**Міністерство науки і освіти України**

**Національний технічний університет**

**«Дніпровська Політехника»**



**Звіт з лабораторної роботи №2**

**З дисципліни «Поглиблене програмування в середовищі Java»**

**Виконав студент:**

**групи 125-21-3**

**Войтович Микита**

**Прийняв:**

**Мінєєв О.С.**

**м. Дніпро**

**2025 р.**

**Войтович Микита 125-21-3**

**Лабораторна робота №2**

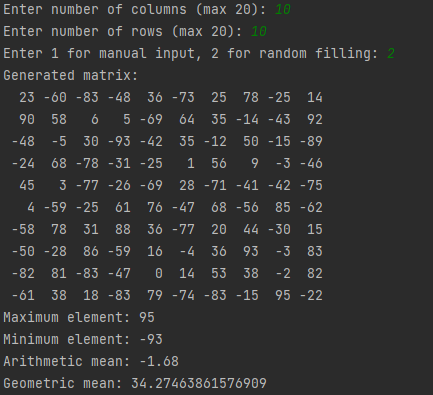
**Основи.**

**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. Диапазон рандомних чисел для створення елементів матриці повинен зверігатись в спеціальних константах.

**Виконання лабораторної роботи:**

package Voytovic\_Mykyta.lab2;  
  
import java.util.Random;  
import java.util.Scanner;  
  
public class Matrix {  
 // Константи для обмеження розміру та значень у матриці  
 private static final int *MIN\_LIMIT* = -100;  
 private static final int *MAX\_LIMIT* = 100;  
 private static final int *MAX\_SIZE* = 20;  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Введення розмірів матриці  
 System.*out*.print("Enter number of columns (max 20): ");  
 int col = scanner.nextInt();  
 System.*out*.print("Enter number of rows (max 20): ");  
 int row = scanner.nextInt();  
  
 // Перевірка коректності введених розмірів  
 if (col <= 0 || col > *MAX\_SIZE* || row <= 0 || row > *MAX\_SIZE*) {  
 System.*out*.println("Error: Invalid matrix size.");  
 return;  
 }  
  
 // Вибір методу заповнення матриці  
 System.*out*.print("Enter 1 for manual input, 2 for random filling: ");  
 int mode = scanner.nextInt();  
  
 int[][] matrix = (mode == 1) ? *userFillMatrix*(scanner, col, row) : *randomFillMatrix*(col, row);  
  
 *displayMatrix*(matrix);  
  
 // Обчислення статистичних характеристик матриці  
 int maxElement = *findMaximum*(matrix);  
 int minElement = *findMinimum*(matrix);  
 double arithmeticMean = *computeArithmeticMean*(matrix);  
 double geometricMean = *computeGeometricMean*(matrix);  
  
 // Виведення результатів  
 System.*out*.println("Maximum element: " + maxElement);  
 System.*out*.println("Minimum element: " + minElement);  
 System.*out*.println("Arithmetic mean: " + arithmeticMean);  
 System.*out*.println("Geometric mean: " + geometricMean);  
 }  
  
 // Метод для ручного введення значень у матрицю  
 private static int[][] userFillMatrix(Scanner scanner, int col, int row) {  
 int[][] matrix = new int[row][col];  
 System.*out*.println("Enter matrix elements row by row:");  
 for (int i = 0; i < row; i++) {  
 for (int j = 0; j < col; j++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
 return matrix;  
 }  
  
 // Метод для випадкового заповнення матриці  
 private static int[][] randomFillMatrix(int col, int row) {  
 int[][] matrix = new int[row][col];  
 Random random = new Random();  
 for (int i = 0; i < row; i++) {  
 for (int j = 0; j < col; j++) {  
 matrix[i][j] = random.nextInt(*MAX\_LIMIT* - *MIN\_LIMIT* + 1) + *MIN\_LIMIT*;  
 }  
 }  
 return matrix;  
 }  
  
 // Метод для виведення матриці у консоль  
 private static void displayMatrix(int[][] matrix) {  
 System.*out*.println("Generated matrix:");  
 for (int[] row : matrix) {  
 for (int value : row) {  
 System.*out*.printf("%4d", value);  
 }  
 System.*out*.println();  
 }  
 }  
  
 // Метод для знаходження максимального значення у матриці  
 private static int findMaximum(int[][] matrix) {  
 int max = Integer.*MIN\_VALUE*;  
 for (int[] row : matrix) {  
 for (int value : row) {  
 if (value > max) max = value;  
 }  
 }  
 return max;  
 }  
  
 // Метод для знаходження мінімального значення у матриці  
 private static int findMinimum(int[][] matrix) {  
 int min = Integer.*MAX\_VALUE*;  
 for (int[] row : matrix) {  
 for (int value : row) {  
 if (value < min) min = value;  
 }  
 }  
 return min;  
 }  
  
 // Метод для обчислення середнього арифметичного  
 private static double computeArithmeticMean(int[][] matrix) {  
 int sum = 0, count = 0;  
 for (int[] row : matrix) {  
 for (int value : row) {  
 sum += value;  
 count++;  
 }  
 }  
 return (double) sum / count;  
 }  
  
 // Метод для обчислення середнього геометричного  
 private static double computeGeometricMean(int[][] matrix) {  
 double product = 1.0;  
 int count = 0;  
 for (int[] row : matrix) {  
 for (int value : row) {  
 product \*= Math.*abs*(value) + 1; // Додаємо 1, щоб уникнути нулів  
 count++;  
 }  
 }  
 return Math.*pow*(product, 1.0 / count) - 1;  
 }  
}

**Результат:**

****

**Висновок:** за цю лабораторну роботу, я навчився працювати з рандомними числами, роботи так, щоб була можливысть вводу з клавіатури, працювати з двовимірними масивами та реалізовувати методи для вичислення максимального та мінімального значення, розрахунок середнього арефметичного та геометричного числа