

LAB 1 REPORT

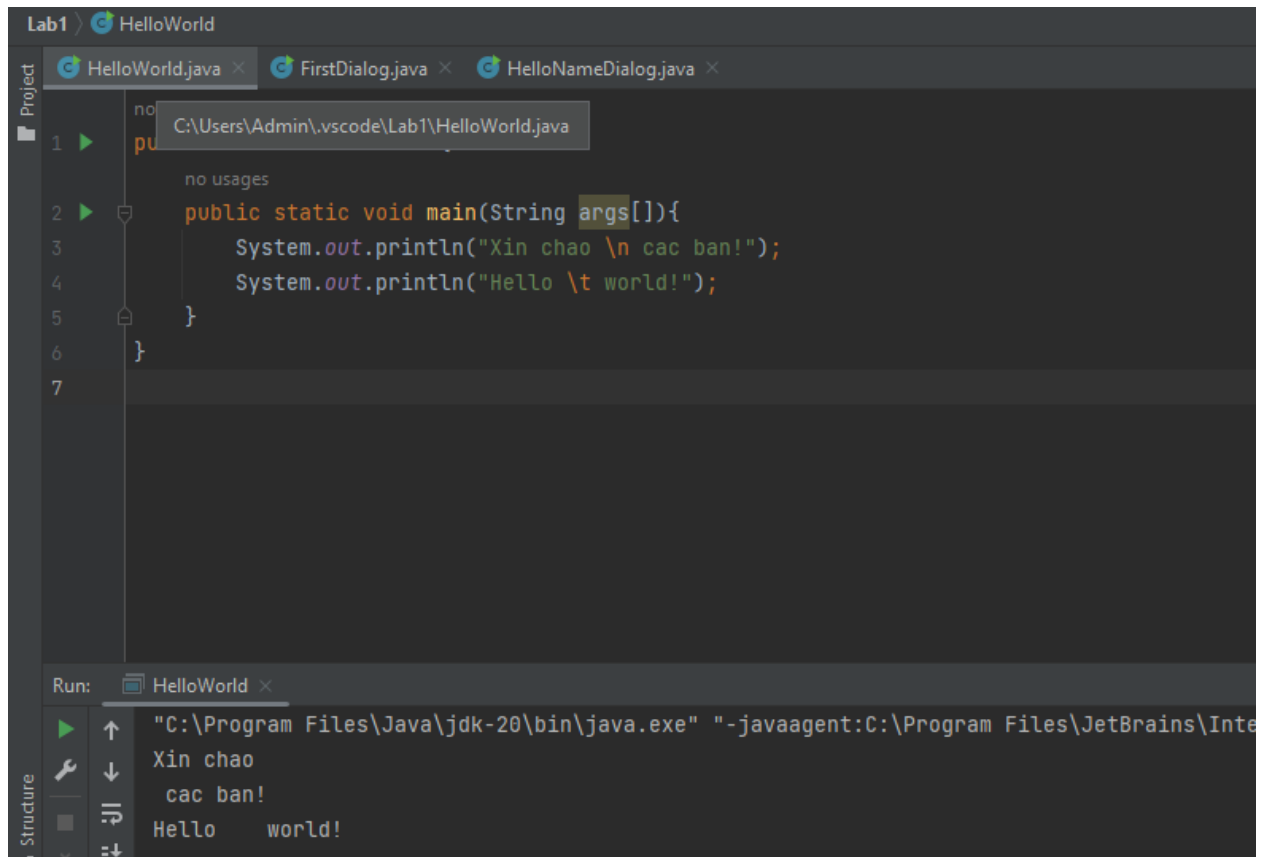
OBJECT ORIENTED PROGRAMMING

The Very First Java Programs

2.2.1 Write, compile the first Java application:

```
1 //Example 1: HelloWorld.java
2 //Text-printing program
3 public class HelloWorld {
4
5     public static void main(String args[]){
6         System.out.println("Xin chao \n cac ban!");
7         System.out.println("Hello \t world!");
8
9     } // end of method main
10 }
```

Result



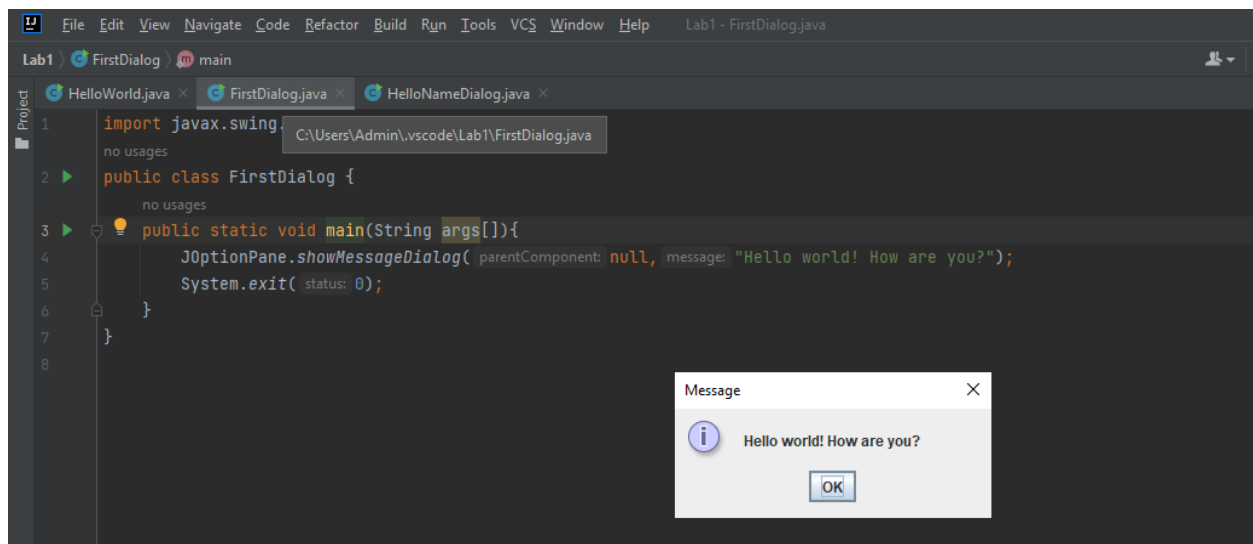
2.2.2 Write, compile the first dialog Java program

```

1 // Example 2: FirstDialog.java
2 import javax.swing.JOptionPane;
3 public class FirstDialog{
4     public static void main(String[] args){
5         JOptionPane.showMessageDialog(null,"Hello world! How are you?");
6         System.exit(0);
7     }
8 }

```

Result



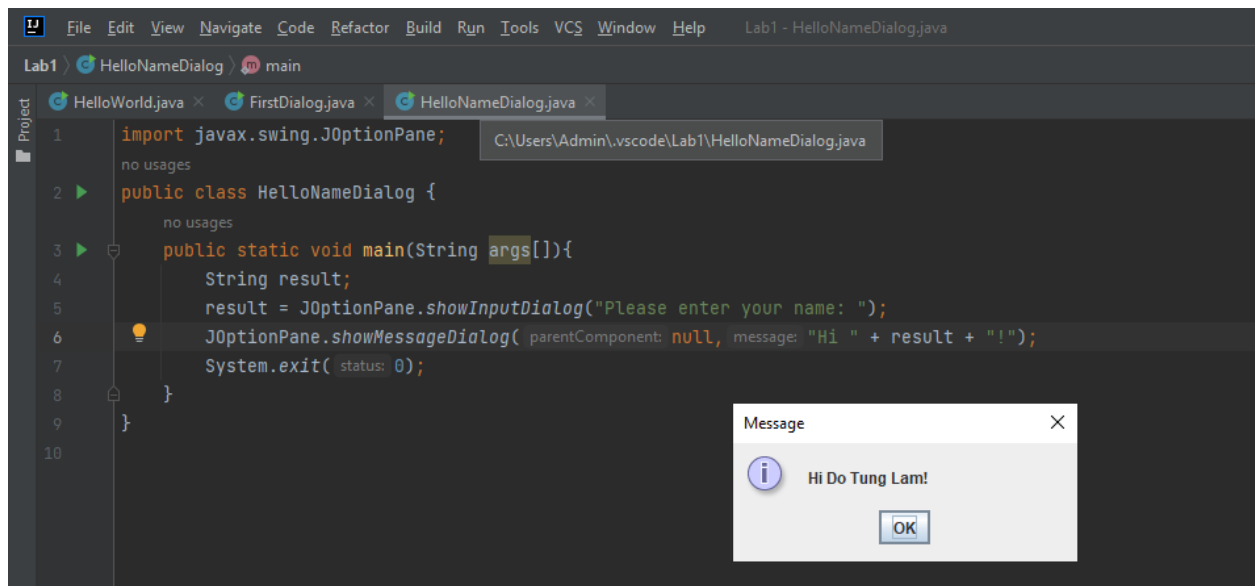
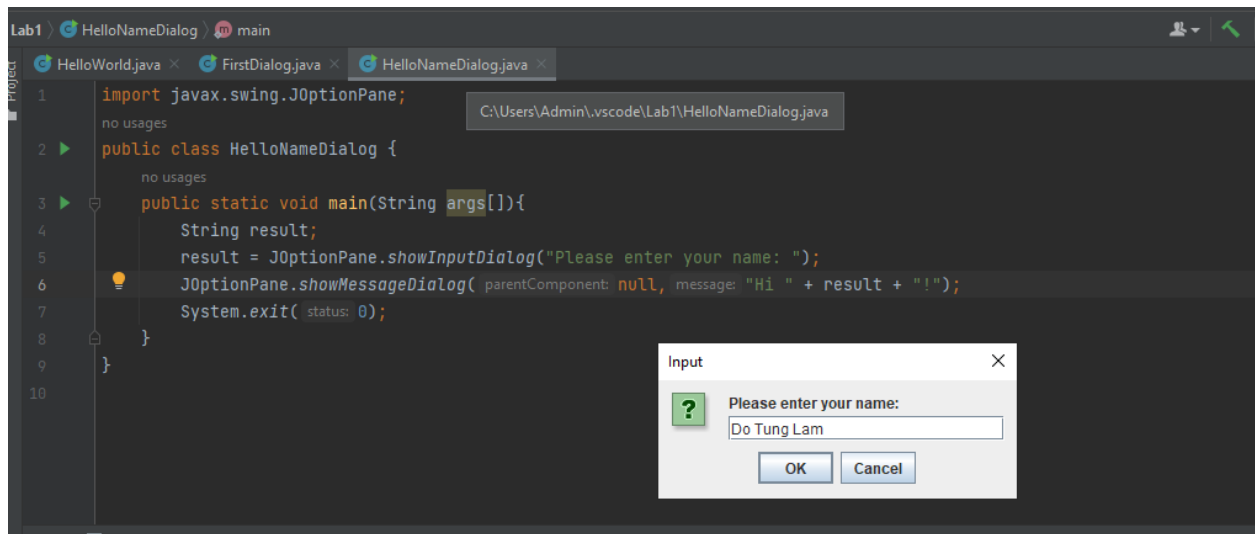
2.2.3 Write, compile the first input dialog Java application

```

1 // Example 3: HelloNameDialog.java
2 import javax.swing.JOptionPane;
3 public class HelloNameDialog{
4     public static void main(String[] args){
5         String result;
6         result = JOptionPane.showInputDialog("Please enter your name:");
7         JOptionPane.showMessageDialog(null, "Hi " + result + "!");
8         System.exit(0);
9     }
10 }

```

Result



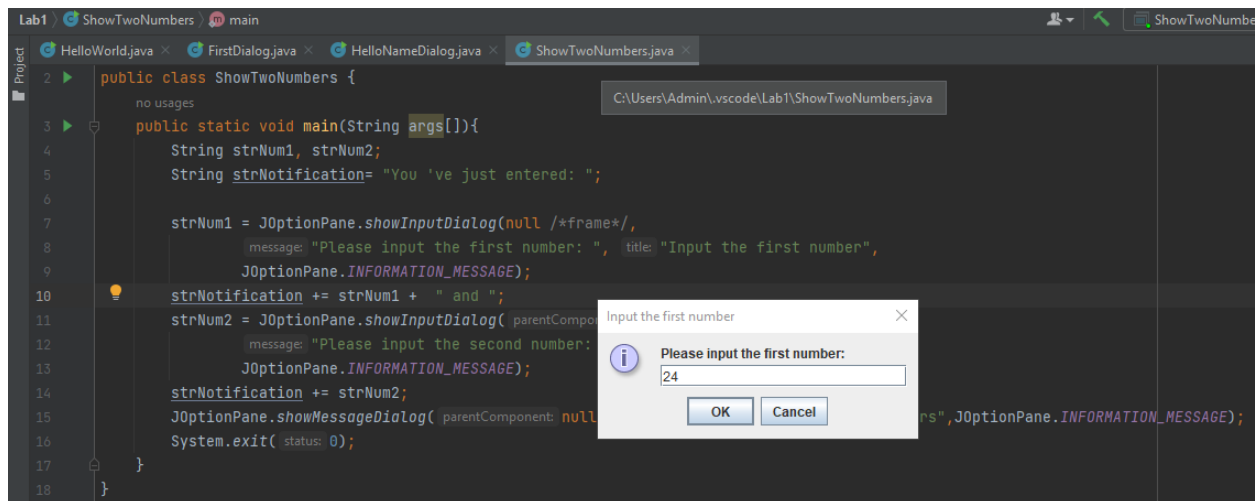
2.2.4 Write, compile, and run the following example:

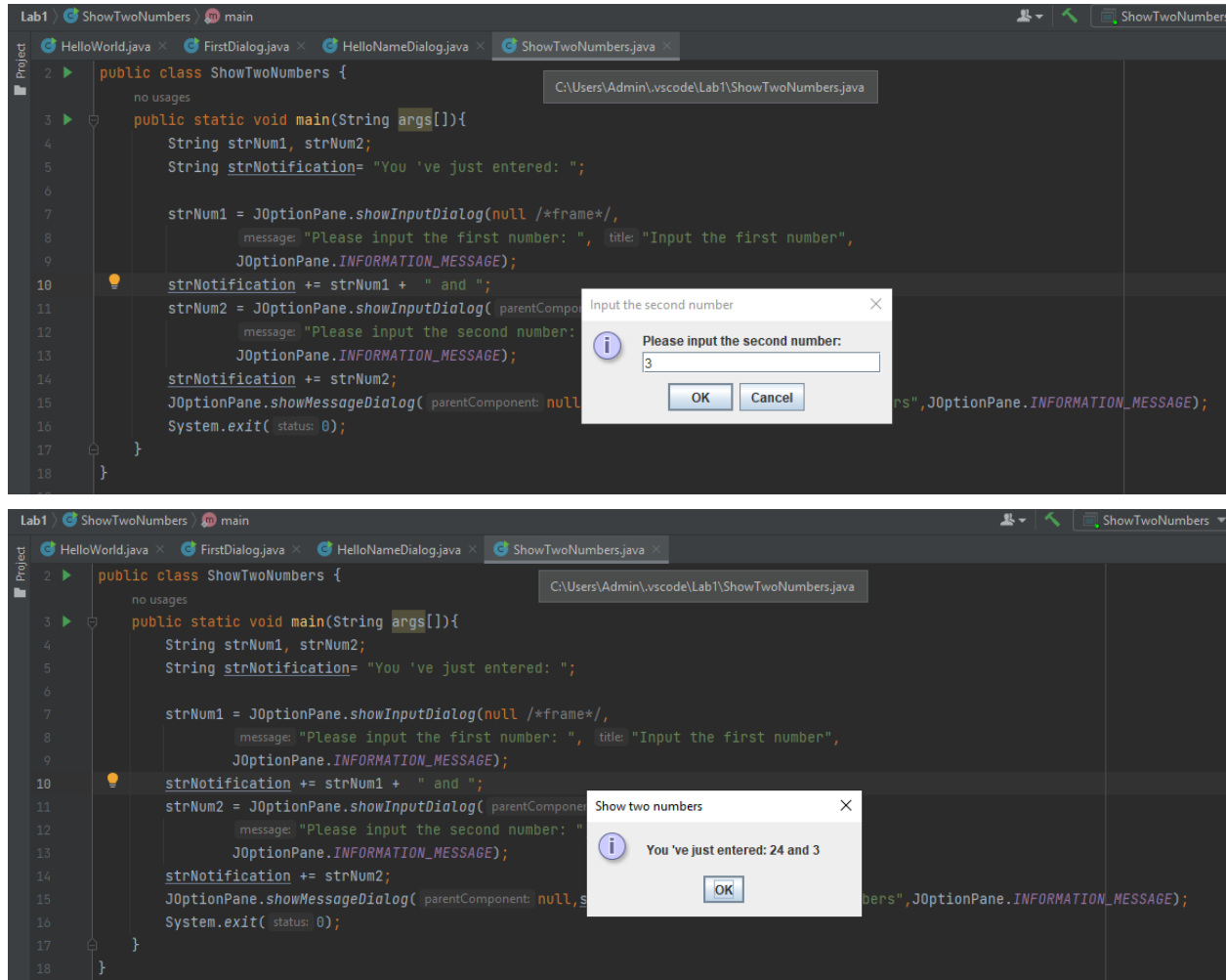
```

1 // Example 5: ShowTwoNumbers.java
2 import javax.swing.JOptionPane;
3 public class ShowTwoNumbers {
4     public static void main(String[] args){
5         String strNum1, strNum2;
6         String strNotification = "You've just entered: ";
7
8         strNum1 = JOptionPane.showInputDialog(null,
9             "Please input the first number: ", "Input the first number",
10             JOptionPane.INFORMATION_MESSAGE);
11         strNotification += strNum1 + " and ";
12
13         strNum2 = JOptionPane.showInputDialog(null,
14             "Please input the second number: ", "Input the second number",
15             JOptionPane.INFORMATION_MESSAGE);
16         strNotification += strNum2;
17
18         JOptionPane.showMessageDialog(null, strNotification,
19             "Show two numbers", JOptionPane.INFORMATION_MESSAGE);
20         System.exit(0);
21     }
22 }

```

Result





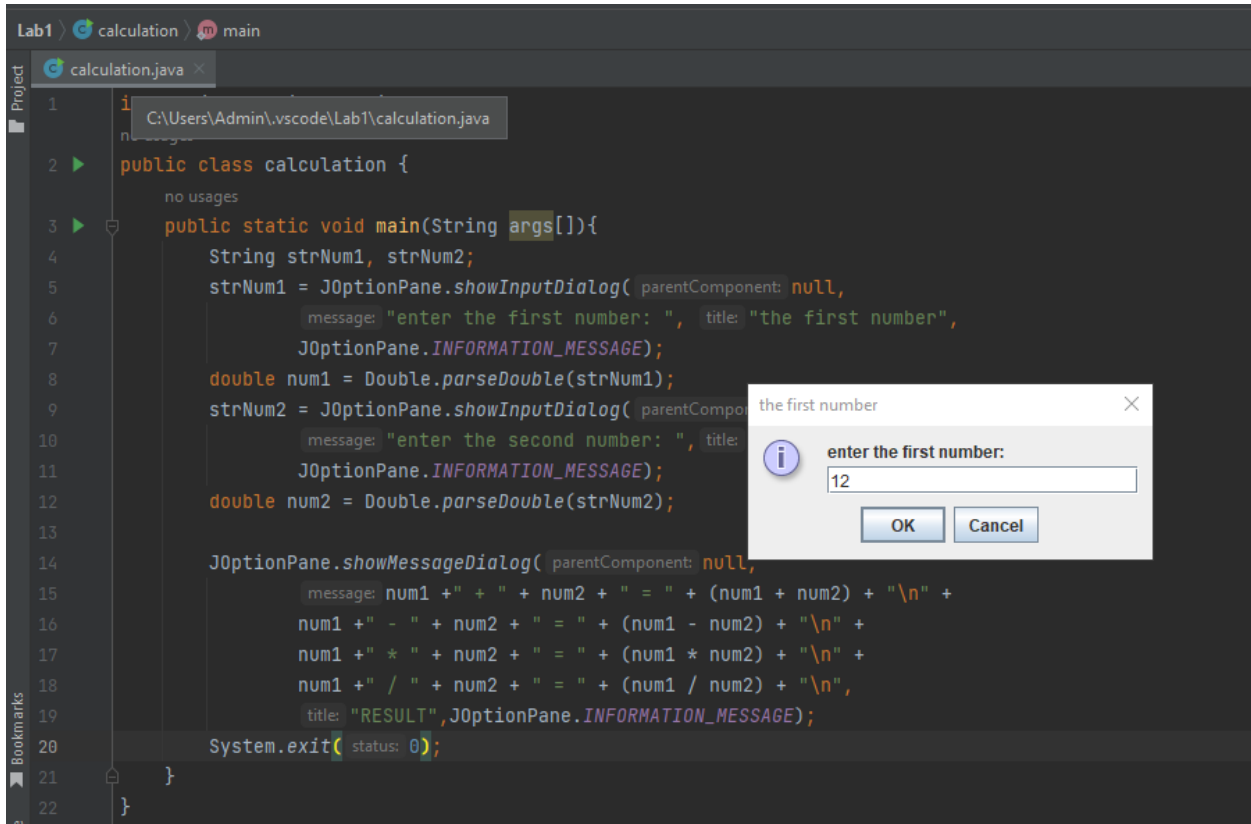
2.2.5 Write a program to calculate sum, difference, product, and quotient of 2 double numbers which are entered by users.

Notes

- To convert from String to double, you can use

```
double num1 = Double.parseDouble(strNum1)
```
- Check the divisor of the division

Result



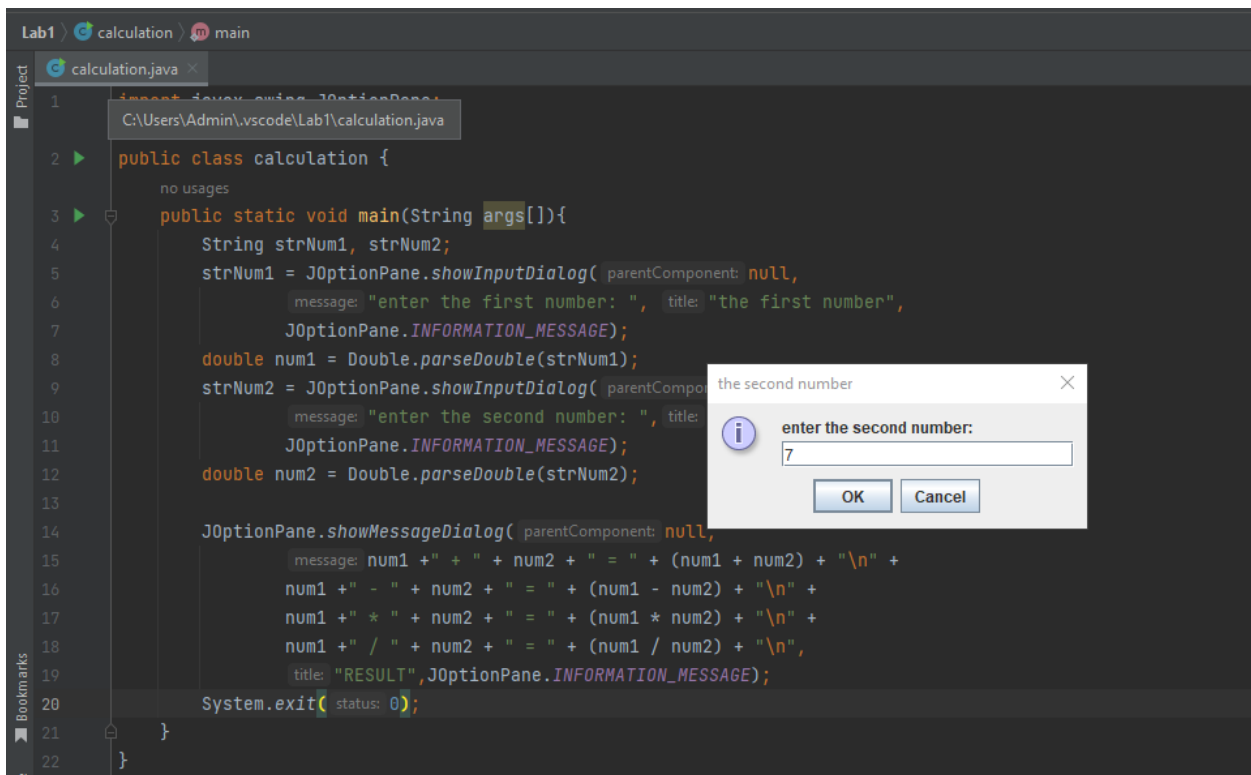
```
1  i C:\Users\Admin\.vscode\Lab1\calculation.java
2  public class calculation {
3      no usages
4      public static void main(String args[]){
5          String strNum1, strNum2;
6          strNum1 = JOptionPane.showInputDialog( parentComponent: null,
7              message: "enter the first number: ", title: "the first number",
8              JOptionPane.INFORMATION_MESSAGE);
9          double num1 = Double.parseDouble(strNum1);
10         strNum2 = JOptionPane.showInputDialog( parentComponent: null,
11             message: "enter the second number: ", title:
12             JOptionPane.INFORMATION_MESSAGE);
13         double num2 = Double.parseDouble(strNum2);
14
15         JOptionPane.showMessageDialog( parentComponent: null,
16             message: num1 + " + " + num2 + " = " + (num1 + num2) + "\n" +
17             num1 + " - " + num2 + " = " + (num1 - num2) + "\n" +
18             num1 + " * " + num2 + " = " + (num1 * num2) + "\n" +
19             num1 + " / " + num2 + " = " + (num1 / num2) + "\n",
20             title: "RESULT",JOptionPane.INFORMATION_MESSAGE);
21         System.exit( status: 0);
22     }
```

the first number

enter the first number:

12

OK Cancel



```
1  i C:\Users\Admin\.vscode\Lab1\calculation.java
2  public class calculation {
3      no usages
4      public static void main(String args[]){
5          String strNum1, strNum2;
6          strNum1 = JOptionPane.showInputDialog( parentComponent: null,
7              message: "enter the first number: ", title: "the first number",
8              JOptionPane.INFORMATION_MESSAGE);
9          double num1 = Double.parseDouble(strNum1);
10         strNum2 = JOptionPane.showInputDialog( parentComponent: null,
11             message: "enter the second number: ", title:
12             JOptionPane.INFORMATION_MESSAGE);
13         double num2 = Double.parseDouble(strNum2);
14
15         JOptionPane.showMessageDialog( parentComponent: null,
16             message: num1 + " + " + num2 + " = " + (num1 + num2) + "\n" +
17             num1 + " - " + num2 + " = " + (num1 - num2) + "\n" +
18             num1 + " * " + num2 + " = " + (num1 * num2) + "\n" +
19             num1 + " / " + num2 + " = " + (num1 / num2) + "\n",
20             title: "RESULT",JOptionPane.INFORMATION_MESSAGE);
21         System.exit( status: 0);
22     }
```

the second number

enter the second number:

7

OK Cancel

The screenshot shows a Java IDE with a file named `calculation.java`. The code defines a `calculation` class with a `main` method. The `main` method prompts the user to enter two numbers, parses them as `double` values, and then displays the results of addition, subtraction, multiplication, and division. A message dialog box titled "RESULT" is shown, displaying the following calculations:

```
12.0 + 7.0 = 19.0
12.0 - 7.0 = 5.0
12.0 * 7.0 = 84.0
12.0 / 7.0 = 1.7142857142857142
```

The code in the IDE is as follows:

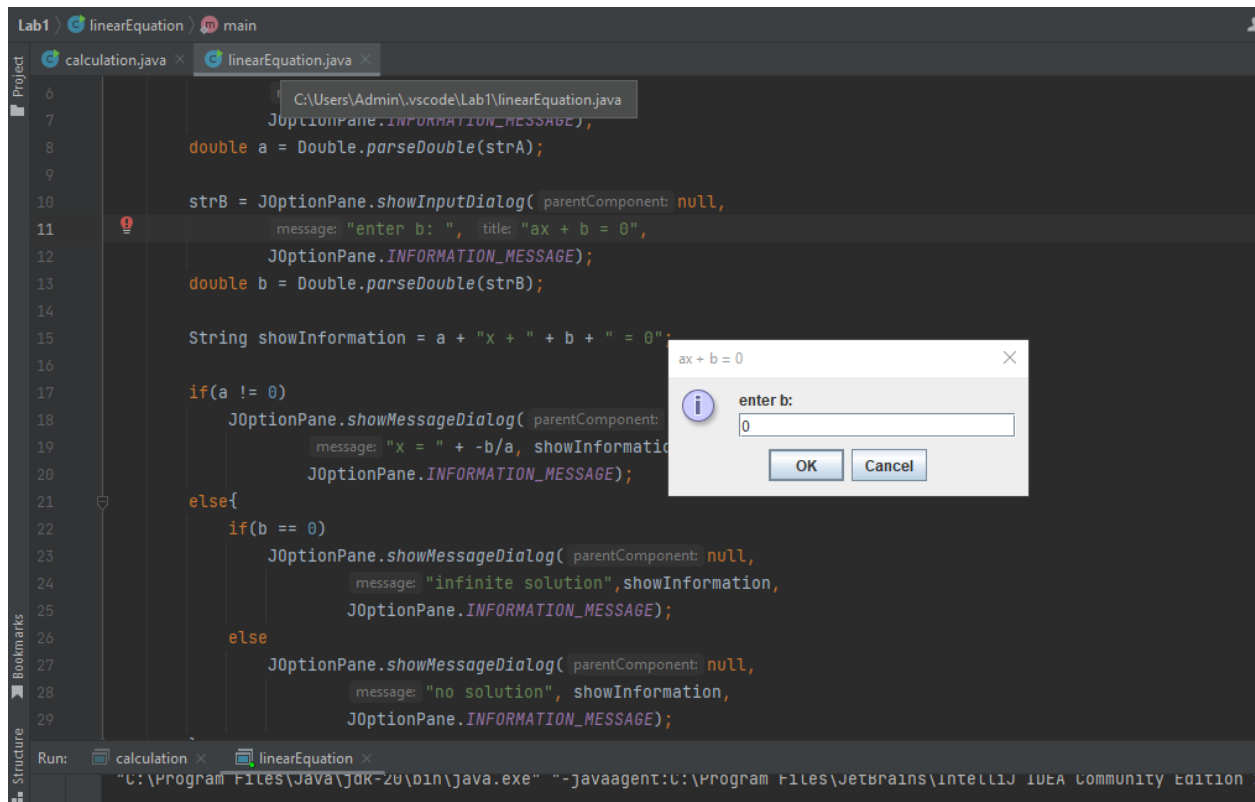
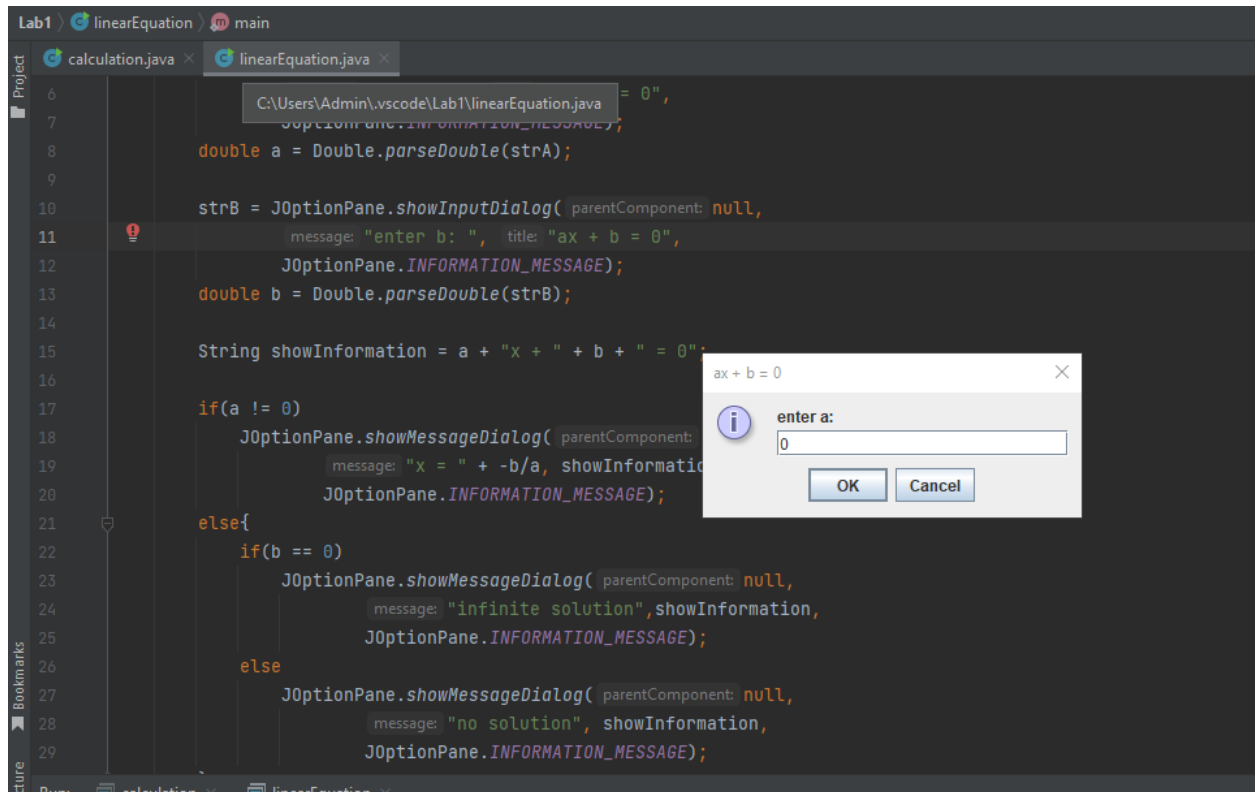
```
1  import javax.swing.JOptionPane;
2  public class calculation {
3      public static void main(String args[]){
4          String strNum1, strNum2;
5          strNum1 = JOptionPane.showInputDialog( parentComponent: null,
6              message: "enter the first number: ", title: "the first number",
7              JOptionPane.INFORMATION_MESSAGE);
8          double num1 = Double.parseDouble(strNum1);
9          strNum2 = JOptionPane.showInputDialog( parentComponent: null,
10             message: "enter the second number: ", title: "the second number",
11             JOptionPane.INFORMATION_MESSAGE);
12          double num2 = Double.parseDouble(strNum2);
13
14          JOptionPane.showMessageDialog( parentComponent: null,
15             message: num1 + " + " + num2 + " = " + (num1 + num2) + "\n" +
16             num1 + " - " + num2 + " = " + (num1 - num2) + "\n" +
17             num1 + " * " + num2 + " = " + (num1 * num2) + "\n" +
18             num1 + " / " + num2 + " = " + (num1 / num2) + "\n",
19             title: "RESULT", JOptionPane.INFORMATION_MESSAGE);
20          System.exit( status: 0);
21      }
22  }
```

2.2.6 Write a program to solve:

The first-degree equation (linear equation) with one variable

Result

$a = 0, b = 0$




```
Lab1 linearEquation linearEquation.java
6 C:\Users\Admin\vscode\Lab1\linearEquation.java",
7
8 double a = Double.parseDouble(strA);
9
10 strB = JOptionPane.showInputDialog( parentComponent: null,
11 message: "enter b: ", title: "ax + b = 0",
12 JOptionPane.INFORMATION_MESSAGE);
13 double b = Double.parseDouble(strB);
14
15 String showInformation = a + "x + " + b + " = 0";
16
17 if(a != 0)
18     JOptionPane.showMessageDialog( parentComponent: null,
19 message: "x = " + -b/a, showInformation,
20 JOptionPane.INFORMATION_MESSAGE);
21 else{
22     if(b == 0)
23         JOptionPane.showMessageDialog( parentComponent: null,
24 message: "infinite solution", showInformation,
25 JOptionPane.INFORMATION_MESSAGE);
26     else
27         JOptionPane.showMessageDialog( parentComponent: null,
28 message: "no solution", showInformation,
29 JOptionPane.INFORMATION_MESSAGE);
30 }
```

0.0x + 0.0 = 0

infinite solution

OK

Run: calculation linearEquation

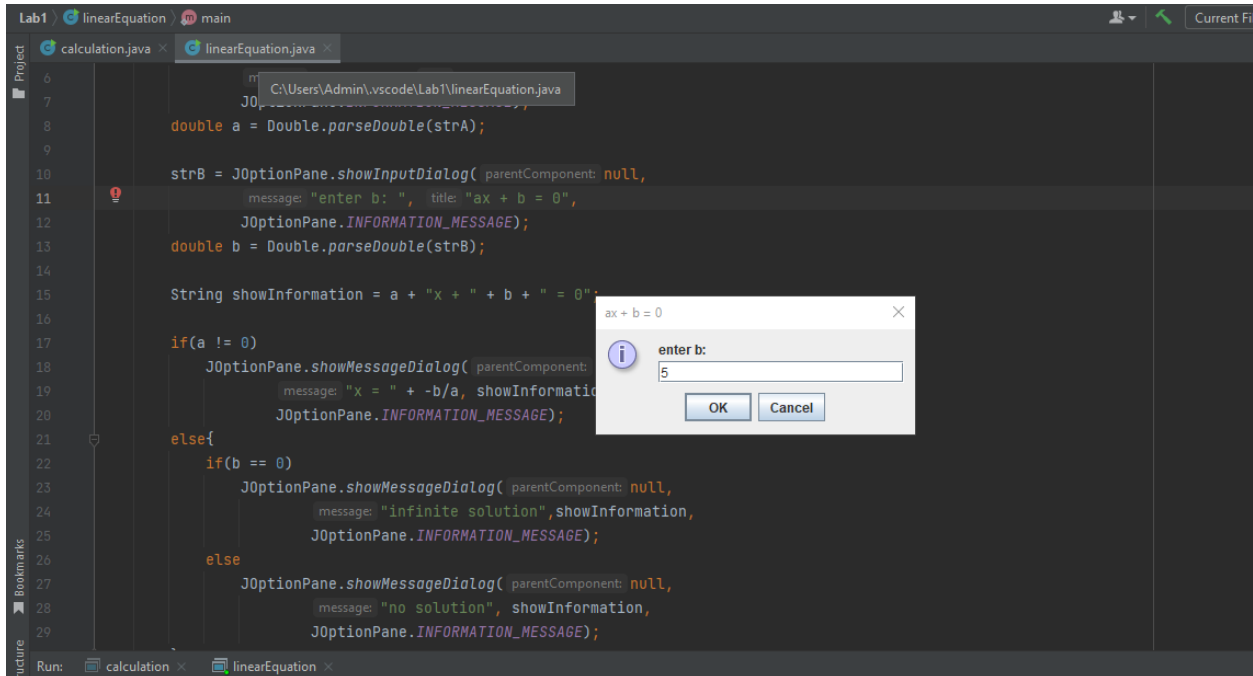
C:\Program Files\Java\jdk-20\bin\java.exe --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition

$a = 0, b \neq 0$

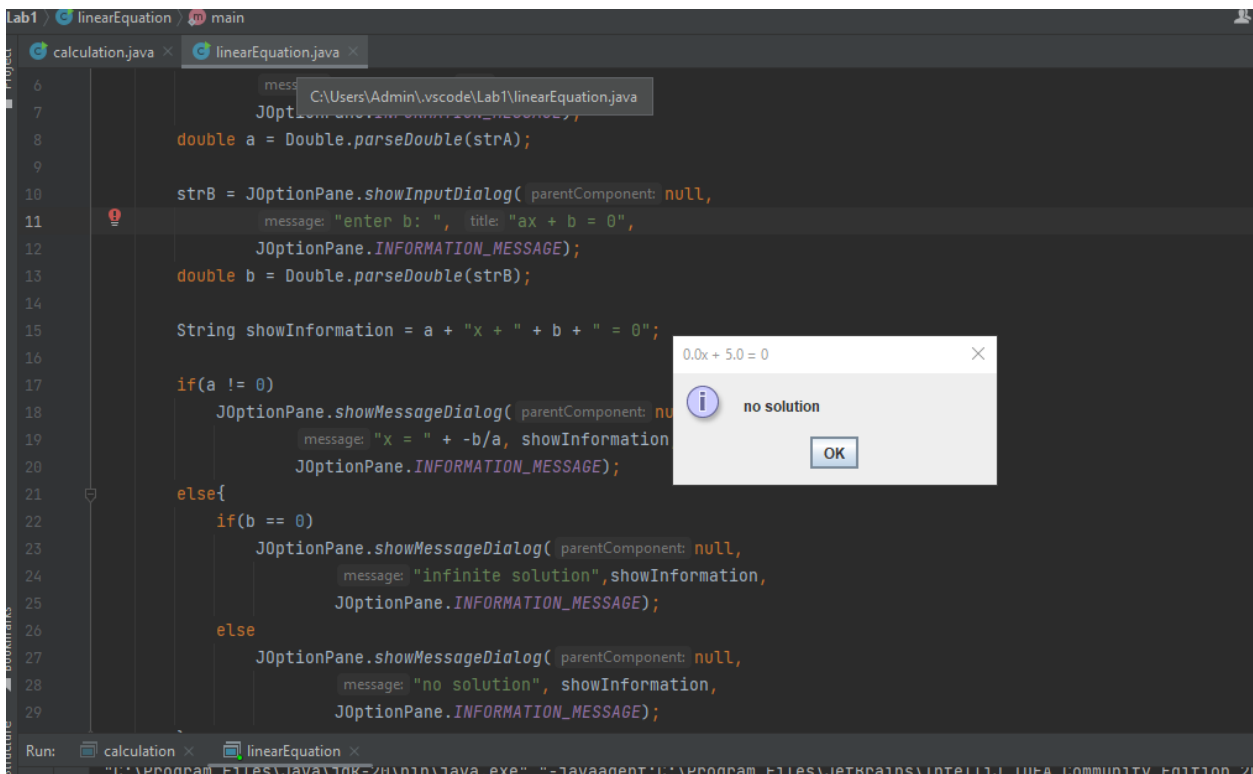
The screenshot shows a Java IDE with a file named `linearEquation.java`. The code is as follows:

```
6  double a = Double.parseDouble(strA);
7
8  double b = Double.parseDouble(strB);
9
10 strB = JOptionPane.showInputDialog( parentComponent: null,
11     message: "enter b: ", title: "ax + b = 0",
12     JOptionPane.INFORMATION_MESSAGE);
13 double b = Double.parseDouble(strB);
14
15 String showInformation = a + "x + " + b + " = 0";
16
17 if(a != 0)
18     JOptionPane.showMessageDialog( parentComponent: null,
19         message: "x = " + -b/a, showInformation,
20         JOptionPane.INFORMATION_MESSAGE);
21 else{
22     if(b == 0)
23         JOptionPane.showMessageDialog( parentComponent: null,
24             message: "infinite solution", showInformation,
25             JOptionPane.INFORMATION_MESSAGE);
26     else
27         JOptionPane.showMessageDialog( parentComponent: null,
28             message: "no solution", showInformation,
29             JOptionPane.INFORMATION_MESSAGE);
```

A dialog box titled `ax + b = 0` is displayed, showing the message `enter a:` and an input field containing `0`. The dialog box has `OK` and `Cancel` buttons.

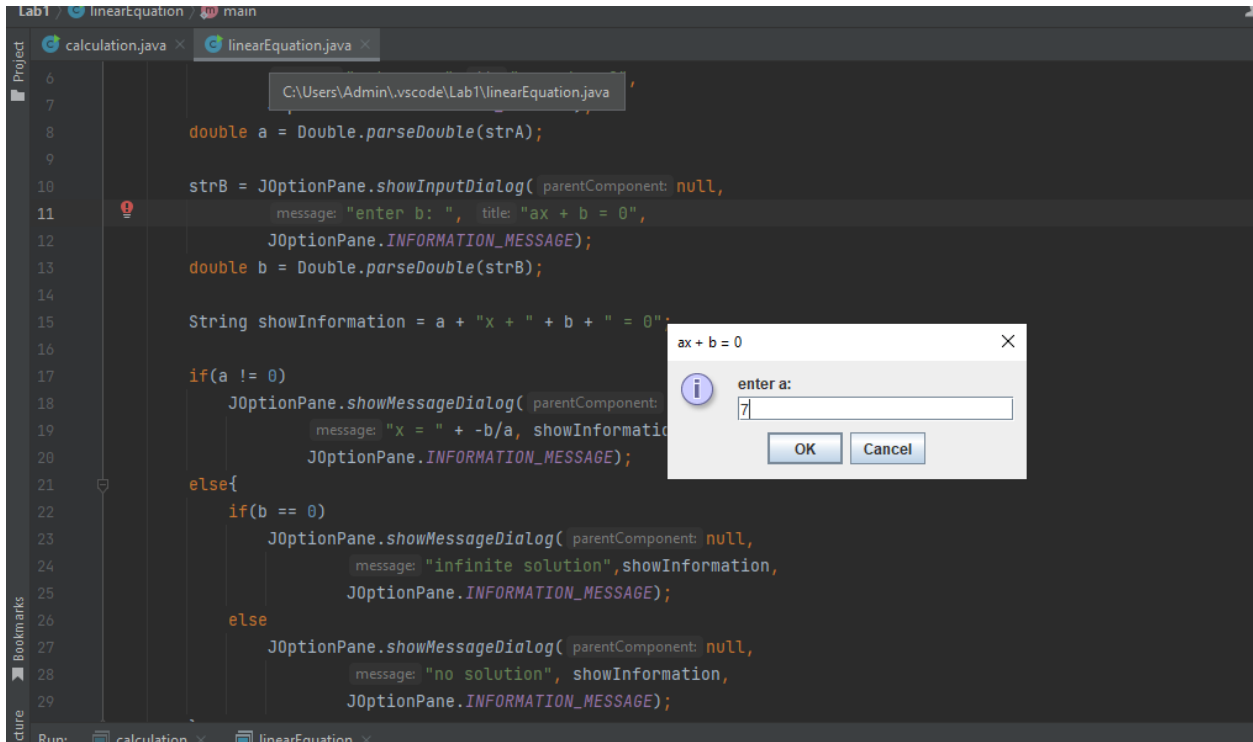


```
6      double a = Double.parseDouble(strA);
7
8      double a = Double.parseDouble(strA);
9
10     strB = JOptionPane.showInputDialog( parentComponent: null,
11     message: "enter b: ", title: "ax + b = 0",
12     JOptionPane.INFORMATION_MESSAGE);
13     double b = Double.parseDouble(strB);
14
15     String showInformation = a + "x + " + b + " = 0";
16
17     if(a != 0)
18         JOptionPane.showMessageDialog( parentComponent: null,
19         message: "x = " + -b/a, showInformation,
20         JOptionPane.INFORMATION_MESSAGE);
21     else{
22         if(b == 0)
23             JOptionPane.showMessageDialog( parentComponent: null,
24             message: "infinite solution", showInformation,
25             JOptionPane.INFORMATION_MESSAGE);
26         else
27             JOptionPane.showMessageDialog( parentComponent: null,
28             message: "no solution", showInformation,
29             JOptionPane.INFORMATION_MESSAGE);
```



```
6      double a = Double.parseDouble(strA);
7
8      double a = Double.parseDouble(strA);
9
10     strB = JOptionPane.showInputDialog( parentComponent: null,
11     message: "enter b: ", title: "ax + b = 0",
12     JOptionPane.INFORMATION_MESSAGE);
13     double b = Double.parseDouble(strB);
14
15     String showInformation = a + "x + " + b + " = 0";
16
17     if(a != 0)
18         JOptionPane.showMessageDialog( parentComponent: null,
19         message: "x = " + -b/a, showInformation,
20         JOptionPane.INFORMATION_MESSAGE);
21     else{
22         if(b == 0)
23             JOptionPane.showMessageDialog( parentComponent: null,
24             message: "infinite solution", showInformation,
25             JOptionPane.INFORMATION_MESSAGE);
26         else
27             JOptionPane.showMessageDialog( parentComponent: null,
28             message: "no solution", showInformation,
29             JOptionPane.INFORMATION_MESSAGE);
```

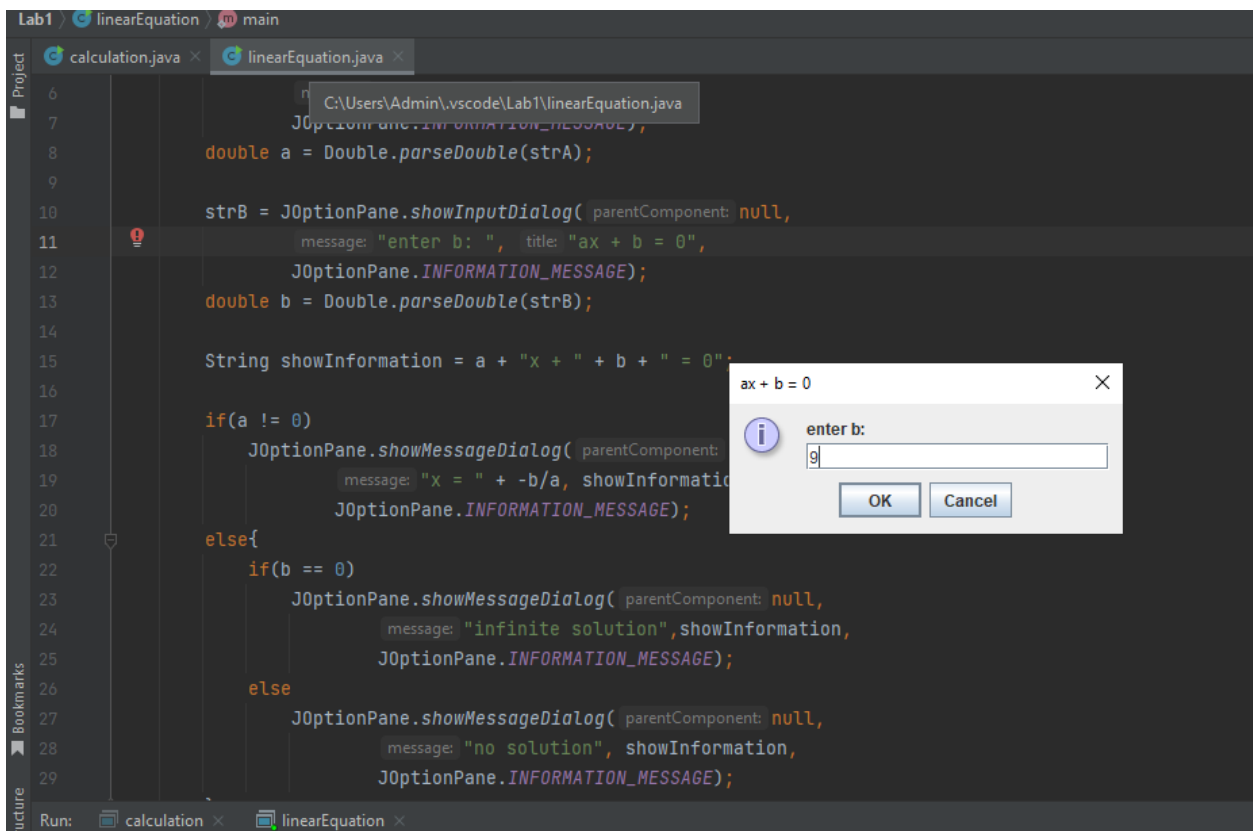
$a \neq 0, b \neq 0$



The screenshot shows the VS Code editor with the file `linearEquation.java` open. The code is as follows:

```
6  
7  
8 double a = Double.parseDouble(strA);  
9  
10  
11 strB = JOptionPane.showInputDialog( parentComponent: null,  
12     message: "enter b: ", title: "ax + b = 0",  
13     JOptionPane.INFORMATION_MESSAGE);  
14  
15 double b = Double.parseDouble(strB);  
16  
17 String showInformation = a + "x + " + b + " = 0";  
18  
19 if(a != 0)  
20     JOptionPane.showMessageDialog( parentComponent:  
21         message: "x = " + -b/a, showInformation,  
22         JOptionPane.INFORMATION_MESSAGE);  
23  
24 else{  
25     if(b == 0)  
26         JOptionPane.showMessageDialog( parentComponent: null,  
27             message: "infinite solution",showInformation,  
28             JOptionPane.INFORMATION_MESSAGE);  
29     else  
30         JOptionPane.showMessageDialog( parentComponent: null,  
31             message: "no solution", showInformation,  
32             JOptionPane.INFORMATION_MESSAGE);
```

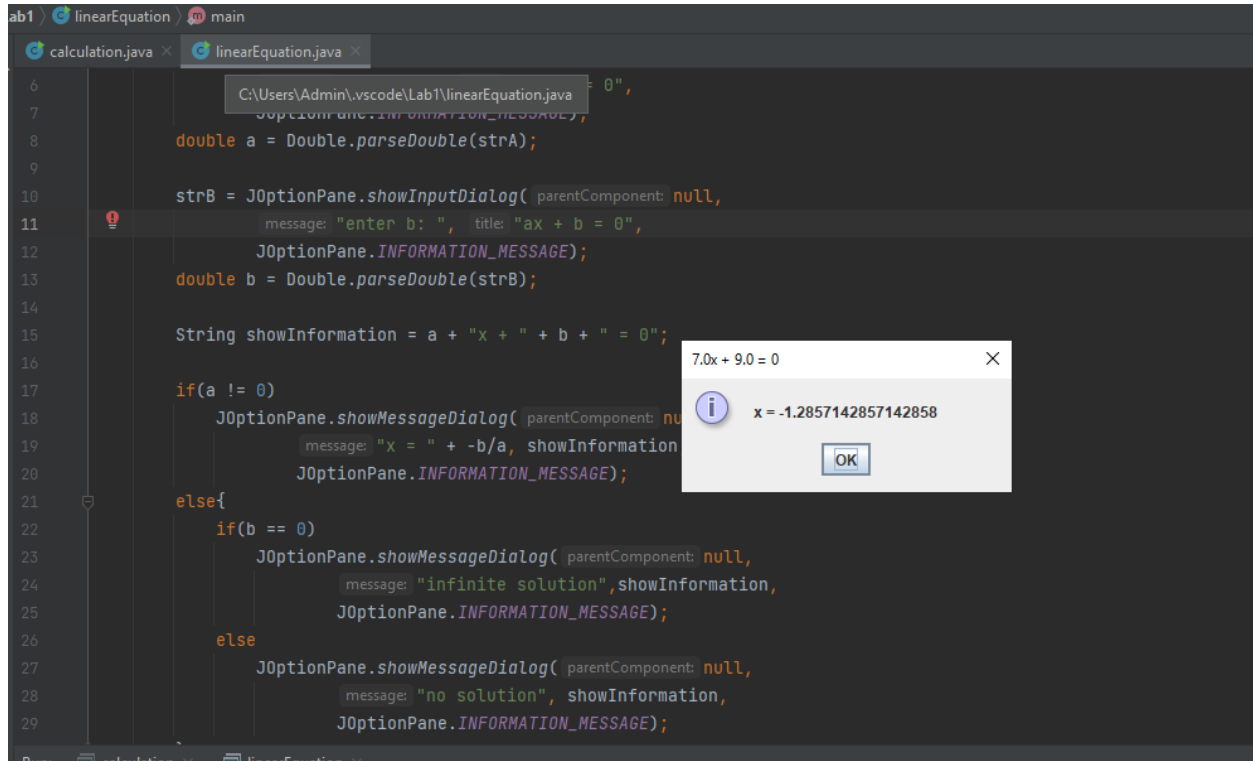
A dialog box titled "ax + b = 0" is displayed, asking to "enter a:". The input field contains the value "7".



The screenshot shows the VS Code editor with the file `linearEquation.java` open. The code is as follows:

```
6  
7  
8 double a = Double.parseDouble(strA);  
9  
10  
11 strB = JOptionPane.showInputDialog( parentComponent: null,  
12     message: "enter b: ", title: "ax + b = 0",  
13     JOptionPane.INFORMATION_MESSAGE);  
14  
15 double b = Double.parseDouble(strB);  
16  
17 String showInformation = a + "x + " + b + " = 0";  
18  
19 if(a != 0)  
20     JOptionPane.showMessageDialog( parentComponent:  
21         message: "x = " + -b/a, showInformation,  
22         JOptionPane.INFORMATION_MESSAGE);  
23  
24 else{  
25     if(b == 0)  
26         JOptionPane.showMessageDialog( parentComponent: null,  
27             message: "infinite solution",showInformation,  
28             JOptionPane.INFORMATION_MESSAGE);  
29     else  
30         JOptionPane.showMessageDialog( parentComponent: null,  
31             message: "no solution", showInformation,  
32             JOptionPane.INFORMATION_MESSAGE);
```

A dialog box titled "ax + b = 0" is displayed, asking to "enter b:". The input field contains the value "9".



The screenshot shows a Java IDE with two tabs: 'calculation.java' and 'linearEquation.java'. The 'linearEquation.java' tab is active, displaying the following code:

```
6      double a = Double.parseDouble(strA);
7
8      double b = Double.parseDouble(strB);
9
10     String showInformation = a + "x + " + b + " = 0";
11
12     if(a != 0)
13     {
14         JOptionPane.showMessageDialog( parentComponent: null,
15             message: "x = " + -b/a, showInformation,
16             JOptionPane.INFORMATION_MESSAGE);
17     }
18     else{
19         if(b == 0)
20         {
21             JOptionPane.showMessageDialog( parentComponent: null,
22                 message: "infinite solution", showInformation,
23                 JOptionPane.INFORMATION_MESSAGE);
24         }
25         else
26         {
27             JOptionPane.showMessageDialog( parentComponent: null,
28                 message: "no solution", showInformation,
29                 JOptionPane.INFORMATION_MESSAGE);
30         }
31     }
32 }
```

Overlaid on the code is a Java Swing window titled "7.0x + 9.0 = 0". It contains an information icon and the text "x = -1.2857142857142858". An "OK" button is at the bottom right.

The system of first-degree equations (linear system) with two variables

Result

```
Lab1 > linearSystem > main
calculation.java x linearEquation.java x linearSystem.java x
4 Scanner input = new Scanner(System.in);
5 double a11, a12, a21, a22,
6 System.out.println("a11*x1 + a12*x2 = b1\na21*x1 + a22*x2 = b2\n");
7 System.out.println("enter a1, a2, b1, b2: ");
8 a11 = input.nextDouble();
9 a12 = input.nextDouble();
10 b1 = input.nextDouble();
11 a21 = input.nextDouble();
12 a22 = input.nextDouble();
13 b2 = input.nextDouble();
14 double D = a11 * a22 - a12 * a21;
15 double Dx = b1 * a22 - b2 * a12;
16 double Dy = a11 * b2 - a21 * b1;
17 if(D == 0)
18     if(Dx != 0 || Dy != 0)

Run: linearSystem x linearSystem x
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\Jet
a11*x1 + a12*x2 = b1
a21*x1 + a22*x2 = b2

enter a1, a2, b1, b2:
2 1 11
5 -4 8
unique solution: x1 = 4.0 x2 = 3.0
```

```

4 Scanner input = new Scanner(System.in);
5 double a11, a12, a21, a22, b1, b2;
6 System.out.println("a11*x1 + a12*x2 = b1\na21*x1 + a22*x2 = b2\n");
7 System.out.println("enter a1, a2, b1, b2: ");
8 a11 = input.nextDouble();
9 a12 = input.nextDouble();
10 b1 = input.nextDouble();
11 a21 = input.nextDouble();
12 a22 = input.nextDouble();
13 b2 = input.nextDouble();
14 double D = a11 * a22 - a12 * a21;
15 double Dx = b1 * a22 - b2 * a12;
16 double Dy = a11 * b2 - a21 * b1;
17 if(D == 0){
18     if(Dx != 0 || Dy != 0)

```

Run: linearSystem

```

a11*x1 + a12*x2 = b1
a21*x1 + a22*x2 = b2
enter a1, a2, b1, b2:
6 6 12
3 3 6
Infinite solution

```

```

4 Scanner input = new Scanner(System.in);
5 double a11, a12, a21, a22, b1, b2;
6 System.out.println("a11*x1 + a12*x2 = b1\na21*x1 + a22*x2 = b2\n");
7 System.out.println("enter a1, a2, b1, b2: ");
8 a11 = input.nextDouble();
9 a12 = input.nextDouble();
10 b1 = input.nextDouble();
11 a21 = input.nextDouble();
12 a22 = input.nextDouble();
13 b2 = input.nextDouble();
14 double D = a11 * a22 - a12 * a21;
15 double Dx = b1 * a22 - b2 * a12;
16 double Dy = a11 * b2 - a21 * b1;
17 if(D == 0){
18     if(Dx != 0 || Dy != 0)

```

Run: linearSystem

```

a11*x1 + a12*x2 = b1
a21*x1 + a22*x2 = b2
enter a1, a2, b1, b2:
6 6 12
3 3 8
No solution

```

The second-degree equation with one variable

Result

The image displays three screenshots of an IDE (IntelliJ IDEA) showing the execution of a Java program for solving quadratic equations. The code is in a file named `secondDegreeEquation.java`.

First Screenshot: Shows the source code of `secondDegreeEquation.java`. The code prompts the user to enter coefficients `a`, `b`, and `c` for the equation $ax^2 + bx + c = 0$. It then checks if `a` is zero. If not, it calculates the discriminant $\Delta = b^2 - 4ac$. If $\Delta > 0$, it prints two distinct real roots. If $\Delta == 0$, it prints a double root. If $\Delta < 0$, it prints "No solution".

Second Screenshot: Shows the program running. The input entered is `1 4 3`. The output is `No solution`, indicating that the discriminant is negative.

Third Screenshot: Shows the program running. The input entered is `1 4 3`. The output is `two roots: x1 = -3.732050807568877 x2 = -0.2679491924311228`, indicating that the discriminant is positive and two distinct real roots are found.

```

17 System.out.println("ax^2 + bx + c = 0\nenter a b c:");
18 a = input.nextDouble();
19 b = input.nextDouble();
20 c = input.nextDouble();
21 if (a == 0)
22     linearEquation(b, c);
23 else {
24     double delta = b*b - 4*a*c;
25     if(delta > 0 )
26         System.out.println("two roots: x1 = " + (-b- Math.sqrt(delta)) / (2*a) +
27             " x2 = " + (-b + Math.sqrt(delta)) / (2*a));
28     else if (delta == 0) {
29         System.out.println("double root x1 = x2 = " + -b / (2*a));
30     }
31     else

```

Run: secondDegreeEquation

```

"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2
ax^2 + bx + c = 0
enter a b c:
4 4 1
double root x1 = x2 = -0.5

```

6.1 Write, compile and run the ChoosingOption program:

```

1 import javax.swing.JOptionPane;
2 public class ChoosingOption{
3     public static void main(String[] args){
4         int option = JOptionPane.showConfirmDialog(null,
5             "Do you want to change to the first class ticket?");
6
7         JOptionPane.showMessageDialog(null,"You've chosen: "
8             + (option==JOptionPane.YES_OPTION?"Yes":"No"));
9         System.exit(0);
10    }
11 }

```

```

import javax.swing.JOptionPane;
no usages
public class ChoosingOption {
    no usages
    public static void main(String args[]){
        int option = JOptionPane.showConfirmDialog( parentComponent: null,
            message: "Do you want to change to the first class ticket?");

        JOptionPane.showMessageDialog( parentComponent: null, message: "You 've chosen: "
            + (option == JOptionPane.YES_OPTION ? "YES" : "NO"));
    }
}

```


If user choose “Cancel”, option is equal to 2 then the output is NO

How to customize the options to users, e.g. only two options: “Yes” and “No”, OR “I do” and “I don’t”

The screenshot shows the first version of the `ChoosingOption.java` file. The code defines a `main` method that uses `JOptionPane.showOptionDialog` with a message "Do you want to change to the first class ticket?" and two options: "I do" and "I don't". The output dialog box shows these two options.

```

1 import javax.swing.JOptionPane;
2 public class ChoosingOption {
3     public static void main(String args[]){
4         String options[] = {"I do", "I don't"};
5         int option = JOptionPane.showOptionDialog( parentComponent: null,
6             message: "Do you want to change to the first class ticket?", title: "choose one",
7             JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, icon: null, options, options[0]);
8         System.out.println(option);
9         JOptionPane.showMessageDialog( parentComponent: null, message: "You've chosen "
10             + (option == JOptionPane.YES_NO_OPTION ? "I DO" : "I DON'T"));
11     }
12 }

```

The screenshot shows the second version of the `ChoosingOption.java` file. The code defines a `main` method that uses `JOptionPane.showOptionDialog` with a message "Do you want to change to the first class ticket?" and two options: "YES" and "NO". The output dialog box shows these two options.

```

1 import javax.swing.JOptionPane;
2 public class ChoosingOption {
3     public static void main(String args[]){
4         String options[] = {"YES", "NO"};
5         int option = JOptionPane.showOptionDialog( parentComponent: null,
6             message: "Do you want to change to the first class ticket?", title: "choose one",
7             JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, icon: null, options, options[0]);
8         System.out.println(option);
9         JOptionPane.showMessageDialog( parentComponent: null, message: "You've chosen "
10             + (option == JOptionPane.YES_NO_OPTION ? "I DO" : "I DON'T"));
11     }
12 }

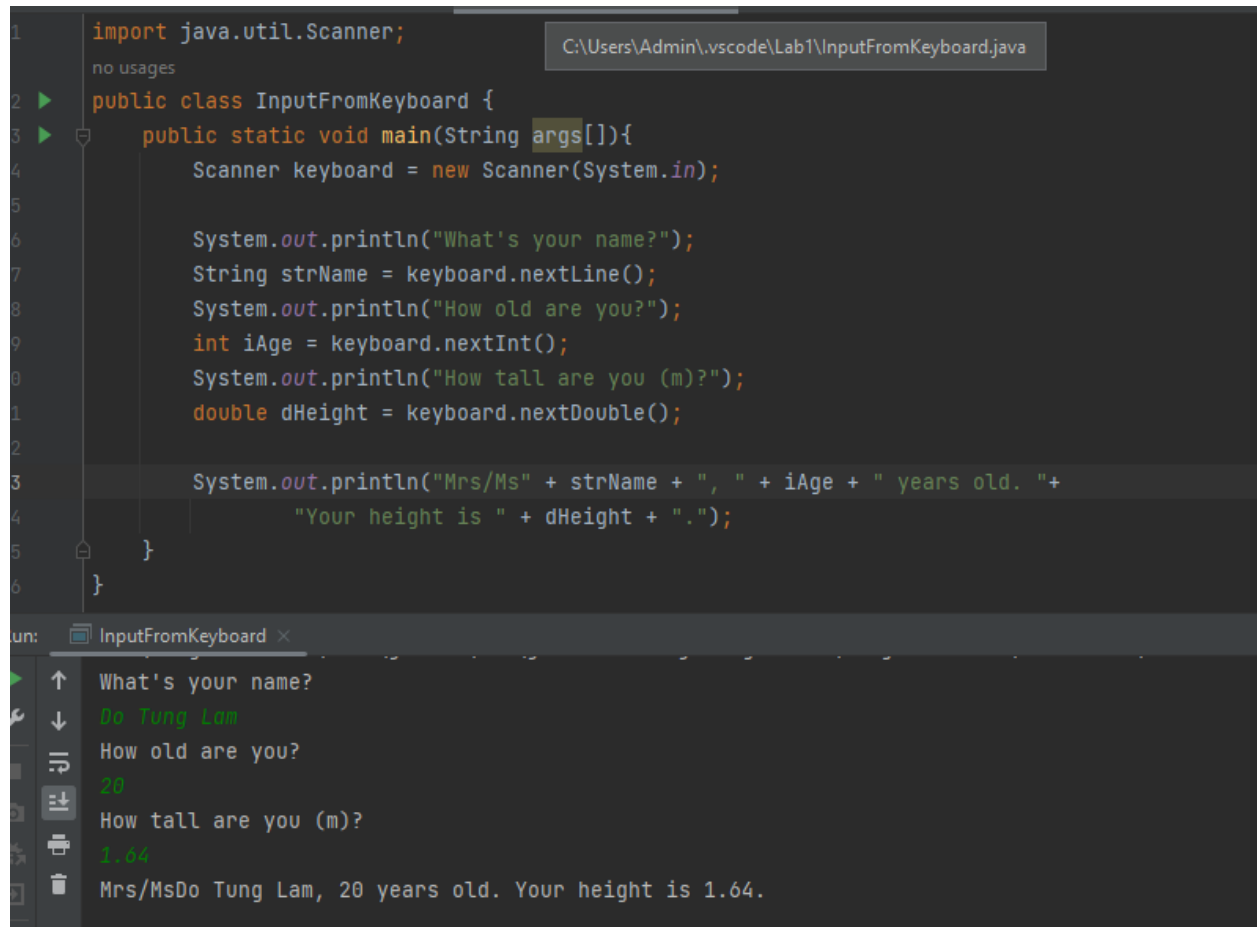
```

6.2 Write a program for input/output from keyboard

```
1 import java.util.Scanner;
2 public class InputFromKeyboard{
3     public static void main(String args[]){
4         Scanner keyboard = new Scanner(System.in);
5
6         System.out.println("What's your name?");
7         String strName = keyboard.nextLine();
8         System.out.println("How old are you?");
9         int iAge = keyboard.nextInt();
10        System.out.println("How tall are you (m)?");
11        double dHeight = keyboard.nextDouble();
12
13        //similar to other data types
14        //nextByte(), nextShort(), nextLong()
15        //nextFloat(), nextBoolean()
16
17        System.out.println("Mrs/Ms. " + strName + ", " + iAge + " years old. "
18                           + "Your height is " + dHeight + ".");
19    }
20 }
21 }
```

Markers Properties Servers Data Source Explorer Snippets Problems Console Search

<terminated> InputFromKeyboard [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_171.jdk/Contents/Home/bin/
What's your name?
Trang
How old are you?
35
How tall are you (m)?
1.65
Mrs/Ms. Trang, 35 years old. Your height is 1.65.



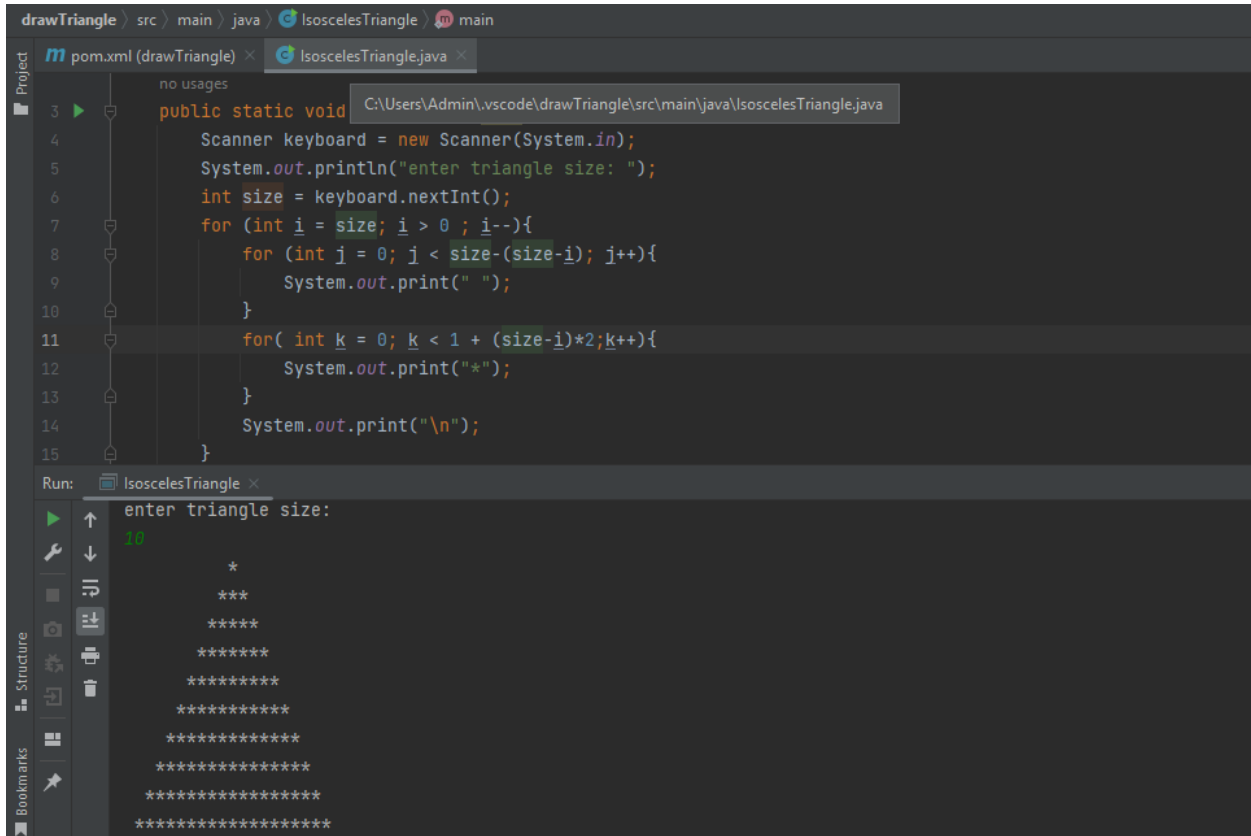
```
1 import java.util.Scanner;  
2  
3 public class InputFromKeyboard {  
4     public static void main(String args[]){  
5         Scanner keyboard = new Scanner(System.in);  
6  
7         System.out.println("What's your name?");  
8         String strName = keyboard.nextLine();  
9         System.out.println("How old are you?");  
10        int iAge = keyboard.nextInt();  
11        System.out.println("How tall are you (m)?");  
12        double dHeight = keyboard.nextDouble();  
13  
14        System.out.println("Mrs/Ms" + strName + ", " + iAge + " years old. "+  
15                           "Your height is " + dHeight + ".");  
16    }  
17 }
```

run: InputFromKeyboard ×

```
↑ What's your name?  
↓ Do Tung Lam  
How old are you?  
20  
How tall are you (m)?  
1.64  
Mrs/MsDo Tung Lam, 20 years old. Your height is 1.64.
```

6.3 Write a program to display a triangle with a height of n stars (*), n is entered by users.

Result



The screenshot shows a Java IDE with a project named 'drawTriangle'. The main file is 'IsoscelesTriangle.java'. The code defines a public static void method that takes no arguments. It uses a Scanner to read an integer 'size' from the user. It then uses nested for loops to print an isosceles triangle of asterisks. The first loop prints spaces, and the second loop prints asterisks. The output shows a triangle with a base of 10 asterisks.

```
public static void  
    Scanner keyboard = new Scanner(System.in);  
    System.out.println("enter triangle size: ");  
    int size = keyboard.nextInt();  
    for (int i = size; i > 0 ; i--){  
        for (int j = 0; j < size-(size-i); j++){  
            System.out.print(" ");  
        }  
        for( int k = 0; k < 1 + (size-i)*2;k++){  
            System.out.print("*");  
        }  
        System.out.print("\n");  
    }
```

Run: IsoscelesTriangle x

enter triangle size:
10

```
      *  
     ***  
    *****  
   *****  
  *****  
 *****  
*****  
*****  
*****  
*****  
*****
```

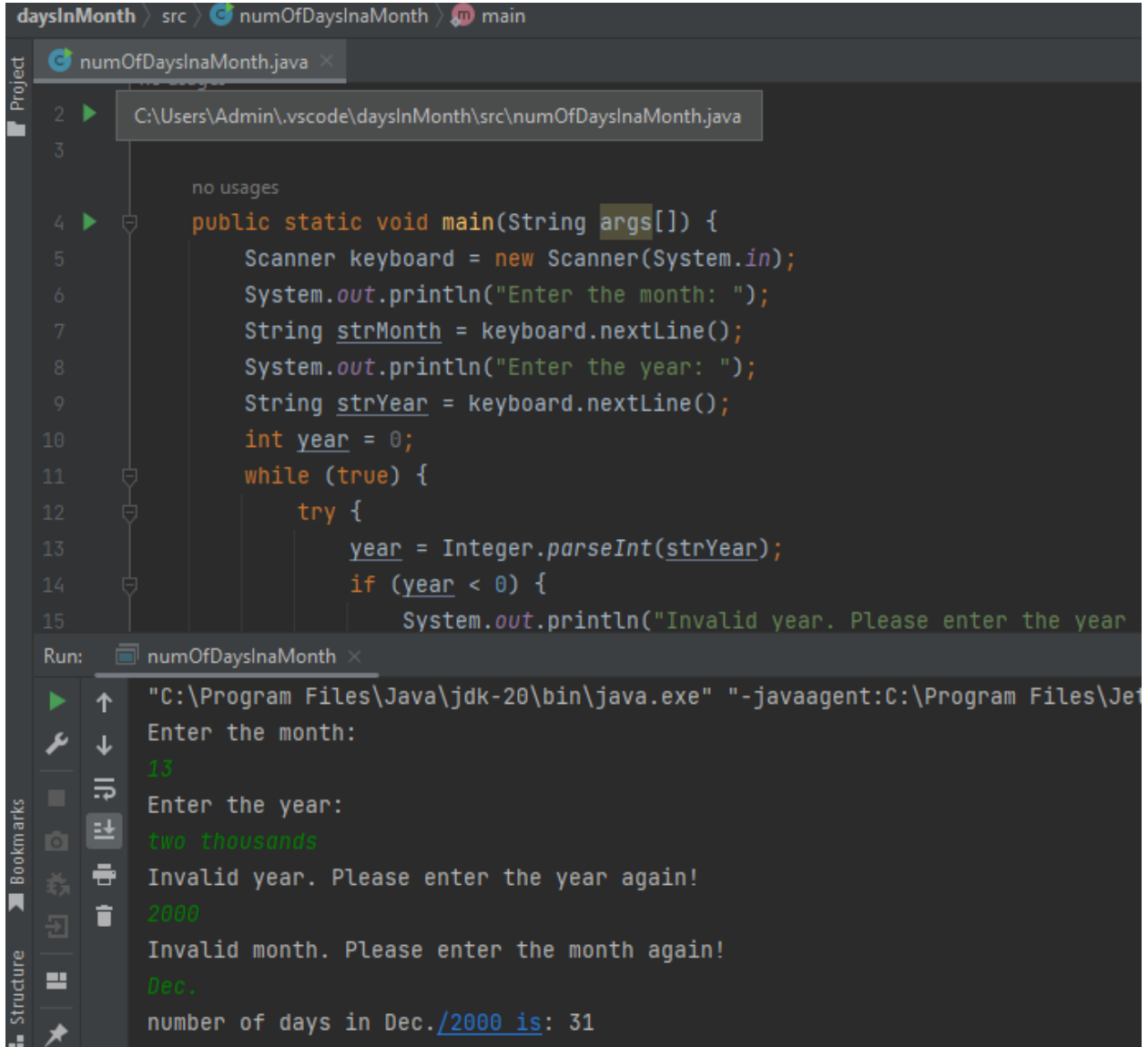
6.4 Write a program to display the number of days of a month, which is entered by users (both month and year). If it is an invalid month/year, ask the user to enter again.

The screenshot displays a Java IDE with the following components:

- Project Explorer:** Shows the project structure with 'daysInMonth' as the root, containing 'src' and 'main' folders. The file 'numOfDayInMonth.java' is selected.
- Source Editor:** Contains the Java code for 'numOfDayInMonth.java'. The code prompts the user to enter a month and a year, then calculates and prints the number of days in that month for the given year. The code is as follows:

```
2 public class numOfDayInMonth {
3
4     public static void main(String args[]) {
5         Scanner keyboard = new Scanner(System.in);
6         System.out.println("Enter the month: ");
7         String strMonth = keyboard.nextLine();
8         System.out.println("Enter the year: ");
9         String strYear = keyboard.nextLine();
10        int year = 0;
11        while (true) {
12            try {
13                year = Integer.parseInt(strYear);
14                if (year < 0) {
15                    System.out.println("Invalid year. Please enter the year again.");
16                    strYear = keyboard.nextLine();
17                }
18            } catch (NumberFormatException e) {
19                System.out.println("Invalid year. Please enter the year again.");
20                strYear = keyboard.nextLine();
21            }
22        }
23        // Logic to calculate the number of days in the month for the given year
24        // (The actual logic is not fully visible in the screenshot)
25    }
26 }
```
- Run Console:** Shows the execution of the program. The output is as follows:

```
Run: numOfDayInMonth
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-20\bin\javaagent.jar" -classpath .;C:\Program Files\Java\jdk-20\bin\javaagent.jar numOfDayInMonth
Enter the month:
12
Enter the year:
2023
number of days in 12/2023 is: 31
```

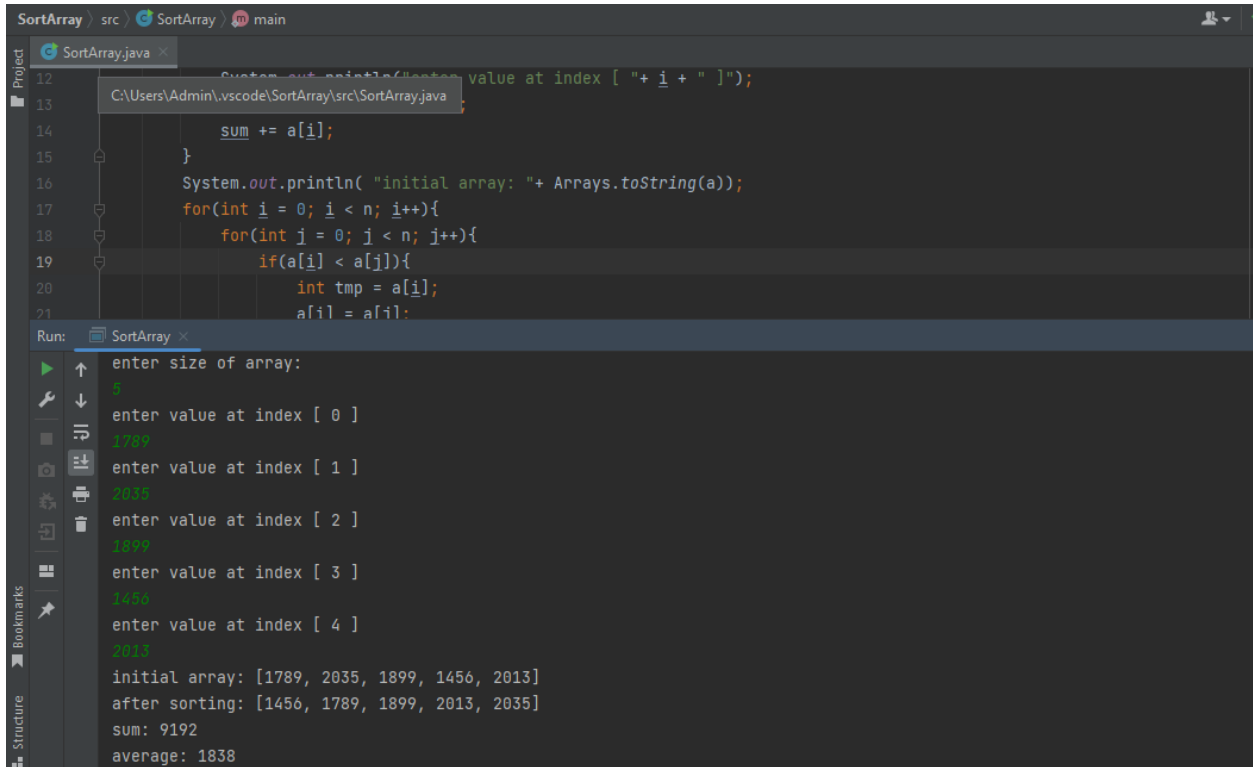


```
daysInMonth > src > numOfDayInMonth > main
numOfDayInMonth.java x
C:\Users\Admin\.vscode\daysInMonth\src\numOfDayInMonth.java
no usages
public static void main(String args[]) {
    Scanner keyboard = new Scanner(System.in);
    System.out.println("Enter the month: ");
    String strMonth = keyboard.nextLine();
    System.out.println("Enter the year: ");
    String strYear = keyboard.nextLine();
    int year = 0;
    while (true) {
        try {
            year = Integer.parseInt(strYear);
            if (year < 0) {
                System.out.println("Invalid year. Please enter the year again!");
                strYear = keyboard.nextLine();
            }
        } catch (NumberFormatException e) {
            System.out.println("Invalid year. Please enter the year again!");
            strYear = keyboard.nextLine();
        }
    }
    // ... (rest of the code for calculating days in month) ...
}
```

Run: numOfDayInMonth x

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\lib\idea_rt.jar=1273.0:C:\Program Files\Java\jdk-20\bin" -Dfile.encoding=UTF-8
Enter the month:
13
Enter the year:
two thousands
Invalid year. Please enter the year again!
2000
Invalid month. Please enter the month again!
Dec.
number of days in Dec./2000 is: 31
```

6.5 Write a Java program to sort a numeric array, and calculate the sum and average value of array elements.



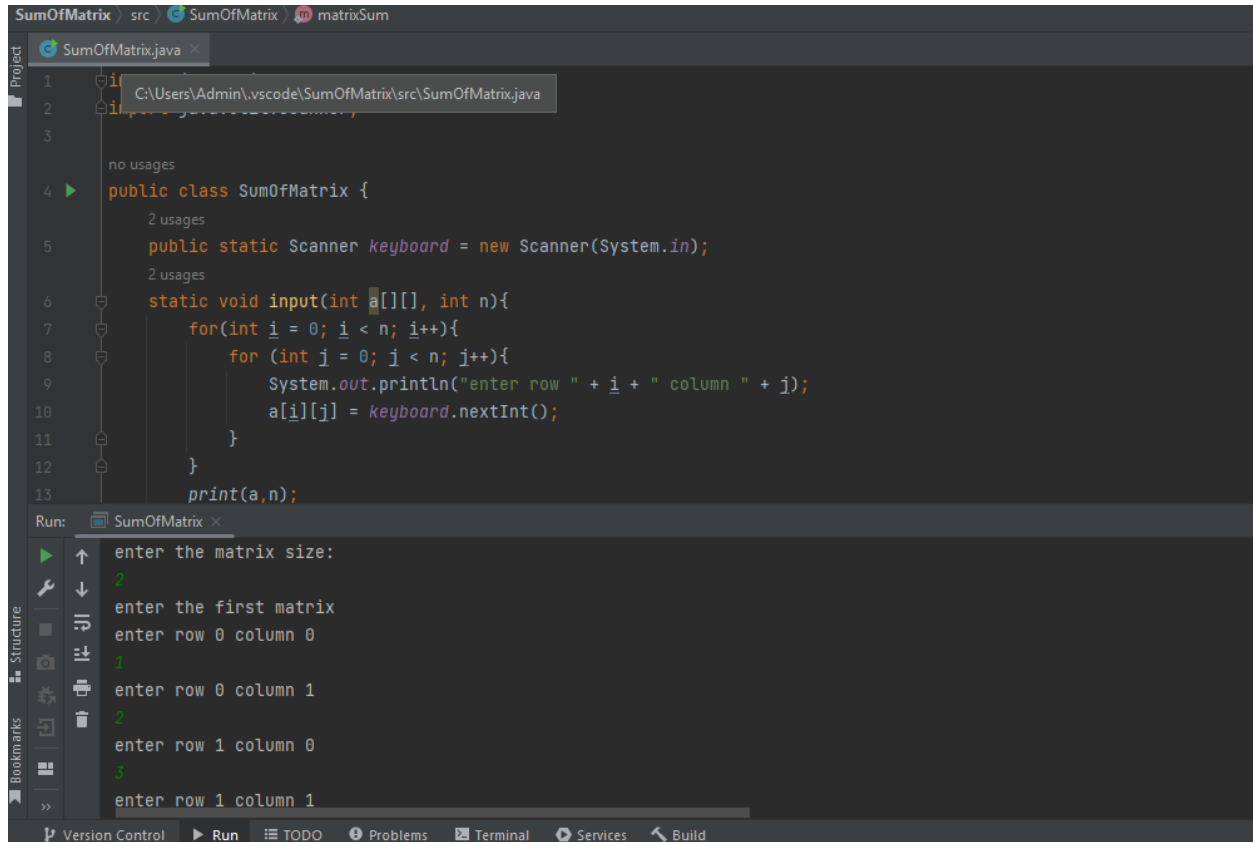
The screenshot shows a Java IDE with a project named 'SortArray'. The main file, 'SortArray.java', contains a bubble sort algorithm. The code prompts the user to enter the size of the array (5) and then five values (1789, 2035, 1899, 1456, 2013). It prints the initial array, sorts it, and prints the sorted array, the sum (9192), and the average (1838).

```
12 System.out.println("enter value at index [ "+ i + " ]");
13 C:\Users\Admin\.vscode\SortArray\src\SortArray.java
14     sum += a[i];
15 }
16 System.out.println( "initial array: "+ Arrays.toString(a));
17 for(int i = 0; i < n; i++){
18     for(int j = 0; j < n; j++){
19         if(a[i] < a[j]){
20             int tmp = a[i];
21             a[i] = a[j];
22             a[j] = tmp;
23         }
24     }
25 }
26 System.out.println("after sorting: "+ Arrays.toString(a));
27 System.out.println("sum: " + sum);
28 System.out.println("average: " + (sum / n));
```

Run: SortArray ×

```
enter size of array:
5
enter value at index [ 0 ]
1789
enter value at index [ 1 ]
2035
enter value at index [ 2 ]
1899
enter value at index [ 3 ]
1456
enter value at index [ 4 ]
2013
initial array: [1789, 2035, 1899, 1456, 2013]
after sorting: [1456, 1789, 1899, 2013, 2035]
sum: 9192
average: 1838
```

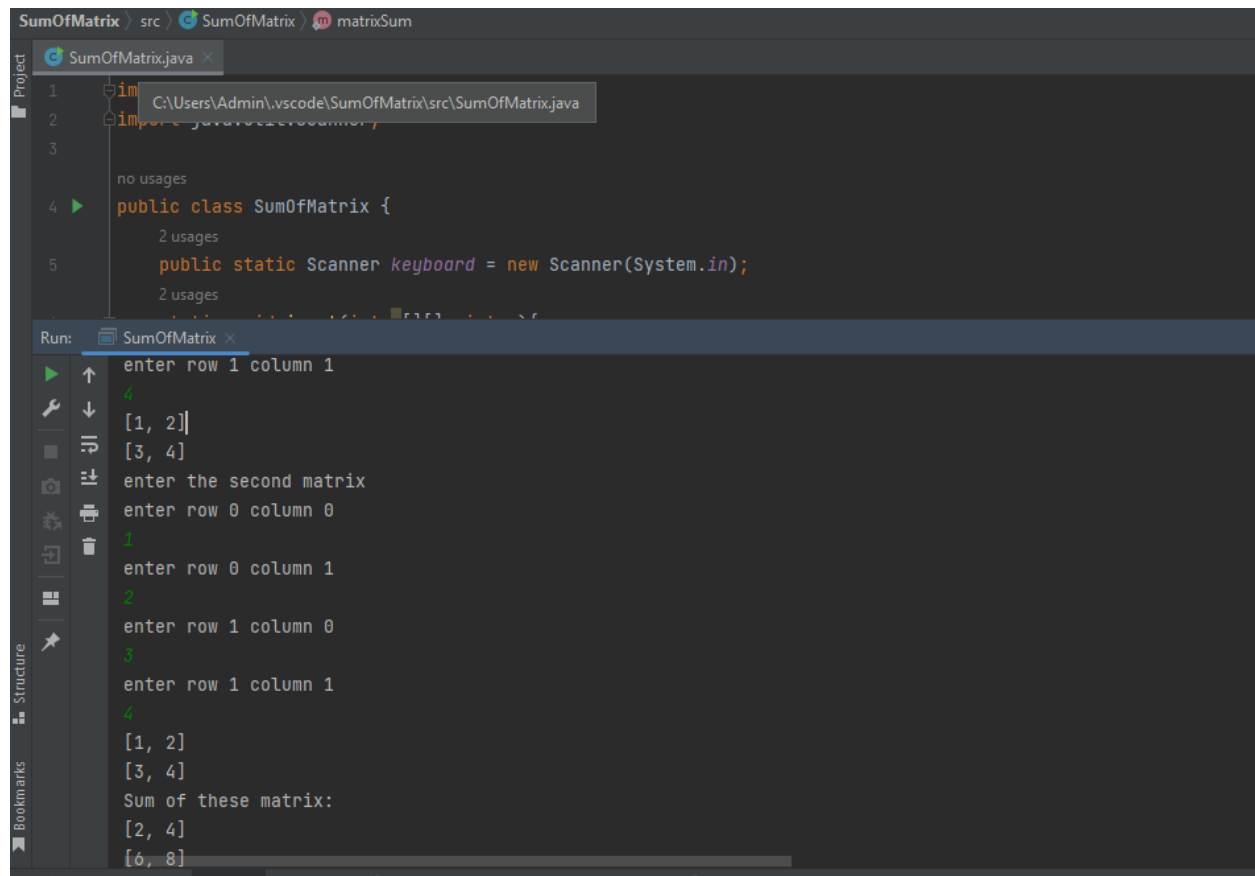
6.6 Write a Java program to add two matrices of the same size



```
SumOfMatrix.java
1  import java.util.Scanner;
2  public class SumOfMatrix {
3
4      public static Scanner keyboard = new Scanner(System.in);
5
6      static void input(int a[][], int n){
7          for(int i = 0; i < n; i++){
8              for (int j = 0; j < n; j++){
9                  System.out.println("enter row " + i + " column " + j);
10                 a[i][j] = keyboard.nextInt();
11             }
12         }
13         print(a,n);
14     }
15 }
```

Run: SumOfMatrix

```
enter the matrix size:
2
enter the first matrix
enter row 0 column 0
1
enter row 0 column 1
2
enter row 1 column 0
3
enter row 1 column 1
3
```

```
SumOfMatrix.java
1  import java.util.Scanner;
2  import java.util.Scanner;
3
4  no usages
5  public class SumOfMatrix {
6      2 usages
7      public static Scanner keyboard = new Scanner(System.in);
8      2 usages
9
10     public static void main(String[] args) {
11         // Enter the first matrix
12         System.out.println("enter row 1 column 1");
13         int row1 = keyboard.nextInt();
14         int col1 = keyboard.nextInt();
15         int[][] matrix1 = new int[row1][col1];
16         for (int i = 0; i < row1; i++) {
17             for (int j = 0; j < col1; j++) {
18                 System.out.println("enter the second matrix");
19                 System.out.println("enter row 0 column 0");
20                 int row0 = keyboard.nextInt();
21                 int col0 = keyboard.nextInt();
22                 int[][] matrix0 = new int[row0][col0];
23                 for (int i = 0; i < row0; i++) {
24                     for (int j = 0; j < col0; j++) {
25                         System.out.println("enter row 0 column 1");
26                         int row0 = keyboard.nextInt();
27                         int col0 = keyboard.nextInt();
28                         System.out.println("enter row 1 column 0");
29                         int row1 = keyboard.nextInt();
30                         int col1 = keyboard.nextInt();
31                         System.out.println("enter row 1 column 1");
32                         int row1 = keyboard.nextInt();
33                         int col1 = keyboard.nextInt();
34                         int[][] matrix1 = new int[row1][col1];
35                         for (int i = 0; i < row1; i++) {
36                             for (int j = 0; j < col1; j++) {
37                                 matrix1[i][j] = keyboard.nextInt();
38                             }
39                         }
40                     }
41                 }
42             }
43         }
44         // Sum of the two matrices
45         int rowSum = row1 + row0;
46         int colSum = col1 + col0;
47         int[][] sumMatrix = new int[rowSum][colSum];
48         for (int i = 0; i < rowSum; i++) {
49             for (int j = 0; j < colSum; j++) {
50                 sumMatrix[i][j] = matrix1[i][j] + matrix0[i][j];
51             }
52         }
53         // Print the sum matrix
54         System.out.println("Sum of these matrix:");
55         for (int i = 0; i < rowSum; i++) {
56             for (int j = 0; j < colSum; j++) {
57                 System.out.print(sumMatrix[i][j] + " ");
58             }
59             System.out.println();
60         }
61     }
62 }
```

Run: SumOfMatrix

```
enter row 1 column 1
4
[1, 2]
[3, 4]
enter the second matrix
enter row 0 column 0
1
enter row 0 column 1
2
enter row 1 column 0
3
enter row 1 column 1
4
[1, 2]
[3, 4]
Sum of these matrix:
[2, 4]
[6, 8]
```