1) **Problem:** In this project, writing a program for executing a series of arithmetic operations in LLVM is given as problem. Arithmetic operations might be summation, subtraction, multiplication and division(+, -, *, /). Also precedence of operations might be determined by parentheses. Students should write a Java/C/C++ program in order to create LLVM code for the given inputs. Also, one should detect the syntax errors in the input. Errors might be unmatching parentheses, two operators in a row, more than one equals sign, illegal variable name, illegal position of equals sign, undefined token (or variable name).
2) **Solution:** I chose Java as programming language. I used 2 classes in order to solve the problem: Main and Components. Main is the class where I had done all of my executions. Component is the class where I stored all of my tokens in arithmetic operation (3, +, *, var1, (etc.) and their corresponding values (variable name, number, allocation number etc.) Further detail can be seen in the Javadoc file or same as below:

Class Main

- java.lang.Object
  - Main

```
public class Main
extends java.lang.Object
```

This class executes the statements of program

- ***Constructor Summary***

Constructors
Constructor and Description
`Main()`

All Methods	Static Methods	Concrete Methods

Modifier and Type	Method and Description
`static int`	`evaluation(java.util.List<java.lang.String> components)` Evaluates a given arithmetic operation
`static void`	`fileReading(java.lang.String fileName)` File is read and stored inside lines List
`static int`	`findMatching(java.util.List<java.lang.String> text, int index)` Finds the matching closing parenthesis of a given opening parenthesis
`static boolean`	`isInteger(java.lang.String str)` Finds whether given string is an integer or not
`static void`	`main(java.lang.String[] args)`

static java.util.List<java.lang.String>	stringtoRegex(java.lang.String str) Splits a string of arithmetic operation into tokens Example: 4 + 3 * (3 - 5) becomes ["4", "+", "3", "*", "(", "3", "-", "5", ")"]

- ***Method Summary***

- **Methods inherited from class java.lang.Object**

  equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ***Constructor Detail***

- **Main**

  public Main()

- ***Method Detail***

- **main**

  public static void main(java.lang.String[] args)
  throws java.lang.Exception

  Throws:
  java.lang.Exception

- **fileReading**

  public static void fileReading(java.lang.String fileName)

  File is read and stored inside lines List

  Parameters:
  fileName - name of file

- **stringtoRegex**

  public
  static java.util.List<java.lang.String> stringtoRegex(java.lang.String str)

  Splits a string of arithmetic operation into tokens Example: 4 + 3 * (3 - 5) becomes ["4", "+", "3", "*", "(", "3", "-", "5", ")"]

  Parameters:
  str - String to split

  Returns:
  list of tokens

- **evaluation**

- ```
  public
  static int evaluation(java.util.List<java.lang.String> components)
                        throws java.lang.Exception
  ```

  Evaluates a given arithmetic operation

  Parameters:

  ```
  components - list of tokens of the operation
  ```

  Returns:

  ```
  value of the operation
  ```

  Throws:

  ```
  java.lang.Exception - thrown if any illegal token is present
  ```

- **findMatching**

- ```
  public static int findMatching(java.util.List<java.lang.String> text,
                                 int index)
  ```

  Finds the matching closing parenthesis of a given opening parenthesis

  Parameters:

  ```
  text - arithmetic operation
  ```

  ```
  index - index of opening parenthesis in the text
  ```

  Returns:

  ```
  index of matching parenthesis, -1 is there is none
  ```

- **isInteger**

  ```
  public static boolean isInteger(java.lang.String str)
  ```

  Finds whether given string is an integer or not

  Parameters:

  ```
  str - String that is assumed to be an integer
  ```

  Returns:

  ```
  true, if the given string can be parsed into integer
  ```

Class Component

- java.lang.Object
  - Component

```
public class Component
extends java.lang.Object
```

Instances of this class represents each token in an arithmetic expression Example: "6", "+", "-", "/", "*", "var1", ")"

Author:

```
2014400102
```

-

- ***Constructor Summary***

Constructors

Constructor and Description
`Component(int num, int alloc)` Number instantiation
`Component(int num, int alloc, boolean isCalculated)` Number instantiation
`Component(java.lang.String oper)` Operator instantiation
`Component(java.lang.String var, int num, int alloc)` Variable instantiation

- ***Method Summary***

- **Methods inherited from class java.lang.Object**

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

-

- ***Constructor Detail***

- **Component**

- `public Component(java.lang.String var,`
- `int num,`
  `int alloc)`

Variable instantiation

Parameters:

`var - Name of variable`

`num - Number`

`alloc - Allocation number`

- **Component**

- `public Component(int num,`
  `int alloc)`

Number instantiation

Parameters:

`num - Number`

`alloc - Allocaiton number`

- **Component**

- ```
  public Component(int num,
                   int alloc,
                   boolean isCalculated)
  ```

  Number instantiation

  Parameters:

  `num - Number`

  `alloc - Allocation number`

  `isCalculated - true, if the allocation number of instance is used during calculation instead of the value of it`

- **Component**

```
public Component(java.lang.String oper)
```

Operator instantiation

Parameters:

`oper - Operator`
```