

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»



ЗВІТ
про виконання практичних робіт
з дисципліни
«Поглиблене програмування в середовищі Java»
Лабораторна робота № 2

Виконав:
студент гр. 121-21-1
Соболев Єгор Романович

Перевірив:
Доц. Мінєєв О.С.

Дніпро
2025

Тема: Основи

Завдання

Розробити програму, що дозволить вам створити, як з клавіатури так і рандомно матрицю цілих чисел типу `int` заданої ширини та висоти(ввести з клавіатури), але не більше 20 на 20. Створити можливість пошуку в цій матриці мінімального і максимального елементу та розрахунок середнього арифметичного. Програма може бути написана в одному класі, обов'язково розбиття на методи. Обов'язкове використання клавіатури, під час вибору ручного чи рандомного створення матриці. Створення системи зчитування з клавіатури зробити будь-яким способом, наприклад завдяки класу `Scanner`. `Scanner` являє собою найпростішу систему сканування клавіатури. Діапазон рандомних чисел для створення елементів матриці повинен зберігатись в спеціальних константах.

Як завдання підвищеної складності додати розрахунок середнього геометричного елементів матриці.

Код програми:

```
package org.example;

> import ...

R Rename usages
public class IMatrix {
    private static final int MAX_SIZE = 20; 1 usage
    private static final int MIN_RANDOM = -100; 2 usages
    private static final int MAX_RANDOM = 100; 1 usage

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введіть висоту матриці (1-20): ");
        int rows = getValidInput(scanner);
        System.out.print("Введіть ширину матриці (1-20): ");
        int cols = getValidInput(scanner);

        int[][] matrix;
        System.out.print("Оберіть спосіб заповнення матриці (1 - вручну, 2 - випадково): ");
        int choice = scanner.nextInt();

        if (choice == 1) {
            matrix = inputMatrix(scanner, rows, cols);
        } else {
            matrix = generateRandomMatrix(rows, cols);
        }
    }
}
```

```

public class IMatrix {
    public static void main(String[] args) {

        printMatrix(matrix);
        int min = findMin(matrix);
        int max = findMax(matrix);
        double avg = calculateArithmeticMean(matrix);
        double geoMean = calculateGeometricMean(matrix);

        System.out.println("Мінімальне значення: " + min);
        System.out.println("Максимальне значення: " + max);
        System.out.println("Середнє арифметичне: " + avg);
        System.out.println("Середнє геометричне: " + (geoMean == -1 ? "Неможливо розрахувати" : geoMean));
    }

    private static int getValidInput(Scanner scanner) { 2 usages
        int value;
        do {
            value = scanner.nextInt();
        } while (value < 1 || value > MAX_SIZE);
        return value;
    }

    private static int[][] inputMatrix(Scanner scanner, int rows, int cols) { 1 usage
        int[][] matrix = new int[rows][cols];
        System.out.println("Введіть елементи матриці:");
        for (int i = 0; i < rows; i++) {

```

```
private static int[][] generateRandomMatrix(int rows, int cols) { 1 usage
    for (int j = 0; j < cols; j++) {
        matrix[i][j] = random.nextInt( bound: MAX_RANDOM - MIN_RANDOM + 1) + MIN_RANDOM;
    }
    return matrix;
}
```

```
private static void printMatrix(int[][] matrix) { 1 usage
    System.out.println("Матрица:");
    for (int[] row : matrix) {
        for (int value : row) {
            System.out.print(value + " ");
        }
        System.out.println();
    }
}
```

```
private static int findMin(int[][] matrix) { 1 usage
    int min = Integer.MAX_VALUE;
    for (int[] row : matrix) {
        for (int value : row) {
            if (value < min) {
                min = value;
            }
        }
    }
    return min;
}
```

```
private static int findMax(int[][] matrix) { 1 usage
    int max = Integer.MIN_VALUE;
    for (int[] row : matrix) {
        for (int value : row) {
            if (value > max) {
                max = value;
            }
        }
    }
}
```

```

public class Matrix {
    private static int findMax(int[][] matrix) { 1 usage
        int max = Integer.MIN_VALUE;
        for (int[] row : matrix) {
            for (int value : row) {
                if (value > max) {
                    max = value;
                }
            }
        }
        return max;
    }

    private static double calculateArithmeticMean(int[][] matrix) { 1 usage
        int sum = 0, count = 0;
        for (int[] row : matrix) {
            for (int value : row) {
                sum += value;
                count++;
            }
        }
        return (double) sum / count;
    }

    private static double calculateGeometricMean(int[][] matrix) { 1 usage
        double product = 1.0;
        int count = 0;
        for (int[] row : matrix) {
            for (int value : row) {
                if (value <= 0) return -1; // Геометричне середнє визначається лише для додатних чисел
                product *= value;
                count++;
            }
        }
        return Math.pow(product, 1.0 / count);
    }
}

```

Результати роботи програми:

```

C:\Users\hosha\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3
Введіть висоту матриці (1-20): 1
Введіть ширину матриці (1-20): 3
Оберіть спосіб заповнення матриці (1 - вручну, 2 - випадково): 4
Матриця:
-32 -68 -72
Мінімальне значення: -72
Максимальне значення: -32
Середнє арифметичне: -57.333333333333336
Середнє геометричне: Неможливо розрахувати
5
Process finished with exit code 0

```