**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

по дисциплине «Технологии разработки программного обеспечения»

на тему: «Основы объектно-ориентированного программирования»

Выполнил: студент группы БВТ1801

Кирилин Игорь Андреевич

Руководитель:

Мосева Марина Сергеевна

Москва 2019

**Цель работы**: изучить основы объектно-ориентированного программирования с помощью нескольких задач. Использовать классы по одному на файл, чтобы описать, как эти объекты работают.

**Выполнение работы:**

Класс Point2d

public class Point2d {  
 //Координата X  
 protected double xCoord;  
 //Координата Y  
 protected double yCoord;  
 public Point2d(double x, double y) {  
 xCoord = x;  
 yCoord = y;  
 }  
 //Конструктор по умолчанию  
 public Point2d() {  
 this(0, 0);  
 }  
 //Возвращение координаты X  
 public double getX() {  
 return xCoord;  
 }  
 //Возвращение координаты Y  
 public double getY() {  
 return yCoord;  
 }  
 //Установка значения координаты X  
 public void setX(double val) {  
 xCoord = val;  
 }  
 //Установка значения координаты Y  
 public void setY(double val) {  
 yCoord = val;  
 }  
 //Метод для сравнения полей объектов  
 public boolean areEqual(Object o) {  
 if (this == o)  
 return true;  
 else return false;  
 }  
}

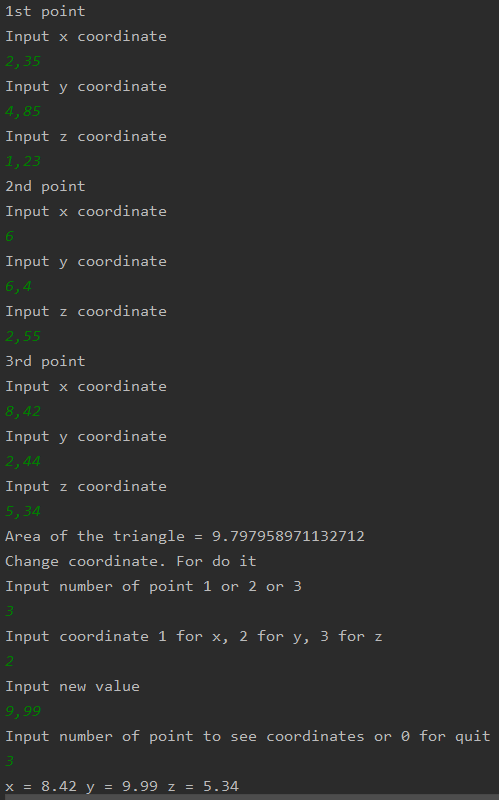
Класс Point3d

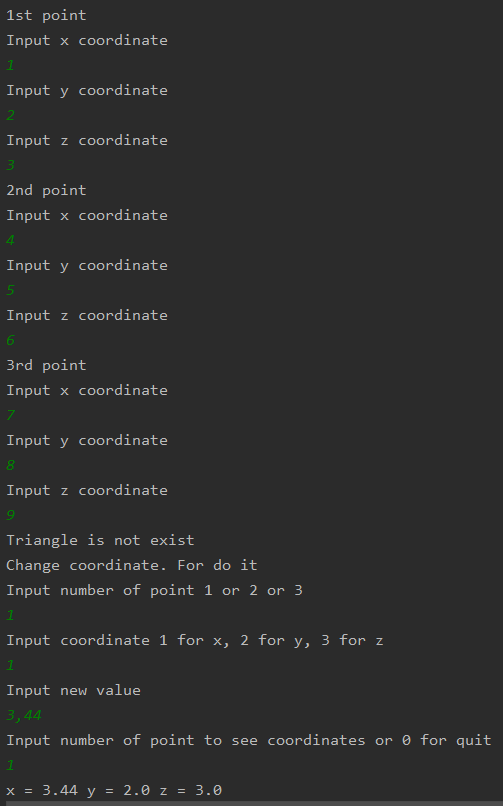
import java.lang.Math;  
  
public class Point3d extends Point2d {  
 /\* Координата Z \*/  
 private double zCoord;  
 public Point3d(double x, double y, double z) {  
 super(x, y);  
 zCoord = z;  
 }  
 /\* Конструктор по умолчанию \*/  
 public Point3d() {  
 this(0, 0, 0);  
 }  
 /\* Возвращение координаты Z \*/  
 public double getZ() {  
 return zCoord;  
 }  
 /\* Установка значения координаты Z \*/  
 public void setZ(double val) {  
 zCoord = val;  
 }  
 /\* Метод для нахождения расстояния между двумя точками \*/  
 public double distanceTo(Point3d o) {  
 double dist = Math.*sqrt*(  
 Math.*pow*(this.getX() - o.getX(), 2) +  
 Math.*pow*(this.getY() - o.getY(), 2) +  
 Math.*pow*(this.getZ() - o.getZ(), 2));  
 return Math.*round*(dist \* 100) / 100;  
 }  
}

Класс Lab1

import java.util.Scanner;  
  
public class Lab1 {  
 public static double computeArea(Point3d o1, Point3d o2, Point3d o3) {  
 double AB = o1.distanceTo(o2);  
 double BC = o2.distanceTo(o3);  
 double AC = o1.distanceTo(o3);  
 double p = 0.5 \* (AB + BC + AC);  
 return Math.*sqrt*(p \* (p - AB) \* (p - BC) \* (p - AC));  
 }  
  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 double x, y, z;  
 Point3d point0 = new Point3d();  
  
 System.*out*.println("1st point");  
 System.*out*.println("Input x coordinate");  
 x = in.nextDouble();  
 System.*out*.println("Input y coordinate");  
 y = in.nextDouble();  
 System.*out*.println("Input z coordinate");  
 z = in.nextDouble();  
 Point3d point1 = new Point3d(x, y, z);  
  
 System.*out*.println("2nd point");  
 System.*out*.println("Input x coordinate");  
 x = in.nextDouble();  
 System.*out*.println("Input y coordinate");  
 y = in.nextDouble();  
 System.*out*.println("Input z coordinate");  
 z = in.nextDouble();  
 Point3d point2 = new Point3d(x, y, z);  
  
 System.*out*.println("3rd point");  
 System.*out*.println("Input x coordinate");  
 x = in.nextDouble();  
 System.*out*.println("Input y coordinate");  
 y = in.nextDouble();  
 System.*out*.println("Input z coordinate");  
 z = in.nextDouble();  
 Point3d point3 = new Point3d(x, y, z);  
  
 if (*computeArea*(point1, point2, point3) == 0)  
 System.*out*.println("Triangle is not exist");  
 else  
 System.*out*.println("Area of the triangle = " + *computeArea*(point1, point2, point3));  
  
 System.*out*.println("Change coordinate. For do it");  
 boolean set = false;  
 while (set == false) {  
 System.*out*.println("Input number of point 1 or 2 or 3");  
 int j = in.nextInt();  
 if (j == 1) {  
 System.*out*.println("Input coordinate 1 for x, 2 for y, 3 for z");  
 int k = in.nextInt();  
 System.*out*.println("Input new value");  
 double val = in.nextDouble();  
 if (k == 1)  
 point1.setX(val);  
 if (k == 2)  
 point1.setY(val);  
 if (k == 3)  
 point1.setZ(val);  
 set = true;  
 }  
 else if (j == 2) {  
 System.*out*.println("Input coordinate 1 for x, 2 for y, 3 for z");  
 int k = in.nextInt();  
 System.*out*.println("Input new value");  
 double val = in.nextDouble();  
 if (k == 1)  
 point2.setX(val);  
 if (k == 2)  
 point2.setY(val);  
 if (k == 3)  
 point2.setZ(val);  
 set = true;  
 }  
 else if (j == 3) {  
 System.*out*.println("Input coordinate 1 for x, 2 for y, 3 for z");  
 int k = in.nextInt();  
 System.*out*.println("Input new value");  
 double val = in.nextDouble();  
 if (k == 1)  
 point3.setX(val);  
 if (k == 2)  
 point3.setY(val);  
 if (k == 3)  
 point3.setZ(val);  
 set = true;  
 }  
 else  
 System.*out*.println("Not existing coordinate");  
 }  
  
 set = false;  
 while (set == false) {  
 System.*out*.println("Input number of point to see coordinates or 0 for quit");  
 int j = in.nextInt();  
 if (j == 1) {  
 System.*out*.println("x = " + point1.getX() + " y = " + point1.getY() + " z = " + point1.getZ());  
 }  
 else if (j == 2) {  
 System.*out*.println("x = " + point2.getX() + " y = " + point2.getY() + " z = " + point2.getZ());  
 }  
 else if (j == 3) {  
 System.*out*.println("x = " + point3.getX() + " y = " + point3.getY() + " z = " + point3.getZ());  
 }  
 else if (j == 0)  
 set = true;  
 else  
 System.*out*.println("Not existing point");  
 }  
 }  
}

Тесты программы





**Вывод**: в ходе лабораторной работы я изучил основы объектно-ориентированного программирования.