Министерство цифрового развития, связи и

Массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Математической кибернетики и информационных технологий»

Отчет по лабораторной работе №2

по дисциплине «Кроссплатформенные технологии программирования»

Выполнил: студент группы БСТ2001

Багдасарян Артём

Москва 2022

Оглавление

[**1.** **Цель и задание** 3](#_Toc96978581)

[**2.** **Ход выполнения лабораторной работы** 3](#_Toc96978582)

[**2.1** **Код программы** 3](#_Toc96978583)

[**2.2** **Результат работы программы** 5](#_Toc96978584)

[**3.** **Вывод** 6](#_Toc96978585)

[**Список использованных источников** 7](#_Toc96978586)

1. **Цель и задание**

В данной лабораторной работе мы изучим основы объектно-ориентированного программирования. Как использовать объекты, которые находятся в другом классе по одному на файл, чтобы описать, как эти объекты работают.

1. **Ход выполнения лабораторной работы** 
   1. **Код программы**

Ниже представлен код Lab2.java

package com.company;  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
  
public class Lab2 {  
 public static void main(String[] args) throws IOException {  
 BufferedReader br = new BufferedReader(new InputStreamReader(System.*in*));  
 System.*out*.println("Enter the first point: (x y z)");  
 String input = br.readLine();  
 String[] str = input.trim().split(" ");  
 Point3d firstPoint = new Point3d(Double.*parseDouble*(str[0]), Double.*parseDouble*(str[1]), Double.*parseDouble*(str[2]));  
 System.*out*.println("Enter the second point: (x y z)");  
 input = br.readLine();  
 str = input.trim().split(" ");  
 Point3d secondPoint = new Point3d(Double.*parseDouble*(str[0]), Double.*parseDouble*(str[1]), Double.*parseDouble*(str[2]));  
 System.*out*.println("Enter the third point: (x y z)");  
 input = br.readLine();  
 str = input.trim().split(" ");  
 Point3d thirdPoint = new Point3d(Double.*parseDouble*(str[0]), Double.*parseDouble*(str[1]), Double.*parseDouble*(str[2]));  
 if ((firstPoint.compareCoord(secondPoint) == true) || (firstPoint.compareCoord(thirdPoint)==true)||(secondPoint.compareCoord(thirdPoint)))  
 System.*out*.println("Одна из точек равна другой, невозможно посчитать площадь");  
 else {  
 double Square = *computeArea*(firstPoint, secondPoint, thirdPoint);  
 System.*out*.println("Площадь равна: " + Square);  
 }  
 }  
 public static double computeArea(Point3d firstPoint, Point3d secondPoint, Point3d thirdPoint){  
 double a = firstPoint.distanceTo(secondPoint);  
 double b = secondPoint.distanceTo(thirdPoint);  
 double c = firstPoint.distanceTo(thirdPoint);  
 double p = 0.5\*(a+b+c);  
 double s = Math.*sqrt*(p\*(p-a)\*(p-b)\*(p-c));  
 double scale = Math.*pow*(10, 2);  
 s = Math.*ceil*(s \* scale) / scale;  
 return s;  
 }  
  
}

Ниже представлен код Point3d.java

package com.company;  
public class Point3d {  
 private double xCoord; // Координата X  
 private double yCoord; // Координата Y  
 private double zCoord; // Координата Z  
 */\*\* Конструктор инициализации \*\*/* public Point3d ( double x, double y, double z) {  
 xCoord = x;  
 yCoord = y;  
 zCoord = z;}  
 */\*\* Конструктор по умолчанию. \*\*/* public Point3d () {  
 this(0.0, 0.0, 0.0);}  
 */\*\* Возвращение координаты X \*\*/* public double getX () {  
 return xCoord;}  
 */\*\* Возвращение координаты Y \*\*/* public double getY () {  
 return yCoord;  
 }  
 */\*\* Возвращение координаты Z \*\*/* public double getZ () {  
 return zCoord;  
 }  
 */\*\* Установка значения координаты X. \*\*/* public void setX ( double val) {  
 xCoord = val;  
 }  
 */\*\* Установка значения координаты Y. \*\*/* public void setY ( double val) {  
 yCoord = val;  
 }  
 */\*\* Установка значения координаты Z. \*\*/* public void setZ ( double val) {  
 zCoord = val;  
 }  
 public double vectorLength(double x, double y, double z) {  
 double num = Math.*sqrt*(Math.*pow*(x,2)+Math.*pow*(y,2)+Math.*pow*(z,2));  
 return num;  
 }  
 public boolean Compare (Point3d a){  
 double num1 = vectorLength(xCoord, yCoord, zCoord);  
 double num2 = vectorLength(a.xCoord, a.yCoord, a.zCoord);  
 if (num1 == num2) return true;  
 else return false;  
 }  
 /\* Метод для сравнения координат точек \*/  
 public boolean compareCoord(Point3d a){  
 if ((a.xCoord==xCoord)&&(a.yCoord==yCoord)&&(a.zCoord==zCoord))  
 return true;  
 else return false;  
 }  
 /\* Вычисление расстояния между двумя точками\*/  
 public double distanceTo(Point3d a){  
 double distance = Math.*sqrt*(Math.*pow*((a.xCoord-xCoord),2)+Math.*pow*((a.yCoord-yCoord),2)+Math.*pow*((a.zCoord-zCoord),2));  
 double scale = Math.*pow*(10, 2);  
 distance = Math.*ceil*(distance \* scale) / scale;  
 return distance;  
 }  
}

Ниже представлен код Point2d.java

package com.company;  
  
public class Point2d {  
 public static void main(String[] args) {  
 Point2d firstPoint = new Point2d(19,5);  
 double x = firstPoint.getX();  
 double y = firstPoint.getY();  
 System.*out*.printf("x=%.0f; y=%.0f \n", x, y);  
  
  
 }  
 /\* координата X \*/  
  
 private double xCoord;  
 */\*\* координата Y \*\*/* private double yCoord;  
 */\*\* Конструктор инициализации \*\*/* public Point2d ( double x, double y) { // метод  
 xCoord = x;  
 yCoord = y;  
 }  
 */\*\* Конструктор по умолчанию. \*\*/* public Point2d () { // метод  
//Вызовите конструктор с двумя параметрами и определите источник.  
 this(0, 0);  
 }  
 */\*\* Возвращение координаты X \*\*/* public double getX () { // функция  
 return xCoord;  
 }  
 */\*\* Возвращение координаты Y \*\*/* public double getY () { // функция  
 return yCoord;  
 }  
 */\*\* Установка значения координаты X. \*\*/* public void setX ( double val) { // функция  
 xCoord = val;  
 }  
 */\*\* Установка значения координаты Y. \*\*/* public void setY ( double val) { // функция  
 yCoord = val;  
 }  
}

* 1. **Результат работы программы**

Результат работы программы показан на рисунке 1

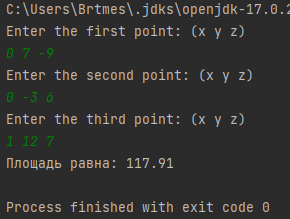


Рисунок 1 – Результат работы программы

* 1. **Программа на Git hub**

Ссылка на репозиторий на Git hub: https://github.com/Artem2406/KTP

1. **Вывод**

В данной лабораторной работе мы изучили основы объектно-ориентированного программирования. Научились использовать объекты, которые находятся в другом классе по одному на файл, для описания работы этих объектов.

# **Список использованных источников**

1. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа, 2006.

2. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004.