**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 «КТП»

Выполнил: студент группы БВТ1802

Самаков Владислав Владимирович Руководитель:

Ксения Андреевна Полянцева

Москва 2020

**1 Цель работы**

Цель работы: реализовать функцию вычисления площади треугольника в трехмерном пространстве.

**2 Задание**

Написать классы Point2D и Point3D, которые будут представлять точки в трехмерном пространстве. Написать функцию подсчета площади. Предусмотреть случай, когда по введенным точкам невозможно построить треугольник.

**3 Текст программы**

**Class Main**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String args[]) {  
  
 // Ввод координат  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите координаты точек треугольника в трехмерном пространстве:");  
  
 System.*out*.println("Координаты первой точки:");  
 point3D first = new point3D(scanner.nextDouble(), scanner.nextDouble(), scanner.nextDouble());  
 System.*out*.println("Введите координаты второй точки:");  
 point3D second = new point3D(scanner.nextDouble(), scanner.nextDouble(), scanner.nextDouble());  
 System.*out*.println("Введите координаты третьей точки:");  
 point3D third = new point3D(scanner.nextDouble(), scanner.nextDouble(), scanner.nextDouble());  
  
 System.*out*.print("Площадь: ");  
 System.*out*.println(*area*(first, second, third));  
 }  
  
 public static double area(point3D first, point3D second, point3D third ) {  
  
 // проверка отсутсвия совпадений координат точек  
 double distance12 = first.distanceTo(second);  
 double distance23 = second.distanceTo(third);  
 double distance13 = first.distanceTo(third);  
  
 if (distance12 == 0 || distance23 == 0 || distance13 == 0) {  
 System.*out*.println("Треугольник с данными точками не существует.");  
 return 0;  
 }  
  
 // вычисление площади  
 double halfPerimeter = (distance12 + distance23 + distance13) / 2;  
  
 double result = Math.*sqrt*(halfPerimeter \* (halfPerimeter - distance12) \* (halfPerimeter - distance23) \*  
 (halfPerimeter - distance13));  
  
 return result;  
 }  
}

**Class Point2d**

public class point2D {  
  
 public point2D(double x, double y) {  
 xCoord = x;  
 yCoord = y;  
 }  
  
 public point2D() {  
 this(0, 0);  
 }  
  
 public double getX() {  
 return xCoord;  
 }  
  
 public void setX(double x) {  
 xCoord = x;  
 }  
  
  
 public double getY() {  
 return yCoord;  
 }  
  
 public void setY(double x) {  
 yCoord = x;  
 }  
  
 protected double xCoord;  
 protected double yCoord;  
  
}

**Class Point3d**

public class point3D extends point2D {  
  
  
 public point3D(double x, double y, double z) {  
 xCoord = x;  
 yCoord = y;  
 zCoord = z;  
 }  
  
 public point3D() {  
 this(0, 0, 0);  
 }  
  
  
 public double distanceTo(point3D point) {  
 int scale = 100;  
  
 double distX = point.getX() - this.xCoord;  
 double distY = point.getY() - this.yCoord;  
 double distZ = point.getZ() - this.zCoord;  
 double distance = Math.*sqrt*(distX\*distX + distY\*distY + distZ\*distZ);  
  
 return Math.*round*(distance\*scale) / (double) scale;  
 }  
  
 public double getZ() {  
 return zCoord;  
 }  
  
 public void setZ(double z) {  
 zCoord = z;  
 }  
  
  
  
 private double zCoord;  
}

**4 Работа программы**

