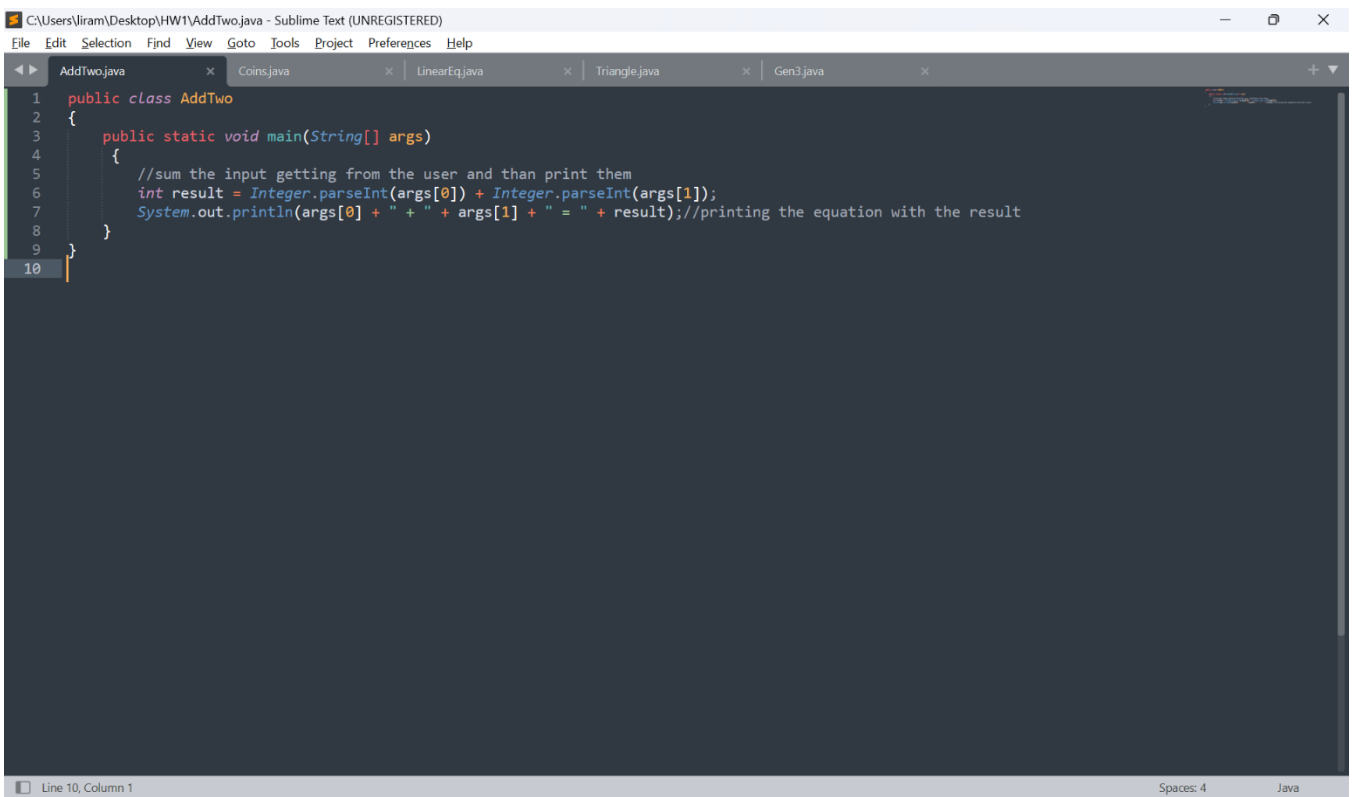


ADDTWO

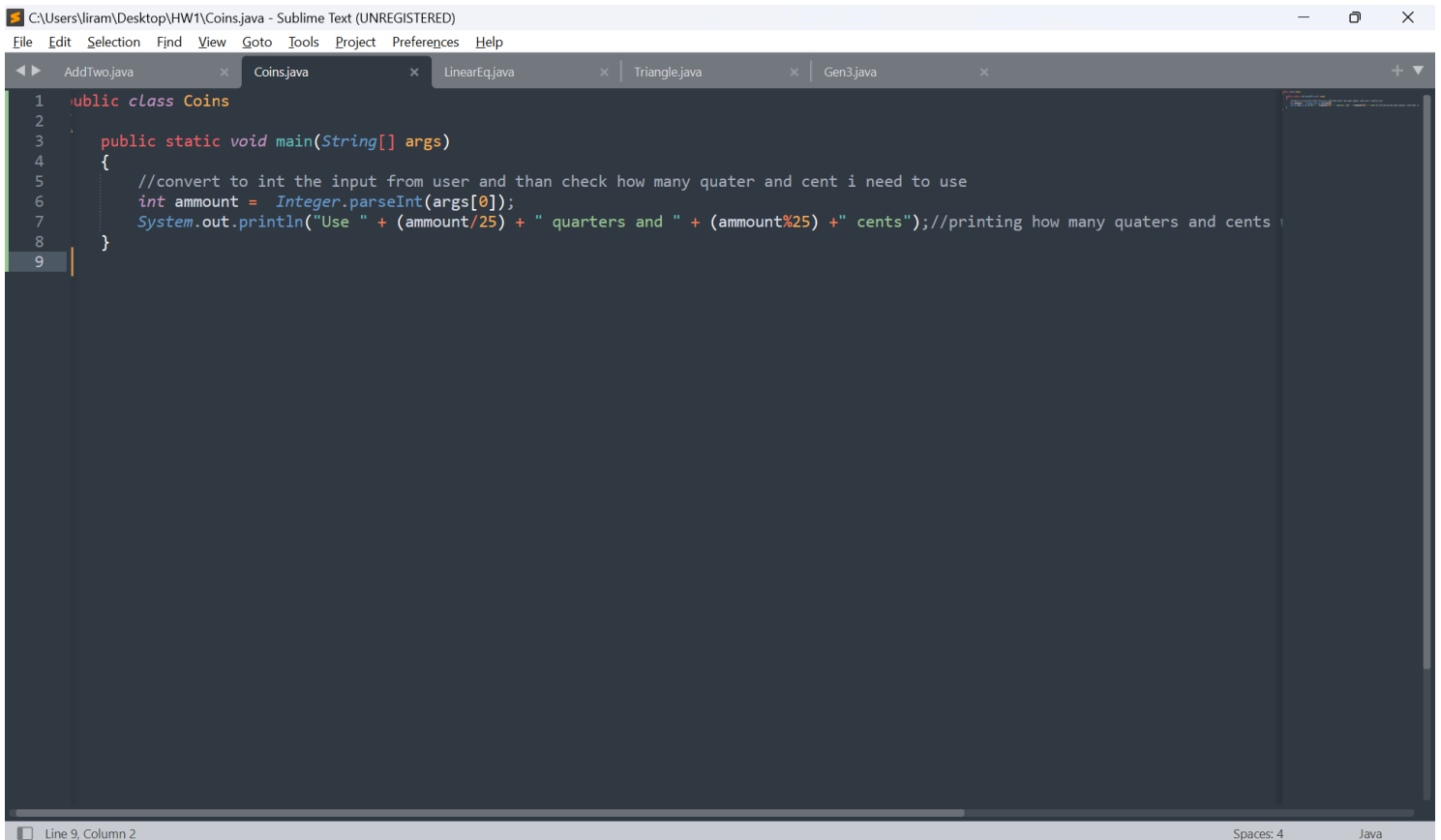
```
public class AddTwo
{
    public static void main(String[] args)
    {
        //sum the input getting from the user and than print them
        int result = Integer.parseInt(args[0]) + Integer.parseInt(args[1]);
        System.out.println(args[0] + " + " + args[1] + " = " + result); //printing the
equation with the result
    }
}
```



```
C:\Users\liram\Desktop\HW1\AddTwo.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
AddTwo.java Coins.java LinearEq.java Triangle.java Gen3.java
1 public class AddTwo
2 {
3     public static void main(String[] args)
4     {
5         //sum the input getting from the user and than print them
6         int result = Integer.parseInt(args[0]) + Integer.parseInt(args[1]);
7         System.out.println(args[0] + " + " + args[1] + " = " + result); //printing the equation with the result
8     }
9 }
10
Line 10, Column 1 Spaces: 4 Java
```

COINS

```
public class Coins
{
    public static void main(String[] args)
    {
        //convert to int the input from user and than check how many quater and
        cent i need to use
        int ammount = Integer.parseInt(args[0]);
        System.out.println("Use " + (ammount/25) + " quarters and " +
        (ammount%25) + " cents");//printing how many quaters and cents we can use
        that we use minimally ammount of cents that we could
    }
}
```



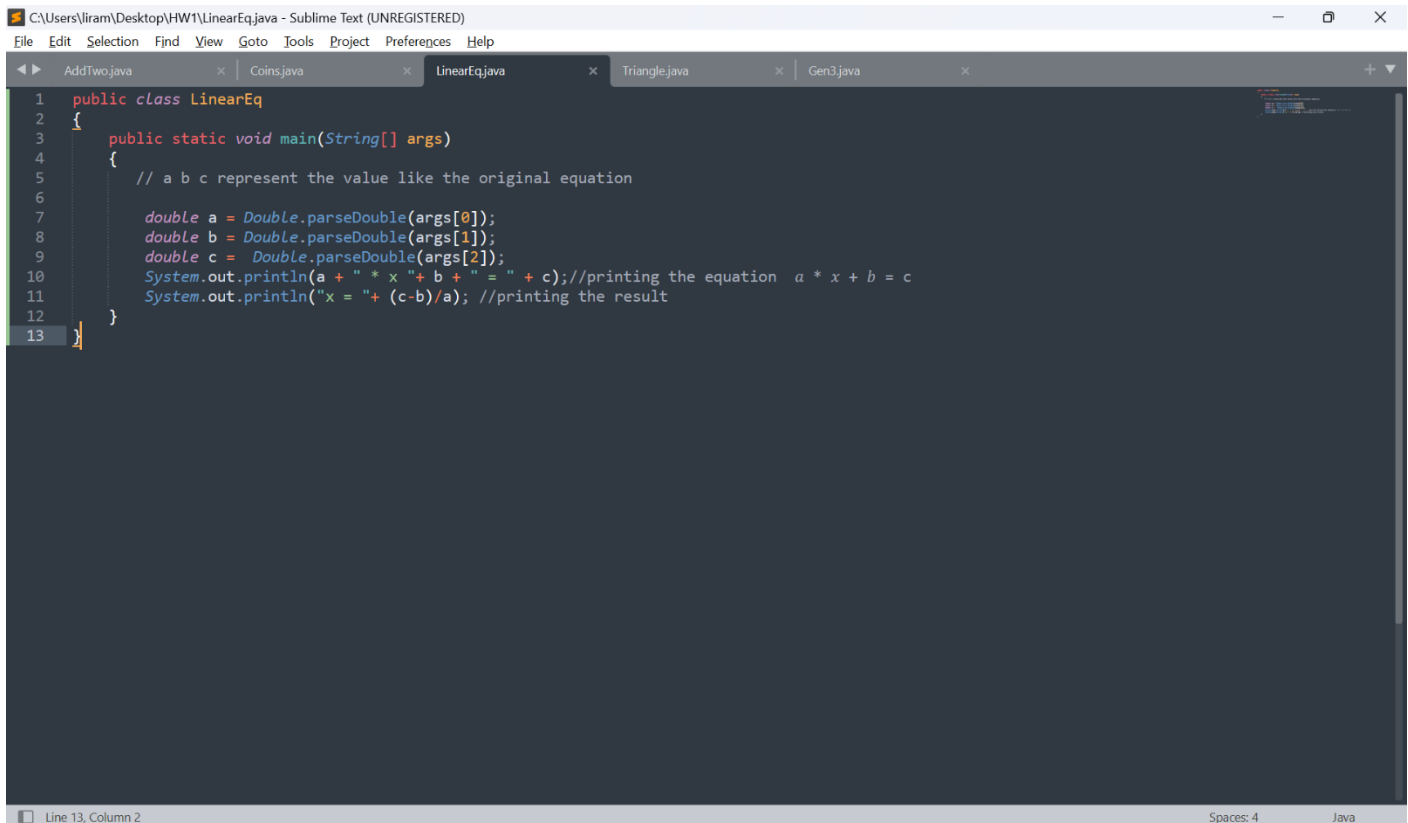
```
C:\Users\liram\Desktop\HW1\Coins.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
AddTwo.java x Coins.java x LinearEq.java x Triangle.java x Gen3.java x
1 public class Coins
2
3     public static void main(String[] args)
4     {
5         //convert to int the input from user and than check how many quater and cent i need to use
6         int ammount = Integer.parseInt(args[0]);
7         System.out.println("Use " + (ammount/25) + " quarters and " + (ammount%25) + " cents");//printing how many quaters and cents
8     }
9 }
```

Line 9, Column 2 Spaces: 4 Java

LinearEq

```
public class LinearEq
{
    public static void main(String[] args)
    {
        // a b c represent the value like the original equation

        double a = Double.parseDouble(args[0]);
        double b = Double.parseDouble(args[1]);
        double c = Double.parseDouble(args[2]);
        System.out.println(a + " * x "+ " + " + b + " = " + c); //printing the equation
         $a * x + b = c$ 
        System.out.println("x = "+ (c-b)/a); //printing the result
    }
}
```

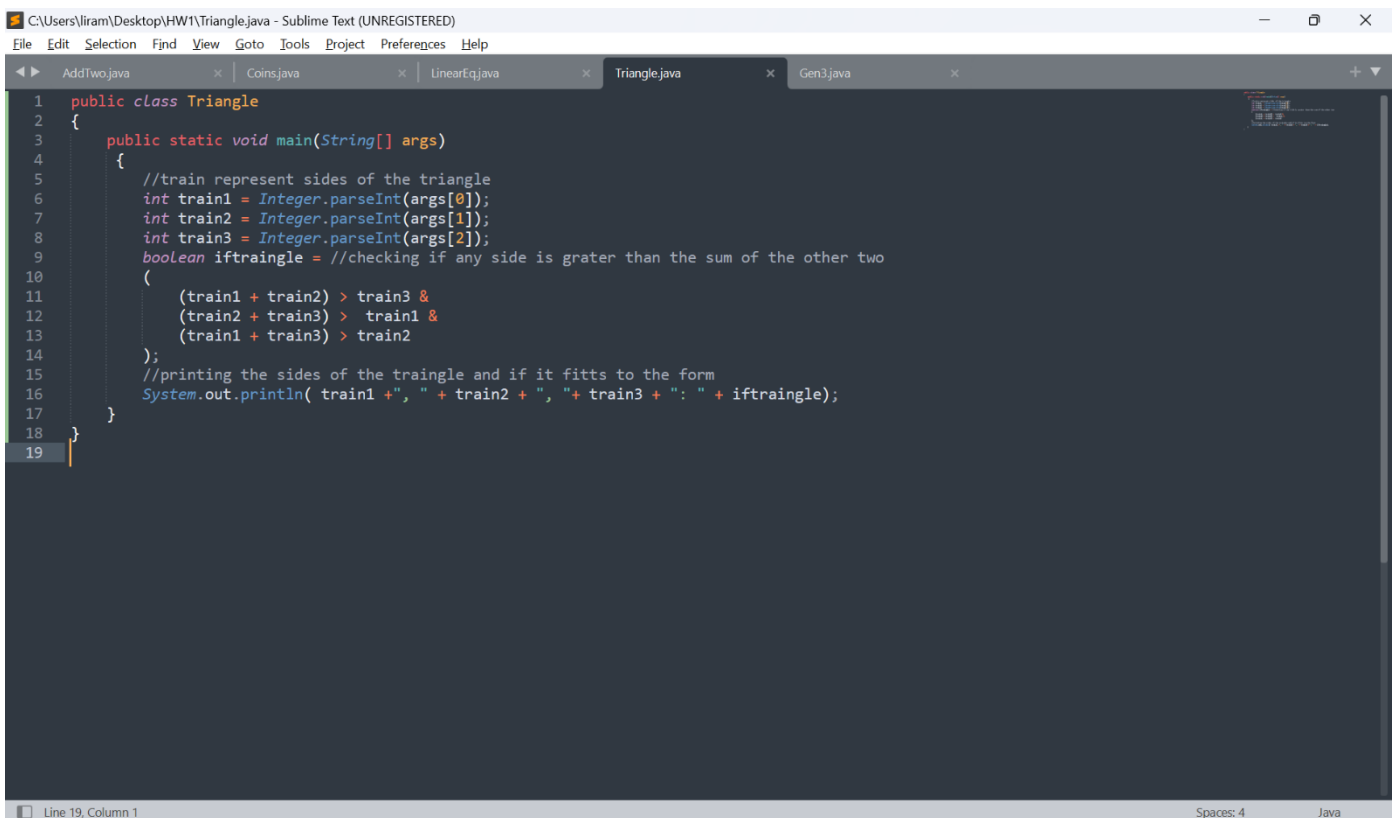


```
C:\Users\jiram\Desktop\HW1\LinearEq.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
AddTwo.java x Coins.java x LinearEq.java x Triangle.java x Gen3.java x
1 public class LinearEq
2 {
3     public static void main(String[] args)
4     {
5         // a b c represent the value like the original equation
6
7         double a = Double.parseDouble(args[0]);
8         double b = Double.parseDouble(args[1]);
9         double c = Double.parseDouble(args[2]);
10        System.out.println(a + " * x "+ " + " + b + " = " + c); //printing the equation  a * x + b = c
11        System.out.println("x = "+ (c-b)/a); //printing the result
12    }
13 }
```

Line 13, Column 2 Spaces: 4 Java

Triangle

```
public class Triangle
{
    public static void main(String[] args)
    {
        //train represent sides of the triangle
        int train1 = Integer.parseInt(args[0]);
        int train2 = Integer.parseInt(args[1]);
        int train3 = Integer.parseInt(args[2]);
        boolean iftriangle = //checking if any side is grater than the sum of the
other two
        (
            (train1 + train2) > train3 &
            (train2 + train3) > train1 &
            (train1 + train3) > train2
        );
        //printing the sides of the traingle and if it fitts to the form
        System.out.println( train1 +", " + train2 + ", "+ train3 + ": " + iftriangle);
    }
}
```



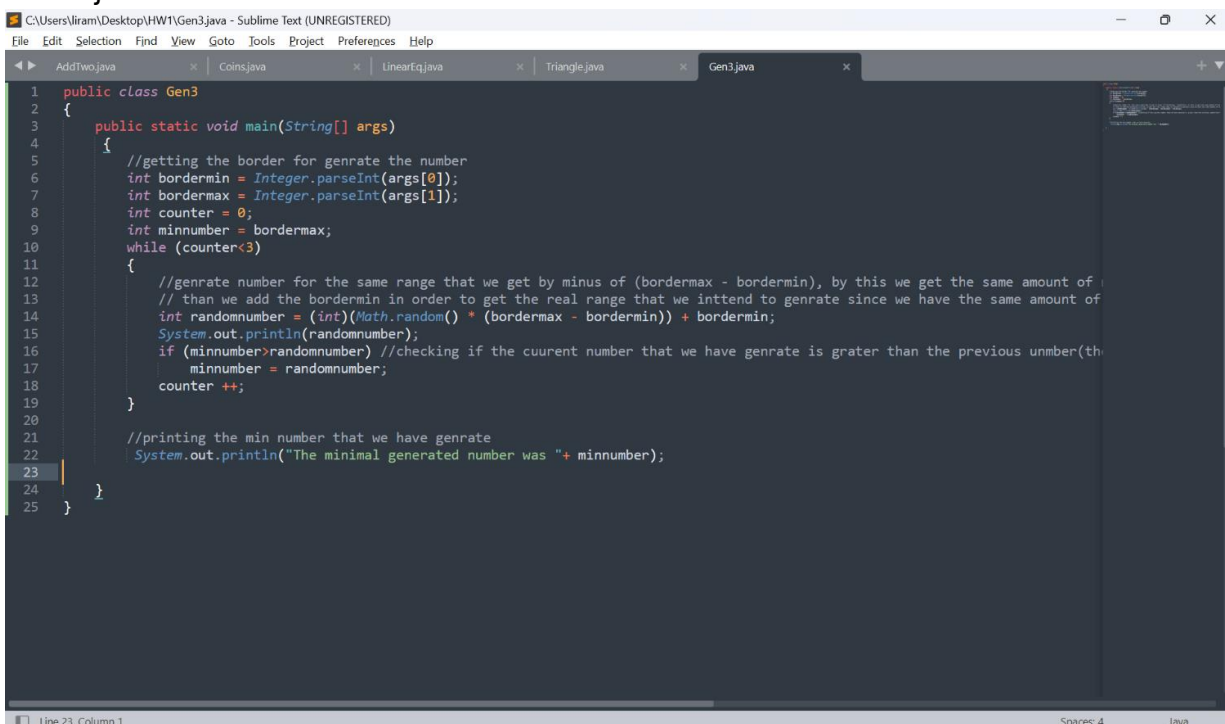
```
C:\Users\liram\Desktop\HW1\Triangle.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
AddTwo.java Coins.java LinearEq.java Triangle.java Gen3.java
1 public class Triangle
2 {
3     public static void main(String[] args)
4     {
5         //train represent sides of the triangle
6         int train1 = Integer.parseInt(args[0]);
7         int train2 = Integer.parseInt(args[1]);
8         int train3 = Integer.parseInt(args[2]);
9         boolean iftriangle = //checking if any side is grater than the sum of the other two
10        (
11            (train1 + train2) > train3 &
12            (train2 + train3) > train1 &
13            (train1 + train3) > train2
14        );
15        //printing the sides of the traingle and if it fitts to the form
16        System.out.println( train1 +", " + train2 + ", "+ train3 + ": " + iftriangle);
17    }
18 }
19
```

Line 19, Column 1 Spaces: 4 Java

GenThree

```
public class GenThree
{
    public static void main(String[] args)
    {
        //getting the border for generate the number
        int bordermin = Integer.parseInt(args[0]);
        int bordermax = Integer.parseInt(args[1]);
        int counter = 0;
        int minnumber = bordermax;
        while (counter<3)
        {
            //generate number for the same range that we get by minus of
            (bordermax - bordermin), by this we get the same amount of number that
            could possibly can been generate
            // then we add the bordermin in order to get the real range that we
            intend to generate since we have the same amount of option we can get
            int randomnumber = (int)(Math.random() * (bordermax - bordermin)) +
bordermin;
            System.out.println(randomnumber);
            if (minnumber>randomnumber) //checking if the cuurent number that
            we have generate is grater than the previous unnumber(the bordermax is the first
            min number but it allways change since this number is not in the range)
                minnumber = randomnumber;
            counter ++;
        }

        //printing the min number that we have generate
        System.out.println("The minimal generated number was "+ minnumber);
    }
}
```



```
1 public class Gen3
2 {
3     public static void main(String[] args)
4     {
5         //getting the border for generate the number
6         int bordermin = Integer.parseInt(args[0]);
7         int bordermax = Integer.parseInt(args[1]);
8         int counter = 0;
9         int minnumber = bordermax;
10        while (counter<3)
11        {
12            //generate number for the same range that we get by minus of (bordermax - bordermin), by this we get the same amount of
13            // then we add the bordermin in order to get the real range that we intend to generate since we have the same amount of
14            int randomnumber = (int)(Math.random() * (bordermax - bordermin)) + bordermin;
15            System.out.println(randomnumber);
16            if (minnumber>randomnumber) //checking if the cuurent number that we have generate is grater than the previous unnumber(th
17                minnumber = randomnumber;
18            counter ++;
19        }
20
21        //printing the min number that we have generate
22        System.out.println("The minimal generated number was "+ minnumber);
23    }
24 }
25 }
```

