МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет

«Дніпровська політехніка»

Изображение выглядит как текст

Автоматически созданное описание

ЗВІТ

з лабораторної роботи №2

з дисципліни

**“Поглиблене програмування в середовищі Java”**

Виконав

студент групи

122-21-2

Бегма Кирило Григорович

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 зверігатись в спеціальних константах.

Як завдання підвищеної складності додати розрахунок середнього геометричного елементів матриці.

Хід роботи

1. Створюю гілку LR\_2.



Зображення, що містить текст, знімок екрана, Шрифт

Вміст, створений ШІ, може бути неправильним.

1. Створюю новий пакет для другої лабораторної роботи.

Створюю новий клас та код програми:

package Lab.LR\_2;  
  
import java.util.Random;  
import java.util.Scanner;  
  
public class MatrixOperations {  
  
 // Constants defining the range for random matrix values  
 private static final int *MIN\_ELEMENT\_VALUE* = -100;  
 private static final int *MAX\_ELEMENT\_VALUE* = 100;  
 private static final int *MAX\_SIZE* = 20;  
  
 // Method for manually inputting matrix values  
 public static int[][] inputMatrixManually(int rows, int cols) {  
 Scanner inputScanner = new Scanner(System.*in*);  
 int[][] matrix = new int[rows][cols];  
  
 System.*out*.println("Please input the matrix elements:");  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 System.*out*.printf("Element at matrix[%d][%d]: ", i, j);  
 matrix[i][j] = inputScanner.nextInt();  
 }  
 }  
  
 return matrix;  
 }  
  
 // Method to generate matrix with random values  
 public static int[][] generateRandomMatrix(int rows, int cols) {  
 Random randomGenerator = 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] = randomGenerator.nextInt(*MAX\_ELEMENT\_VALUE* \* 2 + 1) - *MAX\_ELEMENT\_VALUE*;  
 }  
 }  
  
 return matrix;  
 }  
  
 // Method to find the minimum value in the matrix  
 public static int getMinValue(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;  
 }  
  
 // Method to find the maximum value in the matrix  
 public static int getMaxValue(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;  
 }  
  
 // Method to calculate the arithmetic mean of matrix values  
 public static double calculateMean(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;  
 }  
  
 // Method to calculate the geometric mean of matrix values  
 public static double calculateGeoMean(int[][] matrix) {  
 // Ensure no negative values in the matrix for geometric mean calculation  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[i].length; j++) {  
 if (matrix[i][j] < 0) {  
 System.*out*.println("Geometric mean calculation not possible due to negative values.");  
 return Double.*NaN*;  
 }  
 }  
 }  
  
 // Calculate the product of all matrix elements  
 double product = 1;  
 int totalElements = 0;  
  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[i].length; j++) {  
 product \*= matrix[i][j];  
 totalElements++;  
 }  
 }  
  
 return Math.*pow*(product, 1.0 / totalElements);  
 }  
  
 // Method to display the matrix  
 public static void displayMatrix(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*);  
  
 // Input the dimensions of the matrix  
 System.*out*.print("Enter number of rows for the matrix: ");  
 int rows = scanner.nextInt();  
 System.*out*.print("Enter number of columns for the matrix: ");  
 int cols = scanner.nextInt();  
  
 if (rows > *MAX\_SIZE* || cols > *MAX\_SIZE*) {  
 System.*out*.println("Matrix size cannot exceed 20x20.");  
 return;  
 }  
  
 // Select how the matrix will be created  
 System.*out*.println("Select the matrix creation method:");  
 System.*out*.println("1. Manual input");  
 System.*out*.println("2. Random generation");  
 int choice = scanner.nextInt();  
  
 int[][] matrix = null;  
 if (choice == 1) {  
 matrix = *inputMatrixManually*(rows, cols);  
 } else if (choice == 2) {  
 matrix = *generateRandomMatrix*(rows, cols);  
 } else {  
 System.*out*.println("Invalid option selected.");  
 return;  
 }  
  
 // Display the matrix  
 System.*out*.println("Matrix:");  
 *displayMatrix*(matrix);  
  
 // Calculate results  
 int min = *getMinValue*(matrix);  
 int max = *getMaxValue*(matrix);  
 double arithmeticMean = *calculateMean*(matrix);  
 double geometricMean = *calculateGeoMean*(matrix);  
  
 // Display the results  
 System.*out*.println("Minimum value: " + min);  
 System.*out*.println("Maximum value: " + max);  
 System.*out*.println("Arithmetic mean: " + arithmeticMean);  
 System.*out*.println("Geometric mean: " + (Double.*isNaN*(geometricMean) ? "Not possible to calculate" : geometricMean));  
 }  
}

1. Додаю всі зміни та виконую коміт.
2. Запушую гілку на гітхаб.
3. Створюю Pull Request на GitHub та виконую Merge pull request.
4. Також оновлюю локальну версію гілки:

Зображення, що містить текст, знімок екрана, Шрифт

Вміст, створений ШІ, може бути неправильним.

Бачимо результат:

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Вміст, створений ШІ, може бути неправильним.

Результат виконання створеної програми:

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Вміст, створений ШІ, може бути неправильним.

Висновок. Під час виконання лабораторної роботи №2 з дисципліни «Поглиблене програмування в середовищі Java» було вивчено основи мови Java та створено програму згідно завданню.