

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних технологій та комп'ютерної інженерії



Лабораторна робота №2

з дисципліни:
«Базова Джава»

Виконав:
студент групи 123-21-1
Скоропад Є.В.
Перевірів:
Доцент кафедри САіУ
Мінєєв О.С.

Дніпро
2025

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 зверігатись в спеціальних константах.

Код програми

```
package org.example;
import java.util.Scanner;
import java.util.Random;

public class Main {

    // Constants for the range of random numbers
    private static final int MIN_RANDOM_VALUE = -100;
    private static final int MAX_RANDOM_VALUE = 100;

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Reading matrix dimensions
        System.out.println("Enter the matrix width (maximum 20):");
        int width = readValidDimension(scanner);

        System.out.println("Enter the height of the matrix (maximum 20):");
        int height = readValidDimension(scanner);

        // Choosing a method for creating a matrix
        System.out.println("Choose the method of creating the matrix: 1 - manual, 2 - random:");
        int choice = scanner.nextInt();

        int[][] matrix;
        if (choice == 1) {
            matrix = createMatrixManually(width, height, scanner);
        } else if (choice == 2) {
            matrix = createMatrixRandomly(width, height);
        } else {
            System.out.println("Incorrect selection. The program is ending.");
            return;
        }

        System.out.println("Created matrix:");
        printMatrix(matrix);

        int min = findMin(matrix);
        int max = findMax(matrix);
        System.out.println("Minimum value: " + min);
    }
}
```

```

        System.out.println("Maximum value: " + max);

        double average = calculateAverage(matrix);
        System.out.printf("Arithmetic mean: %.2f%n", average);

        double geometricMean = calculateGeometricMean(matrix);
        System.out.printf("Geometric mean: %.2f%n", geometricMean);

        scanner.close();
    }

    // Method for reading the correct matrix size
    private static int readValidDimension(Scanner scanner) {
        int dimension;
        while (true) {
            dimension = scanner.nextInt();
            if (dimension > 0 && dimension <= 20) {
                break;
            }
            System.out.println("The size must be between 1 and 20. Try again:");
        }
        return dimension;
    }

    // Method for creating a matrix manually
    private static int[][] createMatrixManually(int width, int height, Scanner scanner) {
        int[][] matrix = new int[height][width];
        System.out.println("Enter matrix elements:");

        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                System.out.printf("Element [%d][%d]: ", i + 1, j + 1);
                matrix[i][j] = scanner.nextInt();
            }
        }

        return matrix;
    }

    // Method for creating a random matrix
    private static int[][] createMatrixRandomly(int width, int height) {
        int[][] matrix = new int[height][width];
        Random random = new Random();

        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                matrix[i][j] = random.nextInt(MAX_RANDOM_VALUE - MIN_RANDOM_VALUE + 1) + MIN_RANDOM_VALUE;
            }
        }

        return matrix;
    }

    // Method for deriving a matrix
    private static void printMatrix(int[][] matrix) {
        for (int[] row : matrix) {
            for (int element : row) {
                System.out.print(element + "\t");
            }
            System.out.println();
        }
    }

```

```

    }

    // Method for finding the minimum value
    private static int findMin(int[][] matrix) {
        int min = Integer.MAX_VALUE;
        for (int[] row : matrix) {
            for (int element : row) {
                if (element < min) {
                    min = element;
                }
            }
        }
        return min;
    }

    // Method for finding the maximum value
    private static int findMax(int[][] matrix) {
        int max = Integer.MIN_VALUE;
        for (int[] row : matrix) {
            for (int element : row) {
                if (element > max) {
                    max = element;
                }
            }
        }
        return max;
    }

    // Method for calculating the arithmetic mean
    private static double calculateAverage(int[][] matrix) {
        double sum = 0;
        int count = 0;

        for (int[] row : matrix) {
            for (int element : row) {
                sum += element;
                count++;
            }
        }

        return sum / count;
    }

    // Method for calculating the geometric mean
    private static double calculateGeometricMean(int[][] matrix) {
        double product = 1.0;
        int count = 0;

        for (int[] row : matrix) {
            for (int element : row) {
                if (element != 0) { // Avoid multiplication by zero
                    product *= Math.abs(element);
                    count++;
                }
            }
        }

        if (count == 0) {
            return 0; // If all elements are zero
        }

        return Math.pow(product, 1.0 / count);
    }
}

```

Результат виконання коду

```
Enter the matrix width (maximum 20):
2
Enter the height of the matrix (maximum 20):
3
Choose the method of creating the matrix: 1 - manual, 2 - random:
2
Created matrix:
-42 -100
9 18
46 -2
Minimum value: -100
Maximum value: 46
Arithmetic mean: -11,83
Geometric mean: 19,93
```