**ГБОУ “Президентский ФМЛ 239”**

**Нахождение правильных треугольников**

**Годовой проект по информатике**

**Павлов Михаил 10-1**

**1. Постановка задачи.**

Дано множество точек на плоскости. Определить среди них множество точек наибольшего размера такое, что каждая точка этого множества является вершиной равностороннего треугольника, вершины которого принадлежат этому множеству, вывести на экран это множество и все равносторонние треугольники.

**2. Уточнение исходных и выходных данных.**

Во входном файле points.txt хранятся координаты исходных точек. Количество точек не известно. Координаты точек по модулю не превышают 1. Необходимо вывести на экран все равносторонние треугольники, а также их вершины.

**3. Математическая модель.**

Ax, Ay, Bx, By, Cx, Cy – координаты вершин треугольника.

(Ax - Bx)2 + (Ay – By)2 ==(Ax - Cx)2 + (Ay – Cy)2 ==(Cx - Bx)2 + (Cy – By)2 – проверка треугольника на правильность.

**4. Анализ используемой структуры данных.**

Так как я планирую перебирать все тройки точек, каждую точку нужно хранить. Поскольку количество точек не известно, я буду хранить их в динамическом массиве. Количество точек и треугольников в ответе не известно, значит их тоже надо хранить в динамических массивах.

**5. Выбор метода решения.**

Создадим класс triangle. Переберем все тройки точек, по каждой тройке точек построим треугольник и проверим является ли он равносторонним. Если является, то занесём его и его вершины в ответы.

**6. Комментированный листинг.**

public void solve() {

resultTriangles.clear();  
resultPoints.clear();  
// перебираем тройки точек  
for (int i = 0; i < points.size(); i++) {  
 for (int j = i + 1; j < points.size(); j++) {  
 for (int k = j + 1; k < points.size(); k++) {  
 Point p1 = points.get(i);  
 Point p2 = points.get(j);  
 Point p3 = points.get(k);  
 Triangle t = new Triangle(p1, p2, p3);  
 // для каждого треугольника проверяем является ли он правильным  
 if (t.regular()) {  
 // если треугольник правильный заносим его и его вершины в ответ  
 resultTriangles.add(t);  
 resultPoints.add(p1);  
 resultPoints.add(p2);  
 resultPoints.add(p3);  
 }  
 }  
 }  
}

}

// создаёт случайную точку

public void addRandomPoints(int n) {  
 for (int i = 0; i < n; i++) {  
 Point p = Point.getRandomPoint();  
 points.add(p);  
 }  
}  
  
/\*\*  
 \* Очистить задачу  
 \*/  
public void clear() {  
 points.clear();  
 triangles.clear();  
 resultTriangles.clear();  
 resultPoints.clear();  
}  
  
/\*\*  
 \* Нарисовать задачу  
 \*  
 \* @param gl переменная OpenGL для рисования  
 \*/  
public void render(GL2 gl) {  
 gl.glColor3d(0.5, 0.7, 0.1);  
 for (Point point : points) {  
 point.render(gl);  
 }  
  
 gl.glColor3d(0.3, 0.1, 0.5);  
 for (Triangle triangle : resultTriangles) {  
 triangle.render(gl);  
 }  
 gl.glColor3d(0.3, 0.1, 0.5);  
 for (Point point : resultPoints) {  
 point.render(gl);  
 }  
}

// создаёт треугольник по трём точкам

public Triangle(Point a, Point b, Point c){  
 this.ax=a.x;  
 this.ay=a.y;  
 this.bx=b.x;  
 this.by=b.y;  
 this.cx=c.x;  
 this.cy=c.y;  
}  
// создаёт треугольник по координатам трёх точек  
public Triangle(double ax,double ay,double bx,double by, double cx, double cy){  
 this.ax=ax;  
 this.ay=ay;  
 this.bx=bx;  
 this.by=by;  
 this.cx=cx;  
 this.cy=cy;  
}

// проверяет треугольник на правильность   
public boolean regular(){  
 Vector2 posA =new Vector2(ax,ay);  
 Vector2 posB =new Vector2(bx,by);  
 Vector2 posC =new Vector2(cx,cy);  
 double ABx = posA.x-posB.x;  
 double ABy = posA.y-posB.y;  
 double BCx = posB.x-posC.x;  
 double BCy = posB.y-posC.y;  
 double ACx = posA.x-posC.x;  
 double ACy = posA.y-posC.y;  
 double q=Math.sqrt(ABx\*ABx+ABy\*ABy);  
 double w=Math.sqrt(BCx\*BCx+BCy\*BCy);  
 double e=Math.sqrt(ACx\*ACx+ACy\*ACy);  
 if((Math.abs(q-w)<0.01)&&(Math.abs(q-e)<0.01)&&(Math.abs(e-w)<0.01)) {  
 return true;  
 }else{  
 return false;  
 }  
}

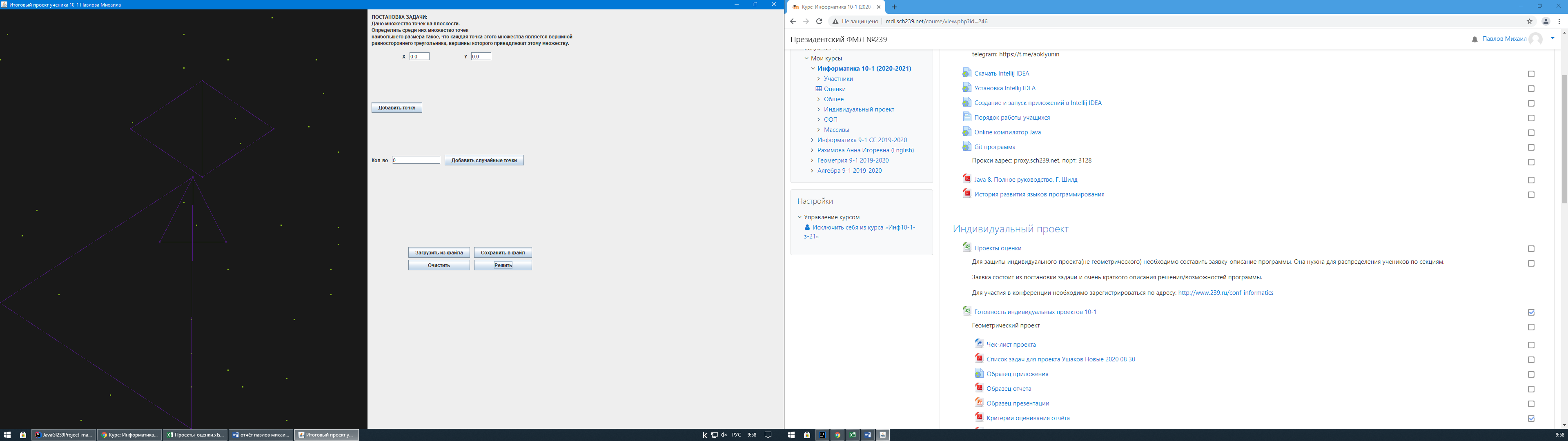
// создаёт случайный треугольник  
public static Triangle getRandomTriangle(){  
 double x1= Math.random()\*2-1;  
 double x2= Math.random()\*2-1;  
 double x3= Math.random()\*2-1;  
 double y1= Math.random()\*2-1;  
 double y2= Math.random()\*2-1;  
 double y3= Math.random()\*2-1;  
 return new Triangle(x1,y1,x2,y2,x3,y3);  
}

**7.**  **Пример работы программы.**

Исходные данные:

-0,80 0,04  
0,04 -0,48  
0,56 -0,76  
-1,00 -0,40  
-0,56 -0,96  
-0,28 0,46  
-0,88 -0,08  
0,04 -0,64  
0,68 0,44  
0,76 0,60  
-0,68 -0,36  
0,00 0,08  
0,80 -0,24  
0,84 -0,04  
0,60 -0,48  
0,00 0,72  
0,24 -0,24  
0,00 0,88  
0,04 -0,80  
0,24 0,76  
0,04 -1,00  
0,80 -0,36  
-1,00 0,76  
0,72 0,76  
0,28 0,48  
0,56 -0,96  
0,32 -0,80  
0,84 0,32  
-0,40 -0,84  
-0,96 0,88  
0,48 0,96  
0,84 -0,12  
0,24 -0,72  
0,49 0,43  
0,10 0,20  
0,10 0,66  
-0,29 0,43  
0,31 0,36  
0,07 -0,03  
0,53 -0,03  
-0,13 -0,11  
0,23 -0,11  
0,05 0,20

Выходные данные:



**8. Анализ правильности решения.**

Переберём все тройки точек из примера вручную и убедимся в правильности работы программы