ГБОУ "Президентский ФМЛ № 239"

**Поиск отрезка наибольшей длины между точками пересечения двух окружностей**

Годовой проект по информатике

Работу выполнила

Ученица 10-1 класса

Лебедева Екатерина

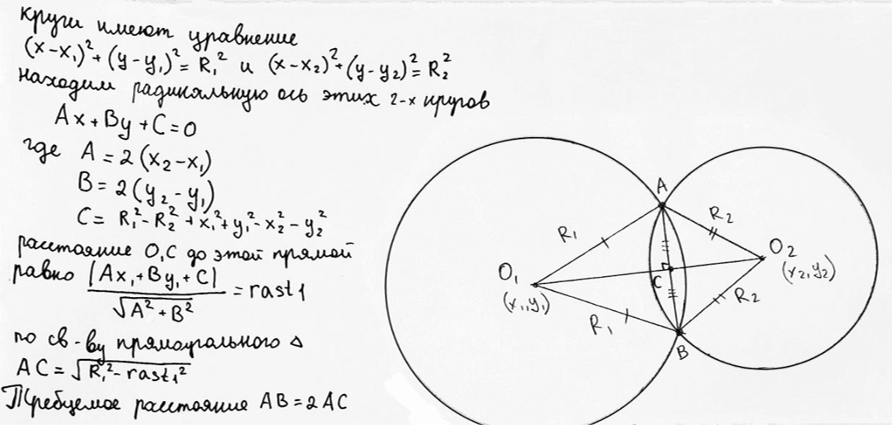
**1. Постановка задачи.**

На плоскости задано множество окружностей. Найти такую пару пересекающихся окружностей, что длина отрезка, проведенного от одной точки пересечения этих двух окружностей до другой, максимальна.

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

Исходные данные - набор из групп действительных чисел, состоящих из двух координат точки центра и длины радиуса круга, количество групп ограничивается лишь возможностями компьютера.

Выходные данные - две группы такого же вида, такие, для которых измеряемое значение максимально.

**3. Математическая модель.** 

**4. Анализ используемой структуры данных.**

Входные данные хранятся в массиве circles в виде точки и действительного числа, так же хранятся данные пары кругов, для которых измеряемое значение максимально среди уже перебранных пар (изначально они равны null). Также для этой пары хранится длина измеряемого отрезка, которая считается в момент перебора этой пары.

**5. Выбор метода решения.**

Для начала создадим метод, считающий расстояние между точками пересечения двух кругов (если такие есть) в классе Circle. Метод подробнее описан в математической модели.

Далее в основной части программы перебираем все возможные пары. Это делается с помощью двух вложенных циклов for, т. е. программа выполняется за n^2. Для каждой пары мы считаем нужный параметр и сравниваем с максимальным из уже известных и заменяем его на полученный если он больше. Таким образом после завершения цикла результирующие переменные будут содержать наибольшее значение расстояния между точками пересечения двух кругов из возможных и данные этих кругов соответственно. Далее мы просто выводим на экран эти круги другим цветом и рисуем отрезок между точками их пересечения.

Рассмотрим основную часть самого кода

public void solve() {

solve = 0; //

resultCircle1 = null; // обнуляем переменные  
 resultCircle2 = null; //  
 double maxLength = -1; //  
  
 // перебираем пары кругов  
 for (int i = 0; i < circles.size(); i++) { // заходим в цикл  
 for (int j = i + 1; j < circles.size(); j++) {   
 Circle c = circles.get(i); // достаем круги из массива  
 Circle c2 = circles.get(j); //  
  
 if (!Double.*isNaN*(c.OLength(c2)) && c.OLength(c2) > maxLength && !c.pos.equals(c2.pos)) { // проверяем не является ли значение функции NaN, не являются ли круги совпадающими и больше ли полученное значение максимального   
 resultCircle1 = c; // записываем круги в результирующие переменные  
 resultCircle2 = c2; //  
 maxLength = c.OLength(c2); // и расстояние между точками их пересечения тоже

posA = c.A(c2); // а также координаты точек пересечения  
 posB = c.B(c2); //  
 }  
 }  
 }  
 if (maxLength == -1) { // в случае если никакие круги не пересекаются, меняем переменную, отвечающую за вывод сигнала о том что пересечений нет  
 solve = 1; //  
 }   
}

**6. Пример работы программы.**

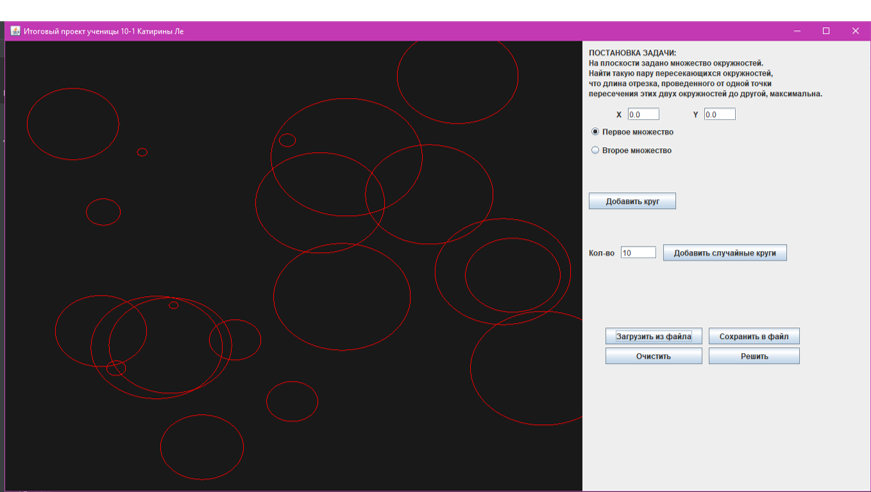
**Входные данные:**

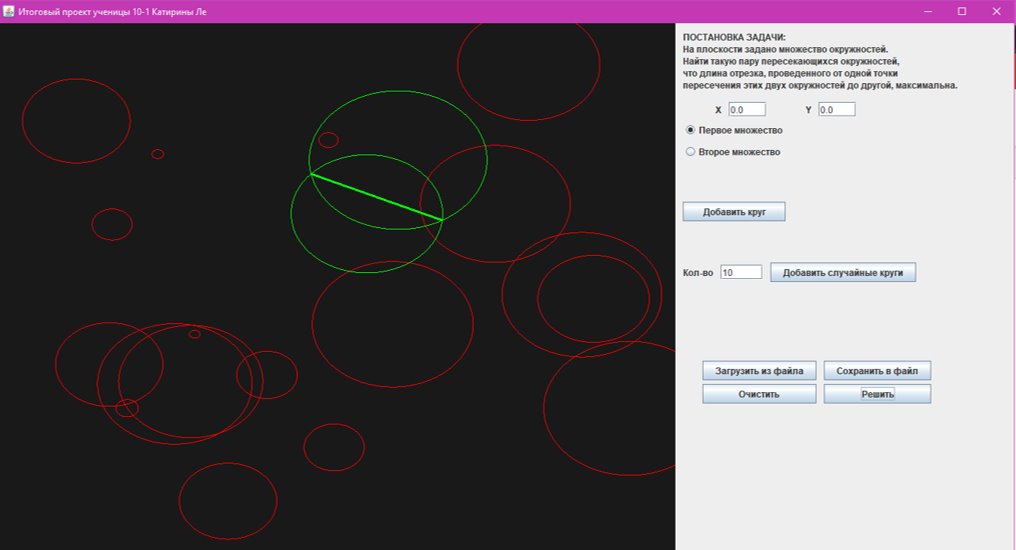
**0,86545263 -0,45471616 0,25304672**  
**0,56796101 0,84276191 0,20979158**  
**0,09119011 0,28011589 0,22323325**  
**-0,02188219 0,55875360 0,02810661**  
**-0,52537131 0,50568534 0,01685170**  
**-0,61507742 -0,45435473 0,03281474**  
**-0,66798501 -0,28935232 0,15803739**  
**-0,20306359 -0,32847766 0,08931976**  
**-0,42768171 -0,35354655 0,21237544**  
**-0,76505362 0,63085913 0,15855563**  
**-0,47490952 -0,36230045 0,22764259**  
**0,46958636 0,31764898 0,22099691**  
**-0,31788579 -0,80551307 0,14353450**  
**0,72462613 -0,02547332 0,23536545**  
**-0,65956773 0,23969925 0,05866007**  
**-0,00533930 -0,60219311 0,08847970**  
**0,18320250 0,48302193 0,26183962**  
**-0,41629442 -0,17449789 0,01517272**  
**0,75900532 -0,04105464 0,16417592**  
**0,16740575 -0,13719859 0,23705115**

**Выходные данные:**

**0.09119011 0.28011589 0.22323325**

**0.1832025 0.26183962 0.48302193**





Для всех наборов кругов, которые я получала случайным образом, программа выдавала правильный результат (это было видно и визуально и подтверждалось расчетами), поэтому я могу сделать вывод, что программа работает правильно.