Физико-математический лицей № 239

**На плоскости задано множество точек. Найти окружность наименьшей площади, внутри которой находятся все точки множества. Если таких окружностей несколько, найти любую. В качестве ответа нарисовать найденную окружность.**

Отчёт о проекте по информатике

Работу выполнил  
Ученик 10-7 класса  
Захаров Владимир

Санкт-Петербург

2021

**1. Постановка задачи**

На плоскости задано множество точек. Найти окружность наименьшей площади, внутри которой находятся все точки множества. Если таких окружностей несколько, найти любую. В качестве ответа нарисовать найденную окружность.

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

**2.1.Исходные данные**Во входном файле points.txt содержится множество точек на плоскости, удовлетворяющих неравенству |x| + |y| ≤ 2

**2.2. Выходные данные**   
Необходимо вывести на экран окружность такую, что все точки лежат внутри неё и её радиус минимален

**3. Выбор метода решения**

**3.1. Анализ исходных данных и выбор используемой структуры данных**  
 В задаче логично будет использовать список точек, так как с ним будет удобнее всего работать

**3.2. Выбор метода решения**Оптимально будет найти геометрический центр множества точек, а затем найти самую удаленную от него точку. Окружность, проходящая через эту точку и геометрический центр и будет искомой

**4. Составление алгоритма**

**4.1. Обобщенная блок-схема алгоритм**

Началоооо

Конец

Чтение данных. Запись в список

Поиск геометрического центра точек

Поиск самой удаленной от него точки

Вывод окружности

**4.2. Блок-схема алгоритма**

**5. Листинг программы**

*public void* solve() {  
 *Point* center = *new* Point(0, 0);  
 *double* max = 0;  
 *Point* p0 = *new* Point(0,0);  
 *for* (*Point* p: points){  
 p0.x += p.x;  
 p0.y += p.y;  
 }  
 p0.x /= points.size();  
 p0.y /= points.size();  
 *for* (*Point* p : points)  
 *if* (dist(p0, p) > max) max = dist(p0, p);  
 *if* (max != 0)  
 resultCircle = *new* Circle(*new* Point(p0.x, p0.y), max + 0);

**6. Пример работы программы**

**6.1. Исходные данные**

1.1 1

1.2 1.2

1.3 -1

1.1 -1

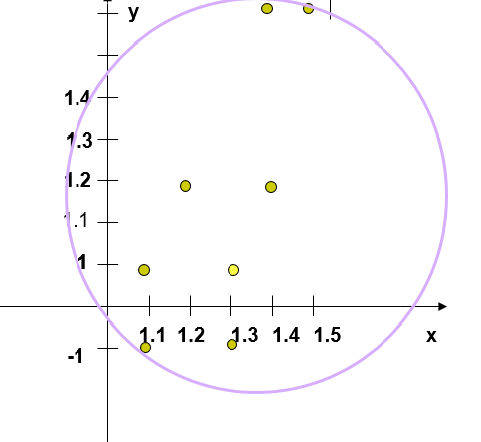
1.3 1

1.4 1.2

1.4 1.6

1.5 1.6

**6.2. Выходные данные**

****