

**Министерство цифрового развития, связи и массовых  
коммуникаций  
Ордена трудового Красного Знамени федеральное  
государственное бюджетное  
образовательное учреждение высшего образования  
«Московский технический университет связи и информатики»**

Кафедра Математическая кибернетика и информационные  
технологии

Отчет по лабораторной работе №2  
“Основы объектно-ориентированного  
программирования”  
по дисциплине «Кроссплатформенные технологии  
программирования»

Выполнил: студент группы  
Бст1904  
Самарина А.В.

Проверил: Мосева М.С.

Москва, 2021

## 1. Point2d

```
public class Point2d {
    //двумерный класс точки.

    private double xCoord; //координата X
    private double yCoord; //координата Y

    // Конструктор инициализации
    public Point2d ( double x, double y) {
        xCoord = x;
        yCoord = y;
    }
    //Конструктор по умолчанию.
    public Point2d () {
        this(0, 0); //Вызовите конструктор с двумя
        параметрами и определите источник.
    }
    //Возвращение координаты X
    public double getX () {
        return xCoord;
    }
    // Возвращение координаты Y
    public double getY () {
        return yCoord;
    }
    // Установка значения координаты X.
    public void setX ( double val) {
        xCoord = val;
    }
    // Установка значения координаты Y.
    public void setY ( double val) {
        yCoord = val;
    }
}
```

## 2. Point 3d

Создайте новый класс Point3d для представления точек в трехмерном Евклидовом пространстве. Необходимо реализовать:

- создание нового объекта Point3d с тремя значениями с плавающей точкой (double);
- метод для сравнения значений двух объектов Point3d;
- метод distanceTo, который в качестве параметра принимает другой объект Point3d, вычисляет расстояние

между двумя точками с точность двух знаков после запятой и возвращает полученное значение.

```
• public class Point3d {  
    //Трехмерный класс точки  
  
    private double xCoord; // Координата X  
    private double yCoord; // Координата Y  
    private double zCoord; // Координата Z  
  
    public Point3d(double x, double y, double z) {  
        //Конструктор инициализации  
        xCoord = x;  
        yCoord = y;  
        zCoord = z;  
    }  
    public Point3d() {  
        /* Конструктор по умолчанию  
        Вызовите конструктор с двумя параметрами и определите  
источник*/  
        this(0, 0, 0);  
    }  
    public double getX() {  
        return xCoord; // Возвращение координаты X  
    }  
    public void setX(double val) {  
        xCoord = val; //Установка значения координаты X  
    }  
    public double getY() {  
        return yCoord; // Возвращение координаты Y  
    }  
    public void setY(double val) {  
        yCoord = val; //Установка значения координаты Y  
    }  
    public double getZ() {  
        return zCoord; // Возвращение координаты Z  
    }  
    public void setZ(double val) {  
        zCoord = val; // Установка значения координаты Z  
    }  
    public boolean sravn(Point3d P) { //сравнение 2 объектов  
пойнт3д  
        return xCoord == P.getX() &&  
            yCoord == P.getY() &&  
            zCoord == P.getZ();  
    }  
    public double distanceTo(Point3d P) { //Вычисляет расстояние  
между 2 точками  
        return Math.sqrt(Math.pow(xCoord - P.getX(), 2) +  
            Math.pow(yCoord - P.getY(), 2) +  
            Math.pow(zCoord - P.getZ(), 2));  
        //округление до 2 знаков после ,  
    }  
}
```

### 3. LAB2

Создать класс под названием Lab2. Этот класс должен иметь следующую функциональность:

- Ввод координат трех точек, находящихся в трехмерном пространстве.
- Создание трех объектов типа Point3d на основании полученных данных.
- Создайте второй статический метод computeArea, который принимает три объекта типа Point3d и вычисляет площадь треугольника, образованного этими точками.
- Посчитайте площадь и выведите полученное значение пользователю.
- Перед вызовом метода computeArea проверьте на равенство значений всех трех объектов Point3d. Если одна из точек равна другой, то выведите соответствующее сообщение пользователю и не вычисляйте площадь.

```
import java.util.Scanner;
public class Lab2 extends Point3d { //наследование
    public static void main(String[] args){
        Point3d p1 = makePoint3d(); //соз. 3 объектов на основе данных
        //пользователя
        Point3d p2 = makePoint3d();
        Point3d p3 = makePoint3d();
        // сравнение точек перед вычислением площади
        if (p1.sravn(p2) || p1.sravn(p3) || p2.sravn(p3)){
            System.out.println("Одна из точек равна другой");
        }
        else{
            double area = computeArea(p1,p2,p3);
            System.out.printf("Площадь треугольника: %.2f", area);
        }
    }
    public static Point3d makePoint3d(){
        Scanner s = new Scanner(System.in); //создаем объект класса сканер,
        return new Point3d(s.nextInt(),s.nextInt(),s.nextInt());
        //возвращаем объект на основе данных пользователя
    }
    public static double computeArea(Point3d p1, Point3d p2, Point3d p3) {
```

```

        /* Вычисляет площадь треугольника, образованного 3 точками, по
        формуле герона */
        double a, b, c, p;
        a = p1.distanceTo(p2); //считает сторону
        b = p1.distanceTo(p3);
        c = p2.distanceTo(p3);
        p = (a + b + c) / 2;
        return Math.sqrt(p * (p - a) * (p - b) * (p - c));
    }
}

```

#### 4.Результат

Вывод при вводе одинаковых значений для двух точек.

```

1 2 1
2 3 4
1 2 1
Одна из точек равна другой

```

Вывод площади треугольника при правильно введенных координатах.

```

1
3
2
4
2
3
2
3
1
Площадь треугольника с точками: 2,12

```

```

1 0 6
4 5 -2
7 3 4
Площадь треугольника: 24,50

```