**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра Информатики

****

**Отчет по лабораторной работе № 2**

по дисциплине «КТП»

на тему:

«**Основы объектно-ориентированного программирования»**

Выполнила: студентка группы БВТ1802

Лаврухина Елена Павловна

Руководитель:

Полянцева Ксения Андреевна

Москва 2020

Цель работы

Изучить основы синтаксиса Java по использованию объектов. В данной лабораторной работе необходимо использовать классы по одному на файл, чтобы описать, как эти объекты работают.

Выполнение

Код программы

1. Point2d

**public class** Point2d  
{  
 **private double xCoord**; *// Координата X* **private double yCoord**; *// Координата Y* **public** Point2d(**double** x, **double** y) *// Конструктор инициализации* {  
 **xCoord** = x;  
 **yCoord** = y;  
 }  
 **public** Point2d()  
 {  
 **this**(0,0);  
 }  
 **public double** getX() *// Возвращение координаты X* {  
 **return xCoord**;  
 }  
 **public double** getY() *// Возвращение координаты Y* {  
 **return yCoord**;  
 }  
 **public void** setX(**double** val) *// Установка значения координаты X* {  
 **xCoord** = val;  
 }  
 **public void** setY(**double** val) *// Установка значения координаты Y* {  
 **yCoord** = val;  
 }  
 **public static boolean** compare(Point2d a, Point2d b) *// Сравнение значений точек* {  
 **if** ((a.getX() == b.getX()) && (a.getY() == b.getY()))  
 **return true**;  
 **else  
 return false**;  
 }  
}

1. Point3d

**import** java.awt.\*;  
**import** java.util.Scanner;  
  
**public class** Point3d **extends** Point2d {  
  
 **static double** sqr (**double** a ) { *// Метод для возведения числа в квадрат* **double** result = a\*a;  
 **return** result;  
 }  
 **public static void** main(String[] args)  
  
 {  
 Scanner in = **new** Scanner(System.***in***);  
 System.***out***.print(**"Введите по очереди координаты для первой точки: "** + **"\n"**);  
 **double** a = in.nextInt();  
 **double** b = in.nextInt();  
 **double** c = in.nextInt();  
 System.***out***.print(**"Введите по очереди координаты для второй точки: "** + **"\n"**);  
 **double** a1 = in.nextInt();  
 **double** b1 = in.nextInt();  
 **double** c1 = in.nextInt();  
 Point3d Point = **new** Point3d(a,b,c);  
 Point3d Point1 = **new** Point3d(a1,b1,c1);  
  
 Point.setX(a);  
 Point1.setX(a1);  
  
 Point.setY(b);  
 Point1.setY(b1);  
  
 in.close(); *// Останавливаем работу Scanner* **if**(*compare*(Point, Point1)) *// Сравниваем координаты точек* System.***out***.println(**"Точки имеют разные координаты."**);  
System.***out***.println(**"Расстояние между точками: "**); *// Рассчитываем расстояние между точками* System.***out***.print(*distanceTo*(Point, Point1));

}  
 }  
  
 **private double zCoord**; *// Координата Z* **public** Point3d(**double** x, **double** y, **double** z) *// Конструктор инициализации* {  
 *// Координата X  
 // Координата Y* **zCoord** = z;  
 }  
 **public** Point3d()  
 {  
 **this**(0, 0, 0);  
 }  
  
 **public double** getZ() *// Возвращение координаты Z* {  
 **return zCoord**;  
 }  
 **public void** setZ(**double** val) *// Установка значения координаты Z* {  
 **zCoord** = val;  
 }  
 **public static boolean** compare(Point3d a, Point3d b) *// Сравнение значений точек* {  
 **return** (a.getX() == b.getX()) && (a.getY() == b.getY()) && (a.getZ() == b.getZ());  
 }  
 **public static double** distanceTo(Point3d a, Point3d b) *// Нахождение расстояния между точками* {  
 **double** Round = Math.*sqrt*(*sqr*((b.getX()-a.getX())) + *sqr*((b.getY()-a.getY())) + *sqr*((b.getZ()-a.getZ())));  
 Round \*= 100;  
 Round = (**int**) Round;  
 **return** (Round / 100);  
 }  
}

1. Lab1

**import** java.text.DecimalFormat;  
**import** java.util.Scanner;  
**public class** lab1 **extends** Point3d  
{  
 **public static void** main(String[] args)  
 {  
 **int** check = 0;  
  
  
 Scanner in = **new** Scanner(System.***in***);  
  
 System.***out***.print(**"Введите по очереди координаты для первой точки: "** + **"\n"**);  
 **double** a = in.nextInt();  
 **double** b = in.nextInt();  
 **double** c = in.nextInt();  
 System.***out***.print(**"Введите по очереди координаты для второй точки: "** + **"\n"**);  
 **double** a1 = in.nextInt();  
 **double** b1 = in.nextInt();  
 **double** c1 = in.nextInt();  
 System.***out***.print(**"Введите по очереди координаты для третьей точки: "** + **"\n"**);  
 **double** a2 = in.nextInt();  
 **double** b2 = in.nextInt();  
 **double** c2 = in.nextInt();  
  
 Point3d Point = **new** Point3d(a,b,c);  
 Point3d Point1 = **new** Point3d(a1,b1,c1);  
 Point3d Point2 = **new** Point3d(a2,b2,c2);  
 in.close();  
  
 Point.setX(a);  
 Point1.setX(a1);  
 Point2.setX(a2);  
  
 Point.setY(b);  
 Point1.setY(b1);  
 Point2.setY(b2);  
  
 Point.setZ(c);  
 Point1.setZ(c1);  
 Point2.setZ(c2);  
  
 **if** (*compare*(Point, Point2)||*compare*(Point2, Point1)||*compare*(Point, Point1)||!*Exist*(Point, Point1, Point2)) *// Проверка координат точек* {  
 System.***out***.println(**"Треугольник не существует."**);  
 check ++;  
  
 }  
  
 **if** (check == 0)  
 {  
 DecimalFormat df = **new** DecimalFormat(**"###.##"**);  
 System.***out***.println(**"Площадь треугольника: "**);  
 System.***out***.print(df.format(*Area*(Point, Point1, Point2)));  
 }  
 }  
  
 **public static double** Area(Point3d a, Point3d b, Point3d c) *// Нахождение площади* {  
 **double** Side1 = *distanceTo*(a, b), Side2 = *distanceTo*(a, c), Side3 = *distanceTo*(c, b);  
 **double** per = (Side1 + Side2 + Side3) / 2;  
 **return** Math.*sqrt*(per \* (per - Side1) \* (per - Side2) \* (per - Side3));  
 }  
  
 **public static boolean** Exist(Point3d a, Point3d b, Point3d c){  
 **if** (*distanceTo*(a, b)>= *distanceTo*(a, c) + *distanceTo*(b, c) ){  
 **return false**;  
 }  
 **else if** (*distanceTo*(a, c)>= *distanceTo*(a, b) + *distanceTo*(b, c)){  
 **return false**;  
 }  
 **else return** !(*distanceTo*(b, c) >= *distanceTo*(a, c) + *distanceTo*(a, b));  
 }  
}

Скриншоты работы программы

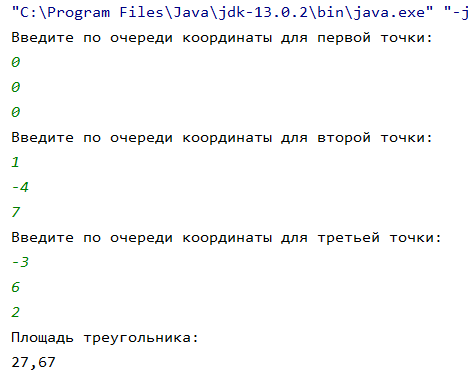


Рисунок 1 – Работа программы

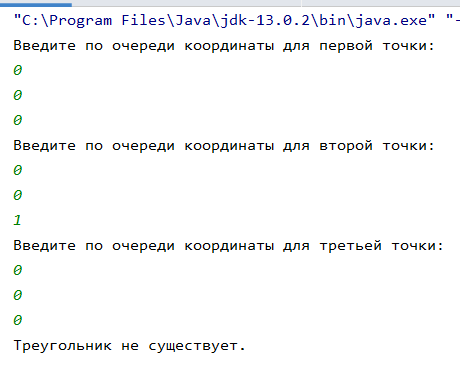


Рисунок 2 – Работа программы

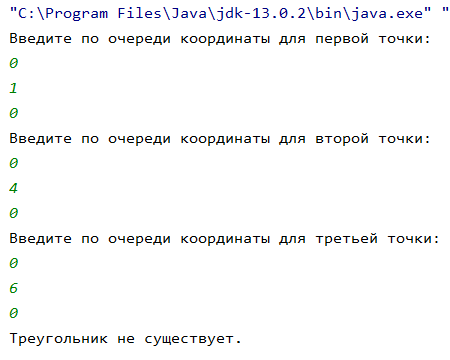


Рисунок 3 – Работа программы

Вывод

В ходе работы были изучены основы синтаксиса Java по использованию объектов. Мы научились использовать и наследовать классы, а, также решать с их помощью простые задачи программирования.