**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Информатики

**Отчет по лабораторной работе №2**

по предмету «Кроссплатформенные технологии программирования»

на тему:

«Основы объектно-ориентированного программирования»

Выполнил: студент группы БВТ 1802

Ганюшкина Елизавета Сергеевна

Руководитель:

Полянцева Ксения Андреевна

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.

**Задания**

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).
3. Скомпилируйте оба исходных файла вместе: javac Point3d.java Lab1.java и затем запустите программу Lab1, тестируя ее с несколькими образцами треугольников.

**Коды программ**

*lab1.java*

import java.text.DecimalFormat;  
import java.util.Scanner;  
public class lab1 extends Point3d{  
public static void main(String[] args){  
int check = 0;  
Scanner in = new Scanner(System.in);  
System.out.print("Введите координаты для первой точки: " + "\n");  
double a = in.nextInt();  
double b = in.nextInt();  
double c = in.nextInt();  
System.out.print("Введите координаты для второй точки: " + "\n");  
 double a1 = in.nextInt();  
double b1 = in.nextInt();  
double c1 = in.nextInt();  
System.out.print("Введите координаты для третьей точки: " + "\n");  
double a2 = in.nextInt();  
double b2 = in.nextInt();  
double c2 = in.nextInt();  
Point3d Point = new Point3d(a,b,c);  
Point3d Point1 = new Point3d(a1,b1,c1);  
Point3d Point2 = new Point3d(a2,b2,c2);  
in.close();  
Point.setX(a);  
Point1.setX(a1);  
Point2.setX(a2);  
Point.setY(b);  
Point1.setY(b1);  
Point2.setY(b2);  
Point.setZ(c);  
Point1.setZ(c1);  
Point2.setZ(c2);  
if (compare(Point, Point2)||compare(Point2, Point1)||compare(Point, Point1)||!Exist(Point, Point1, Point2)) // Проверка координат точек{  
System.out.println("Треугольник не существует.");  
check ++;}  
if (check == 0){  
if (Area(Point, Point1, Point2) == 0){  
System.out.println("Треугольник вырожденный. Все точки лежат на одной прямой.");}  
else{  
DecimalFormat df = new DecimalFormat("###.##");  
System.out.println("Площадь треугольника: ");  
System.out.print(df.format(Area(Point, Point1, Point2)));}}}  
public static double Area(Point3d a, Point3d b, Point3d c) // Нахождение площади{  
double Side1 = distanceTo(a, b), Side2 = distanceTo(a, c), Side3 = distanceTo(c, b);  
double per = (Side1 + Side2 + Side3) / 2;  
return Math.sqrt(per \* (per - Side1) \* (per - Side2) \* (per - Side3));}  
public static boolean Exist(Point3d a, Point3d b, Point3d c){  
if (distanceTo(a, b)> distanceTo(a, c)+distanceTo(b, c) ){  
return false;}  
else if (distanceTo(a, c)> distanceTo(a, b)+distanceTo(b, c)){  
return false;}  
else return !(distanceTo(b, c) > distanceTo(a, c) + distanceTo(a, b));  
}}

*Point2d.java*

public class Point2d{  
private double xCoord; // Координата X  
private double yCoord; // Координата Y  
public Point2d(double x, double y) // Конструктор инициализации{  
xCoord = x;  
yCoord = y;}  
public Point2d(){  
this(0,0);}  
public double getX() // Возвращение координаты X{  
return xCoord;}  
public double getY() // Возвращение координаты Y{  
return yCoord;}  
public void setX(double val) // Установка значения координаты X{  
xCoord = val;}  
public void setY(double val) // Установка значения координаты Y{  
yCoord = val;}  
public static boolean compare(Point2d a, Point2d b) // Сравнение значений точек{  
if ((a.getX() == b.getX()) && (a.getY() == b.getY()))  
return true;  
else  
return false;}}

*Point3d.java*

import java.awt.\*;  
import java.util.Scanner;  
public class Point3d extends Point2d {  
static double sqr (double a ) { //метод для возведения числа в квадрат

double result = a\*a;  
return result;}  
public static void main(String[] args){

Scanner in = new Scanner(System.in);  
System.out.print("Введите координаты для первой точки: " + "\n");  
double a = in.nextInt();  
double b = in.nextInt();  
double c = in.nextInt();  
System.out.print("Введите координаты для второй точки: " + "\n");  
double a1 = in.nextInt();  
double b1 = in.nextInt();  
double c1 = in.nextInt();  
Point3d Point = new Point3d(a,b,c);  
Point3d Point1 = new Point3d(a1,b1,c1);  
Point.setX(a);  
Point1.setX(a1);  
Point.setY(b);  
Point1.setY(b1);  
in.close(); //Останавливаем работу Scanner  
if(compare(Point, Point1)) //Сравниваем координаты точек  
System.out.println("Точки имеют одинаковые координаты.");  
else {  
System.out.println("Точки имеют разные координаты.");//Если расстояние между точками разное  
System.out.println("Расстояние между точками: "); //Рассчитываем расстояние между точками  
System.out.print(distanceTo(Point, Point1));}}  
private double zCoord; // Координата Z  
public Point3d(double x, double y, double z) // Конструктор инициализации{  
// Координата X  
// Координата Y  
zCoord = z;}  
public Point3d(){  
this(0, 0, 0);}  
public double getZ() // Возвращение координаты Z{  
return zCoord;}  
public void setZ(double val) // Установка значения координаты Z{  
zCoord = val;}  
public static boolean compare(Point3d a, Point3d b) // Сравнение значений точек{  
return (a.getX() == b.getX()) && (a.getY() == b.getY()) && (a.getZ() == b.getZ());}  
public static double distanceTo(Point3d a, Point3d b) // Нахождение расстояния между точками{  
double Round = Math.sqrt(sqr((b.getX()-a.getX())) + sqr((b.getY()-a.getY())) + sqr((b.getZ()-a.getZ())));  
Round \*= 100;  
Round = (int) Round;  
return (Round / 100);}}

**Результат работы кода**

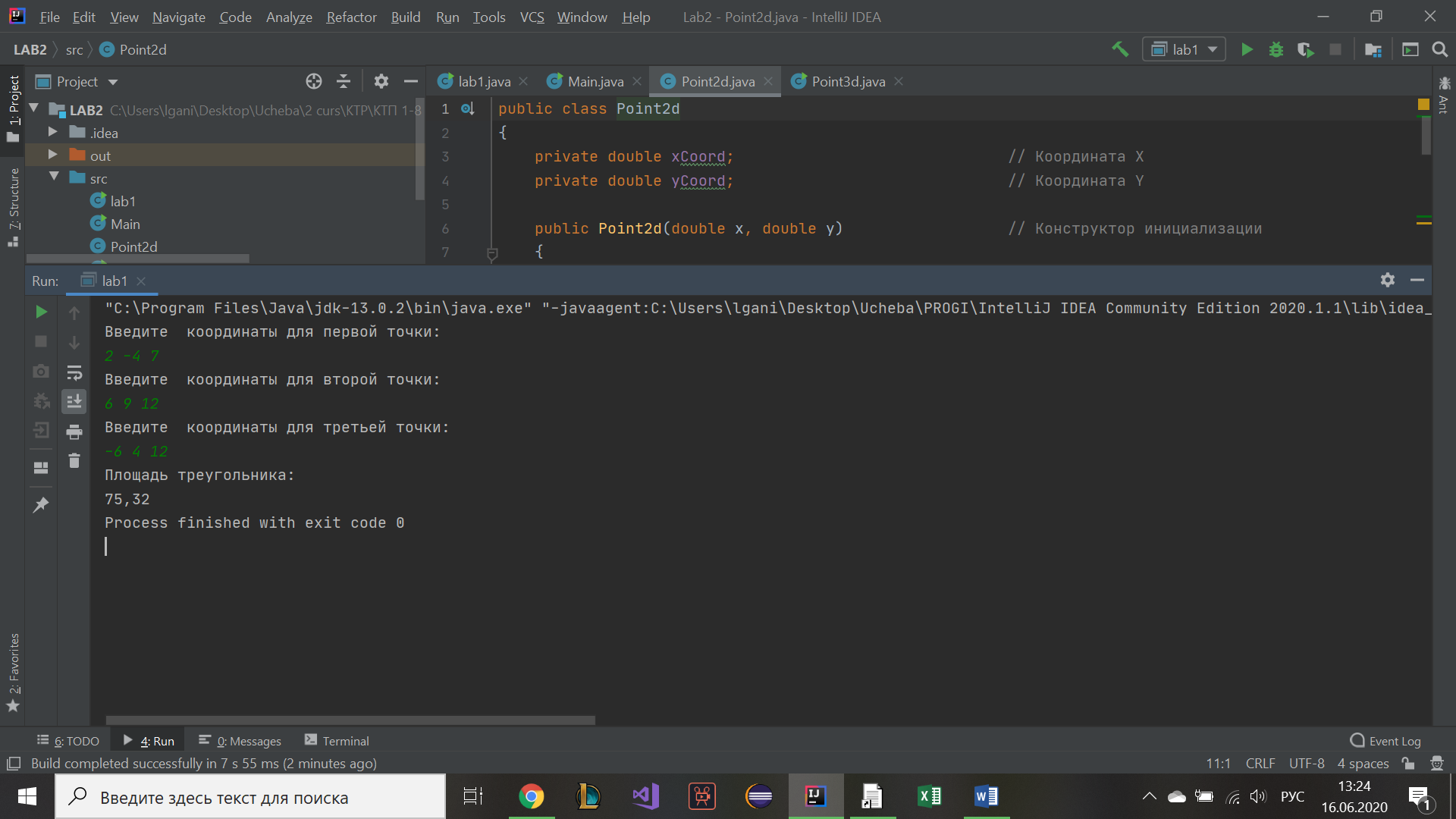


Рисунок 1 - Поиск площади треугольника

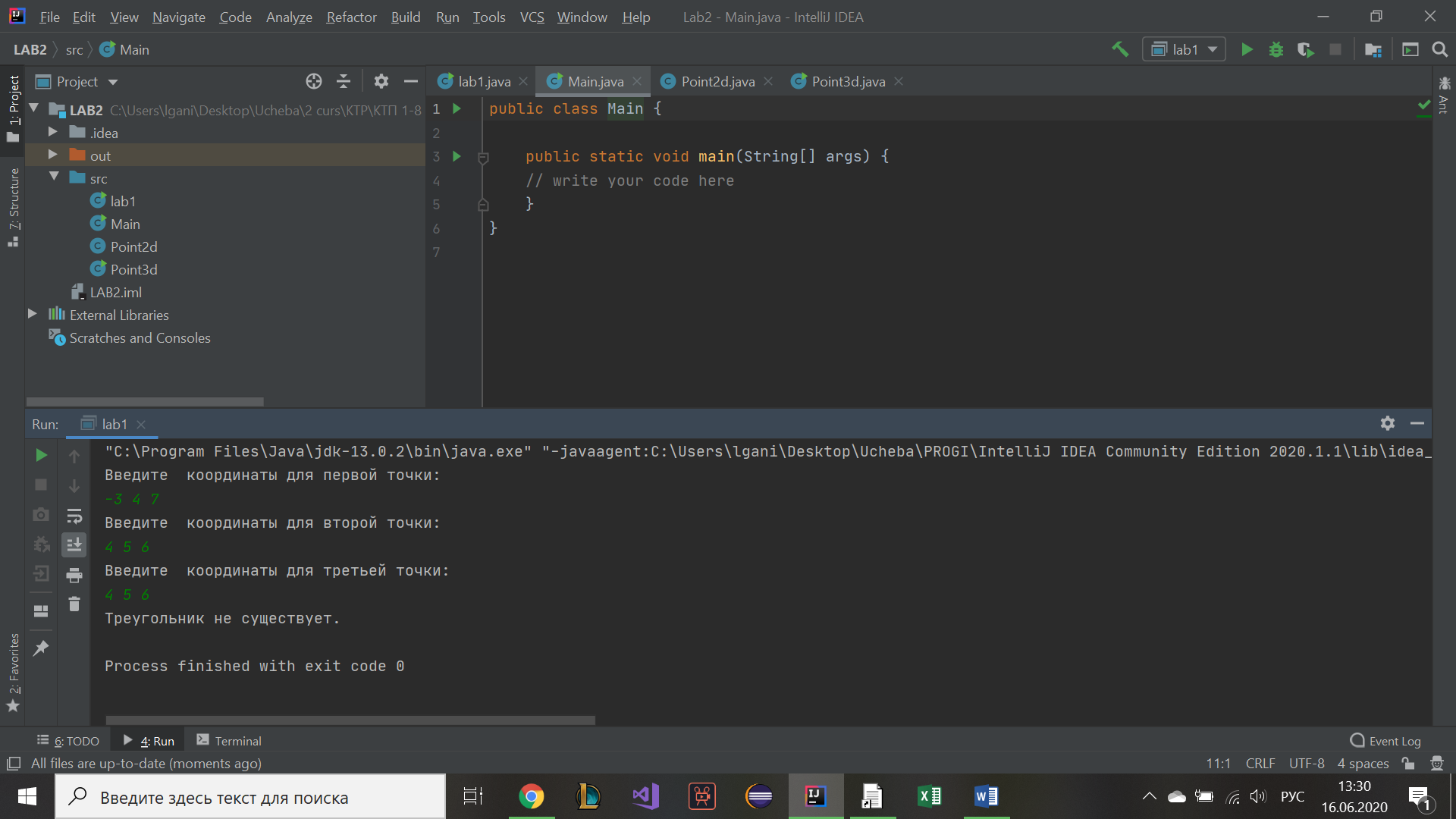


Рисунок 2 - Получение информации о существовании / не существовании треугольника

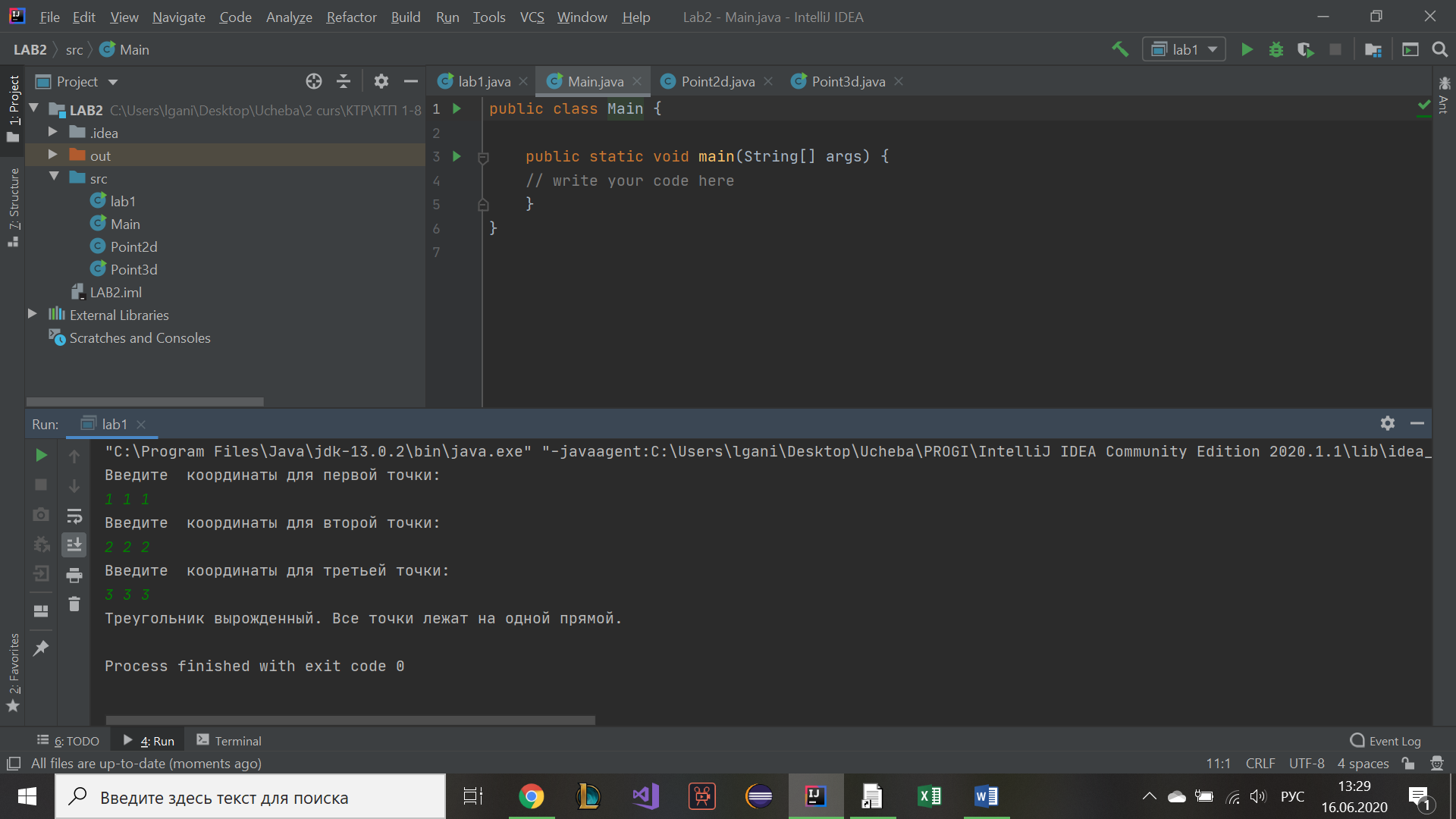


Рисунок 3 - Получение информации о вырожденности / не вырожденности треугольника

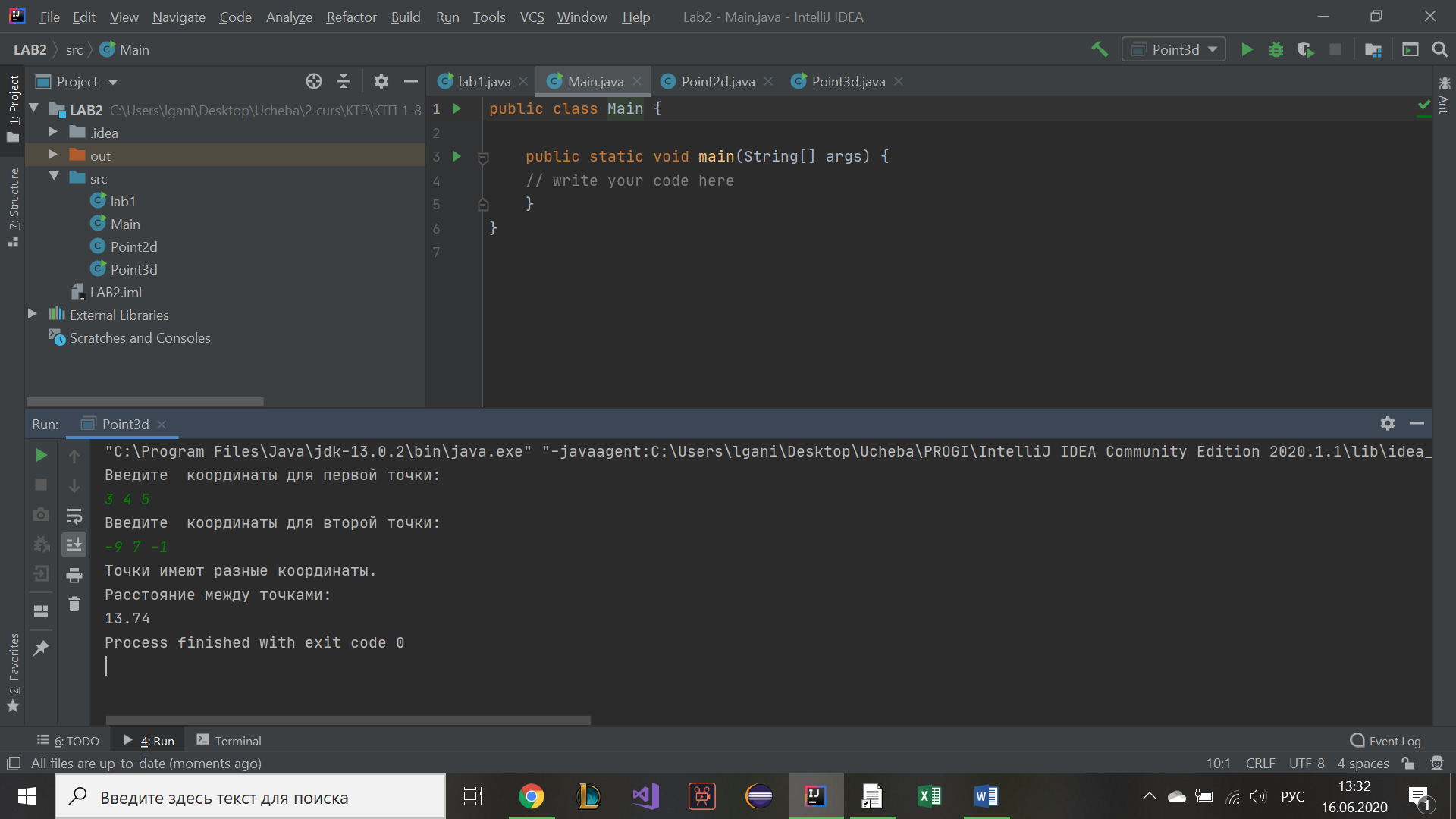


Рисунок 4 – Нахождение расстояния между точками

**Вывод**

В данной лабораторной работе были созданы :  
1) Конструктор инициализации Point3d

2) Методы устанавливающие и возвращающие значение координаты Z   
3) Метод для сравнения координат точек.

4) Метод Area для нахождения площади треугольника по трем точкам в трехмерном пространстве через формулу Герона.

5) 3 объекта класса Point3d.

В методах :  
1) distanceTo нашли расстояние между двумя произвольными точками с помощью формулы.

2) main считали введенные с клавиатуры координаты для точек.

3) Exist проверили, существует ли треугольник.

4) main считывали введенные пользователем координаты для трех точек.

Установили значения координат методами set.