

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ПРЕЗИДЕНТСКИЙ ФИЗИКО-МАТЕМАТИЧЕСКИЙ ЛИЦЕЙ №  
239

ОТЧЁТ ПО ГОДОВОМУ ПРОЕКТУ

Ученик:	Ширшин Даниил
Преподаватель:	Клюнин Алексей Олегович
Класс:	10-3

Санкт-Петербург  
2016

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>3</b>
<b>2</b>	<b>Алгоритм решения задачи</b>	<b>3</b>
2.1	Базовые структуры данных . . . . .	3
2.1.1	Point . . . . .	3
2.1.2	Triangle . . . . .	4
2.2	Построение алгоритма . . . . .	4

# 1 Постановка задачи

Дано  $3n$  точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество из  $n$  треугольников с вершинами в этих точках так, чтобы никакие два треугольника не пересекались и не содержали друг друга.

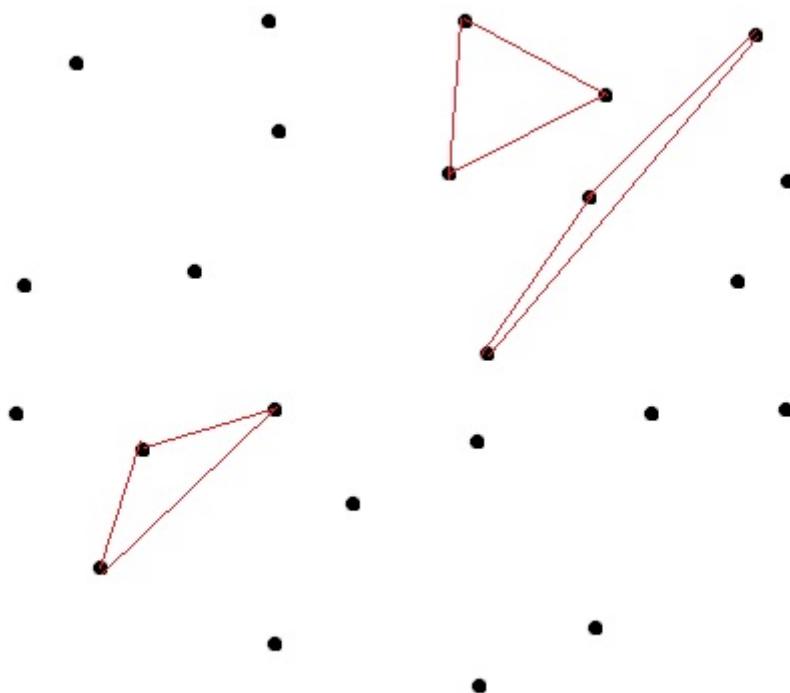


Рис. 1: Пример выделения треугольников из данного мн-ва точек

## 2 Алгоритм решения задачи

### 2.1 Базовые структуры данных

Понадобится 2 класса :

#### 2.1.1 Point

Класс Point описывает поведение точки.

Поля :

double x - отвечает за координату точки по оси "Ox";

double y - отвечает за координату точки по оси "Oy".

1) Сетеры(setX(),setY(),setFlg()).

2) Гетеры(getX(),getY(),getFlg()).

3) Конструкторы :

Point() - создает точку с координатами (0;0);

Point(double x,double y) - создает точку с координатами (x,y).

4) toString - вывод точки в виде : \*название точки\* (x,y).

5) del() - убирает точку из доступных для рассмотрения

Т.к. класс не будет наследоваться, то все поля приватные (private).

### 2.1.2 Triangle

Класс Triangle описывает поведение треугольника.

Поля :

double ax - отвечает за координату 1ой точки по оси "Ox";

double ay - отвечает за координату 1ой точки по оси "Oy".

double bx - отвечает за координату 2ой точки по оси "Ox";

double by - отвечает за координату 2ой точки по оси "Oy".

double cx - отвечает за координату 3ой точки по оси "Ox";

double cy - отвечает за координату 3ой точки по оси "Oy".

1) Конструкторы :

Triangle() - создает треугольник=точке с координатами (0;0);

Triangle(Point a,Point b,Point c) - создает треугольник от 3ех точек.

2) toString - вывод треугольника по координатам вершин.

Т.к. класс не будет наследоваться, то все поля приватные (private).

## 2.2 Построение алгоритма

Задача сводится к более простейшей: при поиске нового треугольника нужно искать такие три точки, которые имеют наименьшую координату по оси "Oy". Соответственно проекция данного треугольника на ось "Oy" будет прямая, концы которой лежат на координатах 2ух точек по "Oy" которые имеют наименьшую координату по оси "Oy" относительно остальных точек, а значит данный треугольник будет лежать ниже прямой, перпендикулярной прямой "Oy" и которой принадлежит точка треугольника с наибольшим значением по "Oy". Тогда проекции всех треугольников не будут иметь больше 1 точки по условию (на одной прямой лежит максимум 2 точки, следовательно при проецировании могут совпасть только координаты двух точек и не более). Значит стороны треугольников не пересекаются и они сами не содержатся друг в друге. Переходим к реализации.

1) Создадим массив для объектов Точки, заполним его, и создадим доп-массив.

2) Найдем точку с наименьшей координатой по "Oy". положим в доп-массив от 0 - индекс объекта этой точки, изменим ее поле flg на false.

3) Аналогично для каждой точки - для следующей точки с наименьшей координатой по "Oy" и имеющей flg=true : положим ее индекс в доп-массив с индексом какой мы по счету рассматриваем эту точку.

4) В цикле: для каждых трех последовательных точек (узнаем из доп-массива их индексы) создаем треугольник по трем точкам.

5) Выводим все треугольники

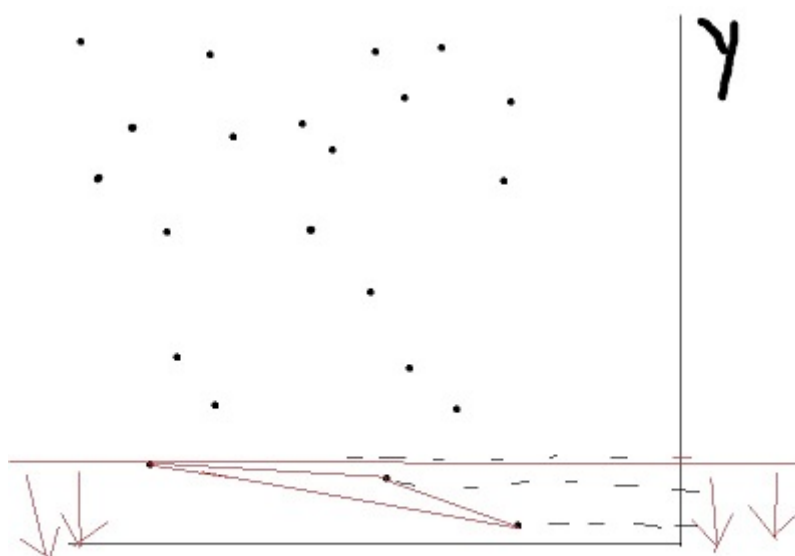


Рис. 2: Пример нахождения 1ого треугольника



```
1 public class Point {
2
3     //поля координат точки
4     private double x;
5     private double y;
6     //поле flg
7     boolean flg;
8
9     //сетеры
10    void setX(double x){
11        this.x=x;
12    }
13    void setY(double y){
14        this.y=y;
15    }
16    void setFlg(boolean flg){
17        this.flg=flg;
18    }
19
20    //гетеры
21    double getX(){
22        return this.x;
23    }
24    double getY(){
25        return this.y;
26    }
27    boolean getFlg(){
28        return this.flg;
29    }
30
31    //конструкторы
32    Point(){
33        this.x=0;
34        this.y=0;
35        flg=true;
36    }
37    Point(double x, double y){
38        this.x=x;
39        this.y=y;
40        flg=true;
41    }
42
43    //формат вывода
44    @Override
45    public String toString(){
46        return "("+this.x+";"+this.y+";";
47    }
48
49    //убирает точку из доступных для рассмотрения
50    void del(){
51        this.flg=false;
52    }
53 }
```

Рис. 3: Класс Point

```

Triangle Triangle()
1  public class Triangle {
2      double ax;
3      double ay;
4      double bx;
5      double by;
6      double cx;
7      double cy;
8
9      Triangle() {
10         ax=0;
11         ay=0;
12         bx=0;
13         by=0;
14         cx=0;
15         cy=0;
16     }
17     Triangle(Point a, Point b, Point c) {
18         ax=a.getX();
19         ay=a.getY();
20         bx=b.getX();
21         by=b.getY();
22         cx=c.getX();
23         cy=c.getY();
24     }
25
26     @Override
27     public String toString() { return "a (" + ax + "; " + ay + ") " + "b (" + bx + "; " + by + ") " + "c (" + cx + "; " + cy + ") "; }
30
31

```

Рис. 4: Класс Triangle