ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное

Бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра информатики

Лабораторная работа №2

По дисциплине: «Технология разработки программного обеспечения»

Тема: «Основы объектно-ориентированногопрограммирования»

Выполнил студент:

Группы: БФИ1902

Шацкий Е. И.

Проверила:

Мосева. М.С.

Москва, 2020 г.

**Содержание**

[1 Задачи на лабораторную работу. 3](#_Toc59353941)

[2 Ход решения лабораторной работы. 3](#_Toc59353942)

[Список используемых источников 9](#_Toc59353943)

# **1 Задачи на лабораторную работу.**

Цель: научиться использовать классы по одному на файл, чтобы описать, как эти объекты работают.

Для выполнения лабораторной работы необходимо:

а) Создать новый класс Point3d для представления точек в трехмерном Евклидовом пространстве.

б) Создать новый объект Point3d с тремя значениями с плавающей точкой (double).

в) Создать новый объект Point3d со значениями (0.0, 0.0, 0.0) по умолчанию.

г) Создать метод для сравнения значений двух объектов Point3d.

д) Создать 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.

е) Создать другой класс под названием Laba2, 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).

ж) Создать 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.

# **2 Ход решения лабораторной работы.**

После написания алгоритма работы программы необходимо скомпилировать исходные файлы с помощью команд javac Point3d.java и javac Laba2.java. Затем запустим программу с помощью команды java Laba2. Этот процесс показан на рисунке 1.

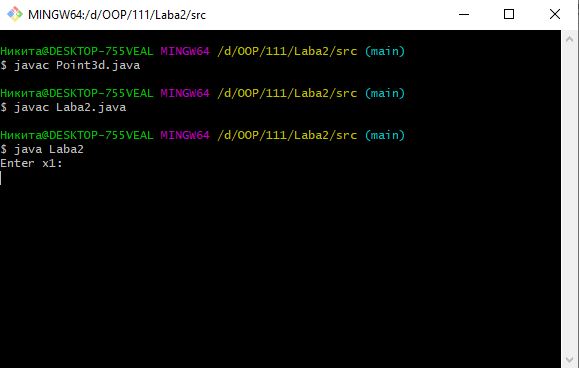


Рисунок 1 – Компиляция файлов и запуск программы.

Далее вводим произвольные данные, которые будут определять местоположение точек в пространстве. По условиям лабораторной работы программа должна выводить площадь треугольника, который получится в результате работы программы. На рисунке 2 представлен результат работы программы.

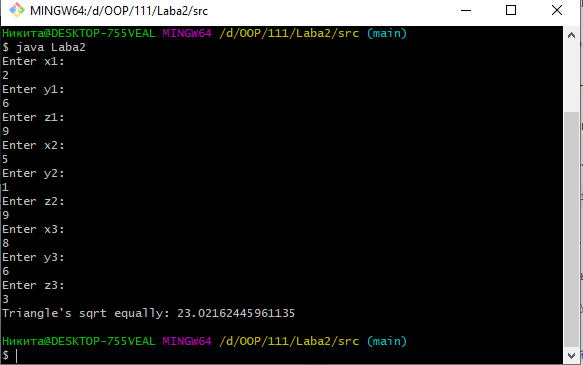


Рисунок 2 – Результат работы программы.

На листинге 1 представлен код файла Laba2.java. На листинге 2 представлен код Point2d.java. На листинге 3 представлен код Point3d.java

Листинг 1 - Код файла Laba2.java

import java.util.Scanner;  
  
public class Laba2 {  
 public static void main(String[] args) {  
 String readstr;  
  
 Scanner reader = new Scanner(System.*in*);  
  
 System.*out*.println("Enter x1:");  
 readstr = reader.nextLine();  
 double x1 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter y1:");  
 readstr = reader.nextLine();  
 double y1 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter z1:");  
 readstr = reader.nextLine();  
 double z1 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter x2:");  
 readstr = reader.nextLine();  
 double x2 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter y2:");  
 readstr = reader.nextLine();  
 double y2 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter z2:");  
 readstr = reader.nextLine();  
 double z2 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter x3:");  
 readstr = reader.nextLine();  
 double x3 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter y3:");  
 readstr = reader.nextLine();  
 double y3 = Double.*parseDouble*(readstr);  
  
 System.*out*.println("Enter z3:");  
 readstr = reader.nextLine();  
 double z3 = Double.*parseDouble*(readstr);  
  
 reader.close();  
  
 Point3d coordinate1 = new Point3d(x1, y1, z1);  
 Point3d coordinate2 = new Point3d(x2, y2, z2);  
 Point3d coordinate3 = new Point3d(x3, y3, z3);  
  
 if (!coordinate1.CompareCoord3d(coordinate2) && !coordinate2.CompareCoord3d(coordinate3) && !coordinate3.CompareCoord3d(coordinate1)) {  
 System.*out*.println("Triangle's sqrt equally: " + *computeArea*(coordinate1, coordinate2, coordinate3));  
 } else {  
 System.*out*.println("Error! Please check your coordinates!");  
 }  
  
  
 }  
  
 public static Double computeArea(Point3d object1, Point3d object2, Point3d object3) {  
 double a = object1.distanceTo(object2);  
 double b = object2.distanceTo(object3);  
 double c = object3.distanceTo(object1);  
 double p = (a + b + c)/2;  
 return Math.*sqrt*(p\*(p-a)\*(p-b)\*(p-c));  
 }  
  
  
}

Листинг 2 - Код файла Point2d.java

public class Point2d {  
 public double xCoord;  
 public 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 boolean CompareCoord(Point2d object){  
 return this.xCoord == object.xCoord && this.yCoord == object.yCoord;  
 }  
}

Листинг 3 - Код файла Point3d.java

public class Point3d extends Point2d{  
 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 getZ(){  
 return zCoord;  
 }  
  
 public void SetZ(double val){  
 zCoord = val;  
 }  
  
 public boolean CompareCoord3d(Point3d object){  
 return this.xCoord == object.xCoord && this.yCoord == object.yCoord && this.zCoord == object.zCoord;  
 }  
  
 public double distanceTo (Point3d object) {  
 return Math.*floor*(Math.*sqrt*(Math.*pow*(object.xCoord - this.xCoord, 2) + Math.*pow*(object.yCoord - this.yCoord, 2) + Math.*pow*(object.zCoord - this.zCoord, 2)) \* 100)/100;  
 }  
}

Вывод: в результате лабораторной работы были изучены принципы использования классов и объектов в Java.

# **Список используемых 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.