**Федеральное агентство связи**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

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

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе № 2

по дисциплине «Объектно-ориентированное программирование»

Выполнил: студент группы БУТ1952

Обухов А.Д.

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2020

Цель работы:

Изучить использование классов в Java с использованием индивидуальных файлов для каждого класса.

Задание.

1. Создайте новый класс Point3d для представления точек в трехмерном Евклидовом пространстве.
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.

Для выполнения поставленной задачи был создан класс Point3d, и класс Lab1.  
Исходный код Point3d:

//Класс трехмерной точки

public class Point3d {

//X координата

private double x;

//Y координата

private double y;

//Z координата

private double z;

//Конструктор по умолчанию, делает точку с координатами (0,0,0)

public Point3d() {

this(0.0, 0.0, 0.0);

}

//Конструктор, делает точку с координатами (x,y,z)

public Point3d(double x, double y, double z) {

this.x = x;

this.y = y;

this.z = z;

}

//Метод для получения X координаты

public double getX() {

return x;

}

//Метод для получения Y координаты

public double getY() {

return y;

}

//Метод для получения Z координаты

public double getZ() {

return z;

}

//Метод для установления X координаты

public void setX(double x) {

this.x = x;

}

//Метод для установления Y координаты

public void setY(double y) {

this.y = y;

}

//Метод для установления Z координаты

public void setZ(double z) {

this.z = z;

}

//Метод для сравнения значений двух объектов Point3d

public boolean equals(Point3d p) {

return x == p.x && y == p.y && z == p.z;

}

//Расстояние между двумя точками

public double distanceTo(Point3d p) {

return Math.sqrt(Math.pow(x - p.x, 2) + Math.pow(y - p.y, 2) + Math.pow(z - p.z, 2));

}

}

Исходный код Lab1:

public class Lab1 {

//Точка входа в программу

public static void main(String[] args) {

Point3d[] points = new Point3d[3];

for (int i = 0; i < args.length; i += 3) {

points[i / 3] = new Point3d(

Double.parseDouble(args[i]),

Double.parseDouble(args[i + 1]),

Double.parseDouble(args[i + 2])

);

}

if ((points[0].equals(points[1])) || (points[0].equals(points[2])) || (points[1].equals(points[2]))){

System.out.println("Some of points are equals, it is not a triangle");

return;

}

System.out.printf("Area of triangle: %.2f \n", computeArea(points[0], points[1], points[2]));

}

//Метод вычисляющий площадь треугольника по трем точкам

public static double computeArea(Point3d p1, Point3d p2, Point3d p3) {

double a = p1.distanceTo(p2);

double b = p1.distanceTo(p3);

double c = p2.distanceTo(p3);

double p = (a + b + c) / 2.0;

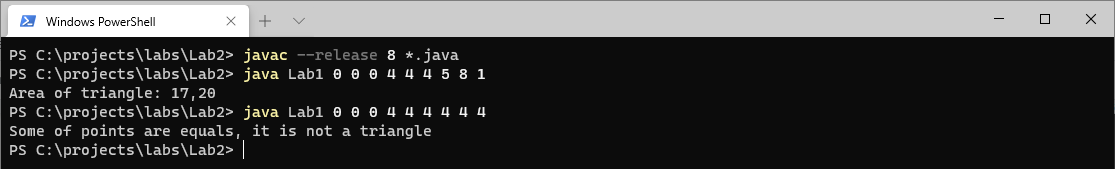
double s = p \* (p - a) \* (p - b) \* (p - c);

return Math.sqrt(s);

}

}

При запуске программы передаются координаты точек как аргументы командной строки и получаем площадь треугольника либо сообщение, что некоторые из точек равны и треугольник не может быть образован.

  
Рис. 1 – выполнение программы.

Заключение.

В ходе выполнения этой лабораторной работы было изучено написание классов Java в индивидуальных файлах.