**Государственное бюджетное общеобразовательное учреждение "Президентский физико-математической лицей № 239"**

**Нахождение пары пересекающихся окружностей, с максимальной длиной отрезка, соединяющего точки их пересечения, в заданном множестве окружностей**

Годовой проект по информатике

Работу выполнила

ученица 10–2 класса

Костюкевич София

Санкт-Петербург

2020

1. **Постановка задачи**

На плоскости задано множество окружностей. Найти такую пару пересекающихся окружностей, что длина отрезка, проведенного от одной точки пересечения этих двух окружностей до другой, максимальна. В качестве ответа: выделить эту пару окружностей, нарисовать отрезок между найденными точками пересечения.

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

**2.1. Исходные данные**

Во входном файле circles.txt содержатся строки с информацией о каждой заданной окружности в формате: x y x1 y1, где x и y — координаты центра задаваемой окружности, а x1 и y1 — координаты точки лежащей на этой окружности. Например, -0,20 -0,88 -0,44 -0,08. Количество строк заранее неизвестно. Каждая из заданных координат представляет из себя вещественное число, в котором дробная часть (если она есть) отделена запятой от целой части. Дробная часть может отсутствовать. Число может быть отрицательным (знак минус перед числом). Ограничения на координаты заданы, это вещественные числа в области [(-1,-1), (1,1)].

**2.2. Выходные данные**

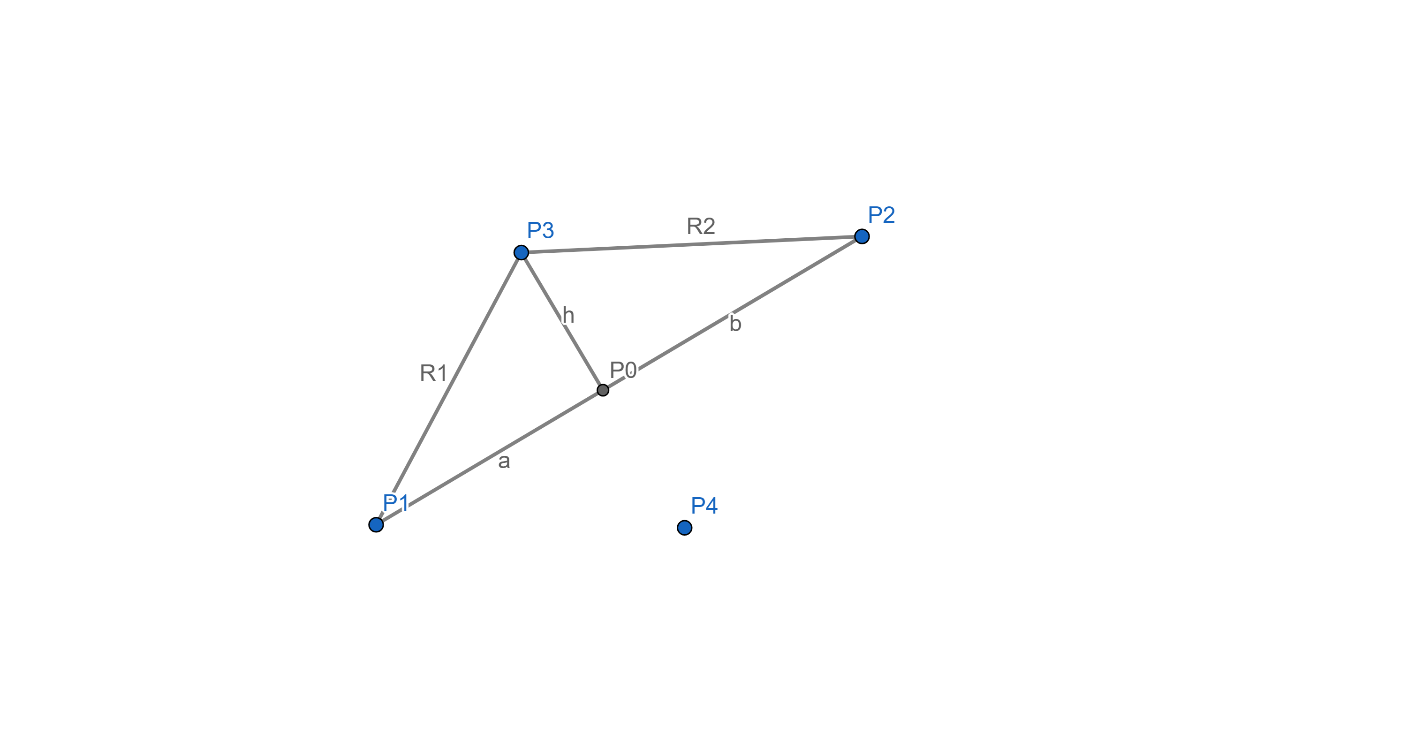
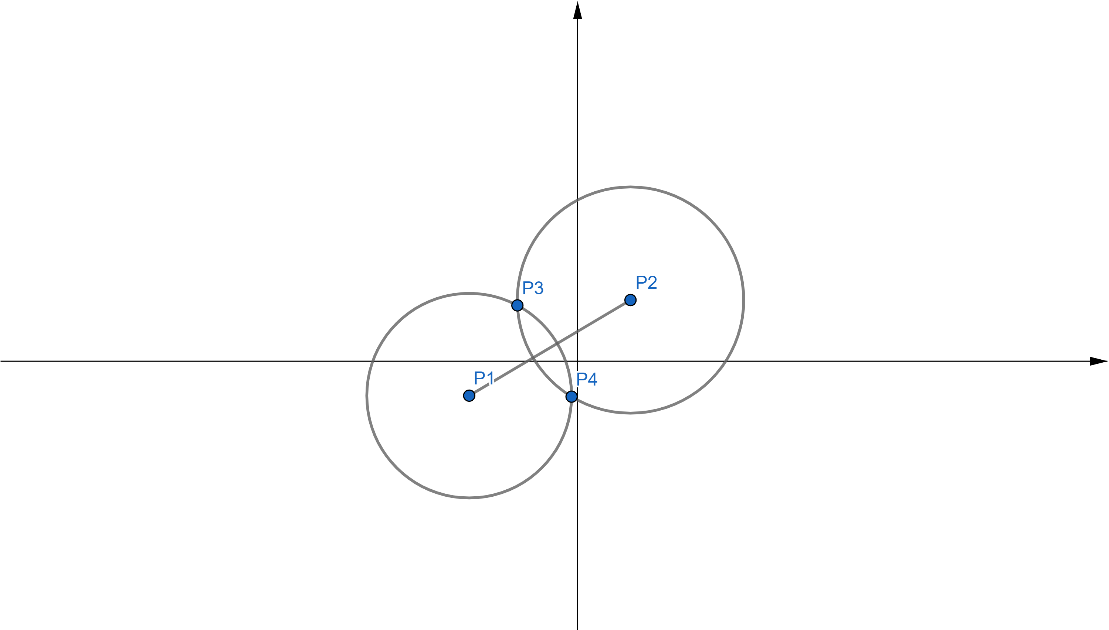
Необходимо выделить на экране пару окружностей — таких, что длина отрезка, соединяющего точки их пересечения, максимальна; и нарисовать этот отрезок. Возможно, таких окружностей не будет, если среди заданных окружностей не будет пересекающихся. В текстовом файле – координаты начала и конца искомого отрезка.

1. **Математическая модель**

Сначала найдём радиус каждой окружности следующим образом:

| r — радиус окружности; x1 — x координата точки на окружности; x — x координата центра окружности; y — y координата центра окружности; y1 — y координата точки на окружности.

(Эта формула использовалась для нахождения любых расстояний между двумя любыми точками.)



Теперь приступим к решению самой задачи. Точка P1 является центром первой окружности радиусом R1 и точка P2 центром второй окружности радиусом R2.

Введём обозначения:

* h - высота треугольника P1P3P2 (расстояние между точками P3 и P0)
* a - расстояние между точками P1 и P0
* b - расстояние между точками P2 и P0
* d = a + b; - расстояние между точками P1 и P2

Проверяем, если R1 + R2 <d, тогда окружности не пересекаются, т. к. лежат отдельно, если |R1 - R2|> d, тогда окружности так же не пересекаются, т. к. одна окружность находится внутри другой.

Рассмотрим 2 прямоугольных треугольника, P1P3P0 и P2P3P0. Получаем:

Так как эти треугольники имеют общий катет (h), тогда:

Выразим a через b:

Теперь выразим b как корень квадратного уравнения, получаемого путём несложных математических преобразований:

Зная b, находим a

Найдем точку P0 с помощью векторного параметрического уравнения прямой:

Зная координаты точки P0 находим координаты точек P3 и P4. Для этого возьмём единичный вектор от P1 до P2, повернем его на +90 градусов для P3 (-90 градусов для P4) и умножим на h.

Таким образом находим координаты точки P3:

Аналогично найдём координаты точки P4.

1. **Анализ используемой структуры данных**

* задаче требуется использовать информацию о конкретных окружностях. Отметим, что данные могут изменяться во время работы программы, у пользователя есть возможность как добавить новые окружности, так и удалит все предыдущие. В связи с этим рационально использование динамического списка, как способа хранения данных. Все входящие данные являются вещественными числами, поэтому имеет смысл использовать в решении тип данных double.

1. **Выбор метода решения**

Считываем построчно данные из файла по каждой окружности, извлекаем из них все необходимые координаты. Сохраняем их в динамический список как объекты класса окружность. Заранее прописываем в классе окружностей функцию, которая находит радиус по заданным точкам.

После чтения исходных данных и сохранения их в массиве перебираем все пары окружностей, чтобы найти искомую. Каждый раз сравниваем длину отрезка, соединяющего точки пересечения этих окружностей (находим его с помощью специальной функции, подробное объяснение которой находится в разделе «математическая модель») с максимальной длиной (изначально задаём её как 0). Если она больше предыдущей максимальной, то мы запоминаем эти новые окружности.

Перебрав все пары, выводим ответ. Т. е. рисуем искомый отрезок и обновляем вывод окружностей из списка, причём меняя цвет найденных окружностей (цвет задаётся как переменная для функции рисования окружностей)

1. **Листинг программы**

{ первая из двух искомых окружностей (текущая) }

Circle max\_c = null;

{ вторая из двух искомых окружностей (текущая) }

Circle max\_c2 = null;

{ длина отрезка между пересечениями max\_c и max\_c2 }

double max\_length = 0;

{ для каждой пары окружностей списка }

for (Circle c : circles) {

for (Circle c2 : circles) {

{ если это различные окружности }

if (c != c2) {

{ если расстояние между их пересечениями больше, чем максимальное}

{ Circle.GetCrossPoints() – реализация математической модели }

if (Circle.GetCrossPoints(c, c2).r > max\_length) {

{ обновляем искомые окружности и максимальную длину (до текущих) }

max\_length = Circle.GetCrossPoints(c, c2).r;

max\_c = c;

max\_c2 = c2;

}

}

}

}

{ если нашлись пересекающиеся окружности }

assert max\_c != null;

{ меняем цвет окружностей, чтобы их выделить }

max\_c.color = 1;

max\_c2.color = 1;

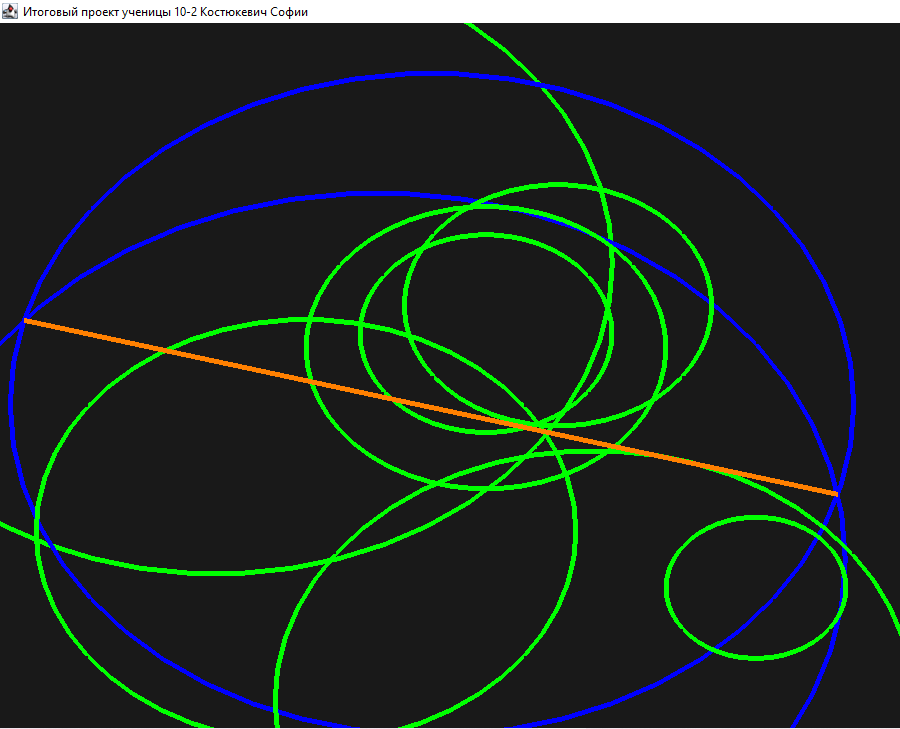
{ задаём искомый отрезок }

lines.clear();

lines.add(Circle.GetCrossPoints(max\_c, max\_c2));

1. **Пример работы программы**

|  |  |
| --- | --- |
| **6.1. Исходные данные** | **6.2 Выходные данные** |
| -0,52 0,32 0,04 -0,36 | 0,86 -0,34 |
| -0,16 -0,52 -0,44 0,48 | -0,95 0,16 |
| 0,08 0,12 0,36 0,12 |  |
| -0,04 -0,08 0,14 -1,00 |  |
| 0,32 -0,92 0,12 -0,24 |  |
| 0,68 -0,60 0,56 -0,44 |  |
| 0,08 0,08 0,32 0,40 |  |
| 0,24 0,20 0,12 -0,12 |  |
| -0,32 -0,44 0,16 -0,08 |  |
| -0,52 -0,80 -0,04 -0,24 |  |



1. **Анализ правильности решения**

Т. к. задача геометрическая, исходные данные и ответ имеют графическое представление, то для доказательства правильности решения достаточно несколько раз получить верный чертёж. Для того чтобы убедиться в правильности чертежей, будем задавать достаточно наглядные примеры вручную и смотреть на результат.

