НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №5

з дисципліни **«**Алгоритми та методи обчислень**»**

Виконав:

студентка 2 курсу

ФІОТ гр. ІО-32

Руденко Тетяна

Перевірив:

Порєв В.М.

Київ 2015р.

Метод визначників Крамера

**Варіант 4**

Матриця:

1 1 1

2 3 1

1 -1 -1

Стовпець вільних членів: 4, 9, -2

**Лістинг:**

package Lab5;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.Rectangle;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Arrays;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

public class myWindow extends JFrame {

public int [][] matrix = new int[3][3];

public int [] vector = new int[3];

public JPanel panel1 = new JPanel();

public JPanel panel2 = new JPanel();

public JPanel panel3 = new JPanel();

public JTextField m11 = new JTextField();

public JTextField m12 = new JTextField();

public JTextField m13 = new JTextField();

public JTextField m21 = new JTextField();

public JTextField m22 = new JTextField();

public JTextField m23 = new JTextField();

public JTextField m31 = new JTextField();

public JTextField m32 = new JTextField();

public JTextField m33 = new JTextField();

public JTextField v1 = new JTextField();

public JTextField v2 = new JTextField();

public JTextField v3 = new JTextField();

public JLabel rootsL = new JLabel(" ");

public JLabel matrixL = new JLabel("Матриця коефіціентів");

public JLabel vectorL = new JLabel("Вільні члени");

public JButton button = new JButton("Розв'язати");

public myWindow(){

super("Labwork 5");

setBounds(500, 130, 700, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.add(panel1, BorderLayout.WEST);

this.add(panel3, BorderLayout.EAST);

panel1.setLayout(new GridLayout(4,3));

panel1.add(new JLabel(" "));

panel1.add(matrixL);

panel1.add(new JLabel(" "));

panel1.add(m11);

m11.setColumns(3);

panel1.add(m12);

m12.setColumns(3);

panel1.add(m13);

m13.setColumns(3);

panel1.add(m21);

m21.setColumns(3);

panel1.add(m22);

m22.setColumns(3);

panel1.add(m23);

m23.setColumns(3);

panel1.add(m31);

m31.setColumns(3);

panel1.add(m32);

m32.setColumns(3);

panel1.add(m33);

m33.setColumns(3);

this.add(panel2, BorderLayout.CENTER);

panel2.setLayout(new GridLayout(4,1));

panel2.setBounds(new Rectangle(20,150));

panel2.add(vectorL);

panel2.add(v1);

v1.setColumns(3);

panel2.add(v2);

v2.setColumns(3);

panel2.add(v3);

v3.setColumns(3);

panel3.add(rootsL);

this.add(button, BorderLayout.SOUTH);

m11.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[0][0] = Integer.parseInt(m11.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[0][0] = Integer.parseInt(m11.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m12.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[0][1] = Integer.parseInt(m12.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[0][1] = Integer.parseInt(m12.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m13.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[0][2] = Integer.parseInt(m13.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[0][2] = Integer.parseInt(m13.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m21.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[1][0] = Integer.parseInt(m21.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[1][0] = Integer.parseInt(m21.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m22.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[1][1] = Integer.parseInt(m22.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[1][1] = Integer.parseInt(m22.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m23.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[1][2] = Integer.parseInt(m23.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[1][2] = Integer.parseInt(m23.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m31.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[2][0] = Integer.parseInt(m31.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[2][0] = Integer.parseInt(m31.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m32.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[2][1] = Integer.parseInt(m32.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[2][1] = Integer.parseInt(m32.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

m33.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

matrix[2][2] = Integer.parseInt(m33.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

matrix[2][2] = Integer.parseInt(m33.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

v1.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

vector[0] = Integer.parseInt(v1.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

vector[0] = Integer.parseInt(v1.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

v2.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

vector[1] = Integer.parseInt(v2.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

vector[1] = Integer.parseInt(v2.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

v3.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void insertUpdate(DocumentEvent ev) {

vector[2] = Integer.parseInt(v3.getText());

}

@Override

public void removeUpdate(DocumentEvent ev) {

vector[2] = Integer.parseInt(v3.getText());

}

@Override

public void changedUpdate(DocumentEvent ev) {

}

});

button.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ev) {

m11.setEnabled(false);

m12.setEnabled(false);

m13.setEnabled(false);

m21.setEnabled(false);

m22.setEnabled(false);

m23.setEnabled(false);

m31.setEnabled(false);

m32.setEnabled(false);

m33.setEnabled(false);

v1.setEnabled(false);

v2.setEnabled(false);

v3.setEnabled(false);

int [] roots = roots();

String solve = "x1 = " + roots[0]+ "; x2 = " + roots[1] + "; x3 = " + roots[2];

if (roots() != null) {

rootsL.setText(solve);

}

else rootsL.setText("Рівняння не має розв'язків");

}

});

}

public int determinant(int [][] m){

int d = 0;

if (m.length == 3)

d = m[0][0]\*m[1][1]\*m[2][2] + m[2][0]\*m[0][1]\*m[1][2] + m[1][0]\*m[2][1]\*m[0][2] -

m[2][0]\*m[1][1]\*m[0][2] - m[0][0]\*m[2][1]\*m[1][2] - m[1][0]\*m[0][1]\*m[2][2];

else d = Integer.MAX\_VALUE;

return d;

}

public int [] roots(){

int[] v = new int [3];

int [][] m = new int[3][3];

int [][] m2 = Arrays.copyOf(matrix, 3);

int det = determinant(matrix);

if ((det != 0) && (det != Integer.MAX\_VALUE)){

for(int i = 0; i < 3; i++){

for(int k = 0; k < 3; k++){

for(int l = 0; l < 3; l++){

m[k][l] = matrix[k][l];

}

}

for(int j = 0; j < 3; j++){

m[j][i] = vector[j];

}

v[i] = determinant(m)/det;

}

}

else v = null;

return v;

}

}

Результат роботи програми:

Висновок: у даній лабораторній роботі ми вивчили основні методи розв'язання систем рівнянь і закріпили отримані знання, реалізувавши програмно метод визначників Крамера.