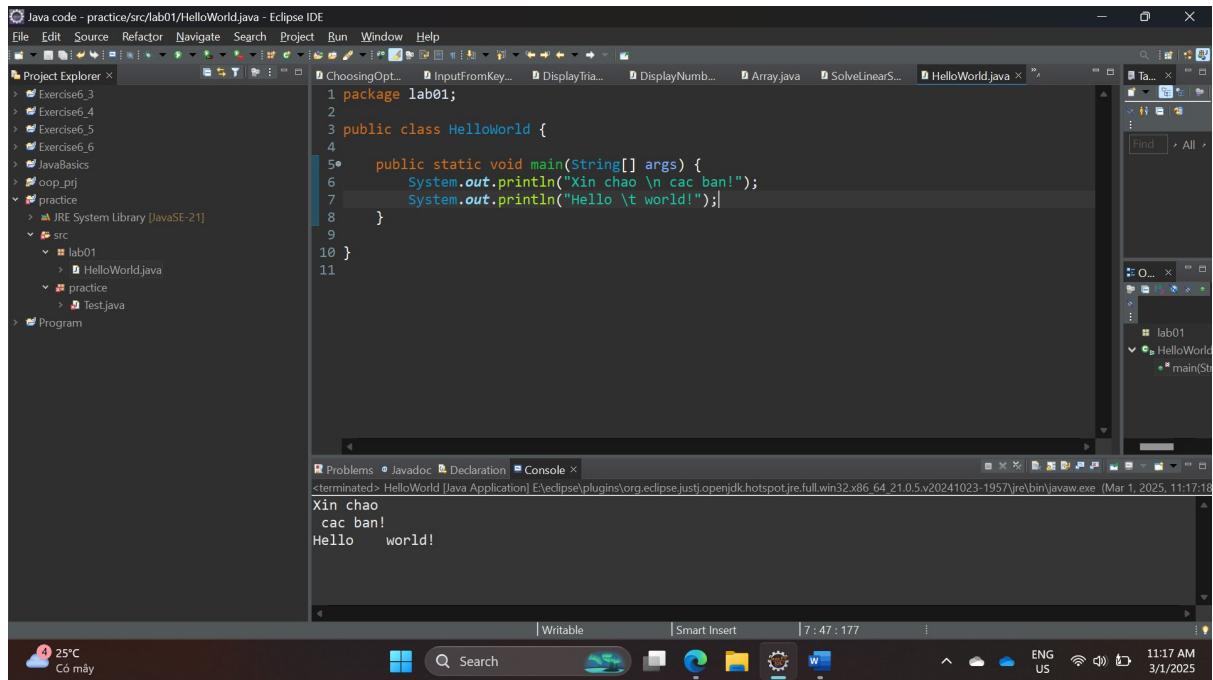


Student name: Lê Ngọc Anh Vũ

Student ID: 20236014

## REPORT LAB 01

### 2.2.1 Write, compile the first Java application:



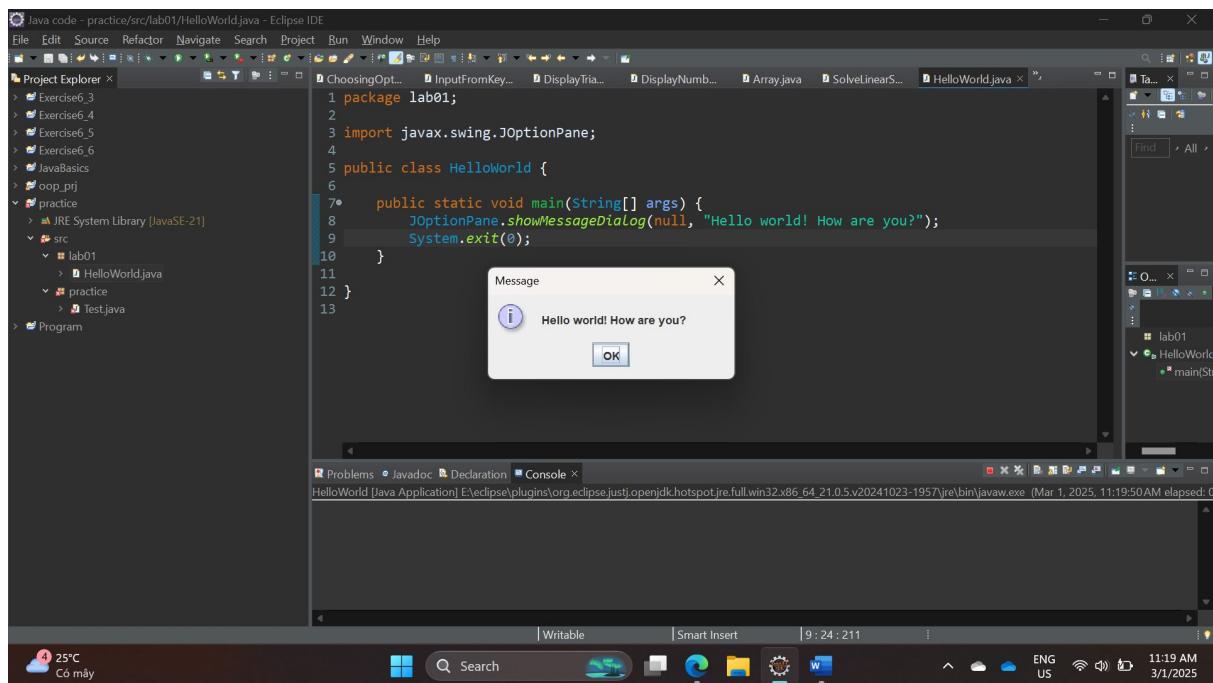
The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left lists several Java projects and files. In the center, the code editor displays the following Java code:

```
1 package lab01;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         System.out.println("Xin chào \n cac ban!");
7         System.out.println("Hello \t world!");
8     }
9
10 }
```

The Console view at the bottom shows the output of the program:

```
Xin chào
cac ban!
Hello    world!
```

### 2.2.2 Write, compile the first dialog Java program

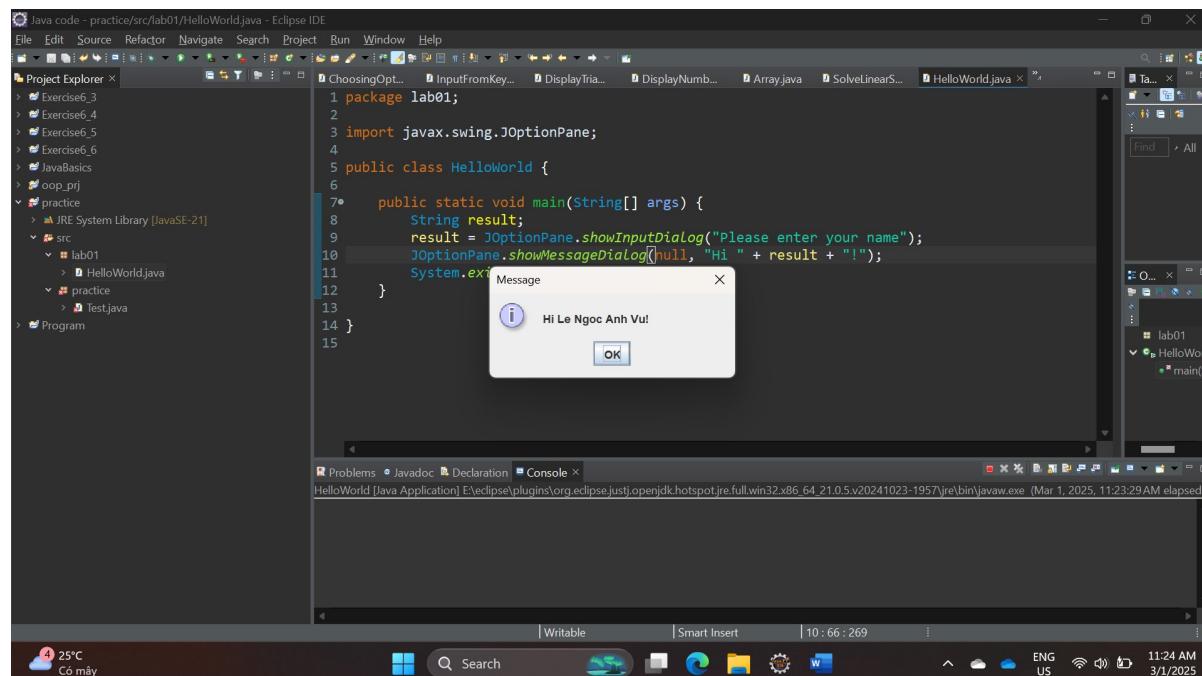
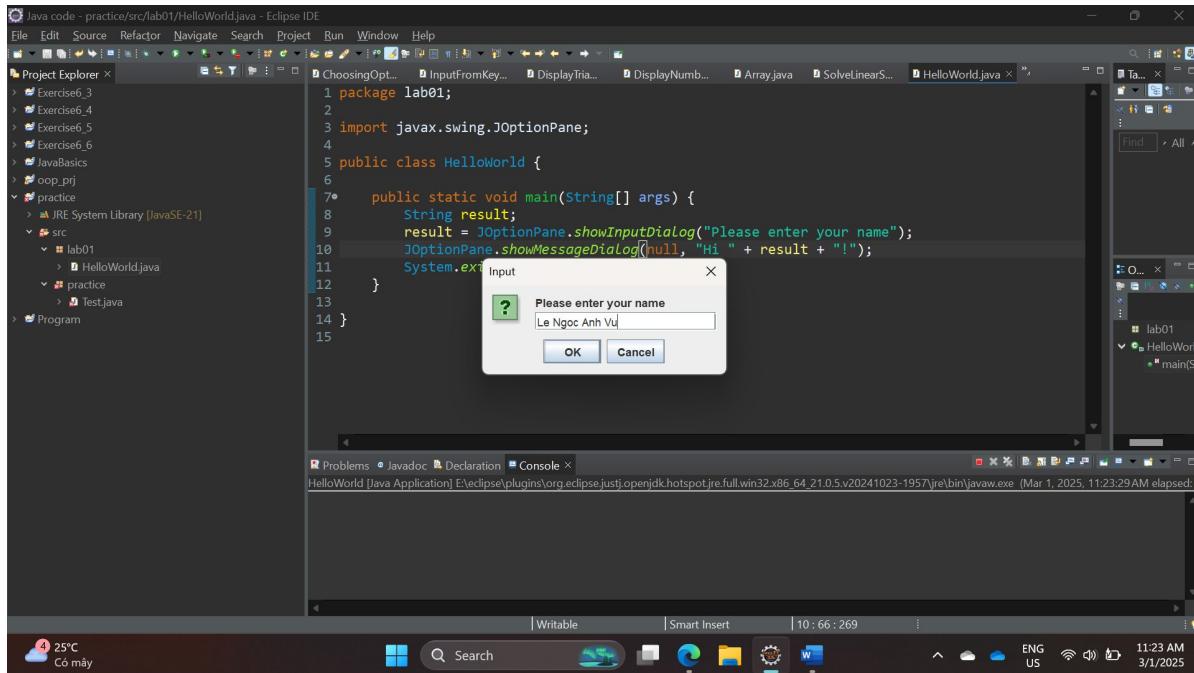


The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left lists several Java projects and files. In the center, the code editor displays the following Java code:

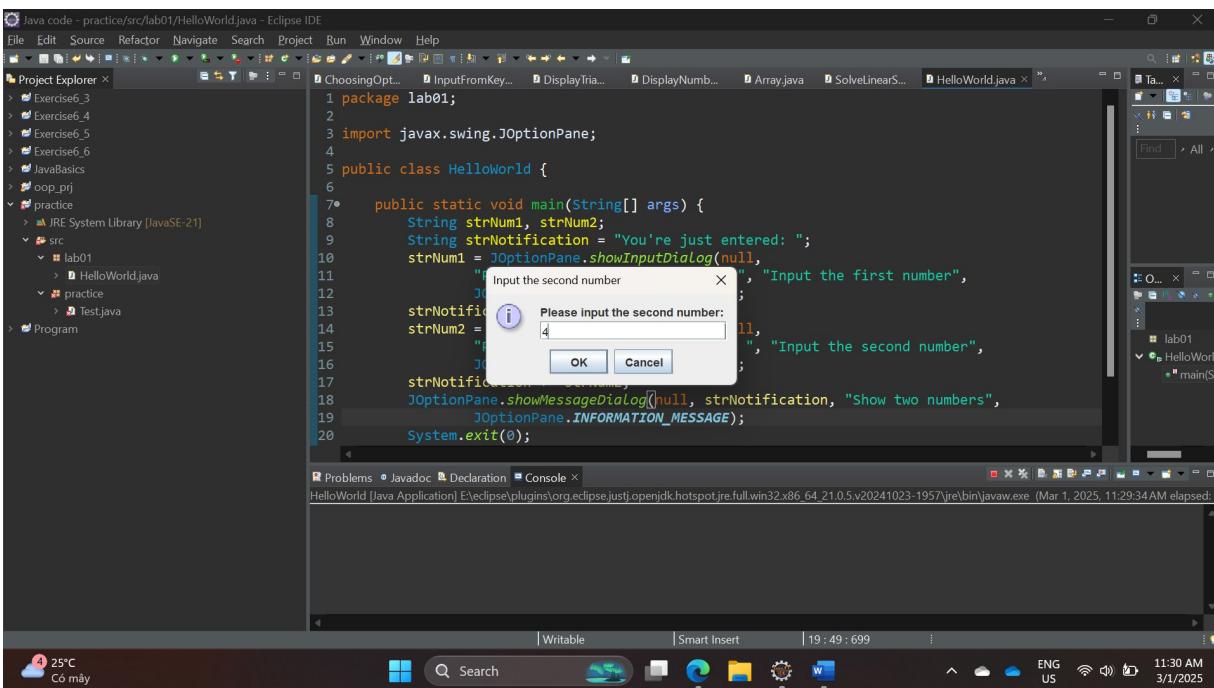
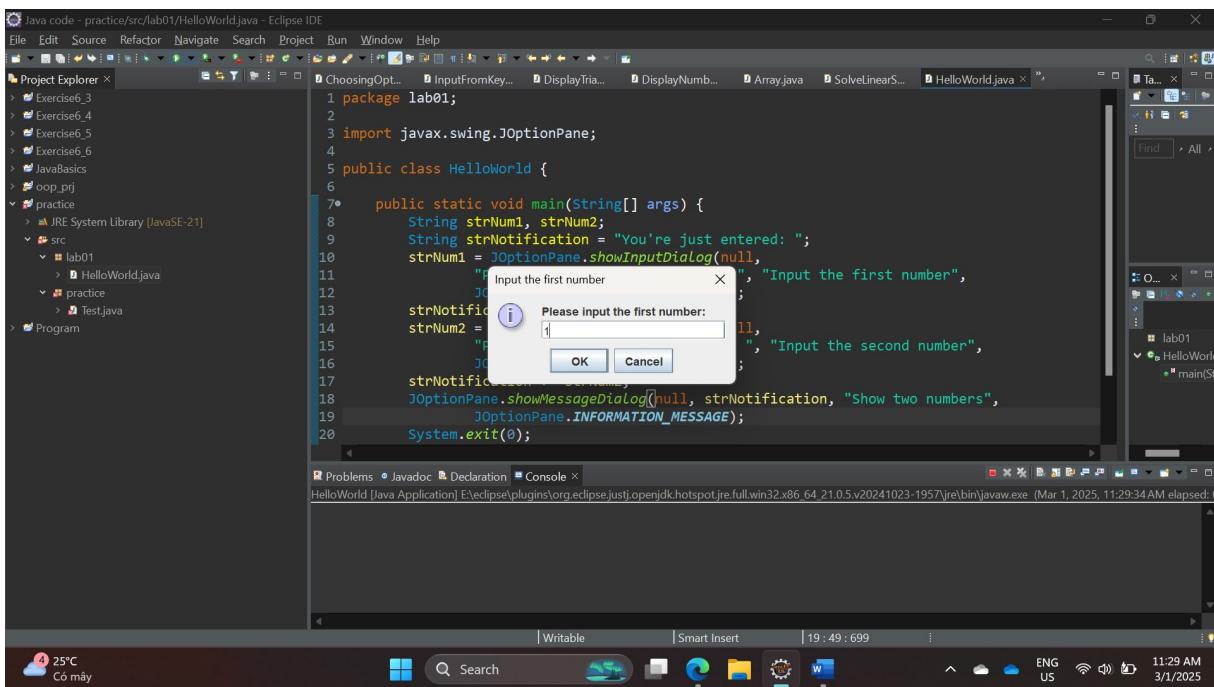
```
1 package lab01;
2
3 import javax.swing.JOptionPane;
4
5 public class HelloWorld {
6
7     public static void main(String[] args) {
8         JOptionPane.showMessageDialog(null, "Hello world! How are you?");
9         System.exit(0);
10    }
11
12 }
```

A message dialog box titled "Message" is displayed in the foreground, containing the text "Hello world! How are you?". The OK button is visible at the bottom of the dialog.

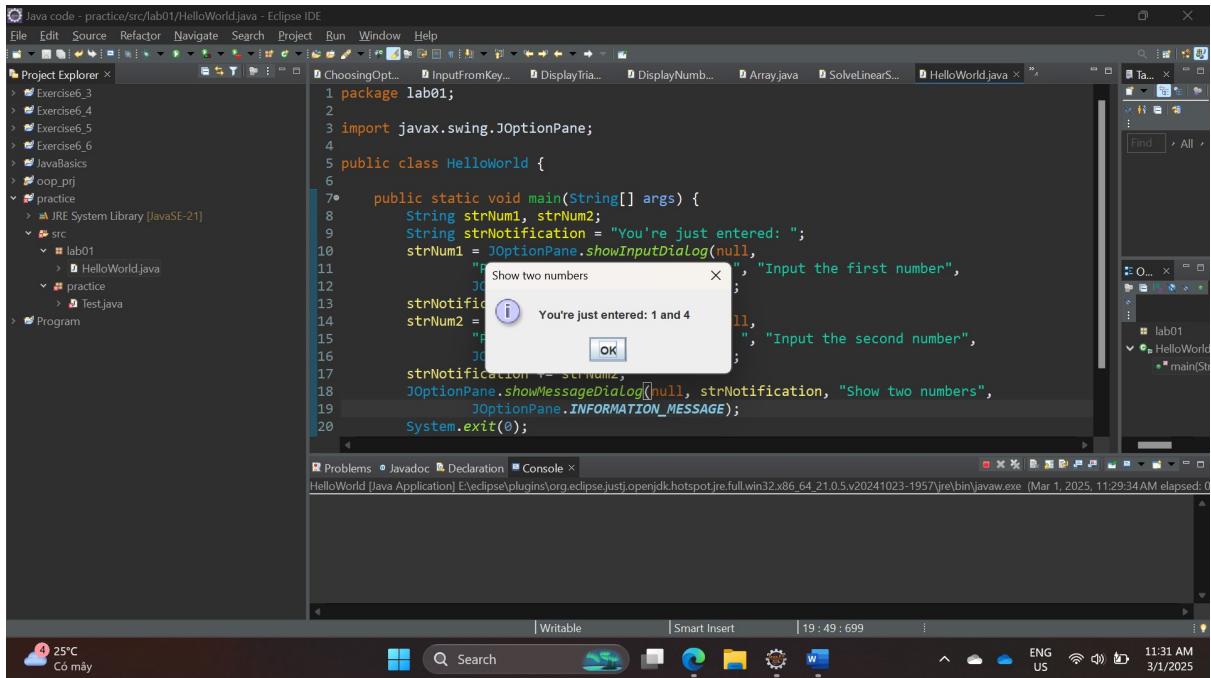
### 2.2.3 Write, compile the first input dialog Java application



### 2.2.4 Write, compile, and run the following example:



And the final dialog:



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

### Notes

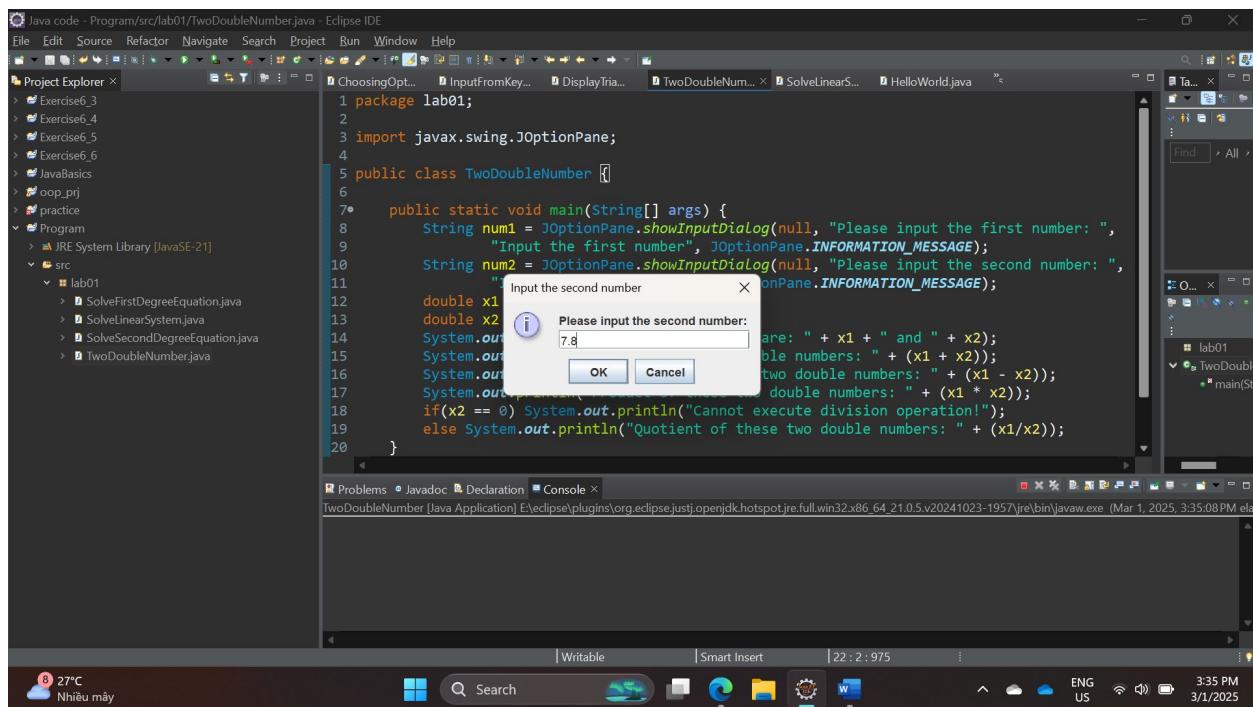
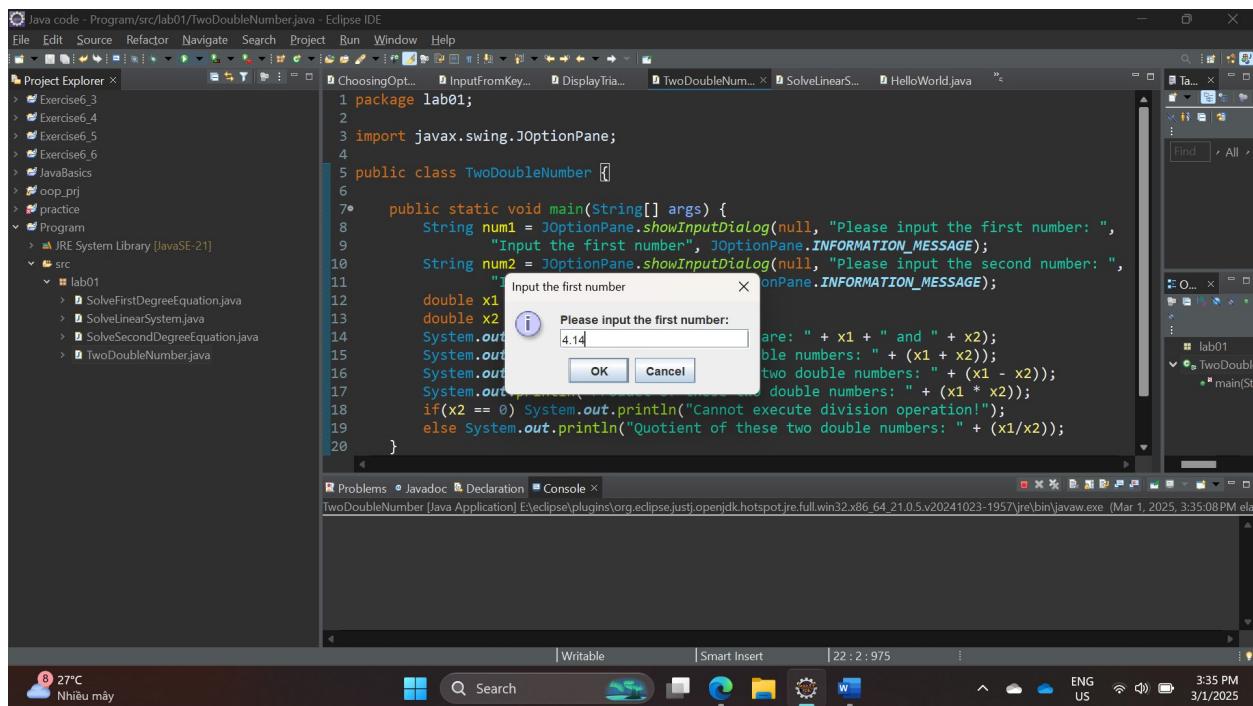
- To convert from String to double, you can use

```
double num1 = Double.parseDouble(strNum1)
```

- Check the divisor of the division

```
1 package lab01;
2
3 import javax.swing.JOptionPane;
4
5 public class TwoDoubleNumber {
6
7     public static void main(String[] args) {
8         String num1 = JOptionPane.showInputDialog(null, "Please input the first number: ",
9             "Input the first number", JOptionPane.INFORMATION_MESSAGE);
10        String num2 = JOptionPane.showInputDialog(null, "Please input the second number: ",
11            "Input the second number", JOptionPane.INFORMATION_MESSAGE);
12        double x1 = Double.parseDouble(num1);
13        double x2 = Double.parseDouble(num2);
14        System.out.println("Two numbers entered are: " + x1 + " and " + x2);
15        System.out.println("Sum of these two double numbers: " + (x1 + x2));
16        System.out.println("Difference of these two double numbers: " + (x1 - x2));
17        System.out.println("Product of these two double numbers: " + (x1 * x2));
18        if(x2 == 0) System.out.println("Cannot execute division operation!");
19        else System.out.println("Quotient of these two double numbers: " + (x1/x2));
20    }
21
22 }|
```

Executing the program:



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like Exercise6\_3, Exercise6\_4, Exercise6\_5, Exercise6\_6, JavaBasics, pop\_pj, practice, Program, JRE System Library [JavaSE-21], and src/lab01 containing SolveFirstDegreeEquation.java, SolveLinearSystem.java, SolveSecondDegreeEquation.java, and TwoDoubleNumber.java.
- Code Editor:** Displays the Java code for TwoDoubleNumber.java. The code prompts the user for two double numbers, calculates their sum, difference, product, and quotient, and handles division by zero.
- Console:** Shows the execution output:
 

```

Two numbers entered are: 4.14 and 7.8
Sum of these two double numbers: 11.94
Difference of these two double numbers: -3.66
Product of these two double numbers: 32.29199999999994
Quotient of these two double numbers: 0.5307692307692308

```
- Bottom Status Bar:** Shows system information including weather (27°C), search bar, taskbar icons, and system status (ENG US, 3:36 PM, 3/1/2025).

## 2.2.6 Write a program to solve:

For simplicity, we only consider the real roots of the equations in this task.

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

Note: A first-degree equation with one variable can have a form such as  $ax + b = 0$  ( $a \neq 0$ ).

You should handle the case where the user input value 0 for a.

```

1 package lab01;
2
3 import java.util.*;
4
5 public class SolveFirstDegreeEquation {
6     public static void main(String[] args) {
7         System.out.println("Consider equation ax + b = 0");
8         Scanner read = new Scanner(System.in);
9         System.out.print("a = ");
10        float a = read.nextFloat();
11        System.out.print("b = ");
12        float b = read.nextFloat();
13        read.close();
14        if(a == 0) {
15            if(b == 0)System.out.println("Equation has many roots");
16            else System.out.println("Equation does not have any roots");
17        }
18        else {
19            System.out.println("Equation has unique root: " + (-b/a));
20        }
21    }
22 }

```

Executing the program:

```

1 package lab01;
2
3 import java.util.*;
4
5 public class SolveFirstDegreeEquation {
6     public static void main(String[] args) {
7         System.out.println("Consider equation ax + b = 0");
8         Scanner read = new Scanner(System.in);
9         System.out.print("a = ");
10        float a = read.nextFloat();
11        System.out.print("b = ");
12        float b = read.nextFloat();
13        read.close();
14        if(a == 0) {
15            if(b == 0)System.out.println("Equation has many roots");
16            else System.out.println("Equation does not have any roots");
17        }
18        else {
19            System.out.println("Equation has unique root: " + (-b/a));
20        }
21    }
22}

```

Output window:

```

Consider equation ax + b = 0
a = 4
b = -2
Equation has unique root: 0.5

```

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

Note: A system of first-degree equations with two variables  $x_1$  and  $x_2$  can be written as follows.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

You should handle the case where the values of the coefficients produce infinitely many solutions and the case where they produce no solution.

```

1 package lab01;
2
3 import java.util.*;
4
5 public class SolveLinearSystem {
6     public static void main(String[] args) {
7         System.out.println("Consider a system of first-degree equations");
8         System.out.println("ax + by = m");
9         System.out.println("cx + dy = n\n");
10        Scanner read = new Scanner(System.in);
11        System.out.print("a = ");
12        float a = read.nextFloat();
13        System.out.print("b = ");
14        float b = read.nextFloat();
15        System.out.print("m = ");
16        float m = read.nextFloat();
17        System.out.print("c = ");
18        float c = read.nextFloat();
19        System.out.print("d = ");
20        float d = read.nextFloat();
21        System.out.print("n = ");
22        float n = read.nextFloat();
23        read.close();
24    }
25}

```

```

24 float D = a*d - b*c;
25 float D2 = a*n - c*m;
26 float D1 = m*d - n*b;
27 if(a == 0 && b == 0 && c == 0 && d == 0 && m == 0 && n == 0)
28     System.out.println("The system of equations has infinitely many roots (x, y) in R*R");
29 else if(D != 0)
30     System.out.printf("The system of equations has unique root (x, y) = (%f/%f, %f/%f)", D1,
31 else if(D == 0 && (D1 != 0 || D2 != 0))
32     System.out.println("The system of equations has no roots");
33 else if(D == 0 && D1 == 0 && D2 == 0){
34     if(n*n + m*m == 0) {
35         if(a == 0 && c == 0)
36             System.out.println("The system of equations has infinitely many roots (x, y) = (",
37         else if(b == 0 && d == 0)
38             System.out.println("The system of equations has infinitely many roots (x, y) = (",
39         else if(a == 0 && b == 0)
40             System.out.printf("The system of equations has infinitely many roots (x, y) = (%f",
41         else if(c == 0 && d == 0)
42             System.out.printf("The system of equations has infinitely many roots (x, y) = (%f",
43     }
44     else if(m == 0) {
45         if(c == 0 && d == 0)
46             System.out.println("The system of equations has no roots");
47
48     else if(c == 0)
49         System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, %f",
50     else if(d == 0)
51         System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, %f",
52     else
53         System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, (%f",
54
55     else if(n == 0) {
56         if(a == 0 && b == 0)
57             System.out.println("The system of equations has no roots");
58         else if(a == 0)
59             System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, %f",
60         else if(b == 0)
61             System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, %f",
62         else
63             System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, (%f",
64
65     else if(a == 0 && b == 0 && c == 0 && d == 0)
66         System.out.println("The system of equations has no roots");
67     else if(a == 0)
68         System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, %f",
69     else if(b == 0)
70         System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, %f",
71 else if(a != 0 && b != 0)
72     System.out.printf("The system of equations has infinitely many roots (x, y) = (%f, (%f",

```

Executing the program:

The screenshot shows the Eclipse IDE interface. In the center, there's a code editor with Java code for solving linear systems. Below it is a terminal window showing the output of the program. The terminal output includes prompts for input values (a, b, m, c, d, n) and the final result: "The system of equations has unique root (x, y) = (-14.000000/-43.000000, -46.000000/-43.000000)". The status bar at the bottom shows system information like weather, battery level, and date/time.

```

1 package lab01;
2
3 import java.util.*;
4
5 public class SolveLinearSystem {
6     public static void main(String[] args) {
7         System.out.println("Consider a system of first-degree equations");
8         System.out.println("ax + by = m");
9         System.out.println("cx + dy = n\n");
10        Scanner read = new Scanner(System.in);
11        System.out.print("a = ");
12        float a = read.nextFloat();
13        System.out.print("b = ");
14        float b = read.nextFloat();
15        System.out.print("m = ");

```

### - The second-degree equation with one variable

Note: A second-degree equation with one variable (i.e., quadratic equation) can have a form such as  $ax^2 + bx + c = 0$ , where x is the variable, and a, b, and c are coefficients ( $a \neq 0$ ).

You should handle the case where the values of the coefficients produce a double root & the case where they produce no root. You should also handle the case where the user input value 0 for a.

```

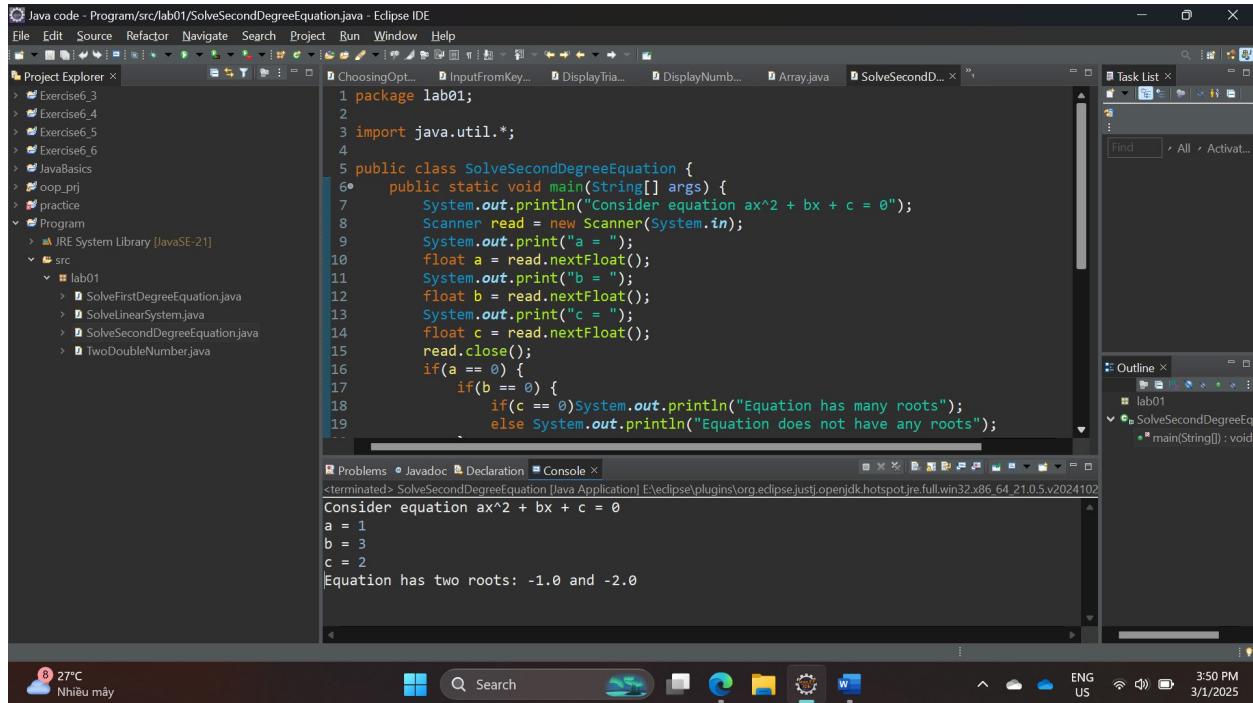
1 package lab01;
2
3 import java.util.*;
4
5 public class SolveSecondDegreeEquation {
6     public static void main(String[] args) {
7         System.out.println("Consider equation ax^2 + bx + c = 0");
8         Scanner read = new Scanner(System.in);
9         System.out.print("a = ");
10        float a = read.nextFloat();
11        System.out.print("b = ");
12        float b = read.nextFloat();
13        System.out.print("c = ");
14        float c = read.nextFloat();
15        read.close();
16        if(a == 0) {
17            if(b == 0) {
18                if(c == 0)System.out.println("Equation has many roots");
19                else System.out.println("Equation does not have any roots");
20            }
21            else {
22                System.out.println("Equation has unique root: " + (-c/b));
23            }
24        }
25        else {
26            float delta = b*b - 4*a*c;

```

```

27         if(delta < 0) System.out.println("Equation does not have any roots");
28     else if (delta == 0) System.out.println("Equation has unique root: " + (-b/(2*a))
29     else {
30         float x1 = (-b + (float)Math.sqrt(delta))/(2*a);
31         float x2 = (-b - (float)Math.sqrt(delta))/(2*a);
32         System.out.println("Equation has two roots: " + x1 + " and " + x2);
33     }
34 }
35 }
36 }
```

Executing the program:



## 6 Exercises

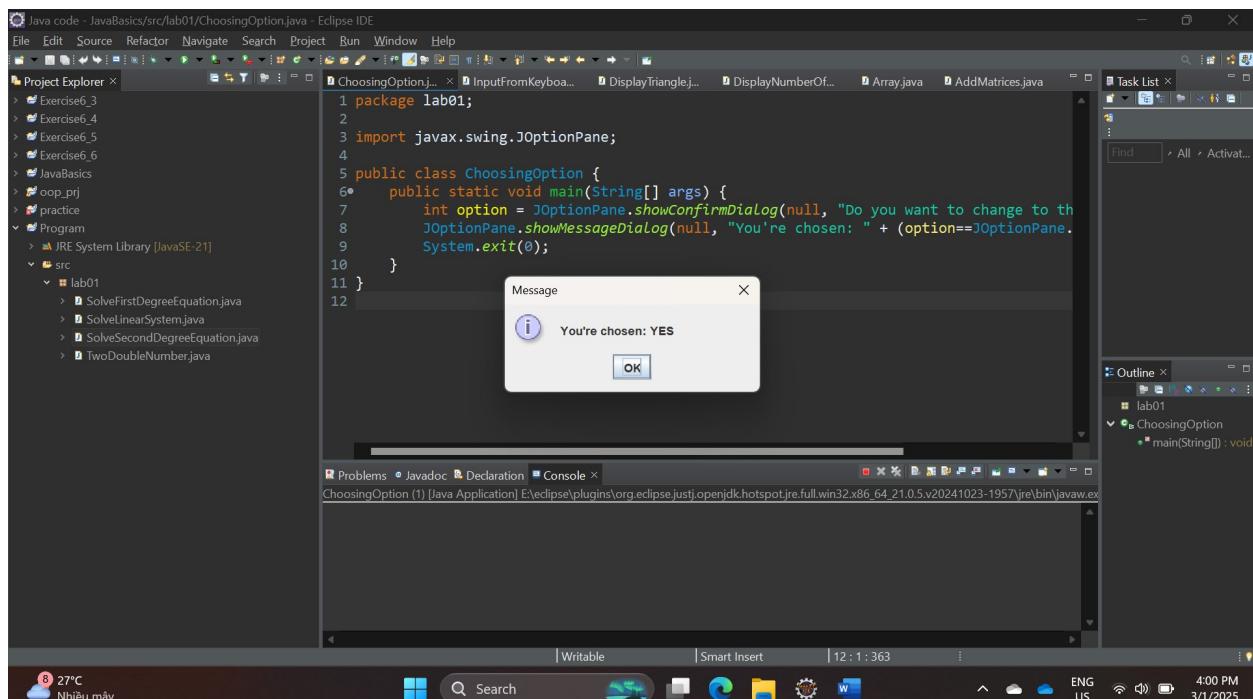
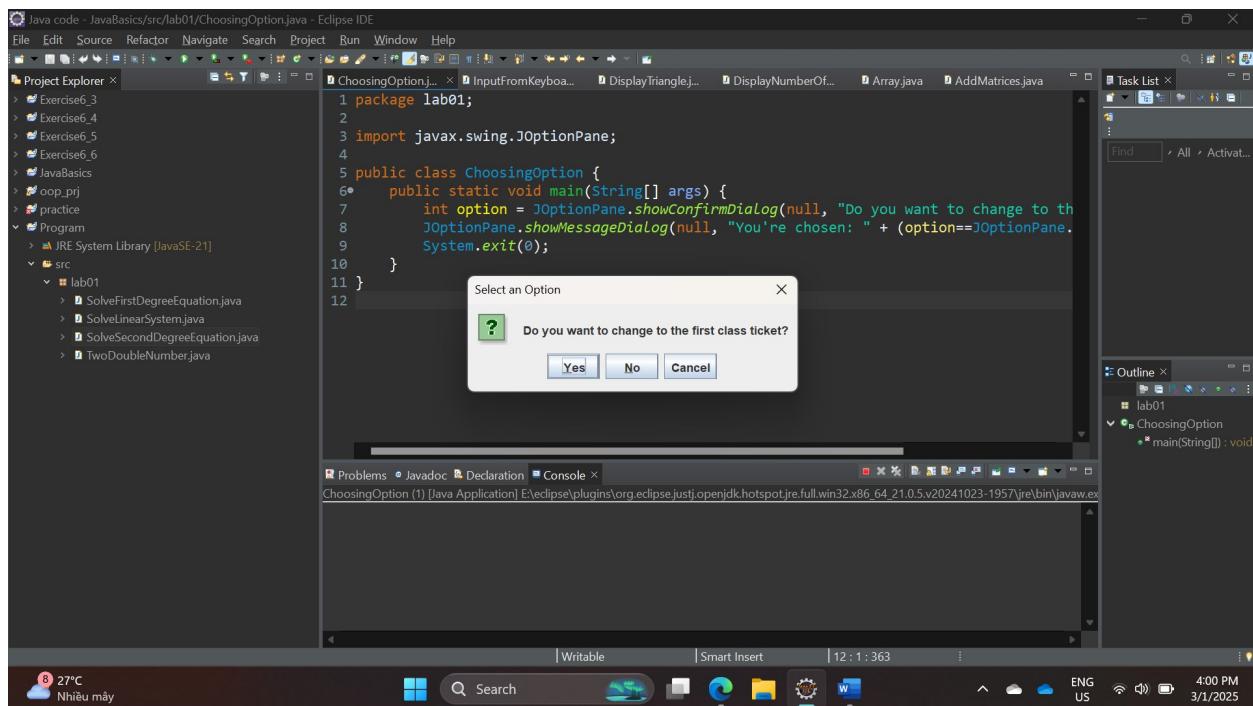
### 6.1 Write, compile and run the ChoosingOption program:

Note: We use the JavaBasics project for this exercise.

```

1 package lab01;
2
3 import javax.swing.JOptionPane;
4
5 public class ChoosingOption {
6     public static void main(String[] args) {
7         int option = JOptionPane.showConfirmDialog(null, "Do you want to change to th
8         JOptionPane.showMessageDialog(null, "You're chosen: " + (option==JOptionPane.
9         System.exit(0);
10    }
11 }
```

Executing the program:



## 6.2 Write a program for input/output from keyboard

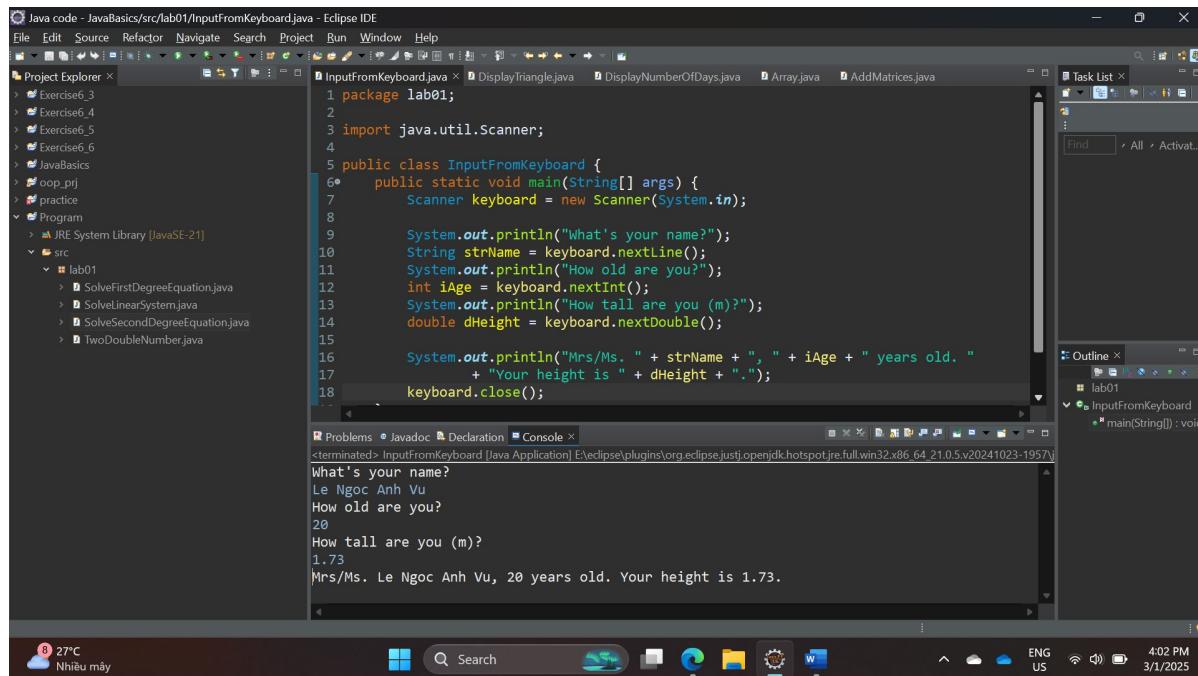
Note: We use the JavaBasics project for this exercise.

```

1 package lab01;
2
3 import java.util.Scanner;
4
5 public class InputFromKeyboard {
6•     public static void main(String[] args) {
7         Scanner keyboard = new Scanner(System.in);
8
9         System.out.println("What's your name?");
10        String strName = keyboard.nextLine();
11        System.out.println("How old are you?");
12        int iAge = keyboard.nextInt();
13        System.out.println("How tall are you (m)?");
14        double dHeight = keyboard.nextDouble();
15
16        System.out.println("Mrs/Ms. " + strName + ", " + iAge + " years old. "
17                            + "Your height is " + dHeight + ".");
18        keyboard.close();
19    }
20}

```

Executing the program:



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

E.g.  $n=5$ :

```

*
 ***
 *****
 ******
 ******

```

Note: You must create a new Java project for this exercise.

```

1 package lab01;
2
3 import java.util.Scanner;
4
5 public class DisplayTriangle {
6    public static void main(String[] args) {
7        Scanner read = new Scanner(System.in);
8        System.out.print("n=");
9        int n = read.nextInt();
10       read.close();
11       for(int i = 1; i <= n; i++) {
12           for(int j = 1; j <= n-i; j++) {
13               System.out.print(" ");
14           }
15           for(int j = 1; j<=2*i-1; j++) {
16               System.out.print("*");
17           }
18           for(int j = 1; j <= n-1; j++) {
19               System.out.print(" ");
20           }
21           System.out.print("\n");
22       }
23   }
24 }
25

```

### Executing the program:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a folder named "lab01" containing several Java files: SolveFirstDegreeEquation.java, SolveLinearSystem.java, SolveSecondDegreeEquation.java, and TwoDoubleNumber.java.
- Editor:** Displays the code for `DisplayTriangle.java`. The code reads an integer `n` from the user, then prints a triangle of asterisks where each row `i` contains  $2i-1$  asterisks.
- Console:** Shows the output of the program when run with `n=9`. The output is:

```

n=9
*
**
*****
*****
*****
*****
*****
*****
*****
*****
```
- Bottom Status Bar:** Shows system information including weather (27°C), search bar, taskbar icons, language (ENG US), date (3/1/2025), and time (4:06 PM).

**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.**

**Note:** You must create a new Java project for this exercise.

- The user can either enter a month in its full name, abbreviation, in 3 letters, or in number. To illustrate, the valid inputs of *January* are January, Jan., Jan, and 1.
- The user must enter a year in a non-negative number and enter all the digits. For instance, the valid input of year *1999* is only 1999, but not 99, “one thousand nine hundred ninety-nine”, or anything else.
- A year is either a common year of 365 days or a leap year of 366 days. Every year that is divisible by 4 is a leap year, except for years that are divisible by 100, but not by 400. For instance, the year 1800 is not a leap year, yet the year 2000 is a leap year. In a year, there are twelve months, which are listed in order as follows.

Month	January	February	March	April	May	June	July	August	September	October	November	December
Abbreviation	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
In 3 letters	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
In Number	1	2	3	4	5	6	7	8	9	10	11	12
Days of Month in Common Year	31	28	31	30	31	30	31	31	30	31	30	31
Days of Month in Leap Year	31	<b>29</b>	31	30	31	30	31	31	30	31	30	31

```

1 package lab01;
2
3 import java.util.Scanner;
4
5 public class DisplayNumberOfDays {
6     public DisplayNumberOfDays() {}
7
8     public boolean checkLeapYear(int year) {
9         if(year % 400 == 0) return true;
10        else if(year % 100 == 0 && year % 400 != 0) return false;
11        else if(year % 4 == 0) return true;
12        return false;
13    }
14
15    public int isValidMonth(String month) {
16        if(month.equals("January") || month.equals("Jan.")) ||
17            month.equals("Jan") || month.equals("1")) return 1;
18        if(month.equals("February") || month.equals("Feb.")) ||
19            month.equals("Feb") || month.equals("2")) return 2;
20        if(month.equals("March") || month.equals("Mar.")) ||
21            month.equals("Mar") || month.equals("3")) return 3;
22        if(month.equals("April") || month.equals("Apr.")) ||
23            month.equals("Apr") || month.equals("4")) return 4;
24        if(month.equals("May") || month.equals("5")) return 5;
25        if(month.equals("June") || month.equals("Jun") || month.equals("6")) return 6;
26        if(month.equals("July") || month.equals("Jul") || month.equals("7")) return 7

```

```

27     if(month.equals("August") || month.equals(" Aug.") ||  

28         month.equals("Aug") || month.equals("8")) return 8;  

29     if(month.equals("September") || month.equals("Sept.") ||  

30         month.equals("Sep") || month.equals("9")) return 9;  

31     if(month.equals("October") || month.equals("Oct.") ||  

32         month.equals("Oct") || month.equals("10")) return 10;  

33     if(month.equals("November") || month.equals("Nov.") ||  

34         month.equals("Nov") || month.equals("11")) return 11;  

35     if(month.equals("December") || month.equals("Dec.") ||  

36         month.equals("Dec") || month.equals("12")) return 12;  

37     return -1;  

38 }  

39  

40 public static void main(String[] args) {  

41     Scanner read = new Scanner(System.in);  

42     DisplayNumberOfDays obj = new DisplayNumberOfDays();  

43     String month;  

44     int m;  

45     int year;  

46     do {  

47         System.out.print("Enter a month: ");  

48         month = read.nextLine();  

49         m = obj.isValidMonth(month);  

50         if(m == -1) {  

51             System.out.println("Wrong Syntax! Please enter again!");  

52             continue;  

53         }  

54         System.out.print("Enter a year: ");  

55         year = read.nextInt();  

56         if(year < 0) {  

57             System.out.println("Enter again!");  

58             continue;  

59         }  

60         break;  

61     }while(true);  

62  

63     if(m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {  

64         System.out.println("The number of days of this month are 31");  

65     }  

66     else if(m == 4 || m == 6 || m == 9 || m == 11) {  

67         System.out.println("The number of days of this month are 30");  

68     }  

69     else if(m == 2 && obj.checkLeapYear(year)) {  

70         System.out.println("The number of days of this month are 29");  

71     }  

72     else if(m == 2 && !obj.checkLeapYear(year)) {  

73         System.out.println("The number of days of this month are 28");  

74     }  

75     read.close();  

76 }  

77 }  

78

```

Executing the program:

```

1 package lab01;
2
3 import java.util.Scanner;
4
5 public class DisplayNumberOfDays {
6     public DisplayNumberOfDays() {}
7
8     public boolean checkLeapYear(int year) {
9         if(year % 400 == 0) return true;
10        else if(year % 100 == 0 && year % 400 != 0) return false;
11        else if(year % 4 == 0) return true;
12        return false;
13    }
14
15    public int isValidMonth(String month) {
16        if(month.equals("January") || month.equals("Jan.")) ||
17            month.equals("Jan") || month.equals("JAN")) return 1;
18
19    }
20
21    public static void main(String[] args) {
22        ArrayList<Integer> array = new ArrayList<Integer>();
23        Scanner read = new Scanner(System.in);
24        System.out.print("Enter the number of elements in an array: ");
25        int n = read.nextInt();
26        int sum = 0;
27        for(int i = 1; i <= n; i++) {
28            int m = read.nextInt();
29            sum += m;
30            array.add(m);
31        }
32        Collections.sort(array);
33        System.out.println("Sorted array:");
34        for(int i = 0; i < n; i++) {
35            System.out.print(array.get(i) + " ");
36        }
37        System.out.println("\nThe sum of all elements in array: " + sum);
38        System.out.println("The average of all elements in array: " + (float)(sum)/n);
39        read.close();
40    }
41}

```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like Exercise6\_3, Exercise6\_4, Exercise6\_5, Exercise6\_6, JavaBasics, pop\_prj, practice, Program, JRE System Library [JavaSE-21], and src/lab01 containing SolveFirstDegreeEquation.java, SolveLinearSystem.java, SolveSecondDegreeEquation.java, and TwoDoubleNumber.java.
- DisplayNumberOfDays.java Content:**

```

1 package lab01;
2
3 import java.util.Scanner;
4
5 public class DisplayNumberOfDays {
6     public DisplayNumberOfDays() {}
7
8     public boolean checkLeapYear(int year) {
9         if(year % 400 == 0) return true;
10        else if(year % 100 == 0 && year % 400 != 0) return false;
11        else if(year % 4 == 0) return true;
12        return false;
13    }
14
15    public int isValidMonth(String month) {
16        if(month.equals("January") || month.equals("Jan.")) ||
17            month.equals("Jan") || month.equals("JAN")) return 1;
18
19    }
20
21    public static void main(String[] args) {
22        ArrayList<Integer> array = new ArrayList<Integer>();
23        Scanner read = new Scanner(System.in);
24        System.out.print("Enter the number of elements in an array: ");
25        int n = read.nextInt();
26        int sum = 0;
27        for(int i = 1; i <= n; i++) {
28            int m = read.nextInt();
29            sum += m;
30            array.add(m);
31        }
32        Collections.sort(array);
33        System.out.println("Sorted array:");
34        for(int i = 0; i < n; i++) {
35            System.out.print(array.get(i) + " ");
36        }
37        System.out.println("\nThe sum of all elements in array: " + sum);
38        System.out.println("The average of all elements in array: " + (float)(sum)/n);
39        read.close();
40    }
41}

```
- Console Output:**

```

Enter a month: february
Wrong Syntax! Please enter again!
Enter a month: fbr
Wrong Syntax! Please enter again!
Enter a month: Feb.
Enter a year: 1900
The number of days of this month are 28

```
- Outline View:** Shows the class structure with methods checkLeapYear(int), isValidMonth(String), and main(String[]).
- Task List:** Shows a single task labeled "Activat...".
- Bottom Bar:** Shows system status including weather (27°C), search bar, and system icons.

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

**Note:** You must create a new Java project for this exercise.

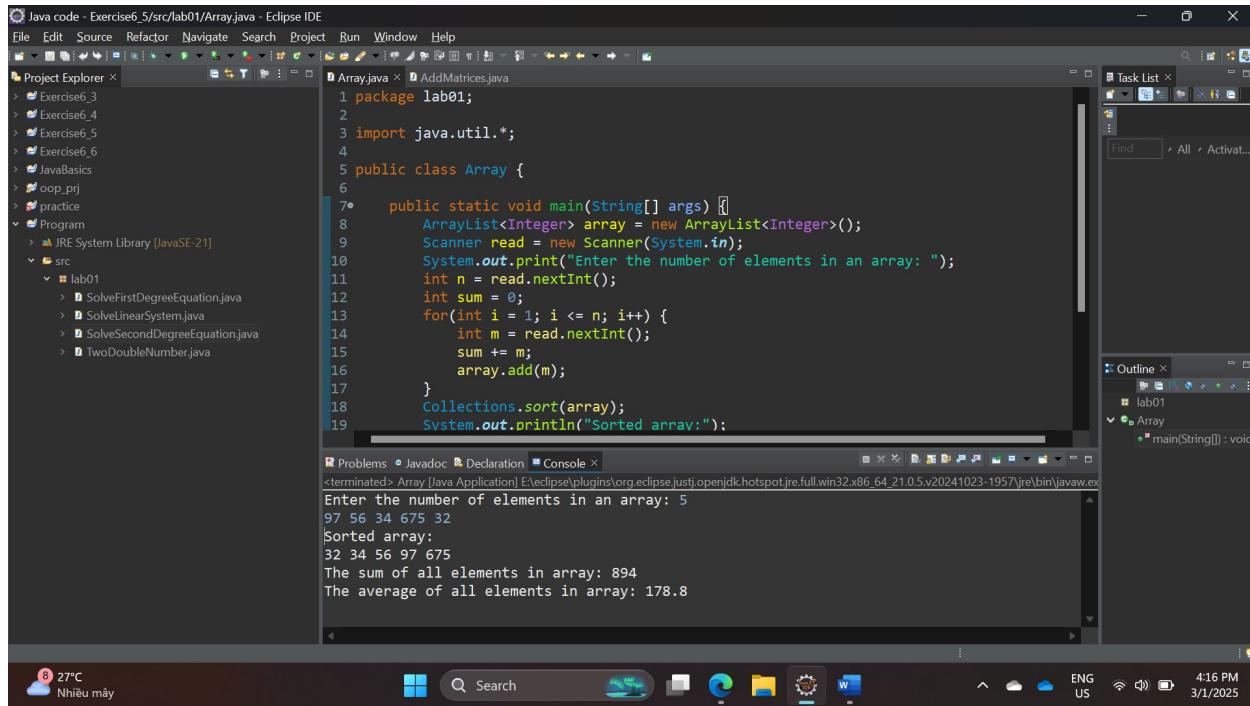
- The array can be entered by the user or a constant.

```

1 package lab01;
2
3 import java.util.*;
4
5 public class Array {
6
7     public static void main(String[] args) {
8         ArrayList<Integer> array = new ArrayList<Integer>();
9         Scanner read = new Scanner(System.in);
10        System.out.print("Enter the number of elements in an array: ");
11        int n = read.nextInt();
12        int sum = 0;
13        for(int i = 1; i <= n; i++) {
14            int m = read.nextInt();
15            sum += m;
16            array.add(m);
17        }
18        Collections.sort(array);
19        System.out.println("Sorted array:");
20        for(int i = 0; i < n; i++) {
21            System.out.print(array.get(i) + " ");
22        }
23        System.out.println("\nThe sum of all elements in array: " + sum);
24        System.out.println("The average of all elements in array: " + (float)(sum)/n);
25        read.close();
26    }
27 }

```

## Executing the program:



```
Java code - Exercise6.5/src/lab01/Array.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Array.java X AddMatrices.java
1 package lab01;
2
3 import java.util.*;
4
5 public class Array {
6
7     public static void main(String[] args) {
8         ArrayList<Integer> array = new ArrayList<Integer>();
9         Scanner read = new Scanner(System.in);
10        System.out.print("Enter the number of elements in an array: ");
11        int n = read.nextInt();
12        int sum = 0;
13        for(int i = 1; i <= n; i++) {
14            int m = read.nextInt();
15            sum += m;
16            array.add(m);
17        }
18        Collections.sort(array);
19        System.out.println("Sorted array:");
20    }
21 }
```

Console output:

```
<terminated> Array [Java Application] E:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe
Enter the number of elements in an array: 5
97 56 34 675 32
Sorted array:
32 34 56 97 675
The sum of all elements in array: 894
The average of all elements in array: 178.8
```

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

Note: You must create a new Java project for this exercise.

- The matrices can be entered by the user or constants.

```
1 package lab01;
2
3 import java.util.*;
4
5 public class AddMatrices {
6     public static void main(String[] args) {
7         Scanner read = new Scanner(System.in);
8         System.out.print("Enter size of the matrices: ");
9         int n = read.nextInt();
10        int[][] A = new int[n][n];
11        int[][] B = new int[n][n];
12        int[][] C = new int[n][n];
13        System.out.printf("Enter matrix A of size %dx%d:\n", n, n);
14        for(int i = 0; i < n; i++) {
15            for(int j = 0; j < n; j++) A[i][j] = read.nextInt();
16        }
17        System.out.printf("Enter matrix B of size %dx%d:\n", n, n);
18        for(int i = 0; i < n; i++) {
19            for(int j = 0; j < n; j++) B[i][j] = read.nextInt();
20        }
21        read.close();
```

```

22         for(int i = 0; i < n; i++) {
23             for(int j = 0; j < n; j++) C[i][j] = A[i][j] + B[i][j];
24         }
25         System.out.println("Sum of two matrices A and B is matrix C:");
26         for(int i = 0; i < n; i++) {
27             for(int j = 0; j < n; j++) System.out.print(C[i][j] + " ");
28             System.out.print("\n");
29         }
30     }
31 }

```

Executing the program:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows multiple Java projects like Exercise6\_3, Exercise6\_4, Exercise6\_5, Exercise6\_6, JavaBasics, oop\_prj, practice, Program, and lab01.
- Code Editor:** Displays the `AddMatrices.java` file with Java code for matrix addition.
- Console:** Shows the execution of the program. It prompts for matrix sizes (3x3), reads matrix A values (12 3 54; 4 0 21; 78 6 0), reads matrix B values (43 6 0; 122 32 5; 0 87 5), and prints the result matrix C (55 9 54; 126 32 26; 78 93 5).
- Task List, Outline, Problems:** Standard Eclipse toolbars.
- System Tray:** Shows weather (27°C), network, battery, and system status.

```

Java code - Exercise6_6/src/lab01/AddMatrices.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer ×
> Exercise6_3
> Exercise6_4
> Exercise6_5
> Exercise6_6
> JavaBasics
> oop_prj
> practice
< Program
> JRE System Library [JavaSE-21]
< src
  < lab01
    > SolveFirstDegreeEquation.java
    > SolveLinearSystem.java
    > SolveSecondDegreeEquation.java
    > TwoDoubleNumber.java
AddMatrices.java ×
1 import java.util.*;
2
3 public class AddMatrices {
4     public static void main(String[] args) {
5         Scanner read = new Scanner(System.in);
6         System.out.print("Enter size of the matrices: ");
7         int n = read.nextInt();
8         int[][] A = new int[n][n];
9         int[][] B = new int[n][n];
10        int[][] C = new int[n][n];
11        System.out.printf("Enter matrix A of size %dx%d:\n", n, n);
12        for(int i = 0; i < n; i++) {
13            for(int j = 0; j < n; j++) A[i][j] = read.nextInt();
14
15
16        }
17        System.out.print("Enter matrix B of size %dx%d:\n", n, n);
18        for(int i = 0; i < n; i++) {
19            for(int j = 0; j < n; j++) B[i][j] = read.nextInt();
20
21
22        }
23        for(int i = 0; i < n; i++) {
24            for(int j = 0; j < n; j++) C[i][j] = A[i][j] + B[i][j];
25        }
26        System.out.println("Sum of two matrices A and B is matrix C:");
27        for(int i = 0; i < n; i++) {
28            for(int j = 0; j < n; j++) System.out.print(C[i][j] + " ");
29            System.out.print("\n");
30        }
31    }
}

```

Enter size of the matrices: 3  
Enter matrix A of size 3x3:  
12 3 54  
4 0 21  
78 6 0  
Enter matrix B of size 3x3:  
43 6 0  
122 32 5  
0 87 5  
Sum of two matrices A and B is matrix C:  
55 9 54  
126 32 26  
78 93 5