**ГБОУ "Президентский ФМЛ № 239"**

**Нахождение окружности, проходящей по крайней мере через три различные точки заданного множества точек на плоскости, и содержащей внутри наибольшее количество точек этого множества**

Годовой проект по информатике

Работу выполнила Ученица 10-3 класса Лабес Алёна

**1. Постановка задачи**

Определить радиус и центр окружности, проходящей по крайней мере через три различные точки заданного множества точек на плоскости, и содержащей внутри наибольшее количество точек этого множества. Вывести на экран радиус и координаты центра искомой окружности.

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

**2.1. Исходные данные**

Во входном файле in.txt в первой строчке содержится одно целое число – количество точек на плоскости, затем список координат точек плоскости в формате: XY где Х - координата точки по оси абсцисс, Y - координата точки по ординат. Например, (0 0) (1 0) (100 239). Каждая координата является целым числом. Число может быть отрицательным (знак минус перед числом). Порядок указания точек в файле не определен, значит будем считать, что он произвольный.

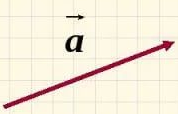
**2.2. Выходные данные**

Необходимо вывести на экран несколько чисел — координаты Х и У центра искомой окружности и ее радиус. Возможно, ответом будет строчка «**No such circle**», если невозможно создать окружность по входным данным

**3. Выбор метода решения**

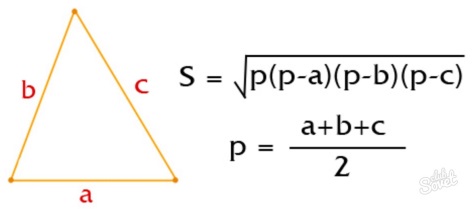
**3.1. Математическая модель**

а) длина вектора



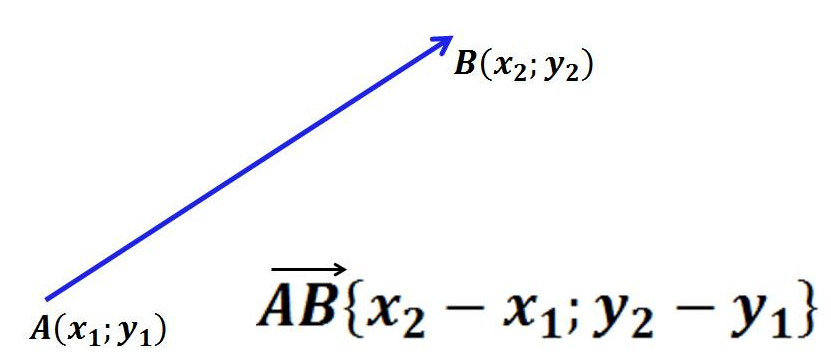
б) формула Герона для нахождения площади треугольника

, где

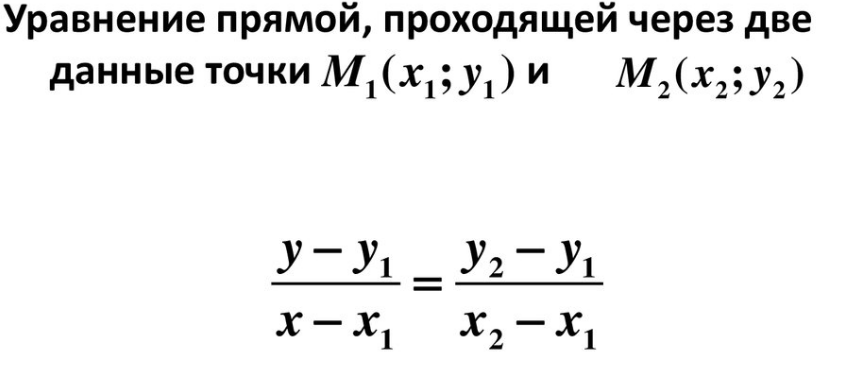


в) нахождение координат вектора через координаты его концов

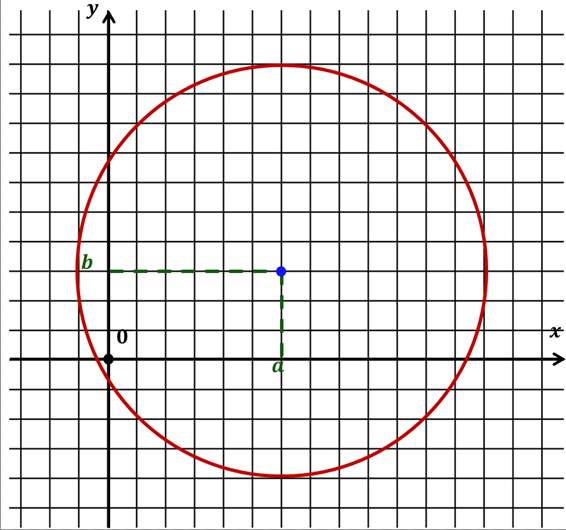
x = x2-x1;  y = y2-y1 ;



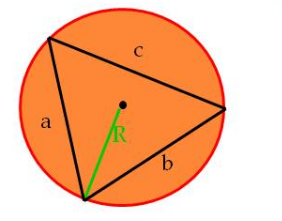
г) уравнение прямой:



д) уравнение окружности



е) радиус описанной около треугольника окружности , где а,b,c стороны вписанного в окружность треугольника, а S – его площадь



**3.2. Анализ исходных данных и выбор используемой структуры данных**

В задаче требуется использовать информацию о координатах каждой из точек. Для этого достаточно одного массива точек

**3.2. Выбор метода решения**

Для решения поставленной задачи целесообразно создать несколько классов, которые описывают сущность проекта: class Point, class Circle, Class Reader, Class main, используя программу IntelliJ IDEA. Каждый класс отвечает за определенную часть проекта.

1. Class Poin:
2. поля типа float

float x; float y - координаты точки на плоскости

1. Метод: we\_are\_triangle, проверяющий можно ли составить из трех точек треугольник, т.е проверяющий, что три точки не лежат на одной прямой. Для этого пишется уравнение прямой через две точки, переданные в аргументе метода, подставляем в уравнение третью точку и смотрим лежит ли третья точка на прямой.
2. Class Vector, содержащий в себе поля типа float, float x; float y – координаты вектора.
3. Class Circle:   
   1)Поля типа float   
   float cx; float cy; float r - радиус окружности и координаты центра   
     
    2)Кострукторы:  
    Конструктор, который будет задавать окружность по трем точкам, возвращая радиус окружности и координаты ее центра: в конструктор передается три точки. Создается три вектора с концами в двух заданных точек, получается треугольник(искомая окружность – есть окружность, описанная вокруг этого треугольника). Используя формулу для нахождения длины(модуля) векторая, узнаем длины сторон треугольника. Посчитав площадь треугольника с помощью формулы Герона находим радиус искомой окружности используя формулу для нахождения радиуса описанной около треугольника окружности.

Для нахождения центра окружности необходимо написать два уравнения прямых, проходящих через середину стороны уже известного треугольника перпендикулярно стороне. Приравнивая два уравнения, находим х, у – искомые координаты центра окружности.

Таким образом, конструктор возвращает радиус и центр окружности по трем точкам.

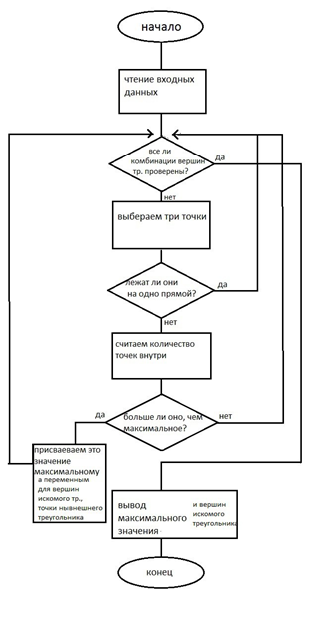
3)Методы:   
  
 1.public bool point\_in\_side, проверяющий принадлежит ли данная точка кругу, ограниченному данной окружностью. Для этого подставляем координаты точки в уравнение окружности и смотрим  меньше ли нуля получившиеся выражение. Если меньше – значит, точка лежит в круге, если больше – точка лежит вне круга, если равняется нулю – точка лежит на окружности(такой случай нас тоже устраивает)

2.static public boolean isAccesible, проверяющий что площадь данного треугольник не равна нулю.

4. Class Reader, который будет считывать из файла заданное множество точек в список, содержащий в себе переменные типа Point, используя метод read, а затем выводить полученный результат в выходной файл.

5. class Main, который реализует проект, тело проекта.

в классе Main перебираем все возможные троеточия, которые задают окружность, используя три вложенных цикла for. С помощью еще одного цикла считаем сколько точек лежит внутри(и на) данной окружности. сравниваем количество точек с максимальным, и если число точек больше максимума, то перезаписываем max и maxInd, где лежат центр и радиус окружности с максимальным количеством точек из данного множества

****

**4. Листинг программы**

**int** max\_kolvo = -1; *// переменная, в которой будет лежать*

*максимальное количество точек лежащее*

*внутри окружности*

**int** maxi = -1;   
**int** maxj = -1; //переменные, где будут храниться три точки,

**int** maxk = -1; через которые проходит искомая окружность

**int** l = *points*.size(); *// количество точек на плоскости*

**for** (**int** i = 0; i < l - 2; i++) { //перебор всех троек точек  
 **for** (**int** j = i + 1; j < l - 1; j++) {  
 **for** (**int** k = j + 1; k < l; k++) {

//вычисление ответа. В случае

существования нужной окружности

возвращает объект класса Circle

**if** (max\_kolvo == -1) {

System.***out***.println(**"No such circle"**);  
 **return new** Circle(0,0,0);  
} **else** {  
 Circle c = **new** Circle(*points*.get(maxi), *points*.get(maxj), *points*.get(maxk));  
 System.***out***.println(max\_kolvo);  
 System.***out***.println(c);  
 System.***out***.println(**"выполнено"**);  
 **return** c;

**new** Circle(*points*.get(i), *points*.get(j), //создается новый

*points*.get(k)).point\_in\_side(*points*.get(t) объект класса Circle

//расчет трех сторон треугольника, полупериметра, площади и, как итог, радиуса

s1 = (**int**) Math.*sqrt*((a1.**x** - a2.**x**) \* (a1.**x** - a2.**x**) + (a1.**y** - a2.**y**) \* (a1.**y** - a2.**y**));

s2 = (**int**) Math.*sqrt*((a3.**x** - a2.**x**) \* (a3.**x** - a2.**x**) + (a3.**y** - a2.**y**) \* (a3.**y** - a2.**y**));  
s3 = (**int**) Math.*sqrt*((a1.**x** - a3.**x**) \* (a1.**x** - a3.**x**) + (a1.**y** - a3.**y**) \* (a1.**y** - a3.**y**));  
p = (s1 + s2 + s3) / 2;  
area = (**int**) Math.*sqrt*(p \* (p - s1) \* (p - s2) \* (p - s3));  
**this**.**r** = s1 \* s2 \* s3 / (4 \* area);

//расчет модуля вектора с координатами X(a1.x – a2.x); Y(a1.y – a2.y)

s1 = (**int**) Math.*sqrt*((a1.**x** - a2.**x**) \* (a1.**x** - a2.**x**) + (a1.**y** - a2.**y**) \* (a1.**y** - a2.**y**));

//метод, проверяющий что точка лежит внутри круга

**public boolean** point\_in\_side(Point a){

**if**(((**this**.**cx** - a.**x**)\*(**this**.**cx** - a.**x**) +(**this**.**cy** - a.**y**)\*(**this**.**cy** - a.**y**)- **this**.**r**\***this**.**r**)<=0){  
 **return true**;  
 }  
 **else**{  
 **return false**;  
 }

**5. Пример работы программы**

**5.1 исходные данные 5.2 выходные данные**

**Пример 1**

13 {219, 324; 181}  
60 411  
285 494  
388 345  
257 369  
188 200  
234 234  
200 180  
231 239  
194 194  
394 276  
315 407  
283 378  
245 283

**Пример 2**

5 {215, 335; 172}   
60 411  
285 494  
388 345  
257 369  
188 315

**6. Анализ правильности решения**

Рассмотрим несколько тривиальных примеров и убедимся в правильности результата

**Пример 1**

21 No such circle

200 500   
200 400  
200 300  
200 200  
200 100  
200 150  
200 50 Как видно из условия, все заданные точки имеют 200 250 одинаковую Х координату, т.е лежат на одной прямой.

200 350 Таким образом, не существует ни одной окружности, 200 450 которая могла бы пройти как минимум через три точки

200 550 плоскости, потому что не существует треугольника,

200 525 который бы мог быть вписан в эту окружность. (нет трех

200 425 точек, которые бы стали его вершинами)

200 325  
200 225  
200 125  
200 275  
200 375  
200 475  
200 175  
200 75

**Пример 2**

3 {250, 250; 71}  
200 200  
200 300  
300 200

Можно заметить, что данные три точки образуют на плоскости прямоугольный треугольник. Искомая окружность – описанная вокруг этого треугольника. Найдем радиус и центр этой окружности.

Известно, что радиус описанной около прямоугольного треугольника окружности равен половине гипотенузы. По теореме Пифагора гипотенуза равна = 141, т.е радиус равен 71 по правилу округления. Также известно, что центр окружности, описанной вокруг прямоугольного треугольника, лежит на середине гипотенузы. Координаты середины отрезка, т.е координаты центра окружности, находятся по формуле ; , т.е в данном случае = 250;   = 250;

