Лабораторная работа №2

по дисциплине

“Объектно-ориентированное программирование”

Выполнил студент

группы БФИ1901

Гасанов Гамзат

Москва 2020

**Цель работы:** Java позволяет использовать объекты. В данной лабораторной работе необходимо использовать классы по одному на файл, чтобы описать, как эти объекты работают.

**Задание:**

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 .

2. 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. 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.

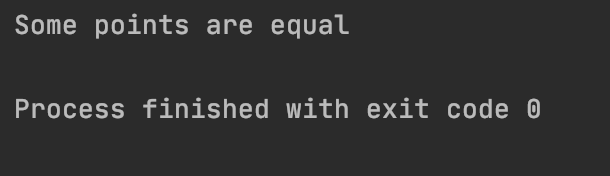
4. Скомпилируйте оба исходных файла вместе: javac Point3d.java Lab1.java и затем запустите программу Lab1, тестируя ее с несколькими образцами треугольников.

**Результат выполнения**

// двумерный класс точки.  
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() { // Конструктор по умолчанию.  
 xCoord = 0;  
 yCoord = 0;  
 }  
  
 public double getxAxis() { // Возвращение координаты X  
 return xCoord;  
 }  
  
 public void setxAxis(double x) { // Установка значения координаты X  
 xCoord = x;  
 }  
  
 public double getyAxis() { // Возвращение координаты Y  
 return yCoord;  
 }  
  
 public void setyAxis(double y) { // Установка значения координаты Y  
 yCoord = y;  
 }  
  
 public boolean equals(Point2d point) {  
 return xCoord == point.xCoord && yCoord == point.yCoord;  
 }  
  
 public double distanceTo(Point2d point) { // Расстояние между точками (длина сторон)  
 return Math.*round*(Math.*sqrt*(Math.*pow*(point.xCoord - xCoord, 2) + Math.*pow*(point.yCoord - yCoord, 2)) \* 100.0) / 100.0;  
 }  
}

public class Point3d extends Point2d {  
 private double zCoord;  
  
 public Point3d(double x, double y, double z) { // Конструктор инициализации  
 super(x, y);  
 zCoord = z;  
 }  
  
 public Point3d() { // Конструктор по умолчанию.  
 super();  
 zCoord = 0;  
 }  
  
 public double getzAxis() { // Возвращение координаты Z  
 return zCoord;  
 }  
  
 public void setzAxis(double z) { // Установка значения координаты Z  
 zCoord = z;  
 }  
  
 public boolean equals(Point3d point) {  
 return getxAxis() == point.getxAxis() && getyAxis() == point.getyAxis() && zCoord == point.zCoord;  
 }  
  
 public double distanceTo(Point3d point) { // Расстояние между точками (длина сторон)  
 return Math.*round*(Math.*sqrt*(Math.*pow*(point.getxAxis() - getxAxis(), 2) + Math.*pow*(point.getyAxis() - getyAxis(), 2) + Math.*pow*(point.zCoord - zCoord, 2)) \* 100.0) / 100.0;  
 }  
}

public class Lab2 {  
 public static double computeArea(Point3d a, Point3d b, Point3d c) throws Exception {  
 if (a.equals(b) || b.equals(c) || c.equals(a)) {  
 throw new Exception("Some points are equal");  
 }  
  
 double sideA = a.distanceTo(b);  
 double sideB = b.distanceTo(c);  
 double sideC = c.distanceTo(a);  
 double semiPerimeter = (sideA + sideB + sideC) / 2;  
  
 return Math.*round*(Math.*sqrt*(semiPerimeter \* (semiPerimeter - sideA) \* (semiPerimeter - sideB) \* (semiPerimeter - sideC)) \* 100.0) / 100.0;  
 }  
  
 public static void main(String[] args) {  
 Point3d a = new Point3d(3, 6, 22);  
 Point3d b = new Point3d(5, 5, 0);  
 Point3d c = new Point3d(16, 1, 1);  
  
 try {  
 System.*out*.println(*computeArea*(a, b, c));  
 }  
 catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

****

