ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский Технический Университет Связи и Информатики» (МТУСИ)

Кафедра МКиИТ

Лабораторная работа по технологиям программирования №1

«Java - Сразу к делу»

Выполнил:

Студент 2 курса

Группы БСТ-1602

Пащенко Алексей

Вариант №12

Москва, 2018

Оглавление

[1. Цель и задачи. 3](#_Toc507377840)

[2. Анализ предметной области и выбор инструментария. 3](#_Toc507377841)

[3. Код программы и объяснение функций. 3](#_Toc507377842)

[3.1. Файл Point3d.java. 3](#_Toc507377843)

[3.2. Файл Lab1.java. 4](#_Toc507377844)

[4. Заключение. 5](#_Toc507377845)

1. Цель и задачи.

Создать класс Point3d, для описания точки в трёхмерном Евклидовом пространстве.

Добавить метод distanceTo для вычисления расстояния между двумя точками.

Создать класс Lab1, который должен содержать основной метод.

Собрать оба исходных файла вместе.

1. Анализ предметной области и выбор инструментария.

В текущей работе я использовал бесплатно распространяемый пакет разработчика Java Development Kit. Однако, этот пакет не предусматривает интегрированную среду разработки, поэтому для написания непосредственно кода мною была выбрана программа Notepad++, так как она обеспечивает достаточно гибкий интерфейс и предоставляет возможность подсветки синтаксиса языка.

1. Код программы и объяснение функций.
   1. Файл Point3d.java.

// Класс точки в трёхмерном пространстве

public class Point3d {

// Переменные координат точки

private double xCoord;

private double yCoord;

private double zCoord;

// Конструктор, чтобы инициализировать точку (x,y,z)

public Point3d (double x, double y, double z){

xCoord=x;

yCoord=y;

zCoord=z;

}

// Конструктор по умолчанию

public Point3d (){

this (0,0,0);

}

// Методы, возвращающие координаты:

public double getX(){

return xCoord;

}

public double getY(){

return yCoord;

}

public double getZ(){

return zCoord;

}

// Методы, позволяющие ввести координаты:

public void SetX(double val){

xCoord=val;

}

public void SetY(double val){

yCoord=val;

}

public void SetZ(double val){

zCoord=val;

}

// Метод, проверяющий равенство двух точек

public boolean areEqual (Point3d point) {

if (xCoord==point.getX() && yCoord==point.getY() && zCoord==point.getZ()) return true;

return false;

}

// Метод, находящий расстояние между двумя точками

public double distanceTo(Point3d p){

return Math.sqrt(Math.pow(xCoord-p.getX(),2) + Math.pow(yCoord-p.getY(),2) + Math.pow(zCoord-p.getZ(),2));

}

}

Метод, проверяющий равенство двух точек берёт попарно координаты каждой оси у обеих точек и сравнивает их. Если все три координаты оказываются одновременно равны, то метод возвращает TRUE, иначе метод возвращает FALSE.

Метод distanceTo используя координаты точек находит расстояние между ними. Для этого он извлекает квадратный корень из суммы квадратов разностей координат точек и возвращает полученное число.

* 1. Файл Lab1.java.

// Импортируем класс Scanner

import java.util.Scanner;

// Класс, управляющий работой программы

public class Lab1{

// Функция получает координаты трёх точек и находит площадь треугольника

public static void main(String[] args){

int eqError=0;

// Создаём массив из трёх точек

Point3d[] p = new Point3d[]{new Point3d(),new Point3d(),new Point3d()};

// Вводим координаты точек

Scanner scanner = new Scanner(System.in);

for(int i=1; i<=3; i++)

{

System.out.println("Введите координаты "+i+" точки:");

p[i-1].SetX(Double.parseDouble(scanner.next()));

p[i-1].SetY(Double.parseDouble(scanner.next()));

p[i-1].SetZ(Double.parseDouble(scanner.next()));

}

// Проверяем точки на совпадение:

if (p[0].areEqual(p[1]) || p[0].areEqual(p[2]) || p[1].areEqual(p[2])) {

System.out.println("Введены точки с одинаковыми координатами!");

eqError++;

}

// Если точки не совпадают, то находим площадь треугольника

if (eqError<1) {

double area=computeArea(p[0],p[1],p[2]);

System.out.println("Площадь треугольника равна "+area);

}

}

// Метод находит площадь треугольника используя формулу Герона

public static double computeArea (Point3d p1, Point3d p2, Point3d p3){

double a=p1.distanceTo(p2);

double b=p1.distanceTo(p3);

double c=p2.distanceTo(p3);

double p=(a+b+c)/2;

return (Math.sqrt(p\*(p-a)\*(p-b)\*(p-c)));

}

}

Метод computeArea находит площадь треугольника, используя формулу Герона, то есть основываясь на длинах сторон. А длины сторон вычисляются как расстояния между точками, то есть, с помощью метода distanceTo.

Функция main получает координаты трёх точек при помощи класса Scanner. Затем, она проверяет, не совпадают ли какие-либо из этих точек попарным сравнением с помощью метода areEqual. Если никакие две точки не совпадают, то вычисляется площадь треугольника с помощью метода computeArea.

1. Заключение.

В ходе работы мною была написана программа, позволяющая вычислить площадь треугольника по координатам его вершин в трёхмерном Евклидовом пространстве.