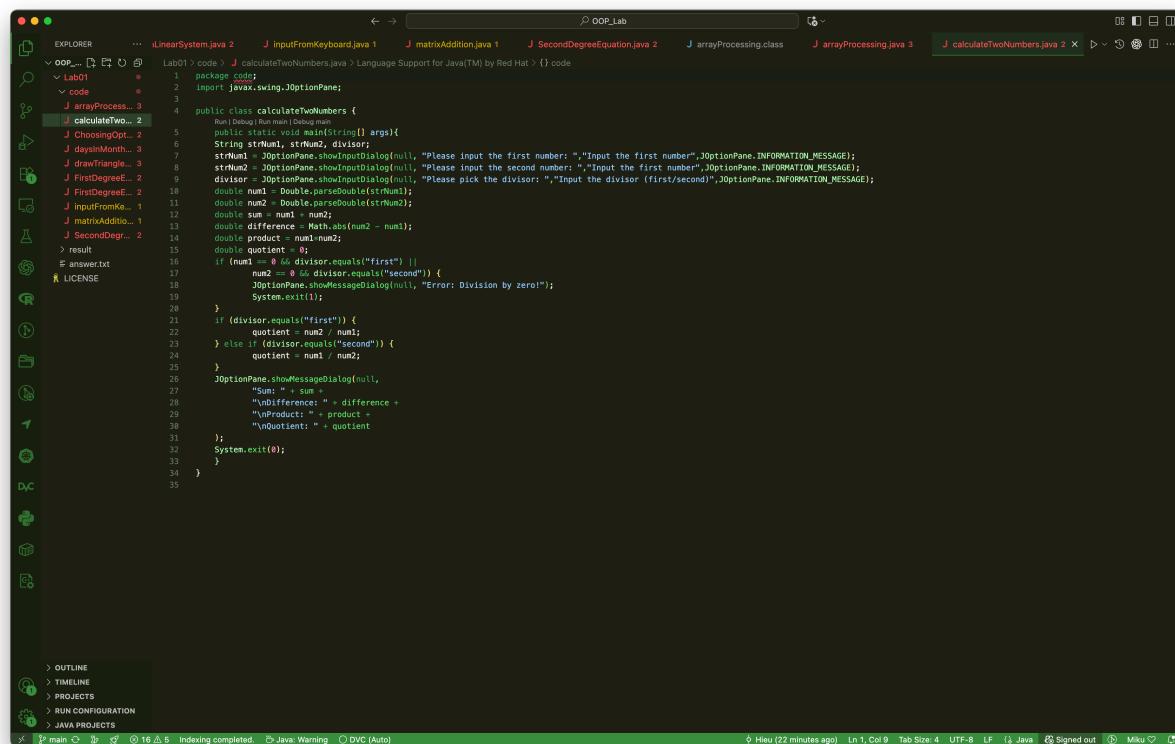


# Lab01 Report

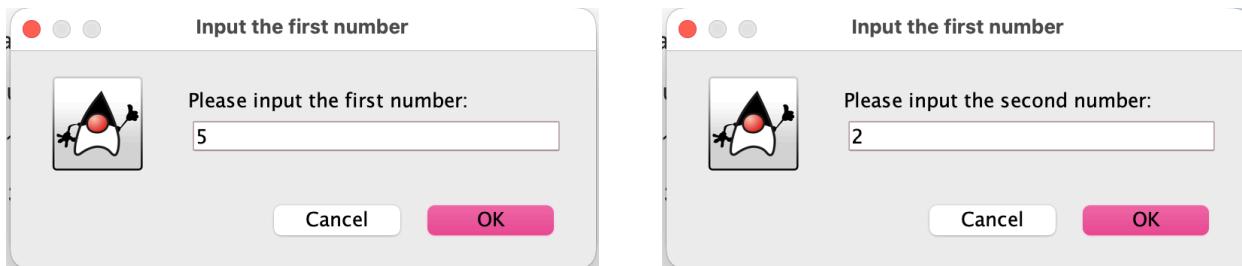
## 2.2.5

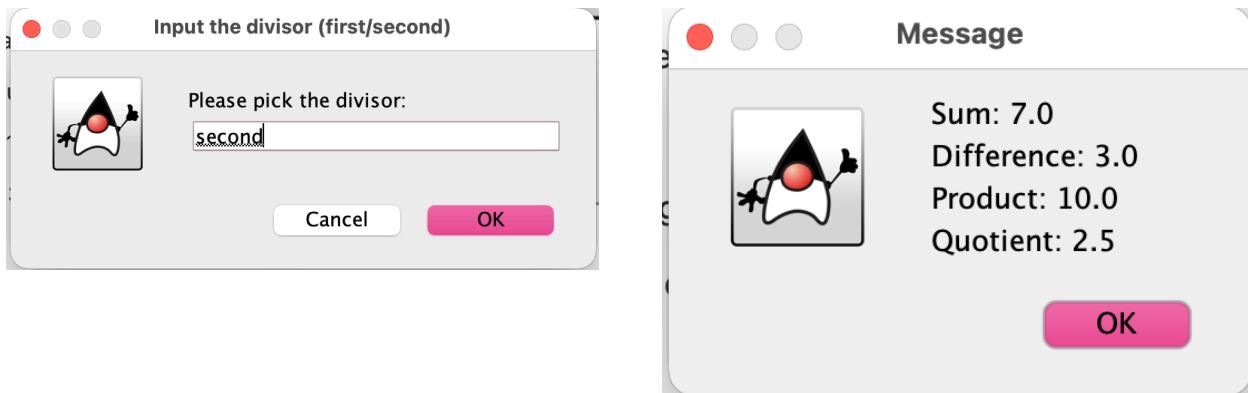


The screenshot shows a Java code editor with several tabs open. The active tab contains the following code:

```
package code;
import javax.swing.JOptionPane;

public class calculateTwoNumbers {
    public static void main (String[] args){
        String strNum1, strNum2, divisor;
        strNum1 = JOptionPane.showInputDialog(null, "Please input the first number:","");
        strNum2 = JOptionPane.showInputDialog(null, "Please input the second number:","");
        divisor = JOptionPane.showInputDialog(null, "Please pick the divisor:","");
        double num1 = Double.parseDouble(strNum1);
        double num2 = Double.parseDouble(strNum2);
        double sum = num1 + num2;
        double difference = Math.abs(num2 - num1);
        double product = num1 * num2;
        double quotient = 0;
        if (num1 == 0 & divisor.equals("first")){
            num2 = 0 & divisor.equals("second")){
                quotient = num1 / num2;
            }
        JOptionPane.showMessageDialog(null,
            "sum: " + sum +
            "\ndifference: " + difference +
            "\nproduct: " + product +
            "\nquotient: " + quotient
        );
        System.exit(0);
    }
}
```





## 2.2.6

Screenshot of an IDE showing multiple Java files in the Explorer panel and their corresponding code in the editor. The code includes various methods for solving equations and performing matrix operations.

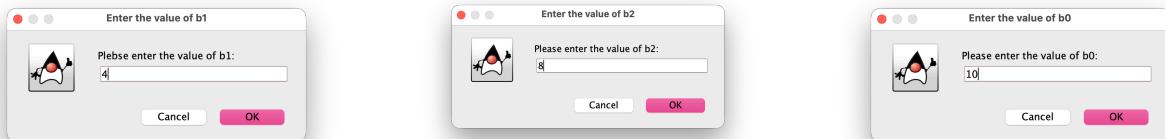
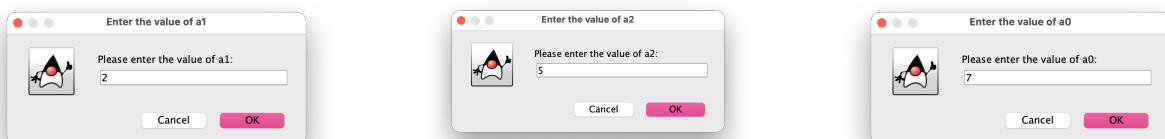
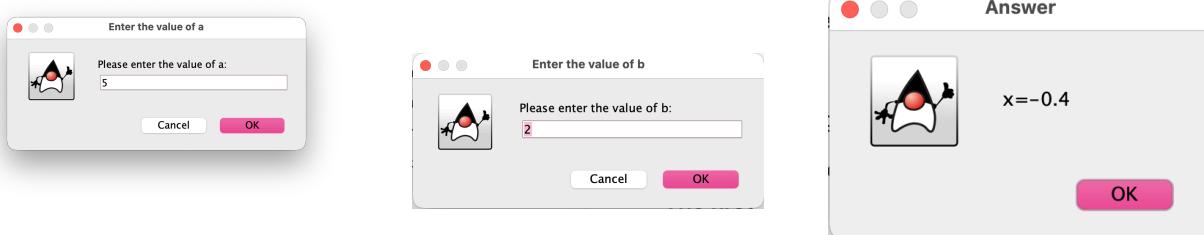
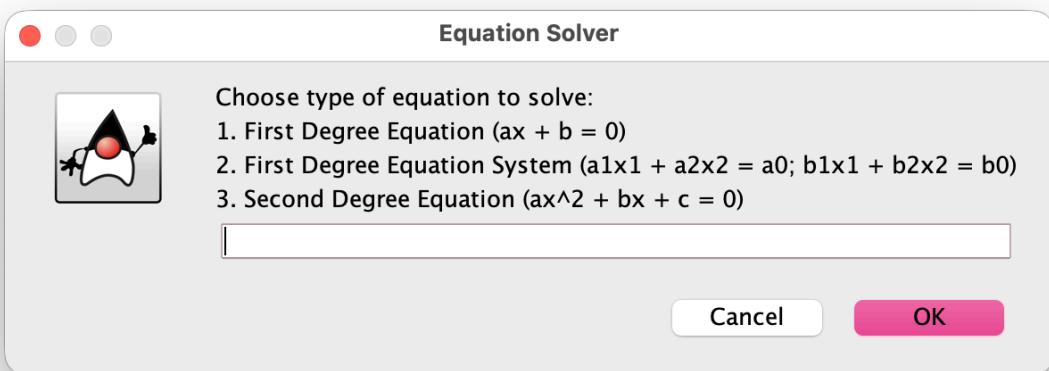
```

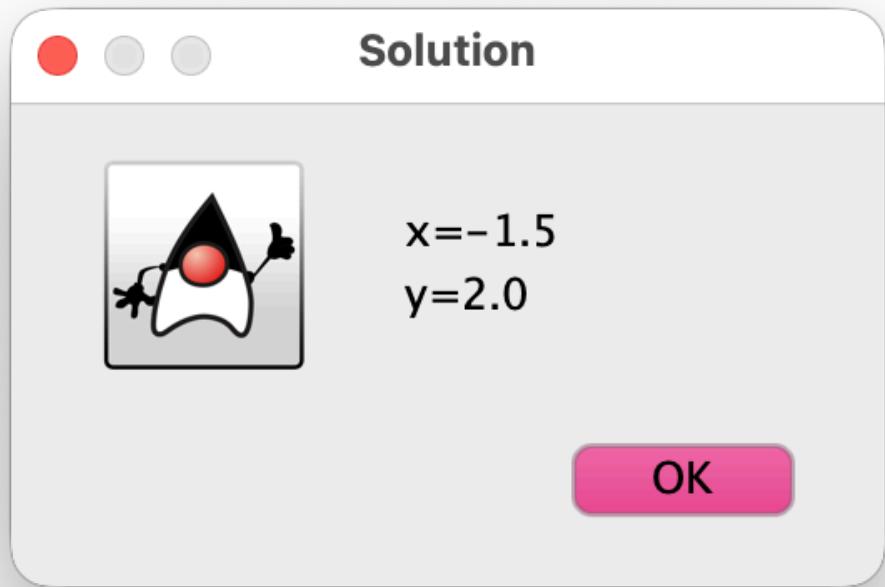
OOP_Lab
Lab01 > code > J_inputFromKeyboard.java 1 J_matrixAddition.java 1 J_arrayProcessing.class J_arrayProcessing.java 3 J_calculateTwoNumbers.java 2 J_solveEquations.java 5, U ...
J_drawTriangle.java 3 J_inputFromKeyboard.java 1 J_matrixAddition.java 1 J_arrayProcessing.class J_arrayProcessing.java 3 J_calculateTwoNumbers.java 2 J_solveEquations.java 5, U ...
J_arrayProcess... 3 J_calculateTwo... 2 J.ChoosingOpt... 2 J.daysInMonth... 3 J.drawTriangle... 3 J.inputFromKe... 1 J.matrixAdditio... 1 J.solveEqua... 5, U ...
result result.pdf # answer.txt LICENSE

public class EquationSolver {
    public static void main(String[] args) {
        String choiceStr = JOptionPane.showInputDialog(
            null,
            "Choose type of equation to solve:\n"
            + "+1: First Degree Equation (ax + b = 0)\n"
            + "+2: First Degree Equation System (ax1 + a2x2 = a0; b1x1 + b2x2 = b0)\n"
            + "+3: Second Degree Equation (ax^2 + bx + c = 0)\n"
            + "Equation Solver", JOptionPane.INFORMATION_MESSAGE
        );
        int choice = Integer.parseInt(choiceStr);
        switch (choice) {
            case 1: // Phuong trinh bac 1
                String s0 = JOptionPane.showInputDialog(null, "Enter value of a:", "Input", JOptionPane.INFORMATION_MESSAGE);
                String s1 = JOptionPane.showInputDialog(null, "Enter value of b:", "Input", JOptionPane.INFORMATION_MESSAGE);
                double a = Double.parseDouble(s0);
                double b = Double.parseDouble(s1);
                if (a == 0) {
                    if (b == 0) {
                        JOptionPane.showMessageDialog(null, "All real numbers are roots", "Answer", JOptionPane.INFORMATION_MESSAGE);
                    } else {
                        JOptionPane.showMessageDialog(null, "No real roots", "Answer", JOptionPane.INFORMATION_MESSAGE);
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "x = " + (-b / a), "Answer", JOptionPane.INFORMATION_MESSAGE);
                }
            case 2: // Phuong trinh bac 2
                Enter matrix A:
                Enter number of rows: 3
                Enter number of cols: 3
                Enter matrix A:
                1 2 3 4 5 6
                2 4 5 3 4 5
                Result matrix C:
                3 6 8
                0 9 6
                9 8 6
            case 3: // Phuong trinh bac 2
                Enter matrix B:
                Enter matrix B:
                Result matrix C:
                3 6 8
                0 9 6
                9 8 6
        }
    }
}

```

The terminal window shows the execution of several Java files from the Lab01 directory, including arrayProcessing, matrixAddition, and calculateTwoNumbers. It also shows an error message related to the first-degree equation solver.



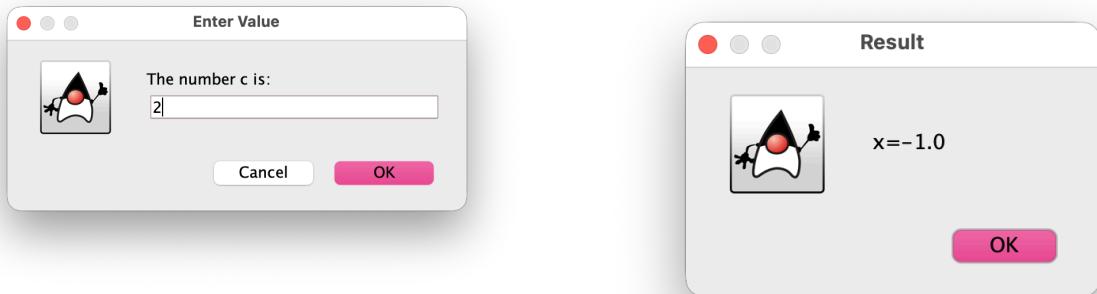


An "Enter Value" dialog window asking for the number  $a$ .

The window has three red, grey, and white circular buttons in the top-left corner. It is titled "Enter Value". Inside, there is a small square icon of the cartoon character. The text "The number a is:" is followed by a text input field containing the value "2". At the bottom are "Cancel" and "OK" buttons.

An "Enter Value" dialog window asking for the number  $b$ .

The window has three red, grey, and white circular buttons in the top-left corner. It is titled "Enter Value". Inside, there is a small square icon of the cartoon character. The text "The number b is:" is followed by a text input field containing the value "4". At the bottom are "Cancel" and "OK" buttons.



## 6.1

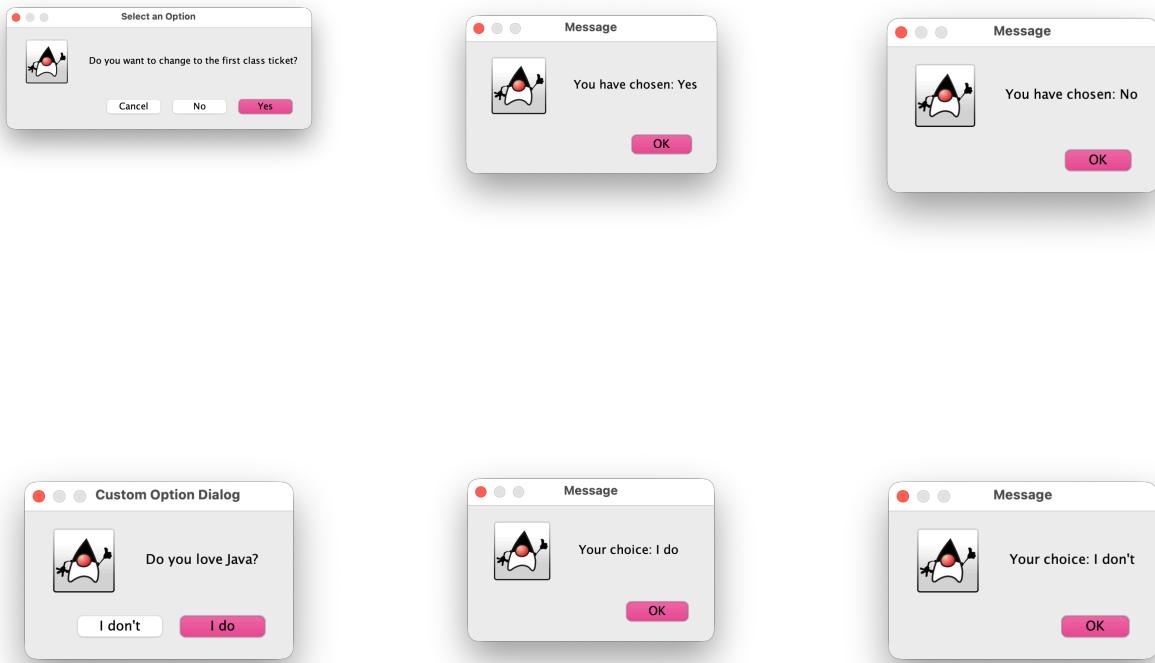
Screenshot of an IDE showing a Java project named 'OOP\_Lab'. The 'EXPLORER' view shows several Java files: ChoosingOption.java, daysinMonth.java, drawTriangle.java, FirstDegreeEquation.java, FirstDegreeEquationLinearSystem.java, inputFromKeyboard.java, and matrixAddition.java. The 'ChoosingOption.java' file is selected and its code is displayed in the central editor:

```

package code;
import javax.swing.*;
public class ChoosingOption {
    public static void main(String[] args) {
        int option = JOptionPane.showConfirmDialog(null,
            "Do you want to change to the first class ticket?");
        JOptionPane.showMessageDialog(null,
            "You have chosen: " + (option == JOptionPane.YES_OPTION ? "yes" :
            option == JOptionPane.NO_OPTION ? "No" : "Cancel"));
        // Custom option
        Object[] options = {"I do", "I don't"};
        int choice = JOptionPane.showOptionDialog(null,
            "Do you love Java?", "Custom Option Dialog",
            JOptionPane.YES_NO_OPTION,
            JOptionPane.QUESTION_MESSAGE,
            null,
            options,
            options[0]);
        JOptionPane.showMessageDialog(null, "Your choice: " + options[choice]);
    }
}

```

The status bar at the bottom shows: 'Hieu (19 minutes ago) Ln 1, Col 9 Spaces: 8 UTF-8 LF Java Signed out Miku'.



## 6.2

```
Lab01 > code > J inputFromKeyboard.java > Language Support for Java(TM) by Red Hat > {} code
1 package com;
2 import java.util.Scanner;
3
4 public class InputFromKeyboard {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.print("What's your name? ");
9         String name = scanner.nextLine();
10
11        System.out.print("How old are you? ");
12        int age = scanner.nextInt();
13
14        System.out.print("How tall are you (m)? ");
15        double height = scanner.nextDouble();
16
17        System.out.println("Mrs/Ms. " + name + ", " + age + " years old. " +
18                           "Your height is " + height + " m.");
19
20        scanner.close();
21    }
22}
```

```
(base) Hieu-MacBook-Pro:00P_Lab hieunguyen$ java Lab01/code/inputFromKeyboard.java
What's your name? Hieu
How old are you? 19
How tall are you (m)? 1.62
Mrs/Ms. Hieu, 19 years old. Your height is 1.62 m.
```

## 6.3

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows a project structure for "OOP\_Lab" containing "Lab01" and several Java files: ChoosingOption.java, drawTriangle.java, daysinMonth.java, FirstDegreeEquation.java, FirstDegreeEquationLinearSystem.java, inputFromKeyboard.java, and matrixAddition.java.
- PROBLEMS**: Displays a warning message about zstandard not being imported.
- OUTPUT**: Shows the command-line output of running the Java code.
- TERMINAL**: Shows the command-line interface with the Java command and user input.
- PORTS**: Shows network port information.
- GITLENS**: Shows Git-related information.
- PROBLEMS**: Shows a warning message about zstandard not being imported.
- OUTPUT**: Shows the command-line output of running the Java code.
- TERMINAL**: Shows the command-line interface with the Java command and user input.
- PORTS**: Shows network port information.
- GITLENS**: Shows Git-related information.

```
(base) Hieu-MacBook-Pro:OOP_Lab hieunguyen$ java Lab01/code/drawTriangle.java
Enter height n: 5
*
***
*****
*****
*****
*****
```

## 6.4

```

J ChoosingOption.java 2 J daysInMonth.java 3 J drawTriangle.java 3 J FirstDegreeEquation.java 2 J FirstDegreeEquationLinearSystem.java 2 J inputFromKeyboard.java 1 J matrixAddition.java 2 ...
OOP_Lab
Lab01 > code > J daysInMonth.java 3 Language Support for Java(TM) by Red Hat > {} code
J arrayProcess... 3
J calculateTwo... 2
J ChoosingOpt... 2
J daysInMonth... 3
J drawTriangle... 3
J FirstDegreeE... 2
J FirstDegreeE... 2
J inputFromKe... 1
J matrixAdditio... 1
J SecondDegr... 2
> result
# answer.txt
LICENSE

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

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] monthNames = {"January", "February", "March", "April", "May", "June",
                               "July", "August", "September", "October", "November", "December"};
        while (true) {
            System.out.print("Enter month (name/abbr/number): ");
            String monthInput = sc.nextLine();
            System.out.print("Enter year: ");
            int year = sc.nextInt();

            if (year < 0) {
                System.out.println("Invalid year. Try again.");
                continue;
            }

            int month = -1;
            // Check month
            try {
                month = Integer.parseInt(monthInput);
            } catch (NumberFormatException e) {}

            // Check string
            if (month == -1) {
                for (int i = 0; i < 12; i++) {
                    if (monthNames[i].equalsIgnoreCase(monthInput)) {
                        monthNames[i].substring(0,3).concat(".").equalsIgnoreCase(monthInput)) {
                            month = i + 1;
                            break;
                        }
                    }
                }
            }

            if (month < 1 || month > 12) {
                System.out.println("Invalid month. Try again.");
                continue;
            }

            int[] daysInMonth = {31,28,31,30,31,30,31,31,30,31,30,31};
            int days = daysInMonth[month-1];
            if (month == 2 && isLeapYear(year)) days = 29;

            System.out.println("Month " + monthNames[month-1] + " " + year + " has " + days + " days.");
            break;
        }
        sc.close();
    }
}

```

```
(base) Hieus-MacBook-Pro:00P_Lab hieunguyen$ java Lab01/code/daysInMonth.java
Enter month (name/abbr/number): august
Enter year: 2007
Month August 2007 has 31 days.
```

## 6.5

The screenshot shows the Red Hat Language Support for Java interface. The code editor displays a Java file named `arrayProcessing.java` with the following content:

```
public class arrayProcessing {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " numbers:");
        for (int i = 0; i < n; i++) arr[i] = sc.nextInt();
        Arrays.sort(arr);
        System.out.println("Sorted array: " + Arrays.toString(arr));
        int sum = 0;
        for (int num : arr) sum += num;
        double avg = (double) sum / n;
        System.out.println("Sum = " + sum);
        System.out.println("Average = " + avg);
        sc.close();
    }
}
```

The terminal window below shows the execution of the program:

```
(base) Hieu-MacBook-Pro:00P_Lab hieunguyen$ java Lab01/code/arrayProcessing.java
Enter number of elements: 4
Enter 4 numbers:
2986 2673 2435 7286
Sorted array: [2435, 2673, 2986, 7286]
Sum = 15380
Average = 3845.0
```

## 6.6

The screenshot shows a Java code editor interface with the following details:

- Project Structure:** The project is named "Lab01". It contains several Java files: Lab01.java, FirstDegreeEquationLinearSystem.java, InputFromKeyboard.java, matrixAddition.java, SecondDegreeEquation.java, arrayProcessing.class, and arrayProc.java.
- Code Editor:** The main editor window displays the content of `matrixAddition.java`. The code implements matrix addition using `Scanner` to read user input for the number of rows and columns, and for the elements of matrices A and B. It then calculates matrix C as the sum of A and B and prints it to the console.
- Toolbars and Menus:** Standard Java IDE menus like File, Edit, View, Tools, Window, and Help are visible at the top.
- Sidebar:** On the left, there's a sidebar with icons for file operations, a search bar, and sections for Outline, Timeline, Projects, Run Configuration, and Java Projects.
- Status Bar:** The bottom status bar shows the current user (You), the date (21 minutes ago), the developer (Hieu), the line number (Ln 1, Col 14), spaces (Spaces: 4), encoding (UTF-8), file type (Java), and signed out status.

```
(base) Hieus-MacBook-Pro:00P_Lab hieunguyen$ java Lab01/code/matrixAddition.java
Enter number of rows: 3
Enter number of cols: 3
Enter matrix A:
1 2 3 5 3 2 4 2 3
Enter matrix B:
2 4 5 3 2 4 5 6 3
Result matrix C:
3 6 8
8 5 6
9 8 6
```