

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики і обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №5 З теорії ймовірностей

Виконав:

Студент групи ІО-71

Чеботаренко А. В.

Залікова книжка №7127

Перевірів:

Марковський О. П.

1. Варіант завдання:

ПМ-3 2 стани Н

2. Лістинги класів проекту:

```
import java.util.ArrayList;
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        double[][] smallArr= {{0,1},{2,1}}; /*{{0.4,0.2,0.4},{0,0.3,0.7},{0.5,0.5,0}};*/

        double[][] bigArr= new double[10][10];

        double l11= smallArr[0][0]*3;
        double l12= smallArr[0][1]*3;
        double l21= smallArr[1][0]*3;
        double l22= smallArr[1][1]*3;

        bigArr[0][2]= l11;
        bigArr[0][6]= l12;
        bigArr[1][4]= l22;
        bigArr[1][9]= l21;
        bigArr[2][3]= l11;
        bigArr[3][0]= l11;
        bigArr[4][5]= l22;
        bigArr[5][1]= l22;
        bigArr[6][7]= l12;
        bigArr[7][1]= l12;
        bigArr[8][0]= l21;
        bigArr[9][8]= l21;

        OperClass op= new OperClass();
        ArrayList nulled= new ArrayList();
        double[][] lighthBigArr= op.simplifyArr(bigArr, nulled);
        double[][] bigSLAU= op.createSLAUarr(lighthBigArr);
        double[] bigResult= new double[bigArr.length];
        double[] result= op.gaus(bigSLAU);

        int pos=0;
        for (int i=0;i<bigResult.length;i++){
            if (!nulled.contains(i)) {bigResult[i]= result[pos]; pos++;}
        }
        bigResult[0]+=bigResult[6]+bigResult[7]+bigResult[2]+bigResult[3];
        bigResult[1]+=bigResult[8]+bigResult[9]+bigResult[4]+bigResult[5];

        System.out.println("Initial matrix :");
        for (int i=0;i<smallArr.length;i++)
            System.out.println(Arrays.toString(smallArr[i]));
        System.out.println("\nIt is calculated theoretically : "+ bigResult[0]+ " "+
bigResult[1]);
        double[] practic= op.calculeteProbability(bigArr);
        System.out.println("It is calculated practically : "+ practic[0]+ " "+
practic[1]);
    }
}
```

```

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Random;

public class OperClass {

    public double[] gaus(double[][] a) {

        double[][] x = new double[a.length][a[0].length];
        for (int i = 0; i < a.length; i++) x[i] = Arrays.copyOf(a[i], a[i].length);
        double p;
        for (int i = 0; i < x.length; i++)
            for (int j = 0; j < x.length; j++)
                if (i != j) {
                    p = x[j][i] / x[i][i];
                    for (int k = i; k <= x.length; ++k) x[j][k] -= p * x[i][k];
                }
        double[] b = new double[x.length];
        for (int i = 0; i < x.length; i++) b[i] = x[i][x.length] / x[i][i];
        return b;
    }

    public double[][] createSLAUarr(double[][] bigArr){

        double[][] slauArr= new double[bigArr.length][bigArr.length+1];
        for (int i=0;i<slauArr[0].length;i++){
            slauArr[0][i]= 1;
            if (i<slauArr.length)slauArr[i][slauArr[0].length-1]= 0;
        }

        for (int i=1;i<bigArr.length;i++){
            for (int j=0;j<bigArr.length;j++){

                //in
                for (int k=0;k<bigArr.length;k++){
                    if (k!=i) slauArr[i][k]= bigArr[k][i];
                }

                //out
                for (int k=0;k<bigArr.length;k++){
                    if (k!=i) slauArr[i][i]+= bigArr[i][k];
                }
            }

            for (int i=1;i<bigArr.length;i++) slauArr[i][i]*=-1;

            return slauArr;
        }

        public double[] calculeteProbability(double[][] bigArr){

            double[] result= new double[bigArr.length];
            Random r= new Random();
            int curPos=0;
            int nextPos=0;
            double allTime=0;

            for (int time=0;time<100;time++){
                double min=Double.MAX_VALUE;

                for (int i=0;i<bigArr.length;i++){
                    if (bigArr[curPos][i]!=0){
                        double tmp= -1/bigArr[curPos][i]*Math.log(r.nextDouble());
                        if (tmp<min) {min= tmp; nextPos= i;}
                    }
                }
            }
        }
    }
}

```

```

        result[curPos]+= min;
        allTime+= min;
        curPos= nextPos;
    }

    for (int i=0;i<result.length;i++) result[i]/=allTime;
    result[0]+=result[6]+result[7]+result[2]+result[3];
    result[1]+=result[8]+result[9]+result[4]+result[5];
    return result;
}

public double[][] simplifyArr(double[][] bigArr, ArrayList nulled){

    //ArrayList nulled= new ArrayList();

    for (int i=0;i<bigArr.length;i++){
        boolean allNull= true;
        for (int j=0;j<bigArr.length;j++){
            if (bigArr[i][j]!=0) allNull= false;
        }
        if (allNull) nulled.add(i);
    }

    ArrayList arrayList= new ArrayList();
    for (int i=0; i<bigArr.length;i++){
        arrayList.add(bigArr[i]);
    }
    for (int i=0; i<nulled.size();i++){
        arrayList.remove(bigArr[(Integer)nulled.get(i)]);
    }

    for (int i=0; i<arrayList.size();i++){
        double[] tmp= new double[arrayList.size()];
        int position=0;
        for (int j=0;j<bigArr.length;j++){
            if (!nulled.contains(j)) {
                double[] temp= (double[])arrayList.get(i);
                tmp[position]= temp[j];
                position++;
            }
        }
        arrayList.set(i,tmp);
    }

    double[][] result= new double[arrayList.size()][];
    for (int i=0;i<result.length;i++){
        result[i]= (double[])arrayList.get(i);
    }

    return result;
}
}

```

Результат роботи програми

Initial matrix :
 [0.0, 1.0]
 [2.0, 1.0]

It is calculated theoretically : 0.5454545454545454 , 0.4545454545454546
 It is calculated practically : 0.5435665719280653 , 0.45643342807193427