

### **Ордена Трудового красного знамени ФГБО УВО Московский технический университет связи и информатики**

**Заочный общетехнический факультет**

**Кафедра «Математической кибернетики и информационных технологий»**

**Дисциплина: Кроссплатформенные технологии программирования**

**Лабораторная работа №2**

«Основы объектно-ориентированного программирования»

Выполнил: Пименов А.А., студент группы БСТ1851

Проверил:

Москва, 2021

Цель работы

1. Создать новый класс Point3d для представления точек в трехмерном евклидовом пространстве. Необходимо реализовать:

* создание нового объекта Point3d с тремя значениями с плавающей точкой,
* 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
* метод distanceTo, который в качестве параметра принимает другой объект Point3d, вычисляет расстояние между двумя точками и возвращает полученное значение

1. Создать другой класс под названием Lab2, содержащий статический метод main. Этот класс должен иметь следующую функциональность:

* ввод координат трех точек, находящихся в трехмерном пространстве. Создание трех объектов Point3d на основании полученных данных.
* метод computeArea, который принимает три объекта Point3d, вычисляет площадь треугольника, образованного этими точками и возвращает полученное значение в формате double.
* расчет площади треугольника и вывод результата пользователю на основе полученных данных и реализованных алгоритмов.

Ход выполнения работы

Текст класса Point3d:

public class Point3d //трехмерный класс точки

{

private double xCoord; //координата X

private double yCoord; //координата Y

private double zCoord; //координата Z

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

{

xCoord = x;

yCoord = y;

zCoord = z;

}

public Point3d () //Конструктор по умолчанию

{

this(0, 0, 0);

}

public double distanceTo (Point3d p) //Метод, считающий расстояние между точками

{

double xx = this.xCoord - p.xCoord;

double yy = this.yCoord - p.yCoord;

double zz = this.zCoord - p.zCoord;

return (Math.sqrt(xx\*xx + yy\*yy + zz\*zz));

}

public double getX () //Возвращение координаты X

{

return xCoord;

}

public double getY () //Возвращение координаты Y

{

return yCoord;

}

public double getZ () //Возвращение координаты Z

{

return zCoord;

}

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

{

xCoord = val;

}

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

{

yCoord = val;

}

public void setZ (double val) //Установка значения координаты Z

{

zCoord = val;

}

}

Текст класса Lab2:

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++) //Просто вывод точек для красоты

{

System.out.println("Point " + 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("Area of a triangle: " + "%.2f", computeArea(myPoint[0], myPoint[1], myPoint[2]));

else

System.out.println("Some of points are equal. Please, enter correct input data");

}

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) //Проверочка на совпадение (равность) 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;

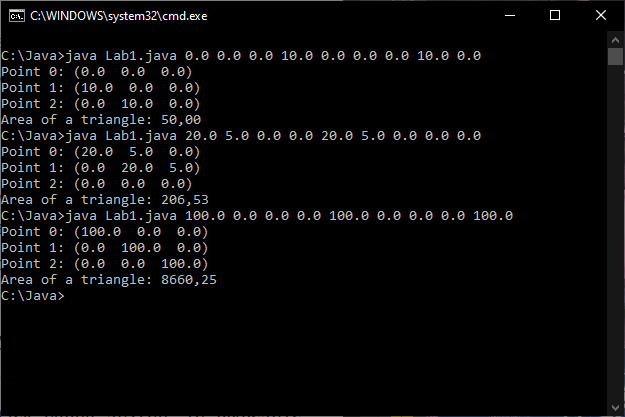
}

}

Скомпилируем эти два файла при помощи команды *javac Point3d.java Lab2.java*.

Запустим программу несколько раз при помощи команды *java Point3d.java Lab2.java.*

Результат выполнения программы:



Вывод

Изучены основы парадигмы объектно-ориентированного программирования.