НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №5

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

Виконав:

студент 2 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Порєв В.М.

Київ 2014 р.

Варіант завдання: метод визначників Крамера

**public** **class** Test {

**public** **static** **void** main(String[] args) {

LabView frame = **new** LabView();

frame.setVisible(**true**);

}

}

**import** java.util.Arrays;

**public** **class** Logic {

**private** **int** size;

**private** **double**[][] matrix;

**private** **double** [] result;

**public** Logic(**int** size, **double** mas [] []) {

calc(size, mas);

}

**private** **void** calc(**int** size, **double** mas [] []){

**this**.size = size;

**this**.matrix = mas;

**this**.result = **new** **double** [size];

**double** [] [] masBuf = **new** **double** [size][size];

**for** (**int** i = 0; i < masBuf.length; i++) {

masBuf [i] = Arrays.*copyOf*(mas[i], size);

}

**double** mainDet = *determinant*(masBuf);

**for** (**int** i = 0; i < result.length; i++) {

**for** (**int** j2 = 0; j2 < masBuf.length; j2++) {

**for** (**int** k = 0; k < masBuf[j2].length; k++) {

masBuf [j2][k] = matrix[j2][k];

}

}

**for** (**int** k = 0; k < masBuf.length; k++) {

masBuf[k][i] = matrix[k][matrix.length];

}

result[i] = *determinant*(masBuf);

}

**for** (**int** i = 0; i < result.length; i++) {

result[i] = result[i]/mainDet;

}

}

**private** **static** **double** determinant(**double**[][] mat) {

// В эту переменную будем складывать промежуточные результаты

**double** result = 0;

// Если у нас матрица из одного столбца и одной строки - возвращаем элемент нулевого столбца нулевой строки, который у этой матрицы единственный (отсчёт ведётся с нуля)

**if**(mat.length == 1) {

result = mat[0][0];

**return** result;

}

// Если у нас матрица 2х2, то возвращаем разность произведений элементов главной и побочной диагоналей. Тоже ничего сложного

**if**(mat.length == 2) {

result = mat[0][0] \* mat[1][1] - mat[0][1] \* mat[1][0];

**return** result;

}

// Вот тут начинается жара))

// Пробегаемся переменной i от 0 до (количество элементов в строке минус один) - в общем, по всем индексам элементов строки

**for**(**int** i = 0; i < mat[0].length; i++) {

// создаём временный массив temp из вещественных чисел размером [кол. элементов с столбце минус один Х кол. эл. в строке минус один]

**double** temp[][] = **new** **double**[mat.length - 1][mat[0].length - 1];

// Раскладываем матрицу по столбцу: Мы должны сформировать матрицу для подсчёта минора для элемента матрицы a[0][i]

// Минор для элемента a[j][i] - это определитель матрицы, получающейся, если выбросить строку и столбец, в которых располагается элемент a[j][i]

// Получается, мы выбрасываем из матрицы верхнюю строчку и столбец i

// Так как мы выбрасываем верхнюю строчку, то переменную j мы прогоняем не с 0, а с 1

**for**(**int** j = 1; j < mat.length; j++) {

// Из каждой последующей строчки нам надо скопировать в новый массив temp элементы, которые идут до i-го элемента и элементы, которые идут после i-го элемента. Но не сам i-й элемент.

// функция System.arraycopy принимает параметры: (массив откуда копировать, смещение откуда копировать, массив куда копировать, смещение куда копировать, количество копируемых элементов)

System.*arraycopy*(mat[j], 0, temp[j-1], 0, i);

System.*arraycopy*(mat[j], i+1, temp[j-1], i, mat[0].length-i-1);

}

// чтобы получить минор, нам надо найти определитель матрицы temp

// чтобы получить знак перестановки, нам надо возвести число (-1) в степень i(индекс, а не комплексное число). Почему это так, я не помню, но это как-то доказывается в теореме о разложении по столбцу/строке

// Ну и вот, прибавляем к уже существующему результату элемент матрицы a[0][i], умноженный на знак перестановки и умноженный на определитель матрицы temp.

result += mat[0][i] \* Math.*pow*(-1, i) \* *determinant*(temp);

}

// Вот такая вот рекурсивная функция получилась. Возвращаем результат.

**return** result;

}

**public** **double**[] getResult (){

**return** **this**.result;

}

}**import** java.awt.BorderLayout;

**public** **class** LabView **extends** JFrame {

**private** JScrollPane scrollPaneSize = **new** JScrollPane();

**private** JScrollPane scrollPaneCouef = **new** JScrollPane();

**private** JButton buttonCoef = **new** JButton(

"\u0412\u0432\u0435\u0441\u0442\u0438 \u043A\u043E\u0435\u0444\u0456\u0446\u0456\u0454\u043D\u0442\u0438");

**private** JButton buttonCalc = **new** JButton(

"\u041E\u0431\u0447\u0438\u0441\u043B\u0438\u0442\u0438");

**private** JList<Integer> listSize;

**private** **int** k;

**private** **double**[][] data;

**private** JPanel contentPane;

**private** Integer size;

**private** **final** JLabel label1 = **new** JLabel("X1 = ");

**private** **final** JLabel label2 = **new** JLabel("X2= ");

**private** **final** JLabel label4 = **new** JLabel("X4 = ");

**private** **final** JLabel label3 = **new** JLabel("X3 = ");

**private** **final** JLabel label5 = **new** JLabel("X5 = ");

**private** **final** JLabel x1 = **new** JLabel("");

**private** **final** JLabel x2 = **new** JLabel("");

**private** **final** JLabel x3 = **new** JLabel("");

**private** **final** JLabel x4 = **new** JLabel("");

**private** **final** JLabel x5 = **new** JLabel("");

/\*\*

\* Create the frame.

\*/

**public** LabView() {

label1.setVisible(**false**);

label2.setVisible(**false**);

label3.setVisible(**false**);

label4.setVisible(**false**);

label5.setVisible(**false**);

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setBounds(100, 100, 462, 303);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel label = **new** JLabel(

"\u041A\u0456\u043B\u044C\u043A\u0456\u0441\u0442\u044C \u0440\u0456\u0432\u043D\u044F\u043D\u044C \u0432 \u0441\u0438\u0441\u0442\u0435\u043C\u0456");

label.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label.setBounds(10, 11, 174, 23);

contentPane.add(label);

buttonCoef.setBounds(248, 12, 149, 23);

contentPane.add(buttonCoef);

JLabel label\_1 = **new** JLabel(

"\u0420\u0435\u0437\u0443\u043B\u044C\u0442\u0430\u0442:");

label\_1.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label\_1.setBounds(10, 45, 101, 23);

contentPane.add(label\_1);

buttonCalc.setBounds(117, 46, 101, 22);

contentPane.add(buttonCalc);

Integer[] listData = { 2, 3, 4, 5 };

listSize = **new** JList<>(listData);

listSize.setVisible(**true**);

listSize.setVisibleRowCount(1);

scrollPaneSize = **new** JScrollPane(listSize);

listSize.addListSelectionListener(**new** ListSelectionListener() {

@Override

**public** **void** valueChanged(ListSelectionEvent even) {

**if** (even.getValueIsAdjusting() == **false**) {

label1.setVisible(**true**);

label2.setVisible(**true**);

label3.setVisible(**true**);

label4.setVisible(**true**);

label5.setVisible(**true**);

LabView.**this**.size = listSize.getSelectedValue();

data = **new** **double**[size][size + 1];

**switch** (size) {

**case** 4:

label5.setVisible(**false**);

**break**;

**case** 3:

label5.setVisible(**false**);

label4.setVisible(**false**);

**break**;

**case** 2:

label3.setVisible(**false**);

label4.setVisible(**false**);

label5.setVisible(**false**);

**break**;

**default**:

**break**;

}

}

}

});

buttonCoef.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**for** (**int** i = 0; i < size; i++) {

InpunData d = **new** InpunData(LabView.**this**, size);

d.setVisible(**true**);

}

}

});

buttonCalc.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**for** (**int** i = 0; i < data.length; i++) {

**for** (**int** j = 0; j < data[0].length; j++) {

System.*out*.print(data[i][j]+"|");

}

System.*out*.println();

}

Logic l = **new** Logic(size, data);

**double**[] res = l.getResult();

**switch** (size) {

**case** 2:

x1.setText(Double.*toString*(res[0]));

x2.setText(Double.*toString*(res[1]));

x1.updateUI();

x2.updateUI();

**break**;

**case** 3:

x1.setText(Double.*toString*(res[0]));

x2.setText(Double.*toString*(res[1]));

x3.setText(Double.*toString*(res[2]));

x1.updateUI();

x2.updateUI();

x3.updateUI();

**break**;

**case** 4:

x1.setText(Double.*toString*(res[0]));

x2.setText(Double.*toString*(res[1]));

x3.setText(Double.*toString*(res[2]));

x4.setText(Double.*toString*(res[3]));

x1.updateUI();

x1.updateUI();

x2.updateUI();

x4.updateUI();

**break**;

**case** 5:

x1.setText(Double.*toString*(res[0]));

x2.setText(Double.*toString*(res[1]));

x3.setText(Double.*toString*(res[2]));

x4.setText(Double.*toString*(res[3]));

x5.setText(Double.*toString*(res[4]));

x1.updateUI();

x1.updateUI();

x1.updateUI();

x2.updateUI();

x5.updateUI();

**break**;

**default**:

**break**;

}

}

});

scrollPaneSize.setBounds(194, 11, 36, 23);

contentPane.add(scrollPaneSize);

label1.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label1.setBounds(10, 79, 46, 14);

contentPane.add(label1);

label2.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label2.setBounds(10, 104, 46, 14);

contentPane.add(label2);

label4.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label4.setBounds(10, 154, 46, 14);

contentPane.add(label4);

label3.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label3.setBounds(10, 129, 46, 14);

contentPane.add(label3);

label5.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label5.setBounds(10, 179, 46, 14);

contentPane.add(label5);

x1.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

x1.setBounds(66, 79, 46, 14);

contentPane.add(x1);

x2.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

x2.setBounds(66, 104, 46, 14);

contentPane.add(x2);

x3.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

x3.setBounds(66, 129, 46, 14);

contentPane.add(x3);

x4.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

x4.setBounds(66, 154, 46, 14);

contentPane.add(x4);

x5.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

x5.setBounds(66, 179, 46, 14);

contentPane.add(x5);

}

**public** **void** update(**double**[] d) {

data[k] = Arrays.*copyOf*(d, d.length);

k++;

}

}

**import** java.awt.BorderLayout;

**public** **class** InpunData **extends** JDialog {

**private** **final** JPanel contentPanel = **new** JPanel();

**private** JTextField textField3 = **new** JTextField();

**private** JTextField textField4 = **new** JTextField();

**private** JTextField textField5 = **new** JTextField();

**private** JTextField textFieldEnd = **new** JTextField();

**private** JTextField textField1 = **new** JTextField();

**private** JTextField textField2 = **new** JTextField();

JLabel label1 = **new** JLabel("X1");

JLabel label2 = **new** JLabel("X2");

JLabel label3 = **new** JLabel("X3");

JLabel label4 = **new** JLabel("X4");

JLabel label5 = **new** JLabel("X5");

**double** [] d;

**int** size;

LabView dataView;

/\*\*

\* Launch the application.

\*/

/\*\*

\* Create the dialog.

\*/

**public** InpunData(LabView data, **int** size) {

**this**.dataView = data;

**this**.size = size;

**this**.d = **new** **double** [size+1];

setBounds(100, 100, 339, 124);

getContentPane().setLayout(**new** BorderLayout());

contentPanel.setBorder(**new** EmptyBorder(5, 5, 5, 5));

getContentPane().add(contentPanel, BorderLayout.*CENTER*);

contentPanel.setLayout(**null**);

setVisibleTrue(size);

{

JPanel buttonPane = **new** JPanel();

buttonPane.setLayout(**new** FlowLayout(FlowLayout.*RIGHT*));

getContentPane().add(buttonPane, BorderLayout.*SOUTH*);

{

JButton okButton = **new** JButton("OK");

okButton.setActionCommand("OK");

buttonPane.add(okButton);

getRootPane().setDefaultButton(okButton);

okButton.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**switch** (InpunData.**this**.size) {

**case** 2:

InpunData.**this**.d = **new** **double** [3];

InpunData.**this**.d[0] = Double.*parseDouble*(textField1.getText());

InpunData.**this**.d[1] = Double.*parseDouble*(textField2.getText());

InpunData.**this**.d[2] = Double.*parseDouble*(textFieldEnd.getText());

**break**;

**case** 3:

InpunData.**this**.d = **new** **double** [4];

InpunData.**this**.d[0] = Double.*parseDouble*(textField1.getText());

InpunData.**this**.d[1] = Double.*parseDouble*(textField2.getText());

InpunData.**this**.d[2] = Double.*parseDouble*(textField3.getText());

InpunData.**this**.d[3] = Double.*parseDouble*(textFieldEnd.getText());

**break**;

**case** 4:

InpunData.**this**.d = **new** **double** [5];

InpunData.**this**.d[0] = Double.*parseDouble*(textField1.getText());

InpunData.**this**.d[1] = Double.*parseDouble*(textField2.getText());

InpunData.**this**.d[2] = Double.*parseDouble*(textField3.getText());

InpunData.**this**.d[3] = Double.*parseDouble*(textField4.getText());

InpunData.**this**.d[4] = Double.*parseDouble*(textFieldEnd.getText());

**break**;

**case** 5:

InpunData.**this**.d = **new** **double** [5];

InpunData.**this**.d[0] = Double.*parseDouble*(textField1.getText());

InpunData.**this**.d[1] = Double.*parseDouble*(textField2.getText());

InpunData.**this**.d[2] = Double.*parseDouble*(textField3.getText());

InpunData.**this**.d[3] = Double.*parseDouble*(textField4.getText());

InpunData.**this**.d[4] = Double.*parseDouble*(textField5.getText());

InpunData.**this**.d[5] = Double.*parseDouble*(textFieldEnd.getText());

**break**;

**default**:

**break**;

}

**for** (**int** i = 0; i < InpunData.**this**.d.length; i++) {

System.*out*.println(InpunData.**this**.d[i]+"|");

}

dataView.update(InpunData.**this**.d);

InpunData.**this**.dispose();

}

});

}

{

JButton cancelButton = **new** JButton("Відміна");

buttonPane.add(cancelButton);

}

}

}

**private** **void** setVisibleTrue(**int** size) {

label2.setVisible(**false**);

label3.setVisible(**false**);

label4.setVisible(**false**);

label5.setVisible(**false**);

textField2.setVisible(**true**);

textField3.setVisible(**true**);

textField4.setVisible(**false**);

textField5.setVisible(**false**);

label1.setVisible(**true**);

textField1.setVisible(**true**);

**switch** (size) {

**case** 2:

label2.setVisible(**true**);

textField2.setVisible(**true**);

**break**;

**case** 3:

label2.setVisible(**true**);

label3.setVisible(**true**);

textField2.setVisible(**true**);

textField3.setVisible(**true**);

**break**;

**case** 4:

textField2.setVisible(**true**);

textField3.setVisible(**true**);

textField4.setVisible(**true**);

label2.setVisible(**true**);

label3.setVisible(**true**);

label4.setVisible(**true**);

**break**;

**case** 5:

textField2.setVisible(**true**);

textField3.setVisible(**true**);

textField4.setVisible(**true**);

textField5.setVisible(**true**);

label2.setVisible(**true**);

label3.setVisible(**true**);

label4.setVisible(**true**);

label5.setVisible(**true**);

**break**;

**default**:

**break**;

}

{

label2.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label2.setBounds(95, 21, 25, 20);

contentPanel.add(label2);

}

{

textField3 = **new** JTextField();

textField3.setColumns(10);

textField3.setBounds(116, 21, 25, 20);

contentPanel.add(textField3);

}

{

label3.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label3.setBounds(144, 21, 25, 20);

contentPanel.add(label3);

}

{

textField4 = **new** JTextField();

textField4.setColumns(10);

textField4.setBounds(166, 21, 25, 20);

contentPanel.add(textField4);

}

{

label4.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label4.setBounds(195, 21, 25, 20);

contentPanel.add(label4);

}

{

textField5 = **new** JTextField();

textField5.setColumns(10);

textField5.setBounds(216, 21, 25, 20);

contentPanel.add(textField5);

}

{

label5.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label5.setBounds(245, 21, 25, 20);

contentPanel.add(label5);

}

{

JLabel label = **new** JLabel("=");

label.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label.setBounds(266, 21, 15, 20);

contentPanel.add(label);

}

{

textFieldEnd = **new** JTextField();

textFieldEnd.setColumns(10);

textFieldEnd.setBounds(280, 21, 25, 20);

contentPanel.add(textFieldEnd);

}

{

textField1 = **new** JTextField();

textField1.setColumns(10);

textField1.setBounds(10, 21, 25, 20);

contentPanel.add(textField1);

}

{

label1.setFont(**new** Font("Tahoma", Font.*PLAIN*, 14));

label1.setBounds(38, 21, 25, 20);

contentPanel.add(label1);

}

{

textField2 = **new** JTextField();

textField2.setColumns(10);

textField2.setBounds(60, 21, 25, 20);

contentPanel.add(textField2);

}

}

}