

## Лабораторная работа №2: Основы объектно-ориентированного программирования

Java позволяет использовать объекты. В данной лабораторной работе необходимо использовать классы по одному на файл, чтобы описать, как эти объекты работают. Вот код для простого класса, который представляет двумерную точку:

```
/**
 * двумерный класс точки.
 **/

public class Point2d {
    /** координата X **/
    private double xCoord;
    /** координата Y **/
    private double yCoord;
    /** Конструктор инициализации **/
    public Point2d ( double x, double y) {
        xCoord = x;
        yCoord = y;
    }
    /** Конструктор по умолчанию. **/
    public Point2d () {
        //Вызовите конструктор с двумя параметрами и определите источник.
        this(0, 0);
    }
    /** 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;
}
}

```

Сохраните данный код в файле с именем Point2d.java, согласно требованиям Java к именам классов и именам файлов.

Экземпляр класса можно также создать, вызвав любой из реализованных конструкторов, например:

```

Point2d myPoint = new Point2d (); //создает точку (0,0)
Point2d myOtherPoint = new Point2d (5,3); //создает точку (5,3)
Point2d aThirdPoint = new Point2d ();

```

Примечание: myPoint != aThirdPoint, несмотря на то, что их значения равны. Объясняется это тем, что оператор равенства == (и его инверсия, оператор неравенства !=) сравнивает ссылки на объекты. Другими словами, == оператор вернет true, если две ссылки указывают на один и тот же объект. В данном случае myPoint и aThirdPoint ссылаются на разные объекты класса Point2d, поэтому операция сравнения myPoint == aThirdPoint вернет false, несмотря на то, что их значения те же!

Для того, чтобы проверить равны ли сами значения, а не ссылки, необходимо создать метод в классе Point2d, который будет сравнивать значения соответствующих полей объектов класса Point2d.

## Рекомендации при программирования

Стиль программирования является неотъемлемой частью при создании программного обеспечения. При создании приложений большая часть времени уходит на отладку программы. Читаемый код и использование комментариев в нем позволяют сэкономить время при отладке программы.

### Ваши задачи:

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

- создание нового объекта `Point3d` с тремя значениями с плавающей точкой (`double`);
- создание нового объекта `Point3d` со значениями (0.0, 0.0, 0.0) по умолчанию,
- возможность получения и изменения всех трех значений по отдельности;
- метод для сравнения значений двух объектов `Point3d`.

Нельзя предоставлять непосредственный доступ к внутренним элементам объекта класса `Point3d`.

2. Добавьте новый 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.

3. Создайте другой класс под названием `Lab1`, который будет содержать статический метод `main`. Помните, что метод `main` должен быть общедоступным (`public`) с возвращаемым значением `void`, а в качестве аргумента должен принимать строку (`String`). Этот класс должен иметь следующую функциональность:

- Ввод координат трех точек, находящихся в трехмерном пространстве. Создание трех объектов типа `Point3d` на основании полученных данных. (Предполагается, что пользователь вводит корректные данные.)

- Создайте второй статический метод `computeArea`, который принимает три объекта типа `Point3d` и вычисляет площадь треугольника, образованного этими точками. (Вы можете использовать формулу Герона.) Верните получившееся значение площади в формате типа `double`.

- На основе полученных данных и с использованием реализованного алгоритма 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.

4. Скомпилируйте оба исходных файла вместе:

```
javac Point3d.java Lab1.java
```

и затем запустите программу `Lab1`, тестируя ее с несколькими образцами треугольников.