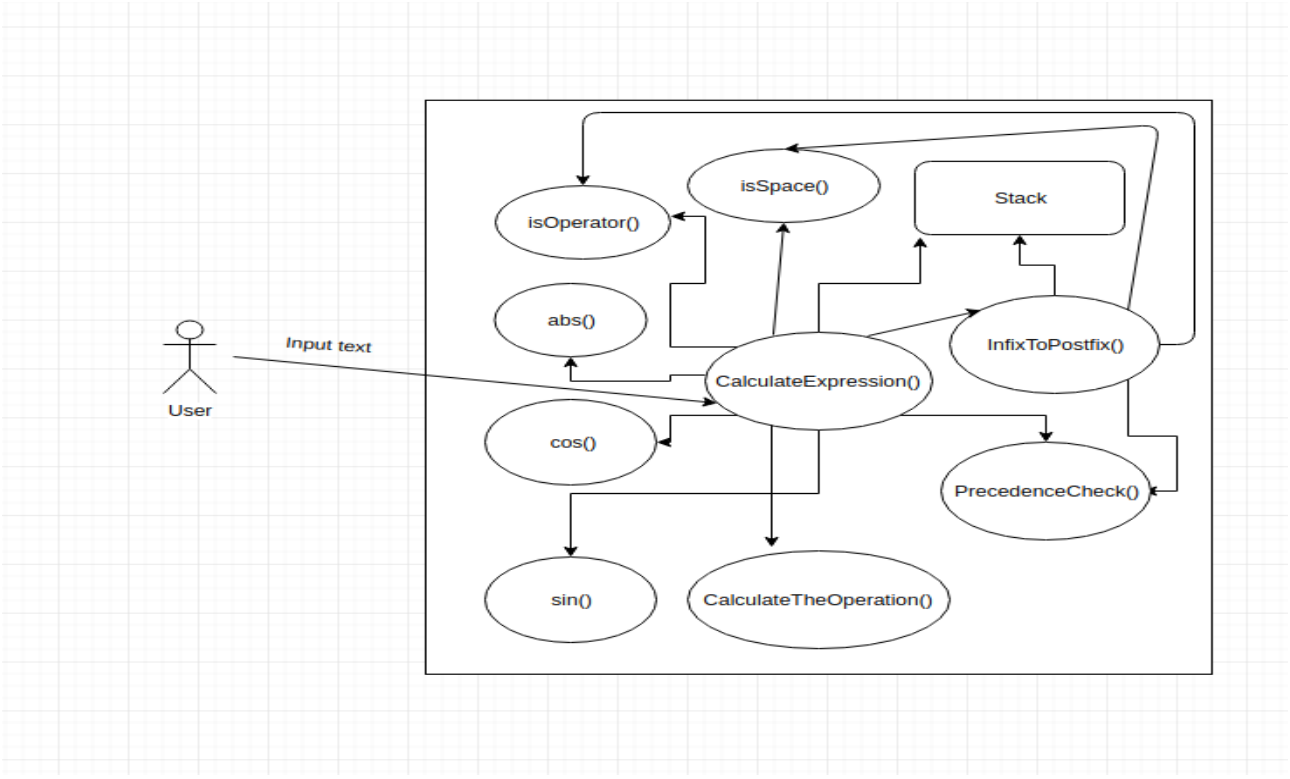

Stack
Capacity int
stack E[]
topOfStack int
Stack()
push(E) void
reallocate() void
pop() E
peek() E
empty boolean
«create»
MainPart2
sin(double) double
abs(double) double
cos(double) double
isOperator(char) boolean
isSpace(char) boolean
InfixToPostfix(String) String
PrecedenceCheck(char, char) boolean
CalculateExpression(String) double
CalculateTheOperation(char, double, double) double
main(String[]) void

2.2 Use Case Diagrams

Part 1



Part 2



2.3 Problem Solution Approach

Part 1

In my programme, I used generic type stack data structure. It was hard to think algorithm about how to find all sets in text file. Firstly I stored all datas in a primitive two dimensional integer array. Then I started to search ones in that array. When I found a one then I checked out of its neighbors if they are also one. If they are ones then I pushed them into stack. I checked that neighbors's neighbors in same way till the all set has been found. To not check again the same location I signed the location that I checked before and It gave me time efficiency.

Time Complexity

- CheckPosition() : It just returns true or false in one instruction. O(1).
- FindSets() : It has one while loop but it is less than n. O(1).
- SetsCounterFunction() : It has nested for loops but it is looping n. O(n).
- main() : It just reads file in a loop sized n. O(n).

Part 2

In my programme, I used generic type stack data structure. Firstly I read lines of text into string and I pushed them to stack. The last input in stack is out expression. Then I made a loop in that stack and I used StringTokenizer for to tokenize the variables in text. I put variables values in that expression. Then I called to evaluate the expression. The evaluation of infix notation called the InfixToPostfix method and it converted infix notation to postfix notation using stack data structure. After that postfix notation is ready to evaluate. CalculateExpression method is worked well and calculated the expression.

Time Complexity

- sin() : It calculates sinus of given value using for loop to n. O(n).
- cos() : It calculates cosinus of given value using for loop to n. O(n).
- abs() : It calculates abs of given value using one instruction. O(1).
- isOperator() : It checks and returns only one instruction. O(1).
- isSpace() : It checks and returns only one instruction. O(1).
- PrecedenceCheck() : It checks and returns only one instruction. O(1).
- InfixToPostfix() : It has while loop given n times. O(n).
- CalculateTheOperation() : It checks and returns only one instruction. O(1).
- CalculateExpression() : It calculates all values in expression in while loop. O(n).
- main() : It just reads file into stack in while loop and calls CalculateExpression. O(n).

3 RESULT

3.1 Test Cases

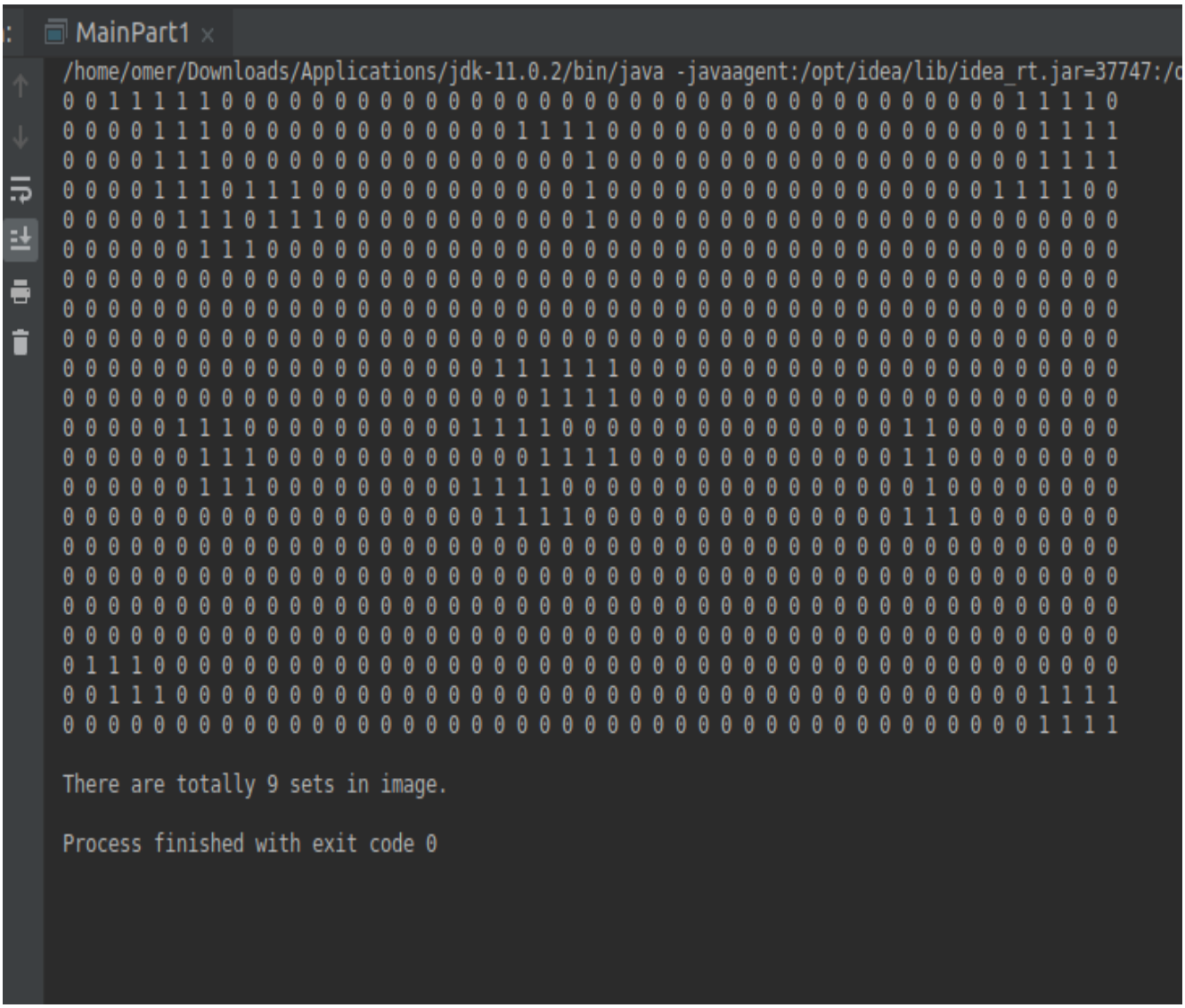
Part 1 & Part 2

Main Test

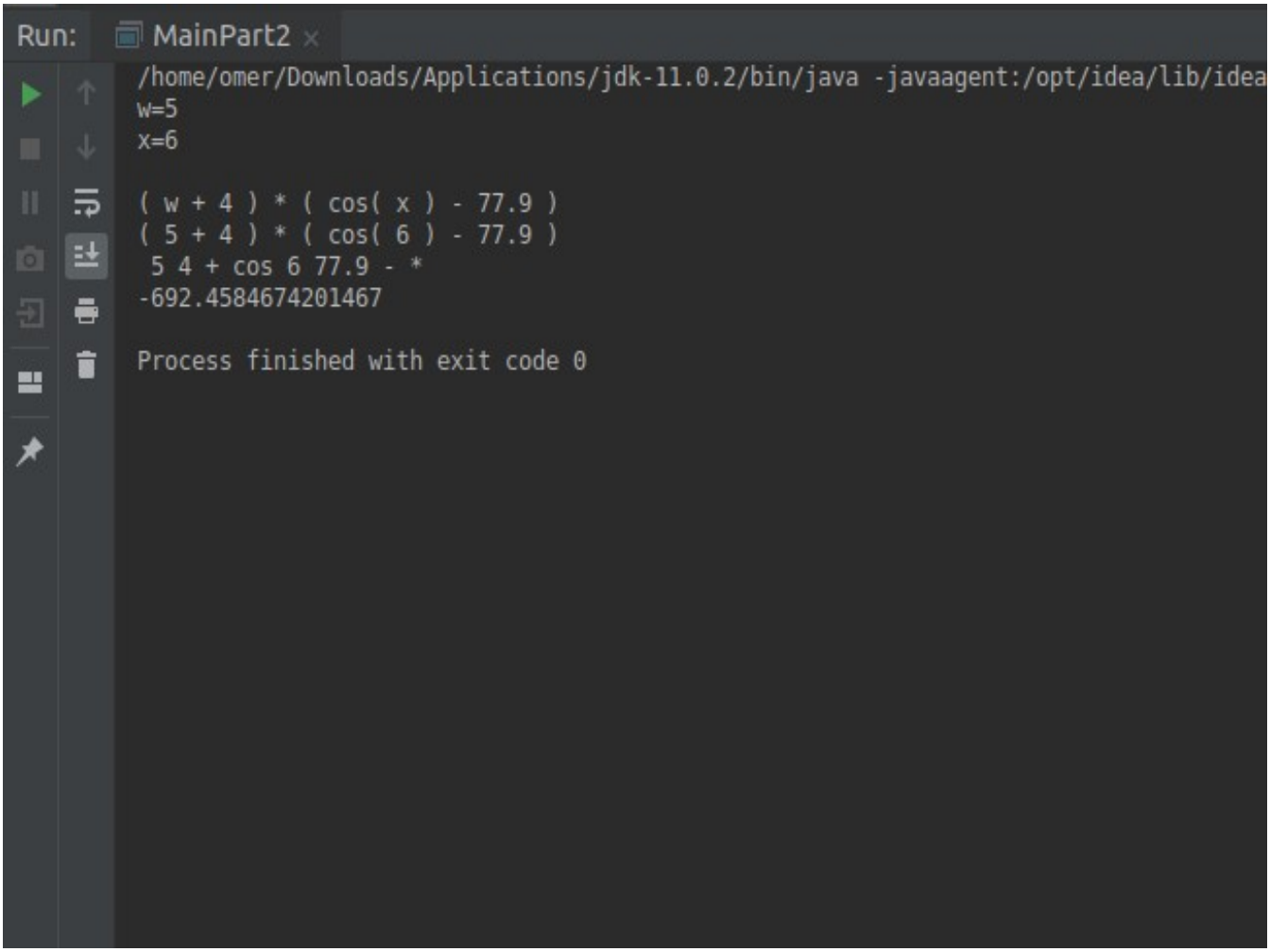
I used main method to test my programme. Given input text files have been tested and that represent it works well. The results are in Running Results.

3.2 Running Results

Part 1 Results



Part 2 Results



For Part 1 Results:

- Programme executed well without any error or warnings. Outputs are correct.
- With handchecking outputs result is verified well.

For Part 2 Results:

- Programme executed well without any errors or warnings. Outputs are correct.
- In output steps firstly printed the input file then I put the variables values inside the expression. Secondly infix to postfix has been executed. Then the result is calculated.
- With handchecking outputs result is verified well.