

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

## Лабораторна робота №3 З теорії ймовірностей

*Виконав:*

Студент групи ІО-32

Довгаль Д.С.

Залікова книжка №3211

*Перевірів:*

Марковський О. П.

## Лістинги класів проекту:

```
import java.util.Random;

public class Main {

    static final int count=5000;
    static final double a=1;
    static final double delta=0.0001;

    public static void main(String[] args) {

        double xMas[]= new double[count];
        double yMas[]= new double[xMas.length];
        Random r= new Random();
        PhiX phiX= new PhiX();
        AfterFunc afterFunc;

        for (int i=0;i<xMas.length;i++){
            double tmp= r.nextDouble();
            xMas[i]= Operations.calcTrap(tmp,phiX);
            afterFunc= new AfterFunc(xMas[i]);
            tmp= r.nextDouble();
            yMas[i]= Operations.calcTrap(tmp,afterFunc);
        }

        double matWx= Operations.matWaiting(xMas);
        double matWy= Operations.matWaiting(yMas);
        System.out.println("Матожидание для X - "+ matWx);
        System.out.println("Матожидание для Y - "+ matWy);

        double xDis= Operations.dispersion(xMas,matWx);
        double yDis= Operations.dispersion(yMas,matWy);
        System.out.println("Дисперсия для X - "+ xDis);
        System.out.println("Дисперсия для Y - "+ yDis);

        double xSigma= Math.sqrt(xDis);
        double ySigma= Math.sqrt(yDis);
        System.out.println("Сигма для X - "+ xSigma);
        System.out.println("Сигма для Y - "+ ySigma);

        double cov= Operations.covariation(xMas, yMas, matWx, matWy);
        System.out.println("Ковариация - "+ cov);

        System.out.println("Коефициент корреляции - "+
        Operations.correlNum(cov,xSigma,ySigma));
    }
}

public class Operations {

    static double calcTrap(double r, MyFunction func){

        double currentStep=0, square=0, step=Main.delta;
        for (;square<r;currentStep+=step) square+=
        (func.doOperation(currentStep)+func.doOperation(currentStep+step))/2*step;
        return currentStep;
    }

    static double afterFunc(double currentX, double y){
        double result=0;
        MyFunction currentFunc;
    }
}
```

```

PhiX phiX= new PhiX();

if (y>=0&&y<=currentX){
    currentFunc= new FirstFunc();
    result= currentFunc.doOperation(currentX)/phiX.doOperation(currentX);
}else if (y>currentX&&y<=Main.a){
    currentFunc= new SecondFunc();
    result= currentFunc.doOperation(y)/phiX.doOperation(currentX);
}

return result;
}

static double matWaiting(double[] argMas){

    double sum=0;
    for (int i=0;i<argMas.length;i++) sum+=argMas[i];
    return sum/argMas.length;
}

static double dispersion(double[] argMas, double matW){

    double sum=0;
    for (int i=0;i<argMas.length;i++) sum+=Math.pow(argMas[i]-matW,2);
    return sum/argMas.length;
}

static double covariation(double[] xMas, double[] yMas, double matWx, double
matWy){

    double sum=0;
    for (int i=0;i<xMas.length;i++) sum+=(xMas[i]-matWx)*(yMas[i]-matWy);
    return sum/xMas.length;
}

static double correlNum(double cov, double xSigma, double ySigma){
    return cov/(xSigma*ySigma);
}

}

public interface MyFunction {
    double doOperation(double arg);
}

public class PhiX implements MyFunction{

    double a= Main.a;
    @Override
    public double doOperation(double x) {
        return 3*(Math.pow(a,2)-Math.pow(x,2))/(2*Math.pow(a,3));
    }
}

public class FirstFunc implements MyFunction{

```

```

double a= Main.a;
@Override
public double doOperation(double x) {
    return 3*(a-x)/Math.pow(a,3);
}
}

public class SecondFunc implements MyFunction{

    double a= Main.a;
    @Override
    public double doOperation(double y) {
        return 3*(a-y)/Math.pow(a,3);
    }
}

public class AfterFunc implements MyFunction{

    private double currentX;

    AfterFunc(double currentX){
        this.currentX= currentX;
    }

    @Override
    public double doOperation(double y) {
        return Operations.afterFunc(currentX,y);
    }
}

```

## Результат работы программы при a=1:

Матожидание для X - 0.37769839999997296  
 Матожидание для Y - 0.3716788799999974  
 Дисперсия для X - 0.059118495137427826  
 Дисперсия для Y - 0.05872965216593369  
 Сигма для X - 0.2431429520620078  
 Сигма для Y - 0.2423420148590287  
 Ковариация - 0.00883270102620617  
 Коэффициент корреляции - 0.14990051704572327