**Федеральное государственное общеобразовательное учреждение высшего образования**

**Московский государственный технический университет**

**им. Н. Э. Баумана**

Факультет "Информатика и системы управления"

Кафедра "Информационные системы и телекоммуникации"

**Отчет по лабораторной работе №2**

**"Формула Герона"**

Студент: Задубин А.А.

Группа: ИУ3-42Б

Преподаватель: Руденкова Ю.С.

**Задание:**

Создали новый класс Point3d который описывает точку в трехмерном евклидовом пространстве. Этот класс должен обеспечивать следующий функционал:

* Создание новой точки Point3d с тремя указанными значениями координат (типа double),
* Создание новой точки Point3d в позиции (0.0, 0.0, 0.0) по умолчанию,
* Чтение и изменение этих трех значений по отдельности, и
* Сравнение двух точек Point3ds на равенство значений координат с помощью метода equals.

Внутренние поля объекта Point3d  должны быть недоступны напрямую.

Далее, добавили новый метод distanceTo, который получает другой объект типа Point3d как аргумент, вычисляет расстояние по прямой до этого объекта, и возвращает это расстояние, используя тип double.

Создали второй класс с именем Lab1, который нужен главным образом для объявления статического метода main. Вспомнили, что метод main должен иметь модификатор доступа public, иметь тип возвращаемого значения void , и иметь один аргумент – массив элементов типа Strings. Добавили в класс следующий функционал:

Пользовательский ввод трех троек значений, каждое из которых задает координаты одной точки в трехмерном пространстве. Создали из этих входных данных объекты Point3d.

Сделали из этой функции метод класса Lab1. Заметили, что этот метод использует классы из пакета java.io, который не виден вашему коду по умолчанию. Добавили в начало нашего файла:

import java.io.\*;

Эта строчка добавит классы из пакета java.io к коду класс Lab1. (Это не требуется делать для классов из пакета java.lang потому что он добавляется к нашему коду автоматически)

Написали второй статический метод computeArea который получает три точки Point3d's и вычисляет площадь треугольника ограниченного этими точками. Вернули эту площадь в значении типа double.

* Использовали данные и код, который вы собрали и написали сами для вычисления и печати на экране площади треугольника.

*Перед* вызовом computeArea проверили на равенство значений все три точки Point3d's. Если в какой либо паре точки “одинаковы”, сообщили об этом пользователю и не вычисляли площадь.

**Код программы:**

*Lab1.java*

import java.io.\*;

public class Lab1 {

public static void main(String[] args) throws Exception {

// Point3d a1 = new Point3d(getDouble(), getDouble(), getDouble());

//

// Point3d a2 = new Point3d(getDouble(), getDouble(), getDouble());

//

// Point3d a3 = new Point3d(getDouble(), getDouble(), getDouble());

Point3d a1 = new Point3d(1, 1, 0);

Point3d a2 = new Point3d(-2, 4, 0);

Point3d a3 = new Point3d(-2, -2, 0);

if (a1.checkEquality(a2) || a2.checkEquality(a3) || a1.checkEquality(a3)) {

throw new Exception("Одинаковые точки");

}

System.***out***.println("Площадь: \t" + (float) *computeArea*(a1, a2, a3));

}

public static double getDouble() {

try {

// Set up a reader tied to standard input.

BufferedReader br = new BufferedReader(new InputStreamReader(System.***in***));

// Read in a whole line of text.

String s = br.readLine();

// Conversion is more likely to fail, of course, if there's a typo.

try {

double d = Double.*parseDouble*(s);

// Return the inputed double.

return d;

} catch (NumberFormatException e) {

// Bail with a 0. (Simple solution for now.)

return 0.0;

}

} catch (IOException e) {

return 0.0;

}

}

public static double computeArea(Point3d a1, Point3d a2, Point3d a3) {

double p = ((a1.distanceTo(a2) + a2.distanceTo(a3) + a1.distanceTo(a3)));

System.***out***.println("Периметр: \t" + p);

p = p / 2;

System.***out***.println("Полупериметр: \t" + p);

return Math.*sqrt*(p \* (p - a1.distanceTo(a2)) \* (p - a2.distanceTo(a3)) \* (p - a1.distanceTo(a3)));

}

}

*Point3d.java*

public class Point3d {

private double PointX, PointY, PointZ;

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

PointX = x;

PointY = y;

PointZ = z;

}

public Point3d() {

this(0, 0, 0);

}

public double getPointX() {

return PointX;

}

public void setPointX(double pointX) {

PointX = pointX;

}

public double getPointY() {

return PointY;

}

public void setPointY(double pointY) {

PointY = pointY;

}

public double getPointZ() {

return PointZ;

}

public void setPointZ(double pointZ) {

PointZ = pointZ;

}

public boolean checkEquality(Point3d a) {

return (a.getPointX() == PointX && a.getPointY() == PointY && a.getPointZ() == PointZ);

}

public double distanceTo(Point3d a) {

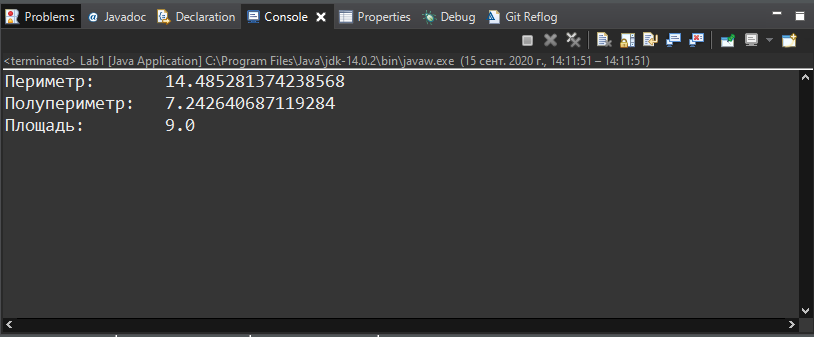
double x = a.getPointX() - PointX, y = a.getPointY() - PointY, z = a.getPointZ() - PointZ;

return (Math.*sqrt*(x \* x + y \* y + z \* z));

}

}

**Вывод в консоль:**

**

**Вывод:**

Мы научились создавать классы, которые обеспечивают работу с точками. С помощью координат этих точек мы смогли вычислить площадь заданного треугольника, если координаты точек не совпадают, иначе площадь вычислять нет необходимости.