Лабораторна робота

з теорії ймовірності №5

Виконала: студент

ФІОТ групи ІО-33

Шуркіна Анастасія

Перевірив:

Марковський О. П.

Київ 2014

Варіант: М, Д, 5 станів

**package** lab5;

**import** java.util.Random;

**import** org.apache.commons.math3.linear.\*;

**public** **class** Class1 {

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

**int** size = 5, jump = 1000;

**double** sum = 0, SumProb = 0;

**int** Position = 0;

**double** SecondArr[][] = **new** **double**[size][size];

**double** ResultArr[] = **new** **double**[size];

**double** FirstArr[][] = { { 0, 0.3, 0.7, 0, 0 }, { 0, 0.4, 0.6, 0, 0 },

{ 0, 0, 0, 1, 0 }, { 0, 0, 0.5, 0, 0.5 },

{ 0.6, 0, 0, 0.3, 0.1 } };

// Вивід початкової матриці

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

**for** (**int** j = 0; j < size; j++)

System.***out***.print(FirstArr[i][j] + " ");

System.***out***.println();

}

// Заповнення і вивід другої матриці

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

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

**for** (**int** j = 0; j < size; j++) {

sum = sum + FirstArr[i][j];

SecondArr[i][j] = sum;

System.***out***.print(SecondArr[i][j] + " ");

}

sum = 0;

System.***out***.println();

}

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

// Заповненя одномірного масиву кількості разів перебування в кожному

// стані

**for** (**int** j = 0; j < jump; j++) {

Random r = **new** Random();

**double** x = r.nextFloat();

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

**if** (x <= SecondArr[Position][i]) {

ResultArr[Position]++;

Position = i;

**break**;

}

}

}

// Підрахування стаціонарних ймовірностей і їх суми

System.***out***.println("Практичний розрахунок:");

System.***out***.println();

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

ResultArr[i] = ResultArr[i] / 1000;

SumProb += ResultArr[i];

System.***out***.println("P" + i + " = " + ResultArr[i]);

}

System.***out***.println("Sum of Probabilites = " + SumProb);

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

// Теореричні розрахунки

**double**[][] Matrix = **new** **double**[size][size];

**double**[] Vector = **new** **double**[size];

// Генерування системи рівнянь

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

Matrix[0][i] = 1;

Vector[0] = 1;

**for** (**int** i = 1; i < size; i++) {

**for** (**int** j = 0; j < size; j++)

**if** (j == i)

**for** (**int** p = 0; p < size; p++) {

**if** (p != j)

Matrix[i][j] -= FirstArr[j][p];

}

**else**

Matrix[i][j] = FirstArr[j][i];

Vector[i] = 0;

}

// Вирішення системи рівнянь

RealMatrix coefficients = **new** Array2DRowRealMatrix(Matrix, **false**);

DecompositionSolver solver = **new** LUDecomposition(coefficients)

.getSolver();

RealVector constants = **new** ArrayRealVector(Vector, **false**);

RealVector solution = solver.solve(constants);

System.***out***.println("Теоретичний розрахунок:");

System.***out***.println();

// Виведення результатів

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

System.***out***.println("P" + i + " = " + solution.getEntry(i));

}

}