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 №3**

З дисципліни «Алгоритми та методи обчислень»

Тема: «Інтерполяція функцій»

Виконала: Перевірив:

Студентка групи ІО-32 ст. викладач

Руденко Т.А. Порєв В. М.

Київ 2015

Тема: Машина Тьюринга

Мета: Ознайомлення з інтерполяційними формулами Лагранжа, Ньютона, рекурентним співвідношенням Ейткена, методами оцінки похибки інтерполяції.

Завдання: Закріплення, поглиблення і розширення знань студентів при вирішенні практичних обчислювальних завдань. Оволодіння обчислювальними методами і практичними методами оцінки похибки обчислень. Придбання умінь і навичок при програмуванні та налагодженні обчислювальних завдань на комп'ютері.

Лістинг:

**package** Lab3;

**import** java.awt.BorderLayout;

**import** javax.swing.JFrame;

**import** org.jfree.chart.ChartFactory;

**import** org.jfree.chart.ChartPanel;

**import** org.jfree.chart.JFreeChart;

**import** org.jfree.chart.plot.PlotOrientation;

**import** org.jfree.data.xy.XYSeries;

**import** org.jfree.data.xy.XYSeriesCollection;

**public** **class** Plotter {

**public** Plotter(**double**[] xInt, **double**[] yInt, **double**[] xF, **double**[] yF, **double**[] delta, String name) {

JFrame frame = **new** JFrame();

frame.setTitle(name);

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

XYSeriesCollection data = **new** XYSeriesCollection();

XYSeries series1 = **new** XYSeries("Вихідна функція");

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

series1.add(xF[i], yF[i]);

data.addSeries(series1);

XYSeries series2 = **new** XYSeries("Інтерпольована функція");

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

series2.add(xInt[i], yInt[i]);

data.addSeries(series2);

JFreeChart chart = ChartFactory.*createXYLineChart*(

"Графіки вихідної та інтерпольваної функцій",

"X",

"Y",

data,

PlotOrientation.*VERTICAL*,

**true**,

**true**,

**false**);

**final** ChartPanel chartPanel = **new** ChartPanel(chart);

chartPanel.setPreferredSize(**new** java.awt.Dimension(600, 600));

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

frame.add(chartPanel, BorderLayout.*WEST*);

XYSeriesCollection mistake = **new** XYSeriesCollection();

XYSeries deltas = **new** XYSeries("Похибка");

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

deltas.add(xF[i], delta[i]);

mistake.addSeries(deltas);

JFreeChart chart2 = ChartFactory.*createXYLineChart*(

"Похибки інтерполяції у невузлових точках",

"X",

"delta",

mistake,

PlotOrientation.*VERTICAL*,

**true**,

**true**,

**false**);

**final** ChartPanel chartPanel2 = **new** ChartPanel(chart2);

chartPanel2.setPreferredSize(**new** java.awt.Dimension(600, 600));

frame.add(chartPanel2, BorderLayout.*EAST*);

frame.pack();

frame.setVisible(**true**);

}

}

**package** Lab3;

**import** java.util.Arrays;

**public** **class** Lab3 {

**private** **static** **final** **double** *a* = 0.0;

**private** **static** **final** **double** *b* = 2.0;

**private** **static** **final** **int** *n* = 11;

**private** **static** **final** **int** *k* = 20;

**private** **static** **double**[] *xInt* = **new** **double**[*n*];

**private** **static** **double**[] *yInt* = **new** **double**[*n*];

**private** **static** **double**[] *yTable* = **new** **double**[*n*];

**private** **static** **double**[] *yExt* = **new** **double**[*n*];

**private** **static** **double**[] *xF* = **new** **double**[*n*\**k*];

**private** **static** **double**[] *yF* = **new** **double**[*n*\**k*];

**private** **static** **double**[] *delta* = **new** **double**[*n*\**k*];

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

**double** h = (*b* - *a*) / (*n*\**k* - 1);

**for**(**int** i = 0; i < *n*\**k*; i++) {

*xF*[i] = *a* + h\*i;

*yF*[i] = *function*(*xF*[i]);

}

h = (*b* - *a*) / (*n* - 1);

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

*xInt*[i] = *a* + h \* i;

*yTable*[i] = *function*(*xInt*[i]);

}

**for**(**int** i = 0; i < *n*; i++)

*yInt*[i] = *interpolateLagrange*(*xInt*[i]);

**for**(**int** i = 0; i < *n*\**k*; i++)

*delta*[i] = Math.*abs*(*yF*[i] - *interpolateLagrange*(*xF*[i]));

Plotter plot = **new** Plotter(*xInt*, *yInt*, *xF*, *yF*, *delta*, "Графіки інтерполяції та похибок(многочлен Лагранжа)");

**for**(**int** i = 0; i < *n*; i++)

*yInt*[i] = *interpolateEitken*(*xInt*[i]);

**for**(**int** i = 0; i < *n*\**k*; i++)

*delta*[i] = Math.*abs*(*yF*[i] - *interpolateEitken*(*xF*[i]));

Plotter plot2 = **new** Plotter(*xInt*, *yInt*, *xF*, *yF*, *delta*, "Графіки інтерполяції та похибок(рекурентне співвідношення Ейткена)");

}

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

**return** Math.*cos*(x+Math.*pow*(Math.*cos*(x), 3));

}

**public** **static** **double** interpolateLagrange(**double** x) {

*yExt* = Arrays.*copyOf*(*yTable*, *n*);

**for**(**int** i = 0; i < *n* - 2; i++)

**for**(**int** j = i + 1; j < *n-1*; j++)

for(int k = j+1; k<n; k++)

*yExt*[k] = ((x-xInt[j])\*(x-xInt[k])/((xInt[i]-xInt[j])\*(xInt[i]-xInt[k])))\*yExt[i]+ ((x-xInt[i])\*(x-xInt[k])/((xInt[j]-xInt[i])\*(xInt[j]-xInt[k])))\*yExt[j] + ((x-xInt[i])\*(x-xInt[j])/((xInt[k]-xInt[i])\*(xInt[k]-xInt[j])))\*yExt[k];

**return** *yExt*[*n* - 2];

}

**public** **static** **double** interpolateEitken(**double** x){

*yExt* = Arrays.*copyOf*(*yTable*, *n*);

**for**(**int** i = 0; i < *n* - 1; i++)

**for**(**int** j = i + 1; j < *n*; j++)

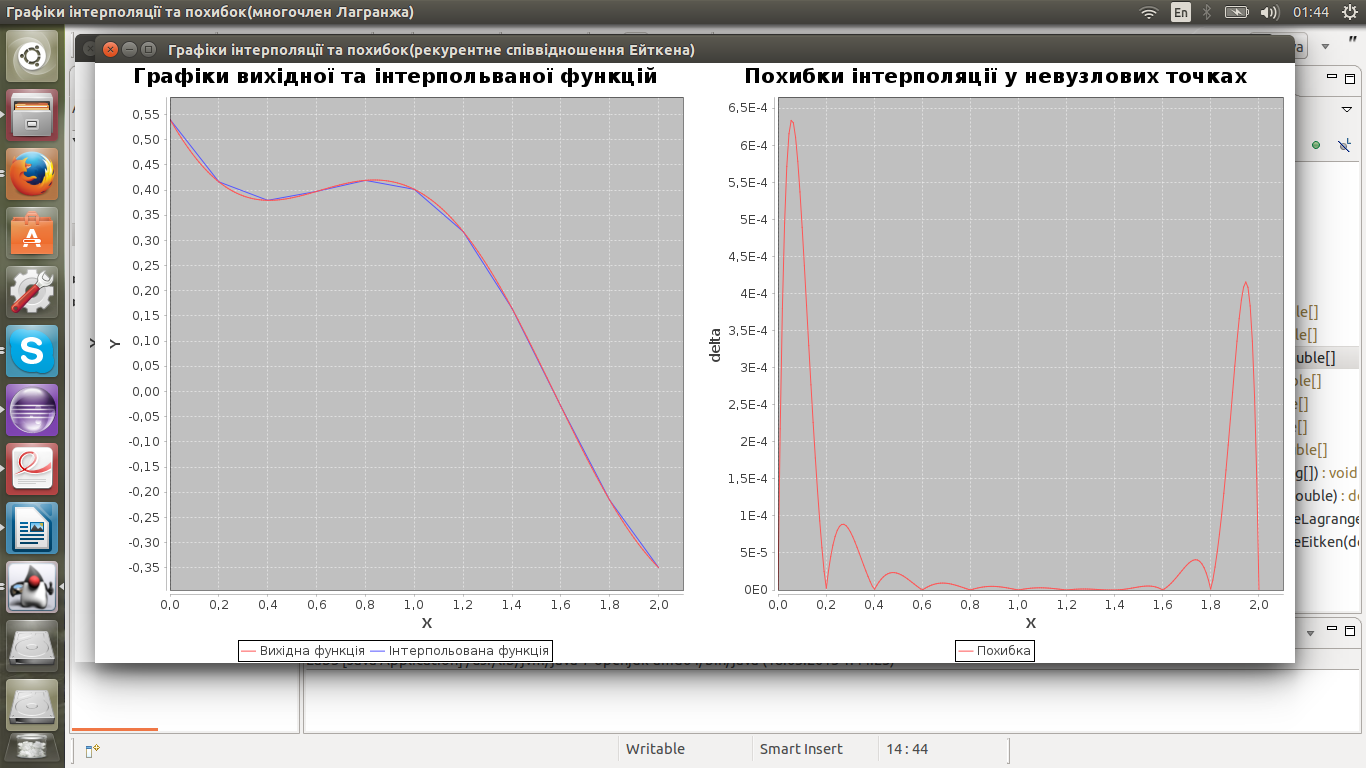
*yExt*[j] = ((*yExt*[i]\*(*xInt*[j]-x))-(*yExt*[j]\*(*xInt*[i]-x)))/(*xInt*[j]-*xInt*[i]);

**return** *yExt*[*n*-1];

}

}

Результат виконання програми:

Висновок:

Виконуючи дану лабораторну роботу, ми ознайомились із принципом роботи машини Тьюринга і створили для неї власний алгоритм, що виконує операцію диз'юнкції двох двійкових чисел.

Висновок:

Під час виконання даної лабораторної роботи ми вивчили основні методи інтерполяції функцій і реалізували програмно два з них — Лагранжа і Ейткена. Також ми здобули навички роботи з бібліотекою JFreeChart