МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет

«Дніпровська політехніка»

ЗВІТ

з лабораторної роботи №2

з дисципліни

“Поглиблене програмування в середовищі Java”

Виконала

студентка групи

122-21-2

Мартиненко Юлія Володимирівна

Дніпро

2025

Лабораторна робота №2

Тема: 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. Диапазон рандомних чисел для створення елементів матриці повинен зберігатись в спеціальних константах.

Як завдання підвищеної складності додати розрахунок середнього геометричного елементів матриці.

Хід роботи

1. Створюю гілку LR_2.

```
PS D:\SAVE\IdeaProjects\HelloWorld> git checkout -b LR_2
Switched to a new branch 'LR_2'
```

```
PS D:\SAVE\IdeaProjects\HelloWorld> git branch
  LR_1
* LR_2
  main
```

2. Створюю новий пакет для другої лабораторної роботи.

Створюю новий клас та код програми:

```
package com.example.LR_2;

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

public class MatrixCalculator {

    // Константи для діапазону випадкових чисел
    private static final int MIN_VALUE = 1;  // Для середнього геометричного
використовуємо лише додатні числа
    private static final int MAX_VALUE = 100;
```

```java
    private static final int MAX_SIZE = 20;

    // Метод для створення матриці вручну
    public static int[][] createMatrixManually(int rows, int cols) {
        Scanner scanner = new Scanner(System.in);
        int[][] matrix = new int[rows][cols];

        System.out.println("Введіть елементи матриці:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.printf("matrix[%d][%d] = ", i, j);
                matrix[i][j] = scanner.nextInt();
            }
        }

        return matrix;
    }

    // Метод для створення матриці випадковим чином
    public static int[][] createMatrixRandomly(int rows, int cols) {
        Random random = new Random();
        int[][] matrix = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix[i][j] = random.nextInt(MAX_VALUE - MIN_VALUE + 1) +
MIN_VALUE;  // Генерація лише додатніх чисел
            }
        }

        return matrix;
    }

    // Метод для знаходження мінімального елемента матриці
    public static int findMin(int[][] matrix) {
        int min = matrix[0][0];
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                if (matrix[i][j] < min) {
                    min = matrix[i][j];
                }
            }
        }
        return min;
    }

    // Метод для знаходження максимального елемента матриці
    public static int findMax(int[][] matrix) {
        int max = matrix[0][0];
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                if (matrix[i][j] > max) {
                    max = matrix[i][j];
                }
            }
        }
        return max;
    }

    // Метод для обчислення середнього арифметичного елементів матриці
    public static double calculateArithmeticMean(int[][] matrix) {
        double sum = 0;
        int totalElements = 0;
        for (int i = 0; i < matrix.length; i++) {
```

```
            for (int j = 0; j < matrix[i].length; j++) {
                sum += matrix[i][j];
                totalElements++;
            }
        }
        return sum / totalElements;
    }

    // Метод для обчислення середнього геометричного елементів матриці
    public static double calculateGeometricMean(int[][] matrix) {
        double product = 1;
        int totalElements = 0;
        boolean hasPositive = false;  // Для перевірки наявності додатніх
чисел

        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                int value = matrix[i][j];

                // Виключаємо від'ємні числа та нулі
                if (value > 0) {
                    product *= value;
                    totalElements++;
                    hasPositive = true;
                }
            }
        }

        if (hasPositive && totalElements > 0) {
            return Math.pow(product, 1.0 / totalElements);  // середнє
геометричне
        } else {
            return Double.NaN; // Якщо немає додатніх чисел
        }
    }

    // Метод для виведення матриці
    public static void printMatrix(int[][] matrix) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print(matrix[i][j] + "\t");
            }
            System.out.println();
        }
    }

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

        // Введення розміру матриці
        System.out.print("Введіть кількість рядків матриці: ");
        int rows = scanner.nextInt();
        System.out.print("Введіть кількість стовпців матриці: ");
        int cols = scanner.nextInt();

        if (rows > MAX_SIZE || cols > MAX_SIZE) {
            System.out.println("Розмір матриці не може бути більшим за
20x20.");
            return;
        }

        // Вибір між ручним вводу або випадковим створенням
        System.out.println("Виберіть спосіб створення матриці:");
        System.out.println("1. Ручне введення");
```

```
        System.out.println("2. Випадкове створення");
        int choice = scanner.nextInt();

        int[][] matrix = null;
        if (choice == 1) {
            matrix = createMatrixManually(rows, cols);
        } else if (choice == 2) {
            matrix = createMatrixRandomly(rows, cols);
        } else {
            System.out.println("Невірний вибір.");
            return;
        }

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

        // Обчислення мінімуму, максимуму, середнього арифметичного
        int min = findMin(matrix);
        int max = findMax(matrix);
        double arithmeticMean = calculateArithmeticMean(matrix);
        double geometricMean = calculateGeometricMean(matrix);

        // Виведення результатів
        System.out.println("Мінімальний елемент: " + min);
        System.out.println("Максимальний елемент: " + max);
        System.out.println("Середнє арифметичне: " + arithmeticMean);
        System.out.println("Середнє геометричне: " +
(Double.isNaN(geometricMean) ? "Неможливо обчислити" : geometricMean));
    }
}
```

Пояснення коду

Константи для матриці:

MIN_VALUE, MAX_VALUE, MAX_SIZE: визначають діапазон випадкових чисел і максимальний розмір матриці (20x20).

Методи:

createMatrixManually: створює матрицю вручну, запитуючи кожен елемент.

createMatrixRandomly: створює матрицю випадковим чином, заповнюючи елементи випадковими числами.

findMin, findMax: шукають мінімальний та максимальний елементи в матриці.

calculateArithmeticMean: обчислює середнє арифметичне елементів матриці.

calculateGeometricMean: обчислює середнє геометричне елементів матриці.

printMatrix: виводить матрицю на екран.

Основний метод (main):

Збирає вхідні дані від користувача: розміри матриці та вибір способу її створення.

Викликає відповідні методи для створення матриці, обчислення статистики і виведення результатів.

3. Додаю всі зміни та виконую коміт:

```
PS D:\SAVE\IdeaProjects\HelloWorld> git add .
PS D:\SAVE\IdeaProjects\HelloWorld> git commit -m "Додаю файли для ЛР2: матриця, мінімум, максимум, середнє"
[LR_2 6250e8e] Додаю файли для ЛР2: матриця, мінімум, максимум, середнє
 1 file changed, 153 insertions(+)
 create mode 100644 src/main/java/com/example/LR_2/MatrixCalculator.java
```

4. Запушую гілку на гітхаб:

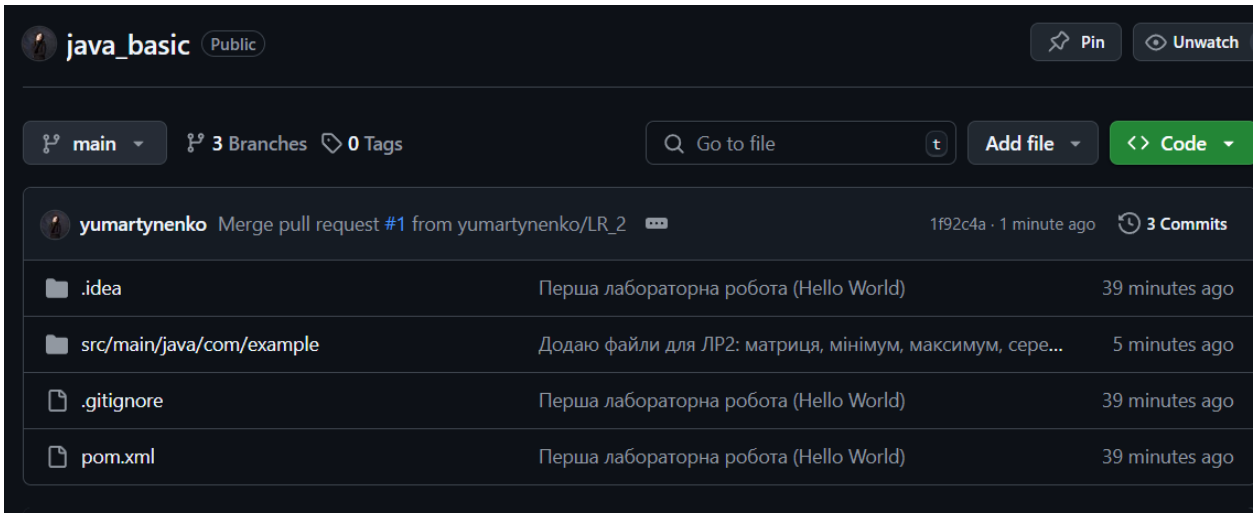
```
PS D:\SAVE\IdeaProjects\HelloWorld> git push origin LR_2
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 1.93 KiB | 1.93 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'LR_2' on GitHub by visiting:
remote:      https://github.com/yumartynenko/java_basic/pull/new/LR_2
remote:
To https://github.com/yumartynenko/java_basic.git
 * [new branch]      LR_2 -> LR_2
PS D:\SAVE\IdeaProjects\HelloWorld>
```

5. Створюю Pull Request на GitHub та виконую Merge pull request.

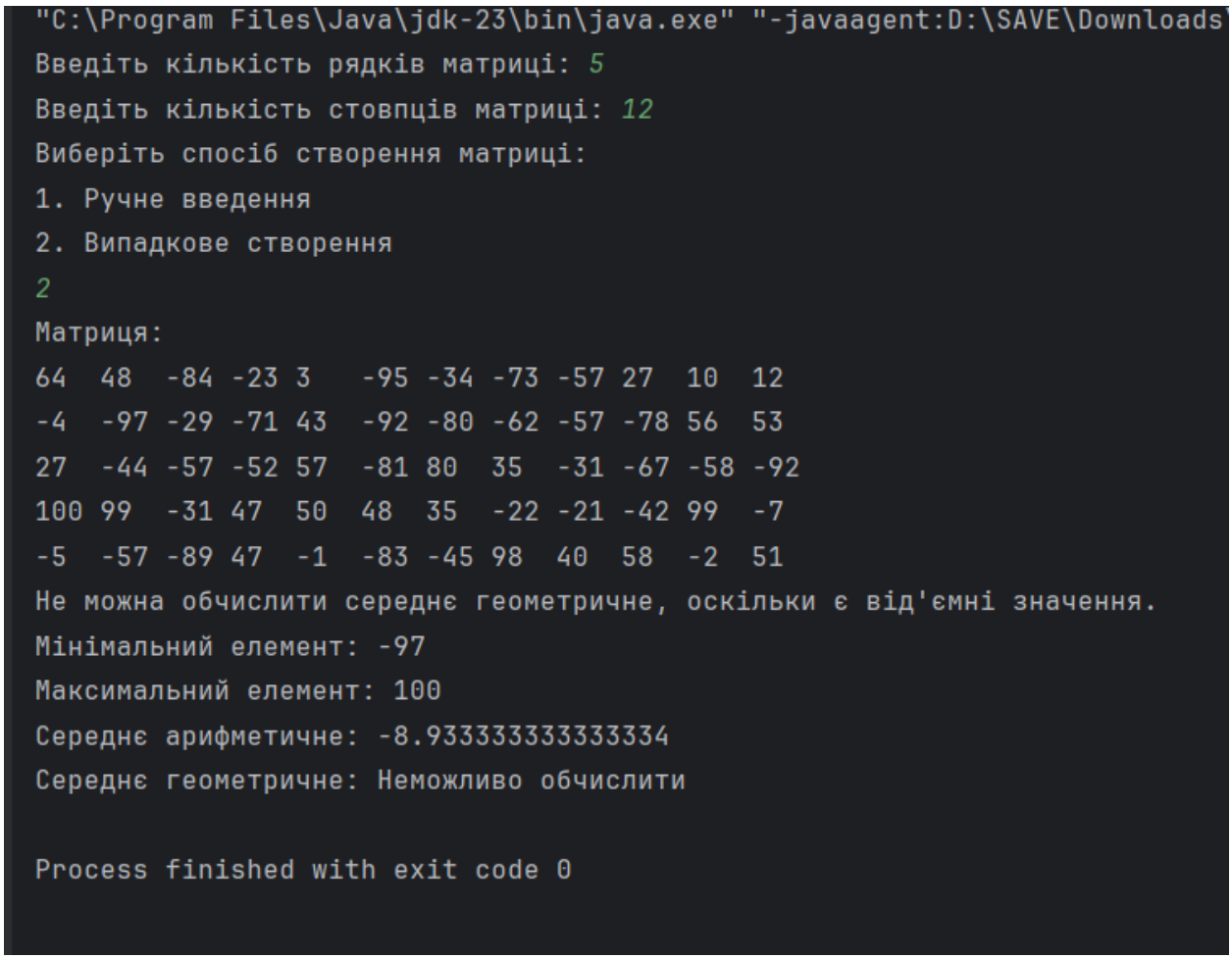
6. Також оновлюю локальну версію гілки:

```
PS D:\SAVE\IdeaProjects\HelloWorld> git checkout main
Switched to branch 'main'
PS D:\SAVE\IdeaProjects\HelloWorld> git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 996 bytes | 28.00 KiB/s, done.
From https://github.com/yumartynenko/java_basic
 * branch            main       -> FETCH_HEAD
   094b6e5..1f92c4a  main       -> origin/main
Updating 094b6e5..1f92c4a
Fast-forward
 .../java/com/example/LR_2/MatrixCalculator.java    | 153 ++++++++++++++++++++
 1 file changed, 153 insertions(+)
 create mode 100644 src/main/java/com/example/LR_2/MatrixCalculator.java
PS D:\SAVE\IdeaProjects\HelloWorld>
```

Бачимо результат:



Результат виконання створеної програми:


"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:D:\SAVE\Downloads
Введіть кількість рядків матриці: 5
Введіть кількість стовпців матриці: 12
Виберіть спосіб створення матриці:
1. Ручне введення
2. Випадкове створення
2
Матриця:
64 48 -84 -23 3 -95 -34 -73 -57 27 10 12
-4 -97 -29 -71 43 -92 -80 -62 -57 -78 56 53
27 -44 -57 -52 57 -81 80 35 -31 -67 -58 -92
100 99 -31 47 50 48 35 -22 -21 -42 99 -7
-5 -57 -89 47 -1 -83 -45 98 40 58 -2 51
Не можна обчислити середнє геометричне, оскільки є від'ємні значення.
Мінімальний елемент: -97
Максимальний елемент: 100
Середнє арифметичне: -8.933333333333334
Середнє геометричне: Неможливо обчислити
Process finished with exit code 0

Або так:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-
Введіть кількість рядків матриці: 2
Введіть кількість стовпців матриці: 2
Виберіть спосіб створення матриці:
1. Ручне введення
2. Випадкове створення
1
Введіть елементи матриці:
matrix[0][0] = 6
matrix[0][1] = 5
matrix[1][0] = 2
matrix[1][1] = 3
Матриця:
6   5
2   3
Мінімальний елемент: 2
Максимальний елемент: 6
Середнє арифметичне: 4.0
Середнє геометричне: 3.6628415014847064

Process finished with exit code 0
```

Висновок. Під час виконання лабораторної роботи №1 з дисципліни «Поглиблене програмування в середовищі Java» було вивчено основи мови Java та створено програму згідно завданню.