НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №4

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

Виконав:

студент 2 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Порєв В.М.

Київ 2014 р.

Варіант №13

Метод: метод хорд.

Програмний код:

**import** java.awt.BorderLayout;

**public** **class** AppView **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField textFieldA;

**private** JTextField textFieldB;

**private** **double** a;

**private** **double** b;

**private** **double** e;

**private** JLabel labelk = **new** JLabel("");

**private** JLabel labelX = **new** JLabel("");

/\*\*

\* Launch the application.

\*/

/\*\*

\* Create the frame.

\*/

**public** AppView() {

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

setBounds(100, 100, 699, 387);

contentPane = **new** JPanel();

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

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel label = **new** JLabel(

"\u041F\u0440\u0430\u0432\u0430 \u0433\u0440\u0430\u043D\u0438\u0446\u044F");

label.setBounds(10, 53, 85, 31);

contentPane.add(label);

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

"\u041B\u0456\u0432\u0430 \u0433\u0440\u0430\u043D\u0438\u0446\u044F");

label\_1.setBounds(10, 11, 85, 31);

contentPane.add(label\_1);

textFieldA = **new** JTextField();

textFieldA.setBounds(105, 16, 86, 20);

contentPane.add(textFieldA);

textFieldA.setColumns(10);

textFieldB = **new** JTextField();

textFieldB.setColumns(10);

textFieldB.setBounds(105, 58, 86, 20);

contentPane.add(textFieldB);

JLabel label\_2 = **new** JLabel(

"\u0422\u043E\u0447\u043D\u0456\u0441\u0442\u044C");

label\_2.setBounds(10, 95, 85, 31);

contentPane.add(label\_2);

JLabel label\_3 = **new** JLabel(

"\u0417\u043D\u0430\u0447\u0435\u043D\u043D\u044F \u043A\u043E\u0440\u0435\u043D\u044F");

label\_3.setBounds(10, 179, 181, 31);

contentPane.add(label\_3);

labelX.setBounds(10, 213, 181, 23);

contentPane.add(labelX);

contentPane.add(**this**.getGrafic(0.01));

JButton button = **new** JButton(

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

button.setBounds(10, 145, 181, 28);

contentPane.add(button);

JLabel label\_4 = **new** JLabel(

"\u041A\u0456\u043B\u044C\u043A\u0456\u0441\u0442\u044C \u0456\u0442\u0435\u0440\u0430\u0446\u0456\u0439");

label\_4.setBounds(10, 251, 181, 31);

contentPane.add(label\_4);

labelk.setBounds(10, 286, 181, 23);

contentPane.add(labelk);

Double [] eps = {0.1,0.01,0.001,0.0001,0.00001,0.000001,0.0000001};

**final** JList<Double> list = **new** JList<>(eps);

list.setVisible(**true**);

list.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION*);

list.setVisibleRowCount(1);

JScrollPane scrollPane = **new** JScrollPane(list);

scrollPane.setBounds(105, 95, 84, 20);

contentPane.add(scrollPane);

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

@Override

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

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

AppView.**this**.e = list.getSelectedValue();

}

}

});

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

@Override

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

**if** (**true**) {

**try** {

Logic l = **new** Logic();

AppView.**this**.a = Double.*parseDouble*(textFieldA.getText());

AppView.**this**.b = Double.*parseDouble*(textFieldB.getText());

**if** (l.func(a)\*l.func(b)<0){

l = **new** Logic(AppView.**this**.a, AppView.**this**.b, AppView.**this**.e);

l.calc();

AppView.**this**.labelk.setText(Integer.*toString*(l.getK()));

AppView.**this**.labelX.setText(Double.*toString*(l.getX()));

AppView.**this**.labelX.updateUI();

AppView.**this**.labelX.updateUI();

}**else**{

JOptionPane.*showMessageDialog*(AppView.**this**, "На таних границях: ["+AppView.**this**.a+";"+AppView.**this**.b+"] коренів немає", "Помилка", JOptionPane.*ERROR\_MESSAGE*);

}

}**catch** (NumberFormatException except){

JOptionPane.*showMessageDialog*(AppView.**this**, "Неправильно введені дані", "Помилка", JOptionPane.*ERROR\_MESSAGE*);

};

}

}

});

}

**private** ChartPanel getGrafic(**double** h) {

XYSeriesCollection dataset = **new** XYSeriesCollection();

XYSeries series = **new** XYSeries("Графік функції y(x) = x^3 - 6x^2+9x-3");

Logic l = **new** Logic(0, 0, 0);

**double** a = -2;

**double** b = 7;

**do** {

series.add(a, l.func(a));

a = a + h;

} **while** (a <= b);

dataset.addSeries(series);

JFreeChart chart = ChartFactory.*createXYLineChart*("", "x", "f(x)",

dataset, PlotOrientation.*VERTICAL*, **true**, **true**, **true**);

ChartPanel pane = **new** ChartPanel(chart);

pane.setLocation(208, 11);

pane.setSize(465, 327);

pane.setVisible(**true**);

**return** pane;

}

}

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

**private** **double** a;

**private** **double** b;

**private** **double** e;

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

**private** **double** x;

**public** Logic(**double** a, **double** b, **double** e) {

**if** (e!=0){

**this**.a = a;

**this**.b = b;

**this**.e=e;

}

}

**public** Logic() {

// **TODO** Auto-generated constructor stub

}

**public** **double** func (**double** x){

**return** Math.*pow*(x, 3)-6\*Math.*pow*(x, 2)+9\*x-3;

}

**private** **double** funcFirstDerivative(**double** x){

**return** 3\*Math.*pow*(x, 2)-12\*x+9;

}

**private** **double** funcSecondDerivative(**double** x){

**return** 6\*x-12;

}

**public** **void** calc(){

**double** a = **this**.a;

**int** k =0;

**double** x =0;

**boolean** flag = **true**;

**do**{

**if** (funcFirstDerivative(a)\*funcSecondDerivative(a)>0){

**double** z=a;

a=b;

b=z;

}

x = a - ((func(a)\*(b-a))/(func(b)-func(a)));

k++;

**if** (Math.*abs*(x-a)<e){

**break**;

}**else**{

**if** (x<a||x>b){

a += (b-**this**.a)\*e;

b = b-(b-**this**.a)\*e;

}

a=x;

}

}**while**(flag);

setX(x);

setK(k);

}

**public** **int** getK() {

**return** k;

}

**public** **void** setK(**int** k) {

**this**.k = k;

}

**public** **double** getX() {

**return** x;

}

**public** **void** setX(**double** x) {

**this**.x = x;

}

}

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

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

AppView frame = **new** AppView();

frame.setVisible(**true**);

}

}