

## Lab 01. Environment Setup & Java Basics

**IT3103 – 7850868 – Thực hành Lập Trình Hướng Đối Tượng**

**Họ và tên: Cao Việt Bách**

**MSSV: 202416777**

**Bài 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)`**

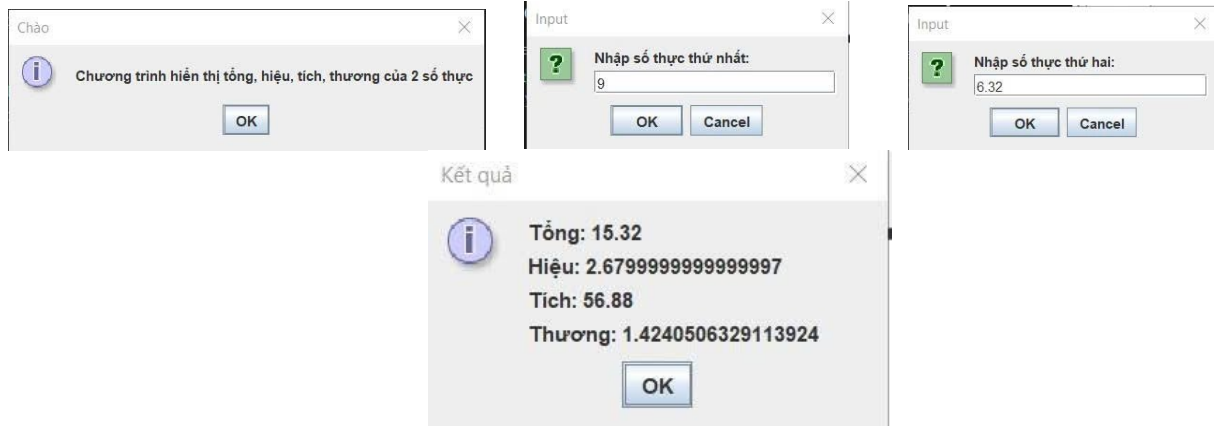
- Check the divisor of the division
- Don't forget to add and commit your work using '`git add .`' and '`git commit -m <message>`' command

## 1. Code

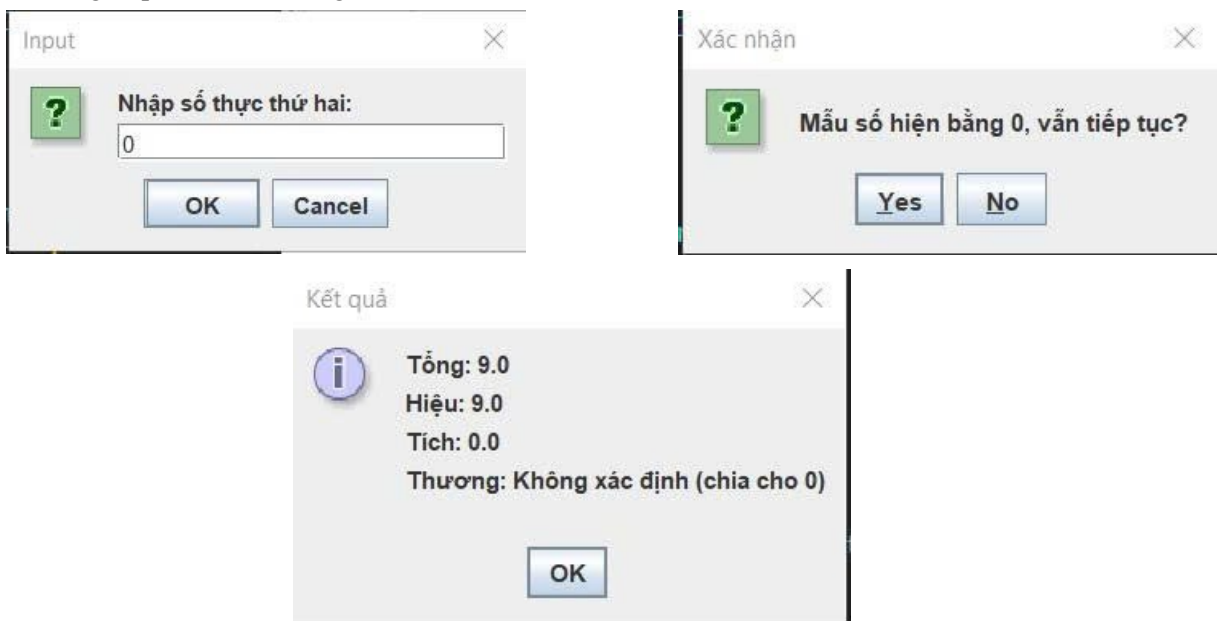
```
1 import javax.swing.JOptionPane;
2 import javax.swing.UIManager;
3 public class exercise2_2_5{
    Run | Debug
4     public static void main(String[] args) {
5         UIManager.put(key: "OptionPane.isAlwaysOnTop", value: true);
6         JOptionPane.showMessageDialog(parentComponent: null,
7             message: "Chương trình hiển thị tổng, hiệu, tích, thương của 2 số thực",
8             title: "Chào",
9             JOptionPane.INFORMATION_MESSAGE
10        );
11        double[] input = new double[2];
12        input[0] = getInput(order: "nhất");
13        input[1] = getInput(order: "hai");
14        String result = calculateInput(input[0], input[1]);
15        JOptionPane.showMessageDialog(parentComponent: null, result,
16            title: "Kết quả", JOptionPane.INFORMATION_MESSAGE);
17    }
18    public static double getInput(String order){
19        double num = 0.0;
20        while(true){
21            String strNum = JOptionPane.showInputDialog(parentComponent: null,
22                "Nhập số thực thứ " + order + " :");
23            try {
24                num = Double.parseDouble(strNum);
25                if (order.equals(anObject: "hai") && num == 0.0) {
26                    int response = JOptionPane.showConfirmDialog(parentComponent: null,
27                        message: "Mẫu số hiện bằng 0, vẫn tiếp tục?",
28                        title: "Xác nhận",JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
29                    if (response == JOptionPane.NO_OPTION)
30                        continue;
31                }
32                break;
33            }
34            catch (NumberFormatException e) {
35                JOptionPane.showMessageDialog(parentComponent: null,
36                    message: "Giá trị nhập vào không hợp lệ. Vui lòng nhập lại.",
37                    title: "Lỗi", JOptionPane.ERROR_MESSAGE);
38            }
39        }
40        return num;
41    }
42
43    private static String calculateInput(double num1, double num2){
44        double sum = num1 + num2;
45        double difference = num1 - num2;
46        double product = num1 * num2;
47        String quotient;
48        if (num2 != 0) quotient = String.valueOf(num1 / num2);
49        else quotient= "Không xác định (chia cho 0)\n";
50        return "Tổng: " + sum + "\n" +
51            "Hiệu: " + difference + "\n" +
52            "Tích: " + product + "\n" +
53            "Thương: " + quotient + "\n";
54    }
55 }
```

## 2. Run

- Trường hợp số thứ hai khác 0



- o Chương trình chạy đúng
- Trường hợp số thứ hai bằng 0



- o Như vậy chương trình đã không thực hiện phép chia cho 0 và in ra “**Không xác định**”
- Trường hợp giá trị nhập vào không hợp lệ



- o Chương trình sẽ yêu cầu nhập lại

### Bài 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.

Don't forget to add and commit your work using 'git add .' and 'git commit -m <message>' command

#### - 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.

#### **Hint:**

Use the following determinants:

$$\begin{aligned} D &= \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12} \\ D_1 &= \begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix} = b_1a_{22} - b_2a_{12} \\ D_2 &= \begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix} = a_{11}b_2 - a_{21}b_1 \end{aligned}$$

#### - 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.

#### **Hint:**

Use the discriminant  $\Delta = b^2 - 4ac$

After completing the code in section 2, you should push all of your changes, including assignment 2.2.1 to 2.2.6 to the **master** branch of the valid repository you have created.

**Hint:** You should use "git push origin <name of the branch that you want to push to the remote repository>", in this case is "**master**", to push all of your works to the **master** branch.

## 1. Code

### a. Menu lựa chọn

```
1 import javax.swing.JOptionPane;
2 public class exercise2_2_6 {
    Run | Debug
3     public static void main(String[] args) {
4         String[] options = {
5             "Giải phương trình bậc nhất ( $ax + b = 0$ )",
6             "Giải hệ phương trình bậc nhất 2 ẩn",
7             "Giải phương trình bậc hai ( $ax^2 + bx + c = 0$ )",
8             "Exit"
9         };
10        while (true) {
11            int choice = JOptionPane.showOptionDialog(parentComponent: null,
12                message: "Chọn loại phương trình:",
13                title: "Trình giải phương trình",
14                JOptionPane.DEFAULT_OPTION,
15                JOptionPane.QUESTION_MESSAGE,
16                icon: null,
17                options,
18                options[0]
19            );
20            switch(choice) {
21                case -1:
22                    System.exit(status: 0);
23                case 0: solveLinearEquation();
24                    break;
25                case 1: solveLinearSystem();
26                    break;
27                case 2: solveQuadraticEquation();
28                    break;
29                case 3: // Exit
30                    JOptionPane.showMessageDialog(parentComponent: null, message: "Thoát chương trình.");
31                    System.exit(status: 0);
32            }
33        }
34    }
35 }
36
```

b. Phương thức giải phương trình bậc nhất

```
53 // Solve  $ax + b = 0$ 
54 private static void solveLinearEquation() {
55     double a;
56     while (true) {
57         a = getInput(heso: "a (a ≠ 0)");
58         if (a != 0) break;
59         JOptionPane.showMessageDialog(parentComponent: null, message: "Hệ số phải khác 0.");
60     }
61     double b = getInput(heso: "b");
62     double x = -b / a;
63     JOptionPane.showMessageDialog(parentComponent: null, String.format(format: "Nghiem: x = %.4f", x));
64 }
```

c. Phương thức giải hệ phương trình bậc nhất hai ẩn

```
66 // Solve system:
67 //  $a_{11}x_1 + a_{12}x_2 = b_1$ 
68 //  $a_{21}x_1 + a_{22}x_2 = b_2$ 
69 private static void solveLinearSystem() {
70     double a11 = getInput(heso: "a11");
71     double a12 = getInput(heso: "a12");
72     double b1 = getInput(heso: "b1");
73     double a21 = getInput(heso: "a21");
74     double a22 = getInput(heso: "a22");
75     double b2 = getInput(heso: "b2");
76
77     double D = a11 * a22 - a12 * a21;
78     double Dx = b1 * a22 - b2 * a12;
79     double Dy = a11 * b2 - a21 * b1;
80
81     if (D != 0) {
82         double x1 = Dx / D;
83         double x2 = Dy / D;
84         JOptionPane.showMessageDialog(parentComponent: null, String.format(
85             format: "Nghiem duy nhât:\nx1 = %.4f\nx2 = %.4f", x1, x2));
86     } else {
87         if (Dx == 0 && Dy == 0) {
88             JOptionPane.showMessageDialog(parentComponent: null, message: "Hệ phương trình vô số nghiệm.");
89         } else {
90             JOptionPane.showMessageDialog(parentComponent: null, message: "Hệ phương trình vô nghiệm.");
91         }
92     }
93 }
```



d. Phương thức giải phương trình bậc hai

```
95 // Solve  $ax^2 + bx + c = 0$ 
96 private static void solveQuadraticEquation() {
97     double a;
98     while (true) {
99         a = getInput(heso: "a (a ≠ 0)");
100         if (a != 0) break;
101         JOptionPane.showMessageDialog(parentComponent: null, message: "Hệ số phải khác 0.");
102     }
103     double b = getInput(heso: "b");
104     double c = getInput(heso: "c");
105
106     double delta = b * b - 4 * a * c;
107     if (delta > 0) {
108         double x1 = (-b + Math.sqrt(delta)) / (2 * a);
109         double x2 = (-b - Math.sqrt(delta)) / (2 * a);
110         JOptionPane.showMessageDialog(parentComponent: null, String.format(
111             format: "Hai nghiệm phân biệt:\nx1 = %.4f\nx2 = %.4f", x1, x2));
112     } else if (delta == 0) {
113         double x = -b / (2 * a);
114         JOptionPane.showMessageDialog(parentComponent: null, String.format(
115             format: "Nghiệm kép:\nx = %.4f", x));
116     } else {
117         JOptionPane.showMessageDialog(parentComponent: null, message: "Không có nghiệm thực.");
118     }
119 }
```

e. Các phương thức hỗ trợ rút gọn khác

```
37 // get valid double input
38 private static double getInput(String heso) {
39     while (true) {
40         String input = JOptionPane.showInputDialog(parentComponent: null, "Nhập hệ số " + heso + ":");
41         if (input == null) {
42             JOptionPane.showMessageDialog(parentComponent: null, message: "Bãi bai!");
43             System.exit(status: 0);
44         }
45         try {
46             return Double.parseDouble(input.trim());
47         } catch (NumberFormatException e) {
48             JOptionPane.showMessageDialog(parentComponent: null, message: "Không hợp lệ. Vui lòng nhập lại.");
49         }
50     }
51 }
```

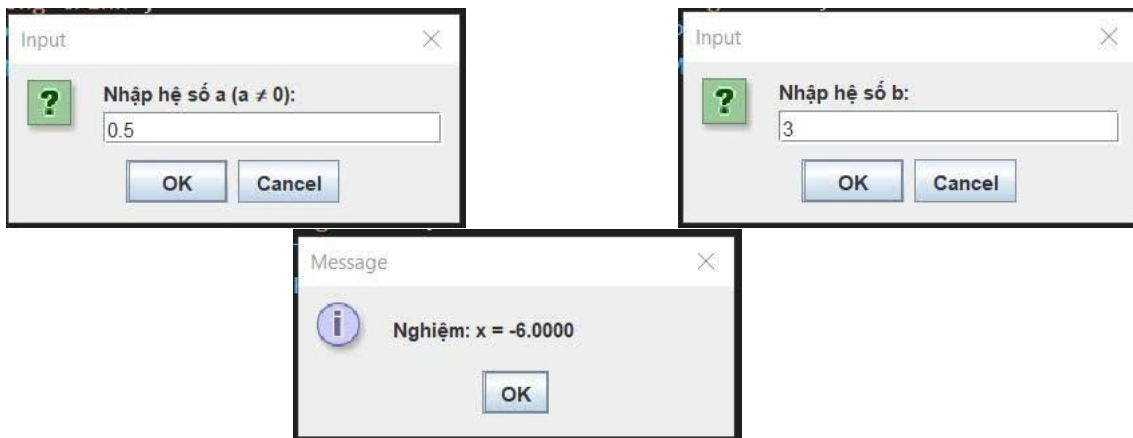
2. Run

a. Đầu tiên cửa sổ lựa chọn hiện ra



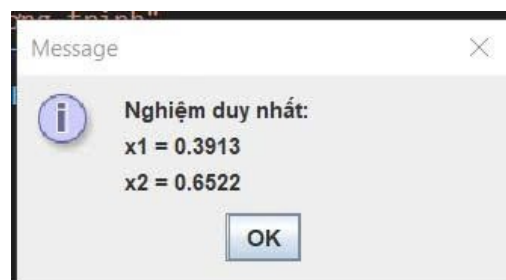
b. Kiểm thử

i. Giải phương trình bậc nhất



ii. Giải hệ phương trình bậc nhất hai ẩn

Nhập vào hệ  $\begin{cases} x + 4y = 3 \\ 7x + 5y = 6 \end{cases}$  ta được kết quả chương trình chạy ra là



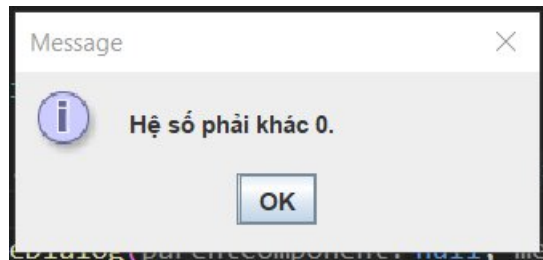
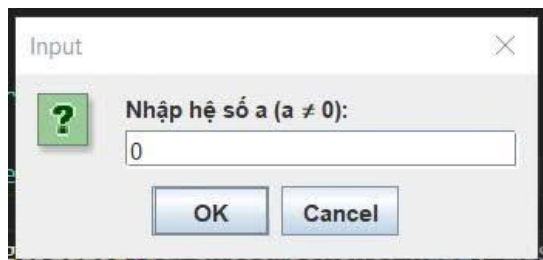


iii. Giải phương trình bậc hai

Nhập vào phương trình  $x^2 + 6x + 9 = 0$  ta được kết quả chương trình chạy ra là



iv. Các trường hợp đặc biệt, chương trình hoạt động đúng như đề yêu cầu !

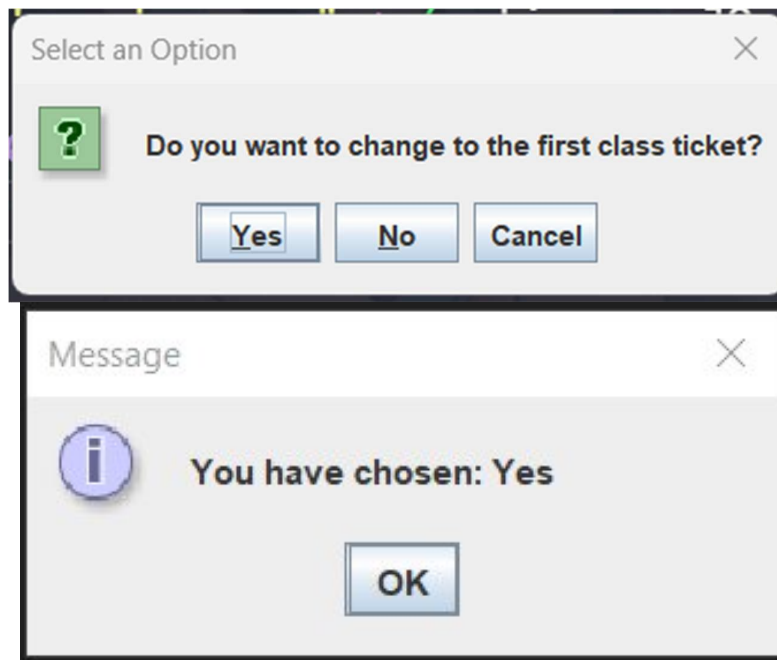


**Bài 6.1.** Write, compile and run the ChoosingOption program

**1. Code**

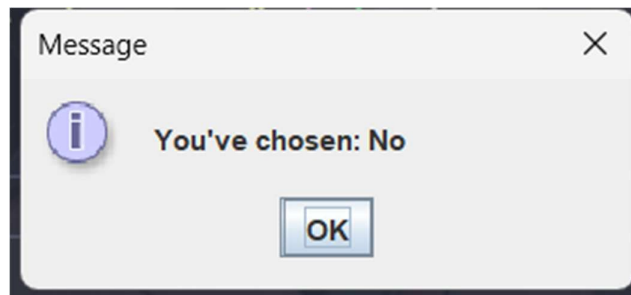
```
1 // This is for exercise 6.1
2
3 import javax.swing.JOptionPane;
4
5 public class Choosing_Option {
6     Run | Debug
7     public static void main(String[] args) {
8         int option = JOptionPane.showConfirmDialog(parentComponent: null,
9             message: "Do you want to change to the first class ticket?"
10        );
11        JOptionPane.showMessageDialog(parentComponent: null,
12            "You have chosen: " + (option == JOptionPane.YES_OPTION? "Yes": "No")
13        );
14        System.exit(status: 0);
15    }
16 }
```

**2. Run**



### 3. Question

- What happens if users choose “Cancel”?
  - o Trong chương trình, JOptionPane.showConfirmDialog() trả về một trong các giá trị
    - JOptionPane.YES\_OPTION (0) nếu chọn Yes
    - JOptionPane.NO\_OPTION (1) nếu chọn No
    - JOptionPane.CANCEL\_OPTION (2) nếu chọn Cancel
  - o Tuy nhiên chương trình chỉ kiểm tra JOptionPane.YES\_OPTION và coi như các lựa chọn khác là No, khi đó nếu người dùng chọn Cancel thì chương trình sẽ hiển thị “You’ve chosen: No”



- How to customize the options to users, e.g. only two options: “Yes” and “No”, OR “I do” and “I don’t” (Suggestion: Use Javadocs or using Eclipse/Netbean IDE help).
  - o Mặc định JOptionPane.showConfirmDialog() hiển thị 3 lựa chọn: Yes, No, Cancel.
  - o Để giới hạn chỉ còn hai lựa chọn “Yes”, “No” :
    - Khai báo cụ thể kiểu cho option là `JOptionPane.YES_NO_OPTION`

```
1 // This is for exercise 6.1
2
3 import javax.swing.JOptionPane;
4
5 public class Choosing_Option {
6     Run | Debug
7     public static void main(String[] args) {
8         int option = JOptionPane.showConfirmDialog(parentComponent: null,
9             message: "Do you want to change to the first class ticket?", title: "Confirm", JOptionPane.YES_NO_OPTION
10        );
11        JOptionPane.showMessageDialog(parentComponent: null,
12            "You have chosen: " + (option == JOptionPane.YES_OPTION? "Yes": "No")
13        );
14        System.exit(status: 0);
15    }
16 }
```

- Chạy thử



- Hiện thị “I do”, “I don’t”
  - Để tùy biến tên nút (ví dụ “**I do**” và “**I don’t**”) thay vì **Yes/No**, ta dùng UIManager để ghi đè nhãn của YES/NO

```
1 // This is for exercise 6.1
2 import javax.swing.UIManager;
3 import javax.swing.JOptionPane;
4
5 public class Choosing_Option {
6     Run | Debug
7     public static void main(String[] args) {
8         UIManager.put(key: "OptionPane.yesButtonText", value: "I do");
9         UIManager.put(key: "OptionPane.noButtonText", value: "I don't");
10        int option = JOptionPane.showConfirmDialog(parentComponent: null,
11        message: "Do you want to change to the first class ticket?",title: "Confirm",JOptionPane.YES_NO_OPTION
12        );
13        JOptionPane.showMessageDialog(parentComponent: null,
14        "You have chosen: " + (option == JOptionPane.YES_OPTION? "Yes": "No")
15        );
16        System.exit(status: 0);
17    }
18 }
```

- Chạy thử



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

Code – Compile – Run

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Admin\OOP.Lab.20251.757055.202416777.Cao-Vi-t-B-ch\OtherProjects\hust\soict\dsai\Lab01\code> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\OOP.Lab.20251.757055.202416777.Cao-Vi-t-B-ch\OtherProjects\hust\soict\dsai\Lab01\code\bin' 'InputFromKeyboard'
What's your name?Bách
How old are you?
19
How tall are you (in meters)?
1.72
Mr/Mrs B?ch, 19 years old. Your height is 1.72m.
PS C:\Users\Admin\OOP.Lab.20251.757055.202416777.Cao-Vi-t-B-ch\OtherProjects\hust\soict\dsai\Lab01\code> 
```



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

E.g. n = 5:

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

#### - Code & Run

```
src > J pyramid6_3.java > ...
1  import java.util.Scanner;
2  public class pyramid6_3 {
3      Run | Debug
4      public static void main(String[] args) {
5          int height;
6          Scanner scanner = new Scanner(System.in);
7          System.out.println(x: "Enter the height of the pyramid:");
8          height = scanner.nextInt();
9          scanner.close();
10
11         for (int i = 1; i <= height; i++) {
12             // Print leading spaces
13             String spaces = " ".repeat(height - i); // Java 11 and above
14             String stars = "*".repeat(2 * i - 1); // Java 11 and above
15             System.out.println(spaces + stars);
16         }
17     }
18 }
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Admin\OOP.Lab.20251.757055.202416777.Cao-Vi-t-B-ch\OtherProjects\hust\soict\dsai\Lab01\code> & 'C:\Program Fi
-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\OOP.Lab.20251.757055.202416777.Cao-Vi-t-B-ch\OtherProjects\
'pyramid6_3'
Enter the height of the pyramid:
4
 *
 ***
 *****
*****
PS C:\Users\Admin\OOP.Lab.20251.757055.202416777.Cao-Vi-t-B-ch\OtherProjects\hust\soict\dsai\Lab01\code> □
```

**Bài 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 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

- **Code**

```
//exercise 6.4
import java.util.Scanner;
public class DayMonthYear {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int month, year = -1;
        while (true) {
            //enter month
            System.out.print("Enter month: ");
            String monthInput = scanner.nextLine().trim().toLowerCase();
            month = parseMonth(monthInput);
            if (month == -1) {
                System.out.println("Invalid month. Please try again.");
                continue;
            }
            //enter year
            System.out.print("Enter year (AD, in full): ");
            String yearInput = scanner.nextLine().trim();
            if (!yearInput.matches("\\d+")) {
                System.out.println("Invalid year. Please try again.");
                continue;
            }
            year = Integer.parseInt(yearInput);
            if (year < 0) {
                System.out.println("Invalid year. Please try again.");
                continue;
            }
            break;
        }
        int days = getDaysInMonth(month, year);
        System.out.println("The month you type has " + days + "days");
        scanner.close();
    }

    private static int parseMonth(String input) {

        switch(input){
            case "january": case "jan." :case "jan" :case "1":return 1;
            case "february": case "feb." :case "feb" :case "2":return 2;
            case "march": case "mar." :case "mar" :case "3":return 3;
            case "april": case "apr." :case "apr" :case "4": return 4;
            case "may": case "may." :case "5":return 5;
            case "june": case "jun." :case "jun" :case "6": return 6;
            case "july": case "jul." :case "jul" :case "7": return 7;
            case "august": case "aug" :case "aug." :case "8": return 8;
            case "september": case "sep." :case "sep" :case "9": return 9;
            case "october": case "oct." :case "oct" :case "10": return 10;
            case "november": case "nov." :case "nov" :case "11":return 11;
            case "december": case "dec." :case "dec" :case "12":return 12;
            default: return -1;
        }
    }

    private static int getDaysInMonth(int month, int year) {
        switch (month) {
            case 2:
                return isLeapYear(year) ? 29 : 28;
            case 4: case 6: case 9: case 11:
                return 30;
            default:
                return 31;
        }
    }

    private static boolean isLeapYear(int year) {
        return (year % 4 == 0) && ((year % 100 != 0) || (year % 400 == 0));
    }

}
```

- Test 1

```
Enter month: 2  
Enter year (AD, in full): 2016  
The month you type has 29 days
```

- Test 2

```
Enter month: 3  
Enter year (AD, in full): 3021  
The month you type has 31 days
```

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

- **Code**

```
import java.util.Arrays;
public class NumericArrays {
    public static void main(String[] args) {
        int[] numbers = {1789, 2035, 1899, 1456, 2013};
        double sum = 0;
        for (int num : numbers) sum += num;
        Arrays.sort(numbers);
        System.out.println("Sorted array: " + Arrays.toString(numbers));
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + sum / numbers.length);
    }
}
```

- **Run**

```
Original Array: [1789, 2035, 1899, 1456, 2013]
Sorted Array: [1456, 1789, 1899, 2013, 2035]
Sum of Array: 9192
Average of Array: 1838.4
```

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

- Code

```
public class AdddaMatrix {
    public static void main(String[] args) {
        int[][] matrixA = {
            {1, 2, 3},
            {4, 5, 6},
            {7, 8, 9}
        };
        int[][] matrixB = {
            {9, 8, 7},
            {6, 5, 4},
            {3, 2, 1}
        };
        if(issameSize(matrixA, matrixB)){
            printMatrix(matrixA, "Matrix A");
            printMatrix(matrixB, "Matrix B");
            printMatrix(addMatrices(matrixA, matrixB), "Sum Matrix");
        }

    }

    public static boolean issameSize(int[][] a, int[][] b){
        if (a.length != b.length || a[0].length != b[0].length) return false;
        return true;
    }

    public static int[][] addMatrices(int[][] a, int[][] b) {
        int rows = a.length;
        int cols = a[0].length;
        int[][] sum = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) sum[i][j] = a[i][j] + b[i][j];
        }
        return sum;
    }

    public static void printMatrix(int[][] matrix, String name) {
        System.out.println(name + ":");
        for (int[] row : matrix) {
            for (int val : row) System.out.print(val + " ");
            System.out.println();
        }
        System.out.println();
    }
}
```



- Run

Matrix A:

1 2 3

4 5 6

7 8 9

Matrix B:

9 8 7

6 5 4

3 2 1

Sum Matrix:

10 10 10

10 10 10

10 10 10