КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

Кафедра мікроелектроніки

**ЛАБОРАТОРНА РОБОТА №10**

з дисципліни «Обчислювальна математика»

Варіант 3

Роботу виконав

Ст. групи ДП-01

Деркач Євген

Роботу перевірив

Татарчук Д. Д.

Київ-2021

**ЛАБОРАТОРНА РОБОТА №10**

**Тема:** Власні вектори і власні числа матриць з елементами-дійсними числами

**Мета роботи:** наочна геометрична інтерпретація власних векторів і власних чисел матриць 2 x 2 з дійсними елементами.

**Що зробити:** в двовимірному просторі (площині) здійснити прямий перебор (з певним малим кроком) одиничних векторів x всіх можливих напрямків і для кожного з них порівняти напрямок результату добутку y = Ax, де A – матриця 2 x 2, з напрямком самого x. Визначити таким чином власні вектори і власні числа матриці A та порівняти їх з величинами, розрахованими аналітично за допомогою характеристичного рівняння.

**Код програми:**

package com.Jeka8833.LabsKPITwo.lab.labs;  
  
import com.Jeka8833.LabsKPITwo.Column;  
import com.Jeka8833.LabsKPITwo.ForceStopException;  
import com.Jeka8833.LabsKPITwo.Reader;  
import com.Jeka8833.LabsKPITwo.Writer;  
import com.Jeka8833.LabsKPITwo.lab.Lab;  
  
public class Laba10 implements Lab {  
  
  
 @Override  
 public String getName() {  
 return "No 10. Власні вектори і власні числа матриць з елементами-дійсними числами";  
 }  
  
 @Override  
 public void run() throws ForceStopException {  
 final double[][] a = Reader.*readMatrix*(2, 2, "Введём матрицу СЛАР A[" + 2 + "," + 2 + "], можно написать random - для авто заполнения матрицы");  
 Writer.*reset*();  
 Writer.*add*(new Column("rad", Double.class), new Column("vecX - x", Double.class), new Column("vecX - y", Double.class)  
 , new Column("vecY - x", Double.class), new Column("vecY - y", Double.class));  
 for (int i = 0; i < 360; i++) {  
 final double rad = Math.*toRadians*(i);  
 final double y1 = a[0][0] \* Math.*cos*(rad) + a[0][1] \* Math.*sin*(rad);  
 final double y2 = a[1][0] \* Math.*cos*(rad) + a[1][1] \* Math.*sin*(rad);  
 Writer.*add*(rad, Math.*cos*(rad), Math.*sin*(rad), y1, y2);  
 }  
 Writer.*add*(new Column("x", Double.class), new Column("y", Double.class));  
 for (int i = 0; i < 360; i++) {  
 final double rad = Math.*toRadians*(i);  
 final double y1 = a[0][0] \* Math.*cos*(rad) + a[0][1] \* Math.*sin*(rad);  
 final double y2 = a[1][0] \* Math.*cos*(rad) + a[1][1] \* Math.*sin*(rad);  
 if (Math.*abs*(Math.*sin*(rad) - y2) < Math.*toRadians*(1) ||  
 Math.*abs*(Math.*abs*(Math.*sin*(rad) - y2) - Math.*PI*) < Math.*toRadians*(1)) {  
 Writer.*add*(0., 0.);  
 Writer.*add*(y1, y2);  
 }  
 }  
 Writer.*saveResult*();  
 }  
}