*Національний технічний університет України*

*«Київський політехнічний інститут»*

***Факультет інформатики та обчислювальної техніки***

Лабораторна робота №2

*з курсу "****Теорія ймовірностей****"*

**Тема: “Генерація випадкової величини згідно заданого розподілу”**

***Виконав:***

*Долинний О.В.*

***Група*** *ІО-31*

***Перевірив:***

*Марковський О.П.*

***Київ - 2014р.***

**Варіант:** Закон розподілу – біноміальний

**Виконання завдання:**

**Клас BinomRand**

**package** lab\_2;

**import** java.util.Random;

**public** **class** BinomRand {

**int** n;

**double** p;

**double**[] arr\_dif;

**double**[] arr\_int;

**int**[] select;

**public** BinomRand(**int** n, **double** p) {

**this**.p = p;

**this**.n = n;

arr\_dif = **new** **double**[n+1];

arr\_int = **new** **double**[n+1];

select = **new** **int**[5000];

**for**(**int** i = 0; i < arr\_dif.length; i++) {

arr\_dif[i] = bernuli( n, i, p);

}

arr\_int[0] = 0;

**for**(**int** i = 1; i < arr\_dif.length; i++) {

arr\_int[i] = arr\_dif[i-1] + arr\_int[i - 1];

}

**this**.generate();

}

**public** **static** **double** combination(**int** n, **int** k) {

**return** *fact*(n)/( *fact*(k)\**fact*(n-k));

}

**public** **static** **double** fact(**int** n) {

**double** result = 1.0;

**if**(n == 0)

**return** result;

**else**

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

result \*= i;

**return** result;

}

**public** **double** bernuli(**int** m, **int** k, **double** p) {

**double** result = *combination*(m, k) \*( (**double**)Math.*pow*(p, k)) \* ((**double**)Math.*pow*(1 - p, m - k));

**return** result;

}

**public** String toString() {

String s = **new** String();

**for**( **int** i = 0; i < 301; i++) {

s += i + " " + *combination*(300,i) + " ";

}

**return** s;

}

**public** **void** generate(){

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

Random bufer = **new** Random();

**double** choice = bufer.nextDouble();

**for** (**int** j = 0; j < arr\_int.length - 1; j++) {

**if** ((choice>=arr\_int[j])&(choice<arr\_int[j+1])) {

select[i] = j;

**break**;

}

}

}

}

**public** **double** getM() {

**double** result = 0;

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

result += select[i];

}

result /= select.length;

**return** result;

}

**public** **double** getD() {

**double** result = 0;

**double** getM = **this**.getM();

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

result = result + (select[i]-getM)\*(select[i]-getM);

}

result /= select.length-1;

**return** result;

}

**public** **double** getMTheor() {

**return** n \* p;

}

**public** **double** getDTheor() {

**return** n \* p \* (1 - p);

}

}

**Клас Working**

**package** lab\_2;

**public** **class** Working {

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

**int** n = 170;

**double** p = 0.9;

BinomRand selection = **new** BinomRand(n,p);

System.***out***.println(" Математичне очікування теор " + selection.getMTheor());

System.***out***.println(" Дисперсія теор " + selection.getDTheor());

System.***out***.println(" Математичне очікування практ " + selection.getM());

System.***out***.println(" Дисперсія практ " + selection.getD());

}

}