ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ “МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ”.

Кафедра МКиИТ

Лабораторная работа №1 по дисциплине: ”Технологии программирования ”

“ Java - Сразу к делу ”.

Выполнила: студентка группы БСТ1602

Лушина Ольга

Проверил: М. Г. Городничев

Москва 2018

**Содержание:**

1. Цели и задачи. 3
2. Анализ предметной области и выбор инструментария. 3
3. Функции и их объяснение. 3
4. Выводы. 6
5. Цели 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 данной лабораторной работе я использовала бесплатный пакет Jdk, также стандартный редактор txt.

1. Файл Point3d.java.

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 берет другой Point3d в качестве параметра и вычисляет двойную точность приближения плавающий точкой расстояния по прямой между двумя точками и возвращает это значение.

/\*\*A two-dimensional point class.

\*/

двумерный

класс

точки

.

public class Point3d {

/\*\* X координата точки\*/

private double xCoord;

/\*\* Y координата точки\*/

private double yCoord;

/\*\* Конструктор, чтобы инициализировать точку к(x, y) значение. \*/

public Point3d(double x, double y) {

xCoord = x;

yCoord = y;

}

/\*\* Конструктор без параметров: значения по умолчанию к точке в источнике. \*/

public Point2d() {

//

this(0, 0);

}

/\*\* Верните X координат точки. \*/

public double getX() {

return xCoord;

}

/\*\* Возвратите координату Y точки. \*/

public double getY() {

return yCoord;

}

/\*\* Набор X координат точки. \*/

public void setX(double val) {

xCoord = val;

}

/\*\* Набор Y координат точки. \*/

public void setY(double val) {

yCoord = 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 distanceTo, который берет другой Point3d в качестве параметра и вычисляет двойную точность приближения

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));

}

}

Файл Lab1.java.

Метод computeArea берет три Point3d и вычисляет область в треугольнике, ограниченном ими.

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.

// Импортируем класс Scanner

import java.util.Scanner;

// Класс, содержащий статический основной метод.

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 j=1; j<=3; j++)

{

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

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

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

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

}

// Проверка на совпадение:

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

System.out.println("Введены одинаковые координаты");

eqError++;

}

// Если не совпало, 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=p2.distanceTo(p1);

double b=p1.distanceTo(p3);

double c=p3.distanceTo(p2);

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

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

}

}

1. **Выводы**:

Создала 2 класса, для описания точки в трёхмерном Евклидовом пространстве и для вычисления расстояния между двумя точками.