Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного знамени

Федеральное государственное бюджетное учреждение

высшего образования

«Московский технический университет связи и информатики»

Кафедра МКиТ

Лабораторная работа № 2

«Основы объектно-ориентированного программирования»

Выполнил студент

Группы БСТ1903

Самоваров А.П.

Вариант № 12

Москва 2021

# **Цель работы**

Целью лабораторной работы является изучение классов и объектов Java.

# **Задание на лабораторную работу**

В данной лабораторной работе необходимо использовать классы по одному на файл, чтобы описать, как эти объекты работают.

# **Код программы**

Код для класса Point2D:

public class Point2d { //двумерный класс точки.

private double xCoord; /\*\* координата X \*\*/

private double yCoord; /\*\* координата Y \*\*/

public Point2d ( double x, double y) { //Конструктор инициализации

xCoord = x;

yCoord = y;

}

public Point2d () { //дефолтный конструктор

this(0, 0); //Вызовите конструктор с двумя параметрами и определите источник.

}

public double getX () { //Возвращение координаты X

return xCoord;

}

public double getY () { //Возвращение координаты Y

return yCoord;

}

public void setX ( double val) { //Установка значения координаты X

xCoord = val;

}

public void setY ( double val) { //Установка значения координаты Y

yCoord = val;

}

}

Код для класса Point3D:

public class Point3d extends Point2d {

private double CrdZ; // Объявление локальной переменной, координата Z

Point3d(double x, double y, double z) // Конструктор инициализации

{

super(x, y);

CrdZ = z; // Присвоение переменной значения z

}

Point3d() //конструктор по умочанию

{

this(0.0, 0.0, 0.0); // Конструктор с тремя параметрами и определение источника.

}

public double getZ()

{

return CrdZ; // Возвращение координаты Z

}

public boolean comparison(Point3d o) // Метод для сравнения двух объектов

{

return ((this.getX() == o.getX()) && (this.getY() == o.getY()) && (this.CrdZ == o.CrdZ)) ? true : false;

}

public double distanceTo(Point3d o)

{

// Вычисление расстояния между точками

return Math.ceil(Math.sqrt(Math.pow((o.getX()-this.getX()), 2)+Math.pow((o.getY()-this.getY()), 2)+Math.pow((o.CrdZ-this.CrdZ), 2))\*100)/100;

}

public void setZ(double value)

{

this.CrdZ = value;

}

}

Код для класса Lab1:

public class lab1 {

public static void main (String[] args)

{

Point3d[] myPoint = new Point3d[3]; //Объявление массива из трёх объектов-точек

for ( int i = 0 ; i <= 2; i++) //Цикл заполнения координат точек

{

myPoint[i] = new Point3d();

myPoint[i].setX(Double.valueOf(args[0+3\*i]));

myPoint[i].setY(Double.valueOf(args[1+3\*i]));

myPoint[i].setZ(Double.valueOf(args[2+3\*i]));

}

for (int i = 0; i <= 2; i++) // Вывод точек

{

System.out.println("Точка " + i + ": (" + myPoint[i].getX() + " " + myPoint[i].getY() + " " + myPoint[i].getZ() + ")");

}

if (checkPoints(myPoint[0], myPoint[1], myPoint[2])) //Проверочка на совпадение (равность) точек

System.out.printf("Площадь треугольника: " + "%.2f", computeArea(myPoint[0], myPoint[1], myPoint[2]));

else

System.out.println("Точки равны, введите правильные координаты точек");

}

public static double computeArea (Point3d p1, Point3d p2, Point3d p3) //Вычисление площади треугольника

{

double a = p1.distanceTo(p2); //Вычисление стороны a

double b = p2.distanceTo(p3); //Вычисление стороны b

double c = p3.distanceTo(p1); //Вычисление стороны c

double per = ((a+b+c)/2); //Вычисление периметра

return (Math.sqrt(per\*(per-a)\*(per-b)\*(per-c))); //Возвращаемое значение - площадь треугольника по формуле Герона

}

public static boolean checkPoints(Point3d p1, Point3d p2, Point3d p3) // Проверка совпадения/равности точек

{

if (((p1.getX() == p2.getX()) && (p1.getY() == p2.getY()) && (p1.getZ() == p2.getZ())) ||

((p2.getX() == p3.getX()) && (p2.getY() == p3.getY()) && (p2.getZ() == p3.getZ())) ||

((p1.getX() == p3.getX()) && (p1.getY() == p3.getY()) && (p1.getZ() == p3.getZ())))

return false;

return true;

}

}

# **Результаты выполнения программы**

Компиляция программы:

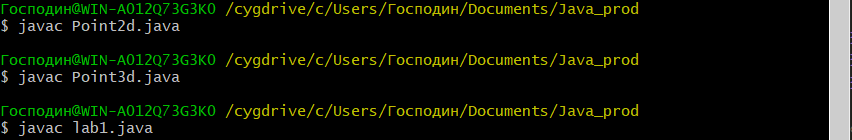


Рисунок 1 – компиляция программы

Расчет площади для точек (1.1.1) (2.2.2) (3.3.3.)

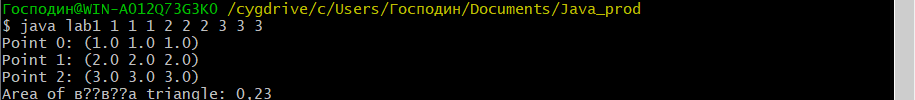


Рисунок 2 – расчет площади треугольника

# **Вывод:**

В ходе проделанной работы были изучены основы работы с объектами Java.