Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное

бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра информатики

Лабораторная работа №4

«Основы объектно-ориентированного

программирования»

Выполнил студент

группы БФИ1902

Соцков И.Н

Москва, 2020 г.

Оглавление

[**1. Цель работы** 4](#_Toc54795543)

[**2. Ход работы** 4](#_Toc54795544)

[**3. Листинг программы** 5](#_Toc54795561)

[**4. Результат программы** 7](#_Toc54795562)

[Вывод 8](#_Toc54795565)

**1. Цель работы**

Цель данной лабораторной работы — использовать классы по одному на файл, чтобы описать, как объекты работают.

**2. Ход работы**

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.

1. 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.
2. 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.

1. Скомпилируйте оба исходных файла вместе: javac Point3d.java Lab1.java. Затем запустите программу Lab1, тестируя ее с несколькими образцами треугольников.

**3. Листинг программы**

Point3d.java:

package com.company;  
import java.util.Objects;  
  
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;  
 }  
  
 Point3d() {  
 xCoord = 0.0;  
 yCoord = 0.0;  
 zCoord = 0.0;  
 }  
  
 public void getX() {  
 System.out.println("x = " + xCoord);  
 }  
  
 public void getY() {  
 System.out.println("y = " + yCoord);  
 }  
  
 public void getZ() {  
 System.out.println("z = " + zCoord);  
 }  
  
 public void setX(double val) {  
 xCoord = val;  
 }  
  
 public void sety(double val) {  
 yCoord = val;  
 }  
  
 public void setz(double val) {  
 zCoord = val;  
 }  
 public void modify (Double x, Double y, Double z) {  
 if (x != null) {  
 xCoord = x;  
 }  
 if (y != null) {  
 yCoord = y;  
 }  
 if (z != null) {  
 zCoord = z;  
 }  
 }  
 public boolean Equals(Point3d object) {  
 return Objects.equals(this.xCoord, object.xCoord) && Objects.equals(this.yCoord, object.yCoord) && Objects.equals(this.zCoord, object.zCoord);  
 }  
 public Double distanceTo(Point3d object) {  
 return Math.floor(Math.sqrt(Math.pow(object.xCoord - this.xCoord, 2) + Math.pow(object.yCoord - this.yCoord, 2) + Math.pow(object.zCoord - this.zCoord, 2)) \* 100)/100;  
 }  
}

Lab2.java:

package com.company;  
import java.util.Scanner;  
  
public class Lab2 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 double x1 = sc.nextDouble();  
 double y1 = sc.nextDouble();  
 double z1 = sc.nextDouble();  
 double x2 = sc.nextDouble();  
 double y2 = sc.nextDouble();  
 double z2 = sc.nextDouble();  
 double x3 = sc.nextDouble();  
 double y3 = sc.nextDouble();  
 double z3 = sc.nextDouble();  
 Point3d x = new Point3d(x1, y1, z1);  
 Point3d y = new Point3d(x2, y2, z2);  
 Point3d z = new Point3d(x3, y3, z3);  
 if (x.Equals(y) == false && y.Equals(z) == false && z.Equals(x) == false) {  
 System.*out*.println("Площадь треугольника по точкам: " + *computeArea*(x, y, z));  
 } else {  
 System.*out*.println("Две или три точки треугольника имеют одинаковые координаты");  
 }  
 }  
 public static double computeArea(Point3d object1, Point3d object2, Point3d object3) {  
 double a = object1.distanceTo(object2);  
 double b = object2.distanceTo(object3);  
 double c = object3.distanceTo(object1);  
 double p = (a + b + c)/2;  
 return Math.*sqrt*(p\*(p-a)\*(p-b)\*(p-c));  
 }  
}

**4. Результат программы**

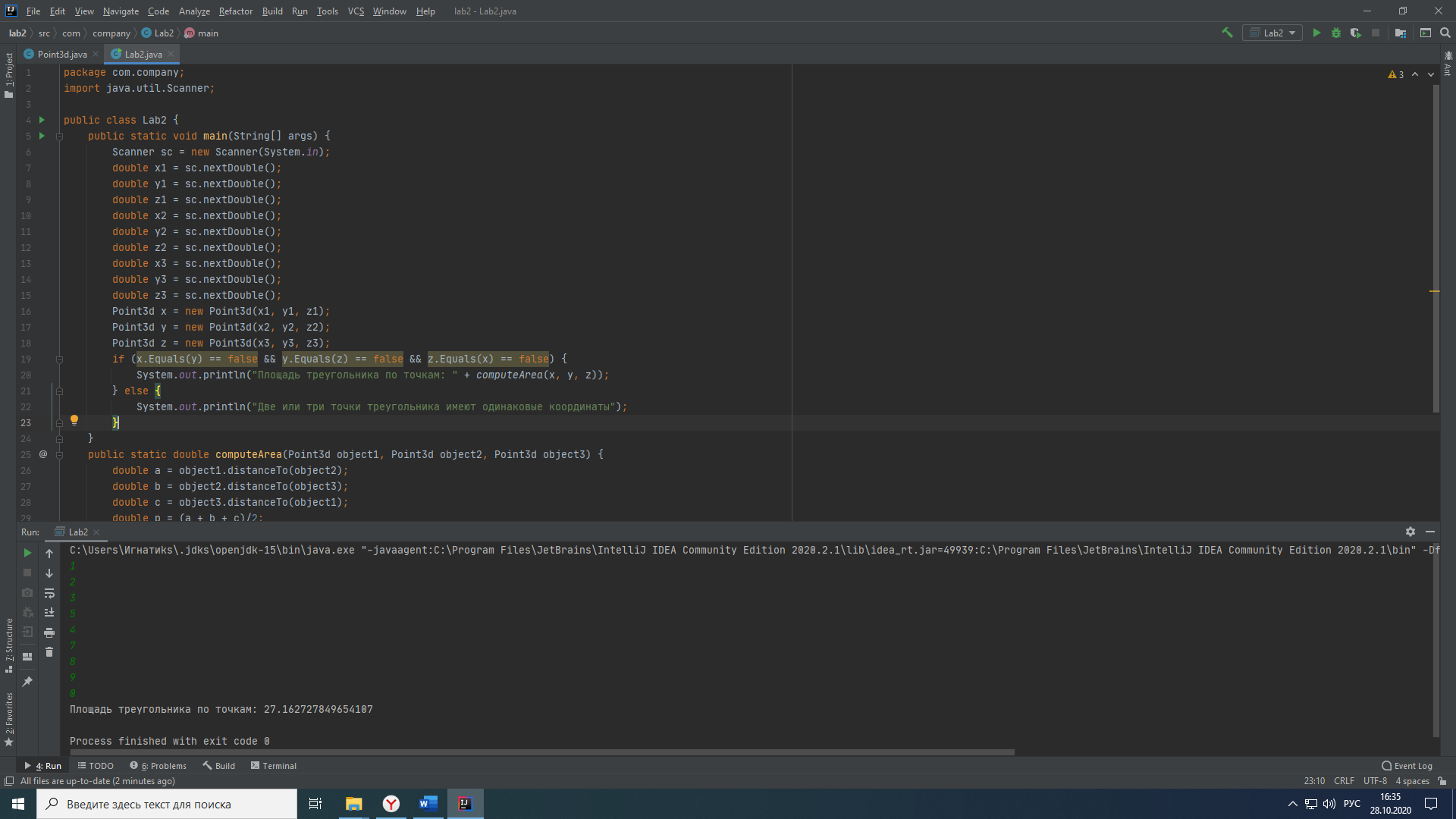


Рисунок 1 - Результат

Вывод

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