ГБОУ "Президентский ФМЛ № 239"

ООпределение наличия у множества точек свойства быть дваждытреугольным

Годовой проект по информатике

Работу выполнила

ученица 10-7

Абросина Даша

Санкт-Петербург

2021

**Постановка задачи**

Множество точек на плоскости назовем дваждытреугольным, если каждая точка этого множества является вершиной хотя бы двух правильных треугольников, построенных по точкам множества. Определить, удовлетворяет ли заданное множество точек этому свойству.

**Уточнение исходных и выходных данных и ограничений на них.**

2.1. **Входные данные**: Координаты точек из вводимого нами множества.

Во входном файле points.txt содержатся строки с информацией о координатах заданных точек множества в формате: х, у, где х- координата точки по оси х, а у- координата по оси у. Количество строк заранее известно. Координаты записаны в десятичной форме. Координата х, а потом через пробел координата у. Координаты могут быть отрицательными, но находятся в пределе (-1; 1).

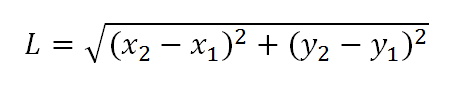
2.2. **Выходные данные.**

Ответ на вопрос, представленный на рисунке, созданном на основе написанной программы, соответствует ли заданное множество точек выбранному свойству.

**3. Математическая модель**

Нужно понять, является ли треугольник, вершины которого есть любые три точки данного множества, равносторонним.

Находим расстояние между двумя выбранными точками:



Где L - длина между выбранными точками, а х1у1 и х2у2- координаты выбранных точек.

**3.1 Анализ используемой структуры данных**

В задаче анализируются лишь координаты введенных из файла точек. Все координаты точек будут храниться в массиве points в формате double.

**3.2 Выбор метода решения**

Считываем построчно данные из файла по всем указанным координатам точек.

После чтения исходных данных выбираем одну любую точку, в паре к ней выбираем еще одну, и ко второй подбираем третью. Получаем три точки, которые и будут являться вершинами данного треугольника. далее, используя выше представленную формулу. используя координаты точек, вычисляем расстояние между каждой парой точек. После выясняем, равны ли полученные величины между собой. таким образом мы определяем, является ли данный треугольник равносторонним. Если это условие выполняется, то появится изображение треугольника, расстояние между вершинами которого равны. Повторяем эти действия, пока не переберем все исходные точки из файла. В итоге, если множество точек будет дваждытреугольным, то мы получим изображение, на котором все точки являются вершинами какого либо треугольника.

**4. Листинг программы**

/\*\*

\* **Добавить точку**

\*

\* @param x координата X точки

\* @param y координата Y точки

\*/

public void addPoint(double x, double y) {

Point point = new Point(x, y);

points.add(point);

}

/\*\*

\* **Решить задачу**

\*/

public void solve() {

triangles.clear();

// **перебираем пары точек**

for (Point p : points) {

for (Point p2 : points) {

for (Point p3 : points) {

Triangle triangle = new Triangle(p, p2, p3);

if (triangle.isRegular(p, p2, p3))

triangles.add(triangle);

}

}

}

}

/\*\*

\* **Загрузить задачу из файла**

\*/

public void loadFromFile() {

points.clear();

try {

File file = new File(FILE\_NAME);

Scanner sc = new Scanner(file);

// пока в файле есть непрочитанные строки

while (sc.hasNextLine()) {

double x = sc.nextDouble();

double y = sc.nextDouble();

sc.nextLine();

Point point = new Point(x, y);

points.add(point);

}

} catch (Exception ex) {

System.out.println("Ошибка чтения из файла: " + ex);

}

}

/\*\*

\* **Сохранить задачу в файл**

\*/

public void saveToFile() {

try {

PrintWriter out = new PrintWriter(new FileWriter(FILE\_NAME));

for (Point point : points) {

out.printf("%.2f %.2f\n", point.x, point.y);

}

out.close();

} catch (IOException ex) {

System.out.println("Ошибка записи в файл: " + ex);

}

}

/\*\*

\* **Добавить заданное число случайных точек**

\*

\* @param n кол-во точек

\*/

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();

}

/\*\*

\* **Нарисовать задачу**

\*

\* @param gl переменная OpenGL для рисования

\*/

public void render(GL2 gl) {

for (Point point : points)

point.render(gl);

for (Triangle triangle : triangles)

triangle.render(gl);

}

}

**5. Входные и выходные данные**

**5.1 Входные данные:**

0,25 0

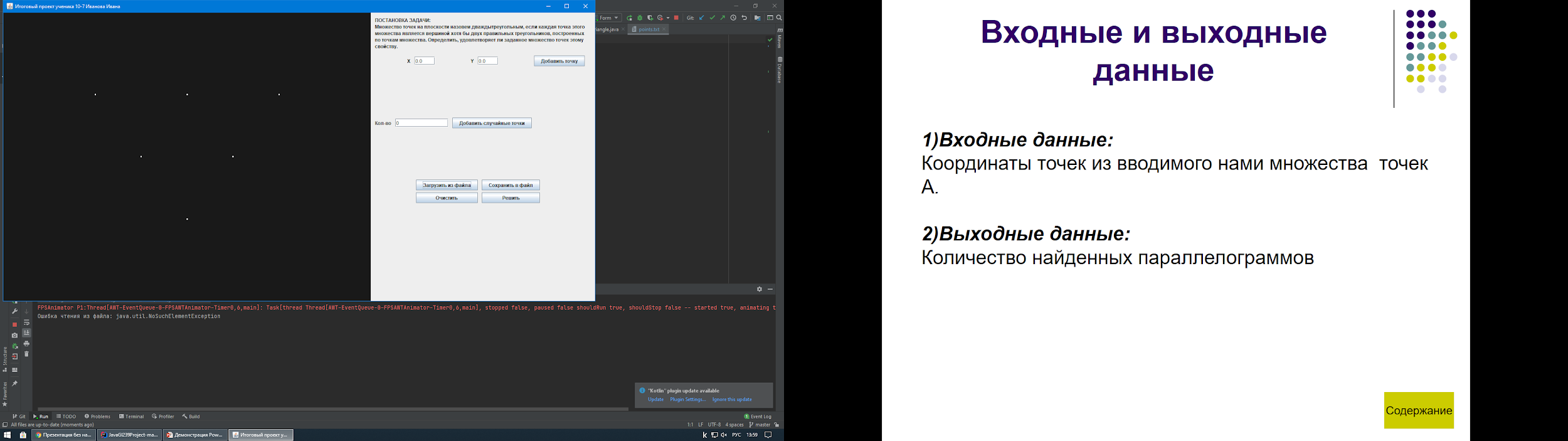
0 -0,4330127019

0,5 0,4330127019

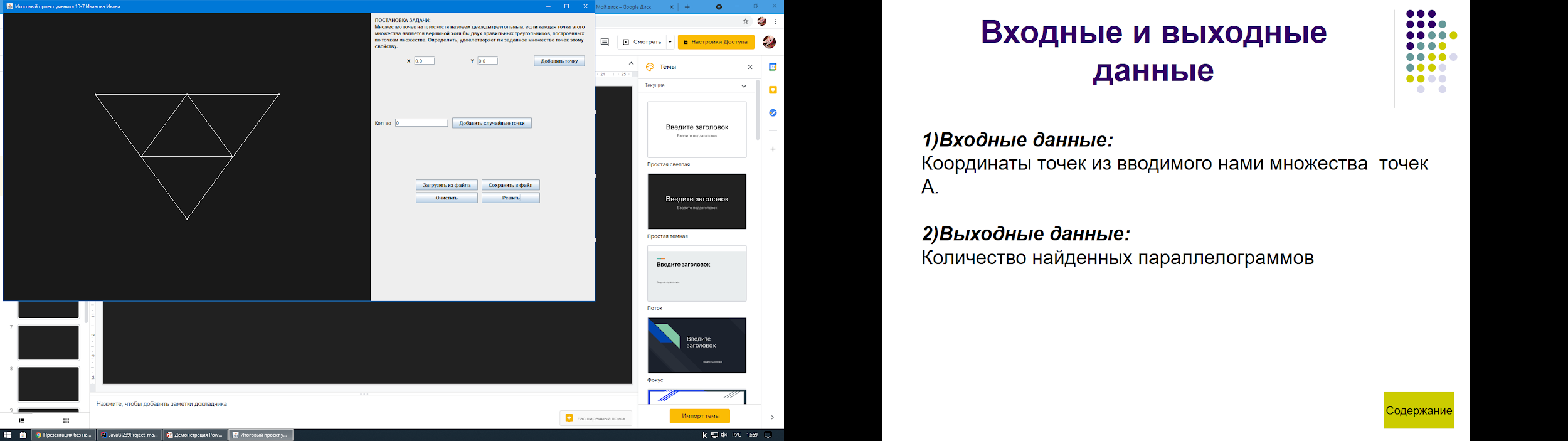
0 0,4330127019

-0,5 0,4330127019

-0,25 0

****

**5.2 Выходные данные:**

****