# *Національний технічний університет України*

***«Київський політехнічний інститут»***

#### ***Факультет інформатики та обчислювальної техніки***

### ***Кафедра обчислювальної техніки***

## ЛАБОРАТОРНА РОБОТА №2

*з курсу «Теорія ймовірності та математична статистика»*

***Виконала:*** *Косейкіна Ганна Сергіївна*

***Факультет***  *ІОТ*

***Група*** *ІО-91* **Залікова книжка****№** *9123*

Задание: Сгенерировать 5000 чисел по функции Симпсона и определить мат. ожидание и дисперсию.

Листинг:

**package** kpi.teorver.lab2;

**public** **class** SimpsonReGenerator {

**private** **double** RESOLUTION = 1.0e-3;

**private** **double** \_m;

**private** **double** \_a;

**public** SimpsonReGenerator(**double** m, **double** a) {

\_m = m;

\_a = a;

}

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

**double** r;

**double** simpsons[] = **new** **double**[count];

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

r = Math.*random*();

**if** (r < 0.5) {

simpsons[i] = \_m - \_a + \_a \* Math.*sqrt*(2.0\*r);

}

**else** {

simpsons[i] = \_m + \_a - \_a \* Math.*sqrt*(2.0\*(1-r));

}

}

**return** simpsons;

}

}

**package** kpi.teorver.lab2;

**public** **class** SimpsonTest {

**private** **static** **final** **int** *COUNT* = 5000;

**private** **static** **final** **int** *SPREAD\_INTERVALS* = 15;

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

**private** **static** **final** **double** *LENGTH* = 1.0;

/\*\*

\* **@param** args

\*/

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

SimpsonReGenerator generator = **new** SimpsonReGenerator(*MEDIAN*, *LENGTH*);

**double** result[] = generator.generate(*COUNT*);

**double** expectation = *expectation*(result, *MEDIAN*, *LENGTH*);

System.*out*.print("Expectation is: ");

System.*out*.println(expectation);

System.*out*.print("Dispersion is: ");

**double** dispersion = *dispersion*(result, expectation, *LENGTH*);

System.*out*.println(dispersion);

System.*out*.print("Standard deviation is: ");

System.*out*.println(Math.*sqrt*(dispersion));

System.*out*.println("Resulting spread: ");

**int** expected\_spread[] = **new** **int**[*SPREAD\_INTERVALS*];

**int** spread[] = *spread*(result, *MEDIAN*, *LENGTH*, expected\_spread);

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

System.*out*.format("%d\n", spread[i]);

}

}

**public** **static** **double** expectation(**double** [] array, **double** m, **double** a) {

**int** intervals = array.length / 20;

**int** counters[] = **new** **int**[intervals];

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

counters[i] = 0;

**int** ix;

**double** dint = (**double**)intervals;

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

ix = (**int**)Math.*round*(Math.*floor*((((array[i]+a-m)/(2.0\*a))\*dint)));

counters[ix]++;

}

**double** expectation = 0.0;

**double** interval\_length = 2.0\*a/((**double**)intervals);

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

**double** probability = ((**double**)counters[i])/((**double**)array.length);

**double** invalue = m - a + ((**double**)i)\*interval\_length + interval\_length/2.0;

expectation += invalue\*probability;

}

**return** expectation;

}

**public** **static** **double** dispersion(**double** [] array, **double** m, **double** a) {

**int** intervals = array.length / 20;

**int** counters[] = **new** **int**[intervals];

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

counters[i] = 0;

**int** ix;

**double** dint = (**double**)intervals;

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

ix = (**int**)Math.*round*(Math.*floor*(Math.*abs*((((array[i]+a-m)/(2.0\*a))\*dint))));

counters[ix]++;

}

**double** dispersion = 0.0;

**double** interval\_length = 2.0\*a/((**double**)intervals);

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

**double** probability = ((**double**)counters[i])/((**double**)array.length);

**double** invalue = m - a + ((**double**)i)\*interval\_length + interval\_length/2.0;

dispersion += (invalue - m) \* (invalue - m) \* probability;

}

**return** dispersion;

}

**public** **static** **int** [] spread(**double** array[], **double** m, **double** a, **int** expected[]) {

**int** counters[] = **new** **int**[expected.length];

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

counters[i] = 0;

}

**int** ix;

**double** dint = (**double**)expected.length;

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

ix = (**int**)Math.*round*(Math.*floor*(Math.*abs*((((array[i]+a-m)/(2.0\*a))\*dint))));

counters[ix]++;

}

**return** counters;

}

}