**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**

****

**ЗВІТ**

**Практична робота №3**

**з дисципліни**

# «Поглиблене програмування в середовищі Java»

**Виконав:**

студент гр. 122-21-3

Кабаченко О. В.

**Прийняв:**

Доцент каф. САУ

Алексєєв О. М.

**м. Дніпро**

**2025  рік**

**Хід роботи**

1) Для початку в існуючому проекті було створено новий пакет «lab2», до якого додано файл – «Matrix». На рисунку 1 наведено результат роботи.

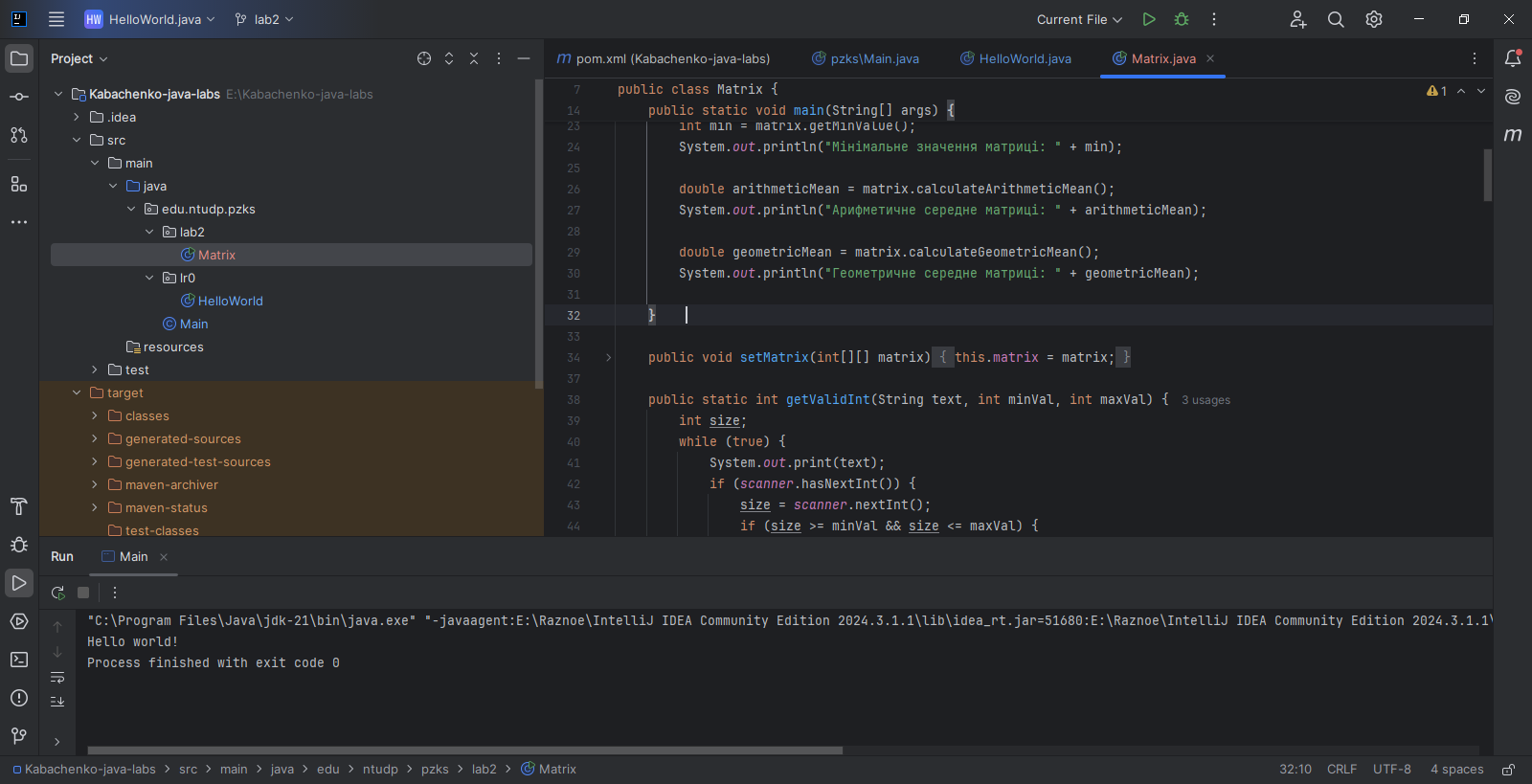


Рисунок 1 – Наповнення нового пакету

2) Файл «Matrix» буде містити однойменний клас, який відповідальний за об’єкт матриці та надає такі методи як створення матриці(різними способами), вивід матриці на екран, а також методи обчислення таких показників як арифметичне і геометричне середнє, знаходження мінімального/максимального елементу матриці. Нижче наведено свторений програмний код.

package edu.ntudp.pzks.lab2;  
  
import java.util.Random;  
import java.util.Scanner;  
import java.util.stream.Stream;  
  
public class Matrix {  
 private static final int *MAX\_SIZE* = 20;  
 private static final int *MIN\_RANDOM* = 1;  
 private static final int *MAX\_RANDOM* = 100;  
 private static final Scanner *scanner* = new Scanner(System.*in*);  
 private int[][] matrix;  
  
 public void setMatrix(int[][] matrix) {  
 this.matrix = matrix;  
 }  
  
 public static int getValidInt(String text, int minVal, int maxVal) {  
 int size;  
 while (true) {  
 System.*out*.print(text);  
 if (*scanner*.hasNextInt()) {  
 size = *scanner*.nextInt();  
 if (size >= minVal && size <= maxVal) {  
 return size;  
 }  
 } else {  
 *scanner*.next();  
 }  
 System.*out*.println("Invalid value, please try again.");  
 }  
 }  
  
 public static int[] getMatrixSizes() {  
 int rows = *getValidInt*("Enter the number of rows of the matrix:", 1, *MAX\_SIZE*);  
 int cols = *getValidInt*("Enter the number of columns of the matrix:", 1, *MAX\_SIZE*);  
 return new int[]{rows, cols};  
 }  
  
 public void createMatrixRandomly() {  
 int[] sizes = *getMatrixSizes*();  
 int rows = sizes[0];  
 int cols = sizes[1];  
  
 int[][] matrix = new int[rows][cols];  
 Random rand = new Random();  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 matrix[i][j] = rand.nextInt(*MAX\_RANDOM* - *MIN\_RANDOM* + 1) + *MIN\_RANDOM*;  
 }  
 }  
 setMatrix(matrix);  
 }  
  
 public void createMatrixManually() {  
 int[] sizes = *getMatrixSizes*();  
 int rows = sizes[0];  
 int cols = sizes[1];  
 int[][] matrix = new int[rows][cols];  
  
 System.*out*.println("Enter the elements of the matrix:");  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 while (true) {  
 System.*out*.print("Element [" + i + "][" + j + "]: ");  
 if (*scanner*.hasNextInt()) {  
 matrix[i][j] = *scanner*.nextInt();  
 break;  
 } else {  
 System.*out*.println("Invalid value, please try again.");  
 *scanner*.next();  
 }  
 }  
 }  
 }  
 this.matrix = matrix;  
 }  
  
 public void createMatrix() {  
 System.*out*.println("Choose how to create the matrix:");  
 System.*out*.println("1. Create the matrix randomly");  
 System.*out*.println("2. Manually enter matrix elements");  
 int usrInput = *getValidInt*("Enter your choice (1 or 2): ", 1, 2);  
  
 if (usrInput ==1){  
 createMatrixRandomly();  
 }  
 else{  
 createMatrixManually();  
 }  
 }  
  
 public void printMatrix() {  
 if (matrix != null) {  
 System.*out*.println("Matrix:");  
 for (int[] row : matrix) {  
 for (int num : row) {  
 System.*out*.printf("%5d", num);  
 }  
 System.*out*.println();  
 }  
 }  
 }  
  
 public int getMaxValue() {  
 int max = matrix[0][0];  
 for (int[] row : matrix) {  
 for (int num : row) {  
 if (num > max) max = num;  
 }  
 }  
 return max;  
 }  
  
 public int getMinValue() {  
 int min = matrix[0][0];  
 for (int[] row : matrix) {  
 for (int num : row) {  
 if (num < min) min = num;  
 }  
 }  
 return min;  
 }  
  
 public double calculateArithmeticMean() {  
 int sum = 0;  
 int elementsCount = Stream.*of*(matrix).mapToInt(m -> m.length).sum();  
 for (int[] row : matrix) {  
 for (int num : row) {  
 sum += num;  
 }  
 }  
 return (double) sum / elementsCount;  
 }  
  
 public double calculateGeometricMean() {  
 double product = 1.0;  
 int elementsCount = Stream.*of*(matrix).mapToInt(m -> m.length).sum();  
 for (int[] row : matrix) {  
 for (int num : row) {  
 if (num < 0) {  
 System.*out*.println("Geometric mean cannot be calculated for negative numbers!");  
 return 0;  
 }  
  
 product \*= num;  
 }  
 }  
 return Math.*pow*(product, 1.0 / elementsCount);  
 }  
  
 public int[][] getMatrix() {  
 return matrix;  
 }  
}

3) Далі до класу додано метод main, який буде здійснювати всі основні операції з матрицею для демонстрації роботи методів.. Нижче наведено програмний код методу, а на рисунку 2 можливо переглянути результат виконання програми.

public static void main(String[] args) {  
  
 Matrix matrix = new Matrix();  
 matrix.createMatrix();  
 matrix.printMatrix();  
  
 int max = matrix.getMaxValue();  
 System.*out*.println("Maximum value of the matrix: " + max);  
  
 int min = matrix.getMinValue();  
 System.*out*.println("Minimum value of the matrix: " + min);  
  
 double arithmeticMean = matrix.calculateArithmeticMean();  
 System.*out*.println("Arithmetic mean of the matrix: " + arithmeticMean);  
  
 double geometricMean = matrix.calculateGeometricMean();  
 System.*out*.println("Geometric mean of the matrix: " + geometricMean);  
  
}

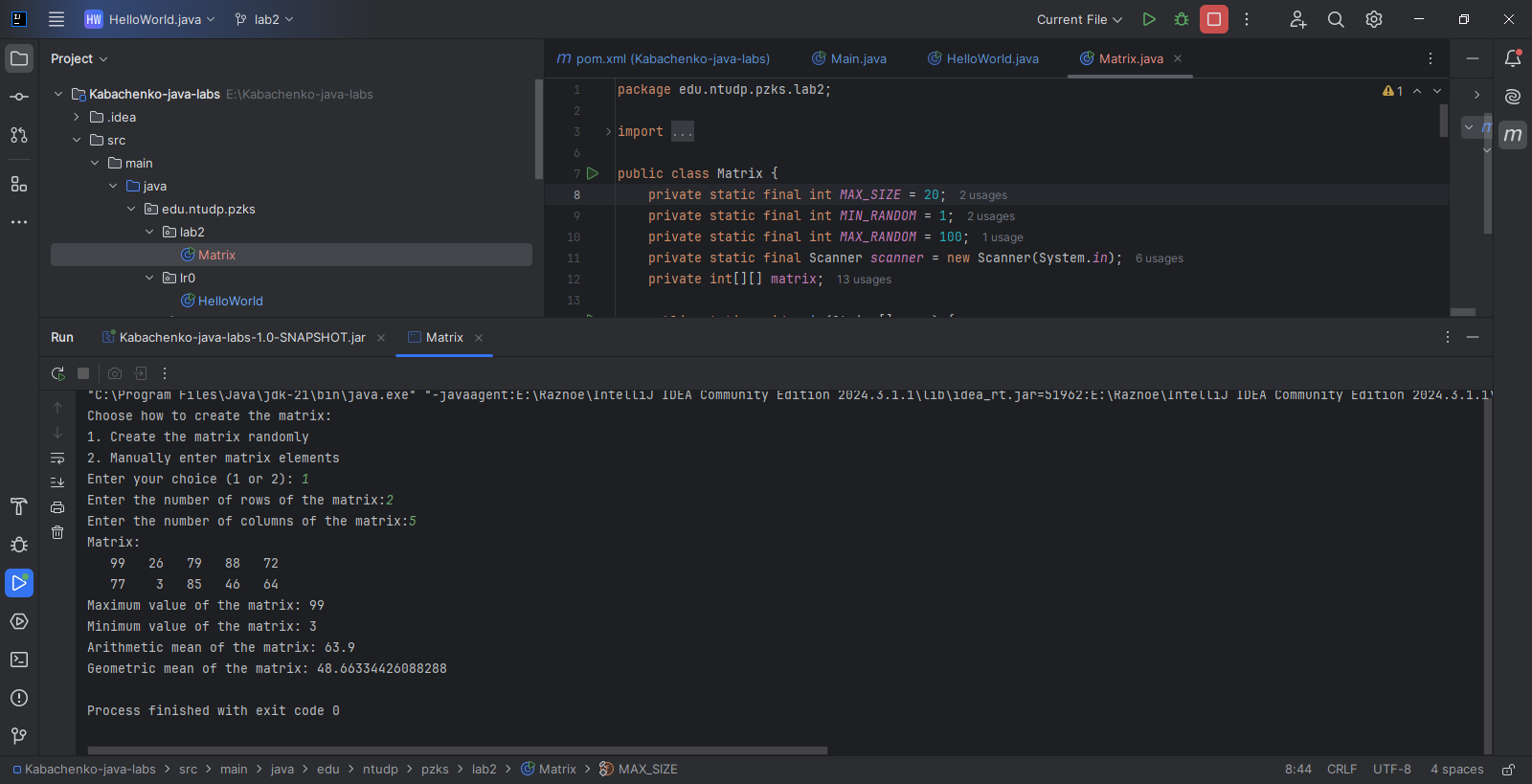


Рисунок 2 – Результат виконання програми.

4) Далі у файлі Main зробимо виклик методу main класу Matrix, як це наведено нижче. На рисунку 3 зображено результат роботи усієї програми.

package edu.ntudp.pzks;  
  
import edu.ntudp.pzks.lr0.HelloWorld;  
import edu.ntudp.pzks.lab2.Matrix;  
  
public class Main {  
 public static void main(String[] args){  
 HelloWorld.*main*(args);  
 Matrix.*main*(args);  
 }  
}

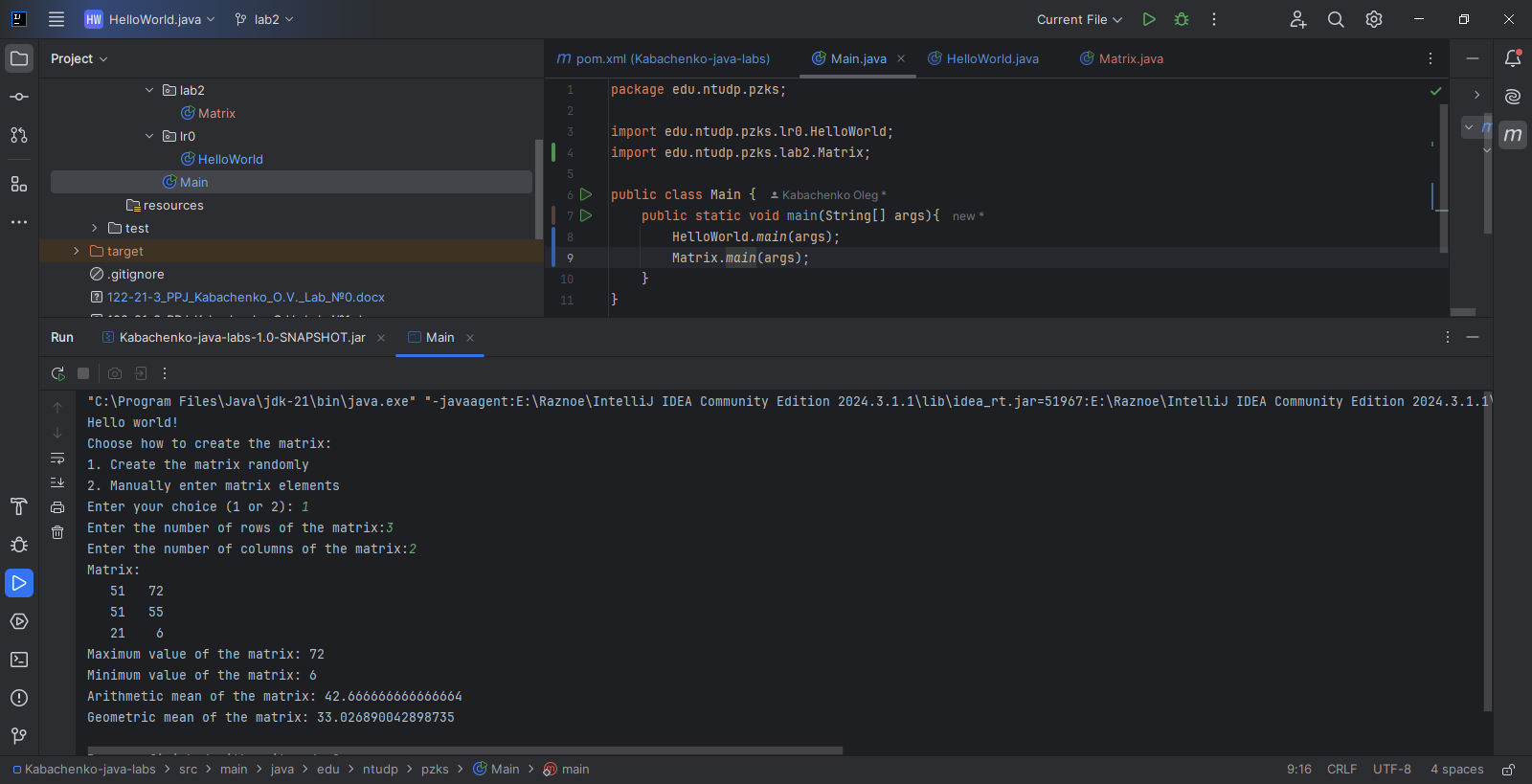


Рисунок 3 – Результат роботи файлу Main

**Висновки**

У ході виконання лабораторної роботи було розглянуто розроблено програму для створення та обробки матриць цілих чисел. Програма дозволяє користувачеві вибирати метод створення матриці через консоль, використовуючи клас Scanner для введення даних. Визначено обмеження на розміри матриці, що не можуть перевищувати 20 на 20 елементів. Для випадкового заповнення матриці використовуються константи, які обмежують діапазон чисел. Усі операції розбиті на окремі методи для полегшення розуміння і підтримки коду.

Лабораторна робота допомогла розвинути навички роботи з масивами та методами, а також з введенням і виведенням даних за допомогою класу Scanner. Виконання цього завдання дозволило здобути досвід у розробці програм з обчисленнями та статистикою на основі введених користувачем даних.