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

(МТУСИ)

Лабораторная работа №2 по теме

«Основы объектно-ориентированного программирования»

Выполнил:

Студент группы БФИ1801

Сиротин Н. С.

**Москва 2020**

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

**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.

**Код класса Extended2dTo3d:**

import java.lang.Math;  
  
  
public class Extended2dTo3d extends Point2d{  
 private double xCoord;  
 private double yCoord;  
 private double zCoord;  
  
 public Extended2dTo3d ( double x, double y, double z) {  
 xCoord = x;  
 yCoord = y;  
 zCoord = z;  
 }  
  
 public Extended2dTo3d () {  
 //Вызовите конструктор с тремя параметрами и определите источник.  
 this(0, 0,0);  
 }  
  
 //новый функционал для нового третьего измерения  
 public double getZ () {  
 return zCoord;  
 }  
  
 public void setZ ( double val) {  
 zCoord = val;  
 }  
  
  
 public boolean comparePoints (Extended2dTo3d point) {  
 return (xCoord == point.getX() || yCoord == point.getY() || zCoord == point.getZ());  
 }  
  
 public double distanceTo(Extended2dTo3d point) {  
 return Math.sqrt(Math.pow(xCoord - point.getX(), 2) + Math.pow(yCoord - point.getY(), 2) + Math.pow(zCoord - point.getZ(), 2));  
 }  
  
 public double heronsArea(Extended2dTo3d firstPoint, Extended2dTo3d secondPoint){  
 double p1 = distanceTo(firstPoint);  
 double BubleX = xCoord;  
 double BubleY = yCoord;  
 double BubleZ = zCoord;  
 double p3 = distanceTo(secondPoint);  
 xCoord = firstPoint.getX();  
 yCoord = firstPoint.getY();  
 zCoord = firstPoint.getZ();  
 double p2 = distanceTo(secondPoint);  
 xCoord = BubleX;  
 yCoord = BubleY;  
 zCoord = BubleZ;  
 double p = (p1 + p2 + p3)/2;  
 return Math.sqrt(p\*(p-p1)\*(p-p2)\*(p-p3));  
 }  
}

**Код класса Point3dMain для тестирования:**

public class Point3dMain {  
  
  
 public static void main(String[] args) {  
 Extended2dTo3d FirstPoint = new Extended2dTo3d(2.424, 6.153, 4.832);  
 Extended2dTo3d SecondPoint = new Extended2dTo3d(5.012, 9.732, 2.512);  
 Extended2dTo3d ThirdPoint = new Extended2dTo3d(1.612, 8.215, 10.561);  
 Extended2dTo3d FirstPointShadow = new Extended2dTo3d(2.424, 6.153, 4.832);  
  
 if (FirstPoint.comparePoints(SecondPoint) || FirstPoint.comparePoints(ThirdPoint) || SecondPoint.comparePoints(ThirdPoint)){  
 System.out.println("Какие-то из точек равны между собой !");  
  
 }else{  
 System.out.println(FirstPoint.comparePoints(SecondPoint));  
 System.out.println(FirstPoint.comparePoints(FirstPointShadow));  
 System.out.println(FirstPoint.distanceTo(SecondPoint));  
 System.out.println(FirstPoint.heronsArea(SecondPoint, ThirdPoint));  
 }  
 }  
}

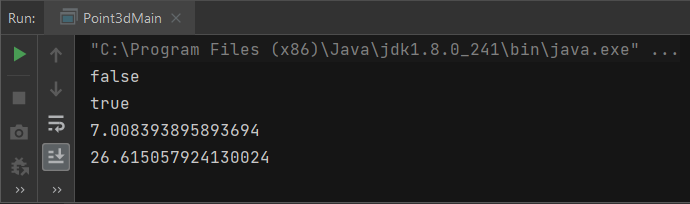


Рисунок 1 — Результаты работы с тестовыми данными

**Вывод:** Мы разработали новый класс на основе уже существующего и расширили его функционал, используя возможности наследования в языке Java.