Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №4**

З дисципліни «Алгоритми та методи обчислень»

Тема: «Розв’язання нелінійних рівнянь на ЕОМ»

Виконала: Перевірив:

Студентка групи ІО-32 ст.викладач

Руденко Т.А. Порєв В.М.

Київ 2015

Мета: Метою даного заняття є ознайомлення з методиками та вивчення різних алгоритмів розв’язання нелінійних рівнянь на ЕОМ.

Завдання: Закріплення знань студентів при вирішенні практичних завданьз розв’язування нелінійних рівнянь. Оволодіння методами і практичними навичками розв’язування нелінійних рівнянь на ЕОМ. Набуття умінь і навичок при програмуванні та налагодженні програм для розв’язування нелінійних рівнянь на комп'ютері.

Варіант 25:

Метод ітерацій

Рівняння: x+lg(x) = 0,5, корінь: 0,672

Лістинг:

**package** lab4;

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**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** Window **extends** JFrame{

**public** **double** a;

**public** **double** b;

**public** **double** x0;

**public** **double** e;

**public** JPanel panel = **new** JPanel();

**public** JTextField aField = **new** JTextField();

**public** JTextField bField = **new** JTextField();

**public** JTextField x0Field = **new** JTextField();

**public** JTextField eField = **new** JTextField();

**public** JButton button = **new** JButton("Знайти корені");

**public** JLabel label = **new** JLabel("Корені:");

**public** Window(){

**super**("Labwork 1");

setBounds(500, 130, 500, 100);

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

**this**.add(panel, BorderLayout.*NORTH*);

panel.add(**new** JLabel("a:"));

panel.add(aField);

aField.setColumns(5);

panel.add(**new** JLabel("b:"));

panel.add(bField);

bField.setColumns(5);

panel.add(**new** JLabel("x0:"));

panel.add(x0Field);

x0Field.setColumns(5);

panel.add(**new** JLabel("e:"));

panel.add(eField);

eField.setColumns(5);

panel.add(button);

**this**.add(label, BorderLayout.*CENTER*);

aField.getDocument().addDocumentListener(**new** DocumentListener() {

@Override

**public** **void** insertUpdate(DocumentEvent ev) {

a = Double.*parseDouble*(aField.getText());

}

@Override

**public** **void** removeUpdate(DocumentEvent ev) {

a = Double.*parseDouble*(aField.getText());

}

@Override

**public** **void** changedUpdate(DocumentEvent ev) {

}

});

bField.getDocument().addDocumentListener(**new** DocumentListener() {

@Override

**public** **void** insertUpdate(DocumentEvent ev) {

b = Double.*parseDouble*(bField.getText());

}

@Override

**public** **void** removeUpdate(DocumentEvent ev) {

b = Double.*parseDouble*(bField.getText());

}

@Override

**public** **void** changedUpdate(DocumentEvent ev) {

}

});

x0Field.getDocument().addDocumentListener(**new** DocumentListener() {

@Override

**public** **void** insertUpdate(DocumentEvent ev) {

x0 = Double.*parseDouble*(x0Field.getText());

}

@Override

**public** **void** removeUpdate(DocumentEvent ev) {

x0 = Double.*parseDouble*(x0Field.getText());

}

@Override

**public** **void** changedUpdate(DocumentEvent ev) {

}

});

eField.getDocument().addDocumentListener(**new** DocumentListener() {

@Override

**public** **void** insertUpdate(DocumentEvent ev) {

e = Double.*parseDouble*(eField.getText());

}

@Override

**public** **void** removeUpdate(DocumentEvent ev) {

e = Double.*parseDouble*(eField.getText());

}

@Override

**public** **void** changedUpdate(DocumentEvent ev) {

}

});

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

@Override

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

aField.setEnabled(**false**);

bField.setEnabled(**false**);

x0Field.setEnabled(**false**);

eField.setEnabled(**false**);

**double** root = solve(a,b,x0,e);

**if** (root !=Double.*MAX\_VALUE*) {

label.setText(String.*valueOf*(root));

}

}

});

}

**public** **double** function(**double** x){

**return** 0.5-Math.*log10*(x);

}

**public** **double** solve(**double** a, **double** b, **double** x0, **double** e){

**if** (function(a) \* function(b) >= 0) {

label.setText("Змініть дані");

aField.setEnabled(**true**);

bField.setEnabled(**true**);

x0Field.setEnabled(**true**);

eField.setEnabled(**true**);

**return** Double.*MAX\_VALUE*;

}

**double** x = x0;

**double** y=0;

**double** flag;

**do**{

y = function(x);

flag = Math.*abs*(y-x);

x = y;

}**while**(flag>=e);

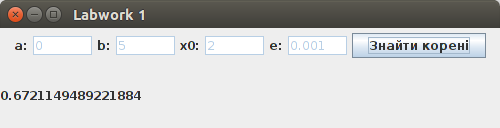
x0 = y;

**return** x0;

}

}

Результат виконання програми:



Висновок:

Виконуючи дану лабораторну роботу, ми вивчили основні алгоритми розв'язання нелінійних рівнянь на ЕОМ і закріпили отримані знання, написавши програму, що обчислює корінь заданого рівняння методом ітерацій (послідовних наближень).