**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. 121-21-2

Ремез Максим Валентинович

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 java.util.Random;  
import java.util.Scanner;  
  
public class Matrix {  
 private static final int *MAX\_SIZE* = 20;  
 private static final int *MIN\_RANDOM* = 1;  
 private static final int *MAX\_RANDOM* = 100;  
  
 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 - рандомно): ");  
 int choice = scanner.nextInt();  
  
 if (choice == 1) {  
 matrix = *inputMatrix*(scanner, rows, cols);  
 } else {  
 matrix = *generateRandomMatrix*(rows, cols);  
 }  
  
 *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) {  
 int value;  
 do {  
 value = scanner.nextInt();  
 } while (value < 1 || value > *MAX\_SIZE*);  
 return value;  
 }  
  
 private static int[][] inputMatrix(Scanner scanner, int rows, int cols) {  
 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++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
 return matrix;  
 }  
  
 private static int[][] generateRandomMatrix(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\_RANDOM* - *MIN\_RANDOM* + 1) + *MIN\_RANDOM*;  
 }  
 }  
 return matrix;  
 }  
  
 private static void printMatrix(int[][] matrix) {  
 System.*out*.println("Матриця:");  
 for (int[] row : matrix) {  
 for (int value : row) {  
 System.*out*.print(value + " ");  
 }  
 System.*out*.println();  
 }  
 }  
  
 private static int findMin(int[][] matrix) {  
 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) {  
 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) {  
 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) {  
 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:**

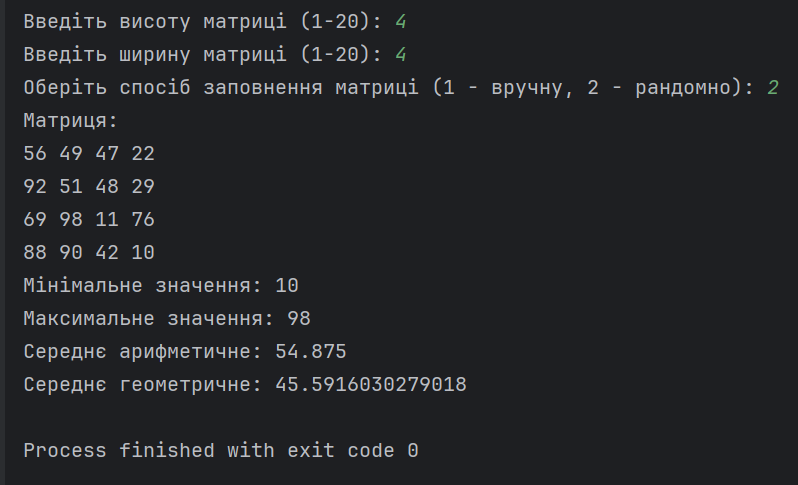


Рисунок 1 – Результат виконання коду ( rand числа 1 — 100 )

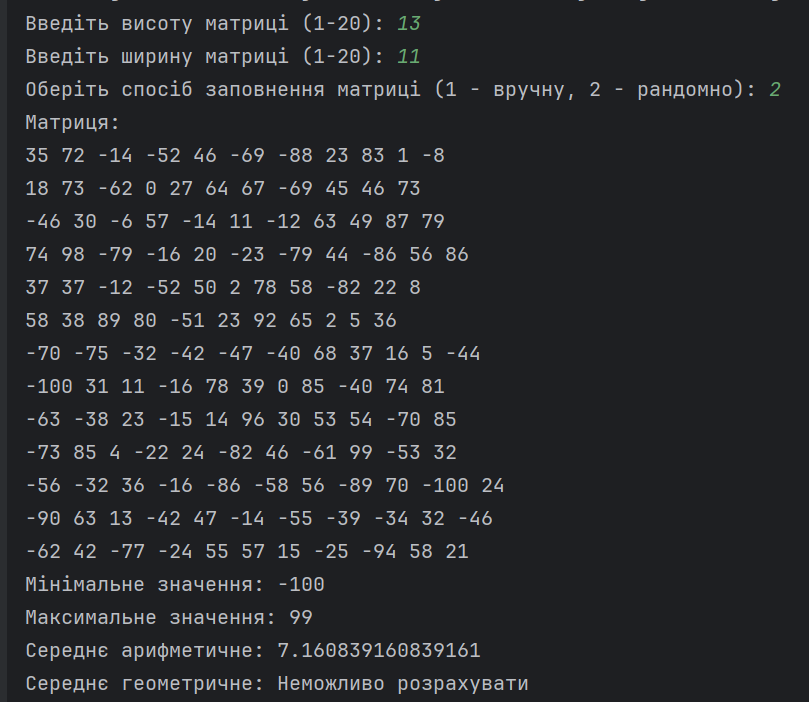


Рисунок 2 – Результат виконання коду ( rand числа -100 – 100 )

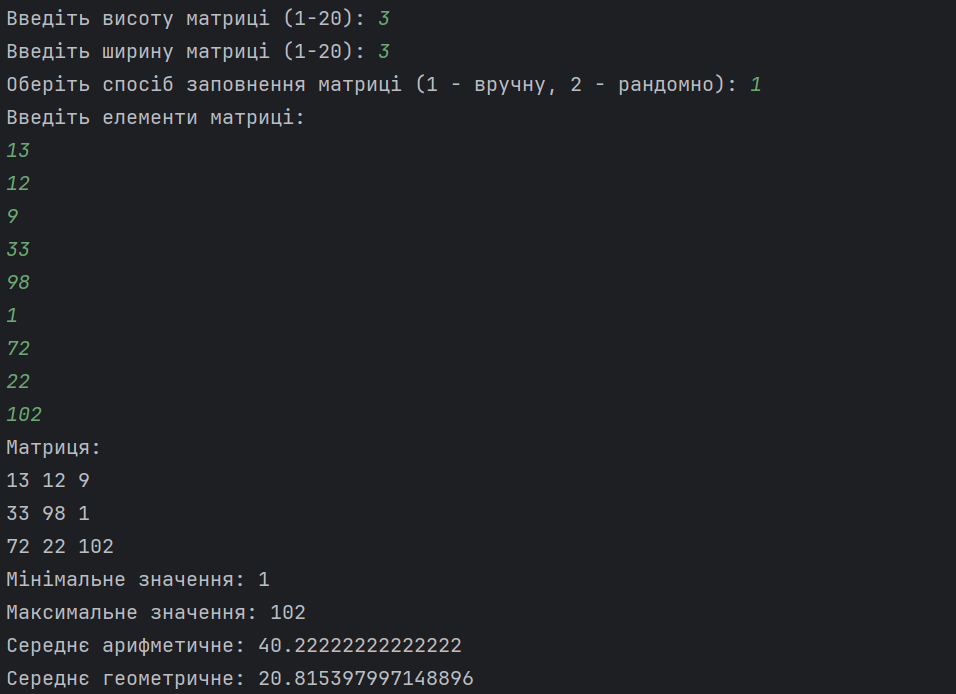


Рисунок 3 – Результат виконання коду ( Ручний ввід даних )