

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



ЗВІТ
Практична робота №2
з дисципліни
«Поглиблене програмування в середовищі Java»

Виконала:

ст. гр. 122-21-3

Беляєва В. В.

Прийняв:

Доцент каф. САіУ

Мінєєв О. С.

м. Дніпро
2024 рік

Практична робота №2

Основи

Мета роботи: навчитися створювати базові програми

Завдання до виконання: Лабораторна робота номер 2. Основи.

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.

Як завдання підвищеної складності додати розрахунок середнього геометричного елементів матриці.

Хід роботи

Було розроблено наступний код:

```
package org.example;

import java.util.Random;
import java.util.Scanner;

public class Main {
    // Константи для генерації випадкових чисел в заданому діапазоні
    private static final int MIN_VALUE = -100; // Мінімальне значення для
елементів матриці
    private static final int MAX_VALUE = 100; // Максимальне значення для
елементів матриці

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

        // Введення розмірів матриці
        System.out.print("Введіть кількість рядків матриці (макс. 20): ");
        int rows = scanner.nextInt();
        System.out.print("Введіть кількість стовпців матриці (макс. 20): ");
```

```

        int cols = scanner.nextInt();

        // Перевірка, щоб розміри не перевищували 20x20
        if (rows > 20 || cols > 20) {
            System.out.println("Розмір матриці не може перевищувати 20x20!");
            return; // Завершуємо виконання програми, якщо розміри не
// відповідають вимогам
        }

        // Вибір способу створення матриці: вручну або випадково
        System.out.print("Виберіть спосіб створення матриці (1 - ручний, 2 -
// випадковий): ");
        int choice = scanner.nextInt();

        // Створення порожньої матриці з введеними розмірами
        int[][] matrix = new int[rows][cols];

        // Заповнення матриці відповідно до вибору користувача
        if (choice == 1) {
            fillMatrixManually(matrix, scanner); // Заповнення вручну
        } else if (choice == 2) {
            fillMatrixRandomly(matrix); // Заповнення випадковими числами
        } else {
            System.out.println("Невірний вибір! Виберіть 1 або 2.");
            return; // Завершуємо програму при неправильному виборі
        }

        // Виведення матриці на екран
        System.out.println("Матриця:");
        printMatrix(matrix);

        // Пошук мінімуму, максимуму, середнього арифметичного і середнього
        // геометричного
        int min = findMin(matrix); // Пошук мінімального елемента
        int max = findMax(matrix); // Пошук максимального елемента
        double avg = calculateArithmeticMean(matrix); // Обчислення
        // середнього арифметичного
        double geoAvg = calculateGeometricMean(matrix); // Обчислення
        // середнього геометричного

        // Виведення результатів
        System.out.println("Мінімальний елемент: " + min);
        System.out.println("Максимальний елемент: " + max);
        System.out.println("Середнє арифметичне: " + avg);
        System.out.println("Середнє геометричне: " + geoAvg);
    }

    // Метод для заповнення матриці вручну
    public static void fillMatrixManually(int[][] matrix, Scanner scanner) {
        // Проходимо по кожному елементу матриці і запитуємо значення у
        // користувача
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print("Введіть елемент матриці [" + i + "][" + j +
                "]: ");

                matrix[i][j] = scanner.nextInt(); // Вводимо значення для
                // кожного елемента
            }
        }
    }

    // Метод для заповнення матриці випадковими числами в заданому діапазоні
    public static void fillMatrixRandomly(int[][] matrix) {

```

```

Random random = new Random();
// Генеруємо випадкові числа для кожного елемента матриці
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix[i].length; j++) {
        matrix[i][j] = random.nextInt(MAX_VALUE - MIN_VALUE + 1) +
MIN_VALUE; // Генерація випадкового числа
    }
}

// Метод для виведення матриці на екран
public static void printMatrix(int[][] matrix) {
    // Виводимо кожен рядок матриці
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            System.out.print(matrix[i][j] + "\t"); // Виводимо елемент з
табуляцією між числами
        }
        System.out.println(); // Переходимо на новий рядок після кожного
ряду
    }
}

// Метод для пошуку мінімального елемента в матриці
public static int findMin(int[][] matrix) {
    int min = matrix[0][0]; // Початкове значення мінімуму – перший
елемент матриці
    // Перебираємо всі елементи матриці, щоб знайти мінімум
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            if (matrix[i][j] < min) { // Якщо поточний елемент менший за
мінімум
                min = matrix[i][j]; // Оновлюємо мінімум
            }
        }
    }
    return min; // Повертаємо мінімальний елемент
}

// Метод для пошуку максимального елемента в матриці
public static int findMax(int[][] matrix) {
    int max = matrix[0][0]; // Початкове значення максимуму – перший
елемент матриці
    // Перебираємо всі елементи матриці, щоб знайти максимум
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            if (matrix[i][j] > max) { // Якщо поточний елемент більший
за максимум
                max = matrix[i][j]; // Оновлюємо максимум
            }
        }
    }
    return max; // Повертаємо максимальний елемент
}

// Метод для обчислення середнього арифметичного елементів матриці
public static double calculateArithmeticMean(int[][] matrix) {
    int sum = 0; // Сума всіх елементів матриці
    int count = 0; // Лічильник елементів
    // Перебираємо матрицю для підсумку всіх елементів
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            sum += matrix[i][j]; // Додаємо поточний елемент до суми

```

```

        count++; // Збільшуємо лічильник
    }
}
// Обчислюємо середнє арифметичне
return (double) sum / count;
}

// Метод для обчислення середнього геометричного елементів матриці
public static double calculateGeometricMean(int[][] matrix) {
    double product = 1; // Початковий добуток всіх елементів (1, щоб не
    змінити результат)
    int count = 0; // Лічильник елементів
    // Перебираємо матрицю для обчислення добутку всіх елементів
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            product *= matrix[i][j]; // Множимо поточний елемент на
            добуток
            count++; // Збільшуємо лічильник елементів
        }
    }
    // Обчислюємо середнє геометричне за формулою: корінь n-го степеня з
    добутку всіх елементів
    return Math.pow(product, 1.0 / count);
}
}

```

Результат роботи програми:

```

C:\Java\jdk-17\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt
.jar=63278:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\Veronica\Desktop\Java\lab2\target\classes org.example.Main
Введіть кількість рядків матриці (макс. 20): 10
Введіть кількість стовпців матриці (макс. 20): 10
Виберіть спосіб створення матриці (1 - ручний, 2 - випадковий): 2
Матриця:
-2 17 -33 -66 54 25 -57 100 -50 26
35 -50 -42 -15 11 -63 -10 25 -66 59
8 -42 -56 -93 50 -96 92 -15 24 34
52 77 76 -83 -61 9 -94 53 20 5
0 40 -35 5 58 -21 56 -55 62 80
-1 39 -48 63 -6 -68 2 92 11 -36
18 -63 14 -53 -27 -42 48 -46 -100 -52
51 -93 2 -35 -33 -65 60 -35 -72 -66
-88 -41 -75 50 33 -4 38 -14 -49 21
-6 0 97 -42 -64 -26 -27 -10 21 14
Мінімальний елемент: -100
Максимальний елемент: 100
Середнє арифметичне: -6.65
Середнє геометричне: 0.0
Process finished with exit code 0

```

Висновок: було вивчено, як створювати базові програми