ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени государственное образовательное учреждение высшего образования

Московский технический университет связи и информатики

Кафедра информатики

**ЛАБОРАТОРНАЯ РАБОТА № 2**

**Основы объектно-ориентированного программирования**

Выполнил студент

группы БСТ1801

КОЛИ.Ж. П.

Москва-2020

public class Lab2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double x;

double y;

double z;

System.out.println("Введите первую точку:");

System.out.print("x: ");

x = scanner.nextDouble();

System.out.print("y: ");

y = scanner.nextDouble();

System.out.print("z: ");

z = scanner.nextDouble();

Point3d point1 = new Point3d(x, y, z);

System.out.println("Введите вторую точку:");

System.out.print("x: ");

x = scanner.nextDouble();

System.out.print("y: ");

y = scanner.nextDouble();

System.out.print("z: ");

z = scanner.nextDouble();

Point3d point2 = new Point3d(x, y, z);

System.out.println("Введите третью точку:");

System.out.print("x: ");

x = scanner.nextDouble();

System.out.print("y: ");

y = scanner.nextDouble();

System.out.print("z: ");

z = scanner.nextDouble();

Point3d point3 = new Point3d(x, y, z);

if (point1.equals(point2) || point1.equals(point3) || point2.equals(point3)) {

System.out.println("Точки совпадают");

} else {

System.out.println("Площадь треугольника: " + computeArea(point1, point2, point3));

}

}

public static double computeArea(Point3d point1, Point3d point2, Point3d point3) {

double a = point1.distanceTo(point2);

double b = point2.distanceTo(point3);

double c = point3.distanceTo(point1);

double p = (a + b + c) / 2;

System.out.println(p);

return Math.sqrt(Math.abs(p \* (p - a) \* (p - b) \* (p - c)));

}

}

public class Point3d {

private double xCoord;

private double yCoord;

private double zCoord;

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

xCoord = x;

yCoord = y;

zCoord = z;

}

public Point3d() {

this(0, 0 ,0);

}

public double getX() {

return xCoord;

}

public void setX(double x) {

xCoord = x;

}

public double getY() {

return yCoord;

}

public void setY(double y) {

yCoord = y;

}

public double getZ() {

return zCoord;

}

public void setZ(double z) {

zCoord = z;

}

@Override

public boolean equals(Object obj) {

if (obj instanceof Point3d) {

Point3d point = (Point3d) obj;

return Double.compare(xCoord, point.xCoord) == 0 && Double.compare(yCoord, point.yCoord) == 0 &&

Double.compare(zCoord, point.zCoord) == 0;

}

return false;

}

public double distanceTo(Point3d point) {

System.out.println(Math.round(Math.sqrt(Math.pow(point.xCoord - xCoord, 2) + Math.pow(point.yCoord - yCoord, 2) +Math.pow(point.zCoord - zCoord, 2)) \* 100) / 100.0);

return Math.round(Math.sqrt(Math.pow(point.xCoord - xCoord, 2) + Math.pow(point.yCoord - yCoord, 2) +Math.pow(point.zCoord - zCoord, 2)) \* 100) / 100.0;

}

}

