Федеральное агенство связи

Ордена Трудового Красного Знамени федеральное государственное

бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и

информационных технологий

Лабораторная работа №2

по дисциплине: «Технология разработки программного обеспечения»

на тему: «Основы объектно-ориентированного программирования»

Выполнил студент

группы БФИ1902

Рахимов Е.К.

Проверила:

Мосева М.С.

Москва, 2020 г.

Оглавление

[1. Цель лабораторной работы 3](#_Toc53502483)

[2. Задание на лабораторную работу 3](#_Toc53502484)

[3. Ход лабораторной работы 3](#_Toc53502485)

[3.1 Алгоритм решение заданий 3](#_Toc53502486)

[3.1.1 Paint3d 3](#_Toc53502487)

[3.1.2 Lab1 4](#_Toc53502488)

[3.2 Листинг программы 4](#_Toc53502489)

[3.3 Результат выполнения программы 8](#_Toc53502490)

[Список использованных источников 10](#_Toc53502491)

# 1. Цель лабораторной работы

Цель данной лабораторной работы — изучить основы синтаксиса Java с помощью нескольких простых задач программирования, научиться использовать компилятор Java и виртуальную машину Java для запуска программы.

# 2. Задание на лабораторную работу

1) **Paint3d**:

Создать класс для представления точек в трехмерном пространтсве.

2) **Lab1**:

Создать класс в котором будет происходить запуск методов Paint3d

# 3. Ход лабораторной работы

## 3.1 Алгоритм решение заданий

### 3.1.1 Paint3d

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.
* 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.

### 3.1.2 Lab1

Создайте другой класс под названием Lab1, который будет содержать статический метод main. Помните, что метод main должен быть общедоступным (public) с возвращаемым значением void, а в качестве аргумента должен принимать строку (String). Этот класс должен иметь следующую функциональность:

* Ввод координат трех точек, находящихся в трехмерном пространстве. Создание трех объектов типа Point3d на основании полученных данных. (Предполагается, что пользователь вводит корректные данные.)
* Создайте второй статический метод computeArea, который принимает три объекта типа Point3d и вычисляет площадь треугольника, образованного этими точками. (Вы можете использовать формулу Герона.) Верните получившееся значение площади в формате типа double.
* На основе полученных данных и с использованием реализованного алгоритма посчитайте площадь и выведите полученное значение пользователю

## 3.2 Листинг программы

package Labs;  
  
public class Point2d {  
 public static void main(String[] args) {  
 Point2d myPoint = new Point2d ();//создает точку (0,0)  
 Point2d myOtherPoint = new Point2d (5,3);//создает точку (5,3)  
 Point2d aThirdPoint = new Point2d ();  
  
 }  
 private double xCoord;  
  
 private double yCoord;  
  
 public Point2d ( double x, double y) {  
 xCoord = x;  
 yCoord = y;  
 }  
  
 public Point2d () {  
 this(0, 0);  
 }  
 public double getX () {  
 return xCoord;  
 }  
  
 public double getY () {  
 return yCoord;  
 }  
  
 public void setX ( double val) {  
 xCoord = val;  
 }  
  
 public void setY ( double val) {  
 yCoord = val;  
 }  
}

public class Point3d extends Point2d{  
 public static void main(String[] args) {  
 Point3d myPoint = new Point3d ();//создает точку (0.0,0.0,0.0)  
 Point3d myOtherPoint = new Point3d (5,3,3);//создает точку (5,3,3)  
 Point3d myOtherPoint3 = new Point3d (5,3,3);//создает точку (5,3,3)  
 Point3d myOtherPoint1 = new Point3d (3,2,1);//создает точку (5,3,3)  
 System.*out*.println(*distanceTo*(myOtherPoint1,myOtherPoint));  
 System.*out*.println(*comparePoint3d*(myOtherPoint,myOtherPoint3)+"=true");  
 System.*out*.println(*comparePoint3d*(myOtherPoint,myOtherPoint1)+"=false");  
  
 }  
  
 private double zCoord;  
  
 public static double distanceTo(Point3d s,Point3d g){  
  
 double D=Math.*sqrt*(Math.*pow*(s.getX() - g.getX(),2)+Math.*pow*(s.getY() - g.getY(),2)+Math.*pow*(s.getZ() - g.getZ(),2));  
 return D;  
 }  
 public Point3d ( double x, double y, double z) {  
 super(x,y);  
 zCoord = z;  
 }  
  
 public Point3d () {  
 this(0.0, 0.0,0.0);  
 }  
  
 public double getZ () {  
 return zCoord;  
 }  
  
 public void setZ (double val) {  
 zCoord = val;  
 }  
  
 public static boolean comparePoint3d (Point3d one,Point3d two){  
 int c1,c2,c3;  
 c1=Double.*compare*(one.getX(),two.getX());  
 c2=Double.*compare*(one.getY(),two.getY());  
 c3=Double.*compare*(one.getZ(),two.getZ());  
 return (c1==0 && c2==0 && c3==0);  
 }  
}

public class Lab1 {  
 public static void main(String[] args) throws IOException {  
 Point3d myPoint1 = new Point3d(Double.*parseDouble*(args[0]), Double.*parseDouble*(args[1]), Double.*parseDouble*(args[2]));  
 Point3d myPoint2 = new Point3d(Double.*parseDouble*(args[3]), Double.*parseDouble*(args[4]), Double.*parseDouble*(args[5]));  
 Point3d myPoint3 = new Point3d(Double.*parseDouble*(args[6]), Double.*parseDouble*(args[7]), Double.*parseDouble*(args[8]));  
 if(Point3d.*comparePoint3d*(myPoint1,myPoint2) ||Point3d.*comparePoint3d*(myPoint1,myPoint3) ||Point3d.*comparePoint3d*(myPoint2,myPoint3) ){  
 System.*out*.println("Не возможно найти площадь");  
 }  
 else {  
 System.*out*.println(*computerArea*(myPoint1,myPoint2,myPoint3));  
 }  
  
 }  
public static double computerArea(Point3d myPint1,Point3d myPint2,Point3d myPint3){  
 double AB=Math.*sqrt*(Math.*pow*(myPint1.getX() - myPint2.getX(),2)+Math.*pow*(myPint1.getY() - myPint2.getY(),2)+Math.*pow*(myPint1.getZ() - myPint1.getZ(),2));  
 double BC=Math.*sqrt*(Math.*pow*(myPint2.getX() - myPint3.getX(),2)+Math.*pow*(myPint2.getY() - myPint3.getY(),2)+Math.*pow*(myPint2.getZ() - myPint3.getZ(),2));  
 double CA=Math.*sqrt*(Math.*pow*(myPint3.getX() - myPint1.getX(),2)+Math.*pow*(myPint3.getY() - myPint1.getY(),2)+Math.*pow*(myPint3.getZ() - myPint1.getZ(),2));  
 double p=(AB+BC+CA)/2;  
 double s=Math.*sqrt*(p\*(p-AB)\*(p-BC)\*(p-CA));  
 return s;  
 }  
}

## 3.3 Результат выполнения программы

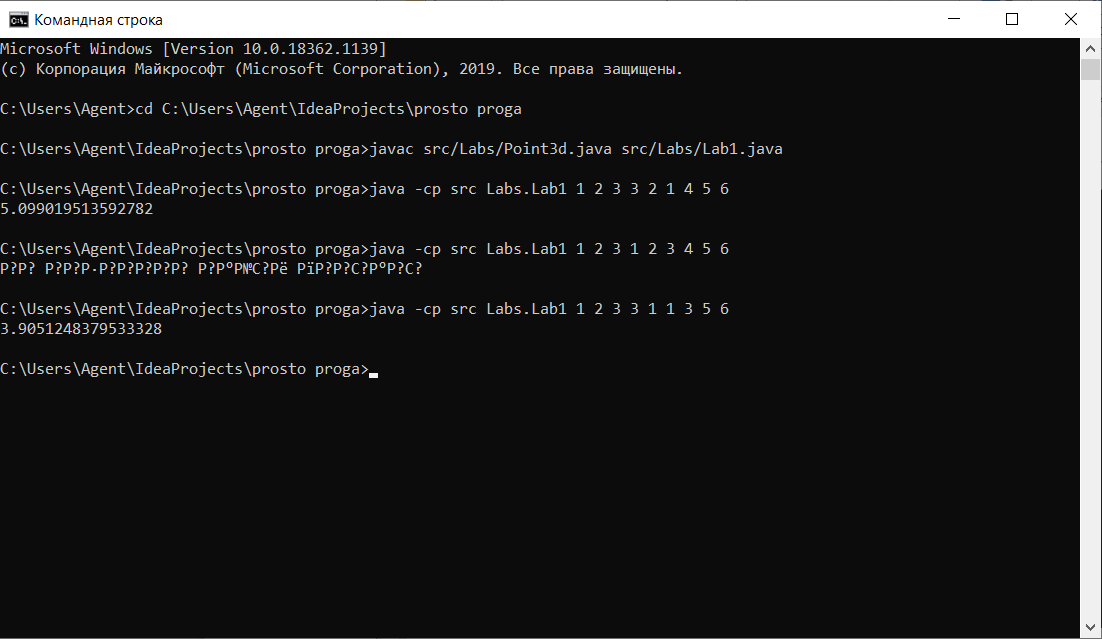


Рисунок 1 – результат выполнения

# Список использованных источников

1) ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления

2) ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления