Лабораторная работа №2

по дисциплине

“Объектно-ориентированное программирование”

Тема:

«Основы объектно-ориентированного программирования»

Выполнила студентка

группы БФИ1901

Киселева Анна

Москва 2020

**Цель работы:**

Необходимо научиться работать с классами программы, создавать экземпляры класса и сравнивать их друг с другом.

**Задачи:**

**1.** Создайте новый класс Point3d для представления точек в трехмерном Евклидовом 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.

Нельзя предоставлять непосредственный доступ к внутренним элементам объекта класса Point3d.

1. **Файл Lab1.java:**

package com.company;

import java.util.Scanner;

public class Lab1 {

public static void main(String[] args){

Scanner in = new Scanner(System.in);

System.out.println("Input Point");

Point3d A = new Point3d(in.nextDouble(), in.nextDouble(), in.nextDouble());

System.out.println("Input Point");

Point3d B = new Point3d(in.nextDouble(), in.nextDouble(), in.nextDouble());

System.out.println("Input Point");

Point3d C = new Point3d(in.nextDouble(), in.nextDouble(), in.nextDouble());

if (A.equals(B)||B.equals(C)||C.equals(A))

System.out.println("Ошибка: Есть равные точки");

else

System.out.println("Площадь треугольника = " + computeArea(A, B, C));

}

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

double a = Math.sqrt(Math.pow(first.getX() - second.getX() , 2) + Math.pow(first.getY() - second.getY(), 2) + Math.pow(first.getZ() - second.getZ(), 2));

double b = Math.sqrt(Math.pow(second.getX() - third.getX() , 2) + Math.pow(second.getY() - third.getY(), 2) + Math.pow(second.getZ() - third.getZ(), 2));

double c = Math.sqrt(Math.pow(first.getX() - third.getX() , 2) + Math.pow(first.getY() - third.getY(), 2) + Math.pow(first.getZ() - third.getZ(), 2));

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

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

}

}

1. 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.
2. **Файл Point2d.java:**

package com.company;

public class Point2d {

private double xCoord;

private double yCoord;

public Point2d ( double x, double y) {

xCoord = x;

yCoord = y;

}

public Point2d () {

//Вызовите конструктор с двумя параметрами и определите источник.

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 boolean equals(Point2d another){

return xCoord==another.yCoord && yCoord==another.yCoord;

}

}

**3.** 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. Перед вызовом метода computeArea проверьте на равенство значений всех трех объектов Point3d. Если одна из точек равна другой, то выведите соответствующее сообщение пользователю и не вычисляйте площадь.

1. **Файл Point3d.java:**

package com.company;

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 double getY () {

return yCoord;

}

public double getZ () {

return zCoord;

}

public void setX ( double val) {

xCoord = val;

}

public void setY ( double val) {

yCoord = val;

}

public void setZ ( double val) {

zCoord = val;

}

public boolean equals(Point3d another){

return xCoord==another.xCoord && yCoord==another.yCoord && zCoord==another.zCoord;

}

public double distanceTo(Point3d another) {

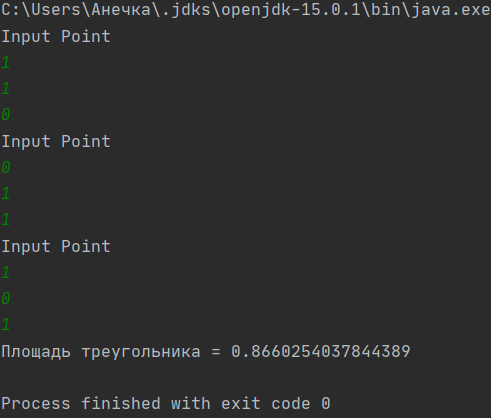
return Math.sqrt(Math.pow(xCoord - another.xCoord, 2) + Math.pow(yCoord - another.yCoord, 2) + Math.pow(zCoord - another.zCoord, 2));

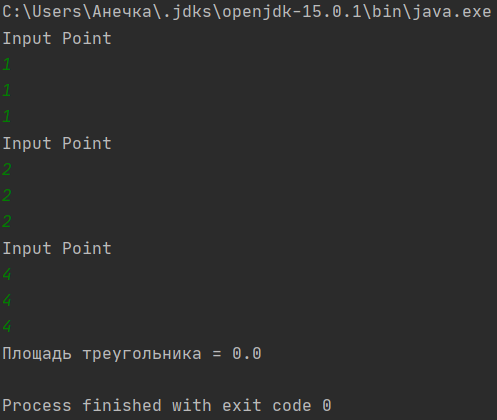
}

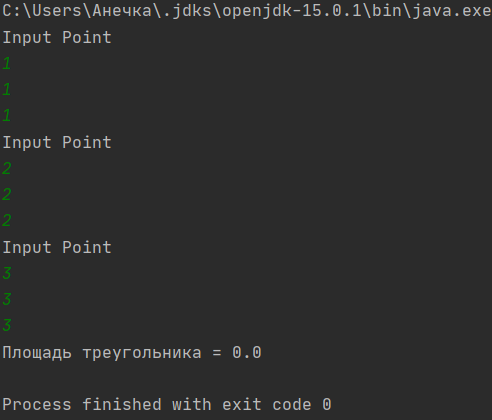
}

**4.** Скомпилируйте оба исходных файла вместе: javac Point3d.java Lab1.java и затем запустите программу Lab1, тестируя ее с несколькими образцами треугольников.

**Скриншоты работы программы:**

****

****

****

**Вывод:** Мы научились работать с классами программы, создавать экземпляры класса и сравнивать их друг с другом.