Министерство Цифрового Развития, Связи и Массовых Коммуникаций Российской Федерации Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования   
“Московский технический университет связи и информатики”

Кафедра «Информационные технологии»

Лабораторная работа №2:   
«Основы объектно-ориентированного программирования**»**

Выполнил: студент гр. БПЗ1902

Кварацхелия Д.Т.

Проверил: Харрасов К.Р.

Москва 2021 г.

Цель работы:

В данной лабораторной работе необходимо использовать классы по одному на файл, чтобы описать, как эти объекты работают.

Задание:

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

Выполнение работы:

1. Создайте новый класс Point3d для представления точек в трехмерном Евклидовом пространстве.
2. public class Point3d {
3. private double xCoord;
4. private double yCoord;
5. private double zCoord;
6. public  Point3d(double xCoord, double yCoord,double zCoord){
7. this.xCoord=xCoord;
8. this.yCoord=yCoord;
9. this.zCoord=zCoord;
10. }
11. public  Point3d(){
12. this(0,0,0);
13. }
14. public double getX(){
15. return xCoord;
16. }
17. public double getY(){
18. return yCoord;
19. }
20. public double getZ(){
21. return zCoord;
22. }
23. public void setX(double xCoord){
24. this.xCoord=xCoord;
25. }
26. public void setY(double yCoord){
27. this.yCoord=yCoord;
28. }
29. public void setZ(double zCoord){
30. this.zCoord=zCoord;
31. }
32. public boolean equals(Point3d anotherPoint) {
33. return xCoord == anotherPoint.getX() &&
34. yCoord == anotherPoint.getY() &&
35. zCoord == anotherPoint.getZ();
36. }

2. Метод distanceTo

1. public double distanceTo(Point3d anotherPoint) {
2. return  Math.sqrt(Math.pow(xCoord - anotherPoint.getX(), 2) +
3. Math.pow(yCoord - anotherPoint.getY(), 2) +
4. Math.pow(zCoord - anotherPoint.getZ(), 2));
5. }}

3. Создайте другой класс под названием Lab1, который будет содержать статический метод main

import java.util.Scanner;

public class laba1 {

    public static Point3d readPoint3dFromUserInput() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Введите координаты точки: ");

        return new Point3d(

                scanner.nextInt(), scanner.nextInt(), scanner.nextInt());

    }

    public static void main(String[] args) {

        Point3d p1 = readPoint3dFromUserInput();

        Point3d p2 = readPoint3dFromUserInput();

        Point3d p3 = readPoint3dFromUserInput();

        if (p1.equals(p2) || p2.equals(p3) || p1.equals(p3)) {

            System.out.println("Одна из точек равна другой");

            return;

        }

        double area = computeArea(p1, p2, p3);

        System.out.printf("Полученная площадь: %.2f\n", area);

    }

    public static double computeArea(Point3d p1, Point3d p2, Point3d p3) {

        double a, b, c, p;

        a = p1.distanceTo(p2);

        b = p1.distanceTo(p3);

        c = p2.distanceTo(p3);

        p = (a + b + c) / 2;

        return Math.sqrt(p \* (p - a) \* (p - b) \* (p - c));

    }

}

4. Скомпилируйте оба исходных файла вместе: javac Point3d.java Lab1.java и затем запустите программу Lab1



