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 №1**

З дисципліни «Комп’ютерне моделювання»

Виконав: Перевірив:

Студент групи ІО-21 доц. Марковський О.П.

Коноз А.О.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

**Лістинг коду**

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileReader;

**import** java.io.ObjectInputStream;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** org.apache.commons.math3.stat.inference.\*;

**public** **class** Main {

File file = **new** File("D:\\result11.dat");

ObjectInputStream reader;

ArrayList<Double> list;

**public** **void** read(){

**try**{

reader = **new** ObjectInputStream(**new** FileInputStream(file));

list =(ArrayList<Double>) reader.readObject();

}**catch**(Exception e){

}

}

**public** **void** findVariables(){

read();

**double** psi;

**double** v;

**double** vect [] = **new** **double**[list.size()];

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

vect[i] = list.get(i);

}

Arrays.*sort*(vect);

psi = vect[list.size()/2];

**double** m = findM(vect);

**double** d = findD(vect, m);

v = Math.*sqrt*(2\*(Math.*log*(m)-Math.*log*(psi)));

Generator gen = **new** Generator(psi, v, 10000);

**double** result [] = gen.generate();

ChiSquareTest test = **new** ChiSquareTest();

**long** clon [] = **new** **long**[result.length];

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

clon[i] =(**long**) result[i];

}

**double** p = test.chiSquareTest(vect, clon);

System.*out*.println("D = " + d);

System.*out*.println("psi = " + psi);

System.*out*.println("m = " + m);

System.*out*.println("v = " + v);

System.*out*.println("p = " + p);

}

**public** **double** findM(**double** vect[]){

**double** m = 0;

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

m += vect[i];

}

m = m/vect.length;

**return** m;

}

**public** **double** findD(**double** vect[], **double** m){

**double** d = 0;

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

d += Math.*pow*((vect[i] - m), 2);

}

d = d/vect.length;

**return** d;

}

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

Main main = **new** Main();

main.findVariables();

}

}

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.ObjectOutputStream;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Random;

**public** **class** Generator {

**private** **double** mu;

**private** **double** v;

**private** Random randomGussian = **new** Random();

**private** **double** [] masLognormal;

**public** Generator(**double** psi, **double** v,**int** n) {

**this**.mu = Math.*log*(psi);

**this**.v = v;

masLognormal = **new** **double** [n];

}

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

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

masLognormal[i]=Math.*exp*(mu + Math.*sqrt*(v)\*randomGussian.nextGaussian());

}

Arrays.*sort*(masLognormal);

**return** masLognormal;

}

}

**Приклад виконання:**

Генератор:

Мат очікування експериментальне 5.193770769233322

Маточікування теоретичне 5.199759053602186

Дисперсія експериментальна 55.4697143884715

Дисперсія теоретична 54.18762727938921

Медіана експериментальна 2.963212226896284

Медіана теоретична 3.0

Запис

Готово

Аналізатор-відтворювач:

D = 55.46416741703256

psi = 2.963212226896284

m = 5.193770769233298

v = 1.059420678290455

p = 1.0