**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

Сасс В.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 я написал класс Point3D, наследующий функционал класса Point2d, и класс Lab1.  
Исходный код Point3D:

public class Point3d extends Point2d{

    // Class description

    // Coordinates z

    private double zCoord;

    // Class contructor

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

        super.setX(x);

        super.setY(y);

        zCoord = z;

    }

    // Class contructor without arguments

    public Point3d(){

        this (0, 0, 0);

    }

    // Get z coordinate

    public double getZ() {

        return zCoord;

    }

    // Set z coordinate

    public void setZ(double z) {

        zCoord = z;

    }

    // Overriding method equals()

    public boolean equals(Point3d objPoint3d){

        double objx = objPoint3d.getX();

        double objy = objPoint3d.getY();

        double objz = objPoint3d.getZ();

        double xCoord = this.getX();

        double yCoord = this.getY();

        if ( (xCoord == objx) && (yCoord == objy) && (zCoord == objz)){

            return true;

        }

        return false;

    }

    // Getting distance between points

    public double distanseTo(Point3d objPoint3d) {

        double objx = objPoint3d.getX();

        double objy = objPoint3d.getY();

        double objz = objPoint3d.getZ();

        double xCoord = this.getX();

        double yCoord = this.getY();

        return Math.sqrt(Math.pow(objx - xCoord, 2) + Math.pow(objy - yCoord, 2) + Math.pow(objz - zCoord, 2));

    }

}

Исходный код Lab1:

public class Lab1 {

    public static void main(String[] args) {

        Point3d[] points = new Point3d[3];

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

            double x = Double.parseDouble(args[i]);

            double y = Double.parseDouble(args[i+1]);

            double z = Double.parseDouble(args[i+2]);

            points[i / 3] = new Point3d(x, y, z);

        }

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

            System.out.println("Some points are equal, there's no triangle");

        }

        else{

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

        }

    }

    public static double computeArea(Point3d first, Point3d second, Point3d third) {

        // Returns area of triangle

        double a = first.distanseTo(second);

        double b = first.distanseTo(third);

        double c = second.distanseTo(third);

        double p = (a + b + c) / 2.0;

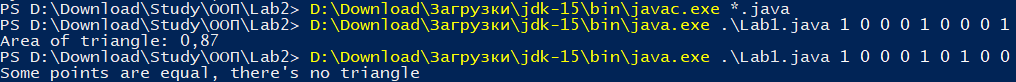
        double s = p \* (p - a) \* (p - b) \* (p - c);

        return Math.sqrt(s);

    }

}

При запуске программы мы передаём координаты точек как аргументы командной строки и получаем площадь треугольника либо сообщение, что существуют равные точки.

  
Рис. 1 – выполнение программы.

Заключение.

В ходе выполнения этой лабораторной работы я научился написанию классов Java в индивидуальных файлах.