

**Федеральное агентство связи**  
**Ордена Трудового Красного Знамени**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе №2**  
по дисциплине «Технологии разработки программного обеспечения»

Выполнил: студент группы  
БВТ1801

Протасова Елена Сергеевна

Руководитель:

Мосева Марина Сергеевна

Москва 2020

**Цель работы:** в данной лабораторной работе необходимо использовать классы по одному на файл, чтобы описать, как эти объекты работают.

**Ход работы:**

```
import java.util.Scanner;
public class Lab1 {
    public static void main(String[] args) {
        //for (int i = 0; i < args.length; i++) {
        //    String s = args[i];
        //}
        double S;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter point 1: ");
        double x = input.nextDouble();
        double y = input.nextDouble();
        double z = input.nextDouble();
        Point3d Point = new Point3d(x,y,z);
        System.out.println("Enter point 2: ");
        double x1 = input.nextDouble();
        double y1 = input.nextDouble();
        double z1 = input.nextDouble();
        Point3d Point1 = new Point3d(x1,y1,z1);
        System.out.println("Enter point 3: ");
        double x2 = input.nextDouble();
        double y2 = input.nextDouble();
        double z2 = input.nextDouble();
        Point3d Point2 = new Point3d(x2,y2,z2);
        S = computeArea(Point, Point1, Point2);
        System.out.println(S);
    }
    public static double computeArea(Point3d point1, Point3d point2, Point3d point3)
    {
        if (point1.isEqual(point2) || (point1.isEqual(point3)) ||
(point3.isEqual(point2)))
        {
            return -1;
        }
        else
        {
            double a, b, c, p, S;
            a = point1.DistanceTo(point2);
            b = point2.DistanceTo(point3);
            c = point3.DistanceTo(point1);
            p = (a + b + c) / 2;
            S = Math.sqrt(p * (p - a) * (p - b) * (p - c));
            return (double)Math.round(S*100)/100;
        }
    }
}
```

```

public class Point3d {
    private double xCoord;
    private double yCoord;
    private double zCoord;
    public Point3d ( double x, double y, double z) {
        xCoord = x;
        yCoord = y;
        zCoord = z;
    }
    public Point3d () {
        this(0, 0, 0);
    }
    public double getX () {
        return xCoord;
    }
    public double getY () {
        return yCoord;
    }
    public double getZ () {
        return zCoord;
    }
    public void setX ( double val) {
        xCoord = val;
    }
    public void setY ( double val) {
        yCoord = val;
    }
    public void setZ ( double val) {
        zCoord = val;
    }
    public double DistanceTo(Point3d pointTwo)
    {
        double a=Math.sqrt(Math.pow(this.xCoord-pointTwo.getX(), 2)
            + Math.pow(this.yCoord-pointTwo.getY(), 2)
            + Math.pow(this.zCoord-pointTwo.getZ(), 2));
        return (double)Math.round(a*100)/100;
    }
    public boolean isEqual(Point3d pointTwo)
    {
        if (this.xCoord == pointTwo.getX() && this.yCoord == pointTwo.getY() &&
            this.zCoord == pointTwo.getZ())
        {
            return true;
        }
        else return false;
    }
}

```

Enter point 1:

0

0

0

Enter point 2:

3

0

0

Enter point 3:

0

3

0

4.5