**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

«Основы объектно-ориенированного программирования»

по дисциплине «Объектно-ориентированное программирование»

Выполнила студентка

группы БУТ 1952

Рюмина К.С.

Москва, 2020

В данной лабораторной работе целью является изучение использования классов для описание работы каждого объекта.

**Задания.**

1. **Создать новый класс Point3d Для представления точек в трехмерном Евклидном 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.**

Для выполнения задания был написан класс Point3d (рис.2), который составлен исходя из классов Point2d и Lab1 (рис.1)

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) {  
 // Возвращение треугольника  
 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);  
 }  
  
}

Рисунок 1 – Исходный код Lab1

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, + Math.pow(objz - zCoord, 2));

}  
}

Рисунок 2 – Исходный код Point3d

Запускаем команду, передав ранее координаты точек как аргументы для командной строки. В результате получим либо площадь треугольника, либо сообщение о том, что присутствуют равные точки.

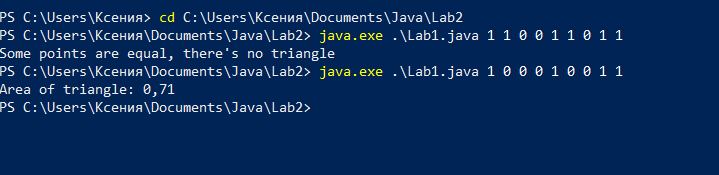


Рисунок 3 – выполнение программы

**Заключение.**

В ходе выполнения данной работы я изучила написание классов Java в индивидуальных файлах