**Комитет по образованию г. Санкт-Петербург**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ПРЕЗИДЕНТСКИЙ ФИЗИКО-МАТЕМАТИЧЕСКИЙ**

**ЛИЦЕЙ №239**

**Отчет о практике**

**«Создание графических приложений на языке Java»**

Учащийся 10-3 класса

Оганесян С.А.

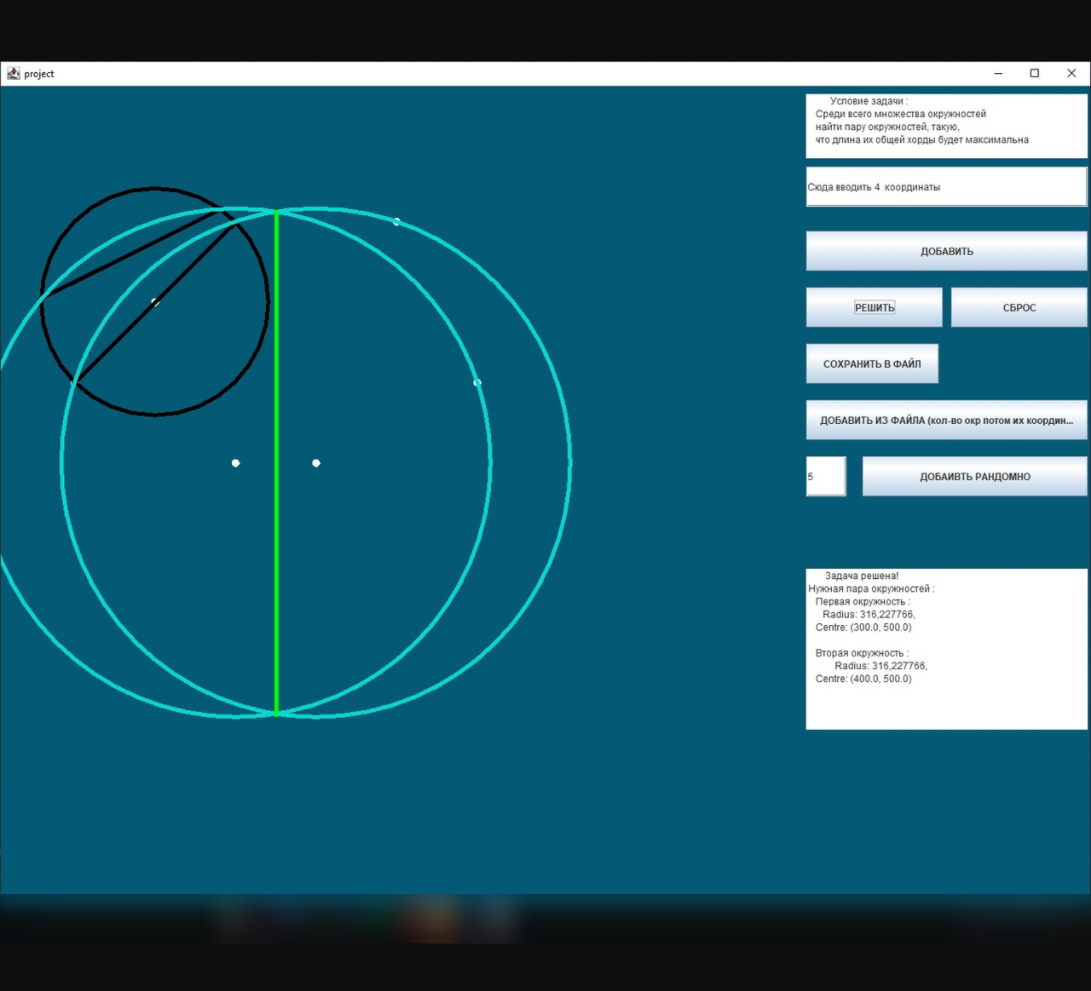
Преподаватель:

Клюнин А.О.

Санкт-Петербург – 2022 год

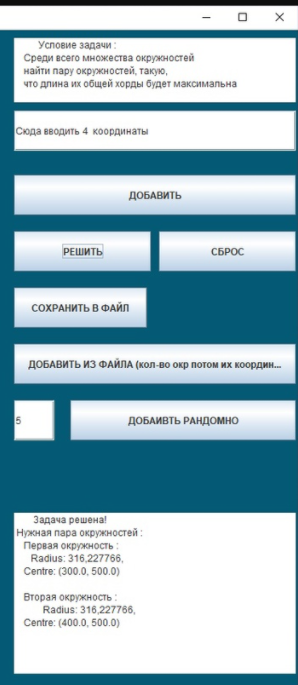
# 1. Постановка задачи

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.



# 2. Элементы управления

Для решения данной задачи необходимо было реализовать следующие элементы управления:



Для добавления точки по координатам было создано поле ввода координат X и Y центра окружности и точки окружности.

Основными обьектами в данной задаче являются окружности , для рандомного добавления некоторого их числа отведено отдельное окно, где надо написать их количество и нажать кнопку “Добавить рандомно”.

Также программа позволяет добавлять точки с помощью клика мышью в области окна, добавлять окружности по их координатам из спец файла.

Также кроме добавления обьектов есть и другие функции панели управления, такие как:

1)Решить задачу для уже заданных обьектов

2)Сбросить все заданные обьекты(Очистить окно)

3)Сохранить все добавленные окружности, а точнее их координаты в спец файл

4)окно с условием задачи и окно с актуальным состоянием программы

3. Структуры данных

Для того чтобы хранить точки, был разработан класс **Point.java.** Лист Point.

В лист точки добавляютя после задания их координа. Хранятся точки центров окружностей, пересечения окружностей. (Точки хранятся=запоминаем их координаты)

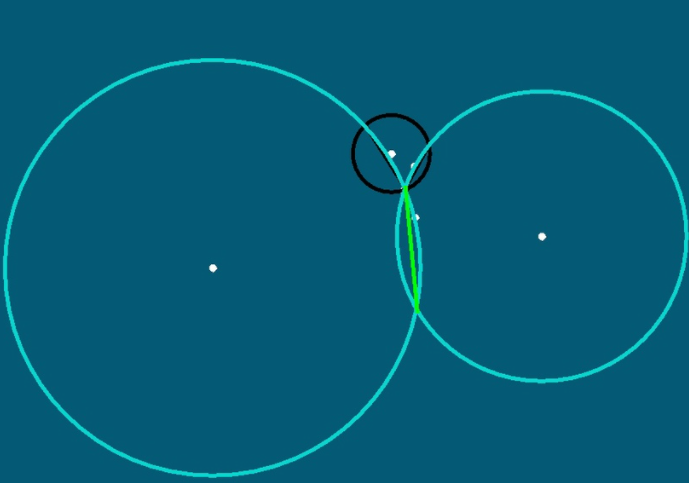
Точки нужны , чтобы по ним задавать окружности, этим занимается класс Circle.java, также есть лист окружностей, очевидно.

4. Рисование

Graphics2D g = (Graphics2D) g1;  
BasicStroke pen1 = new BasicStroke(5);  
g.setStroke(pen1);  
g.setColor(Color.*WHITE*);

g.setColor(new Color(8, 215, 207));(выбрали цвет пера)  
drawCircle(g, line.c1);  
drawCircle(g, line.c2);(нарисовали окружности)  
g.setColor(new Color(0, 255, 0));(снова выбрали цвет)  
g.drawLine((int) line.p1.x, (int) line.p1.y, (int) line.p2.x, (int) line.p2.y); (нарисовали отрезок по координатам его концов)

Обьекты нарисованы с помощью drawCircle(для как окружностей и точек), drawLine (для отрезков)



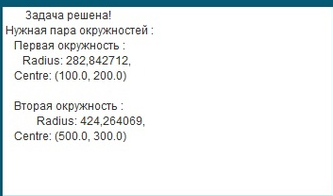
# 5. Решение задачи

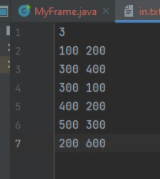
Само решение задачи , вычислительный процесс описаны в отдельном классе Solution.java

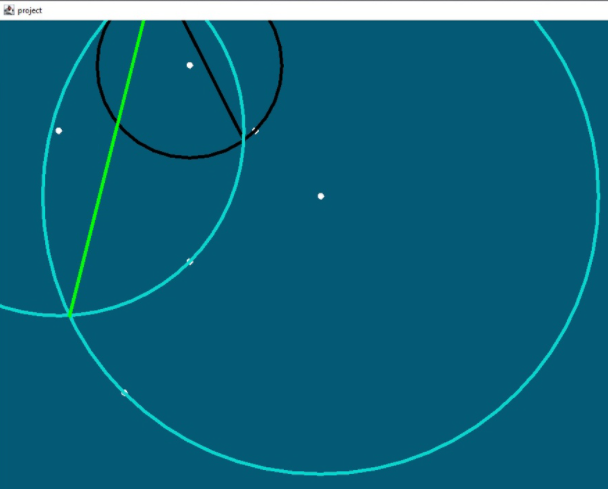
Мы рассматриваем окружности попарно. Сначала метод hasIntersection, принимая на вход 2 окружности , возвращает нам тру или фолз (пересекаются ли они), далее метод findIntersections получает на вход так же 2 окружности и возвращает две/одну(/может и ноль) точек их пересечения. Потом по полученным точкам пересечения в лист Lines помещаются отрезки (общие хорды окружностей). Далее с помощью findMaxLength, просматривая лист, находим отрезок максимальной длины, он и будет искомым.

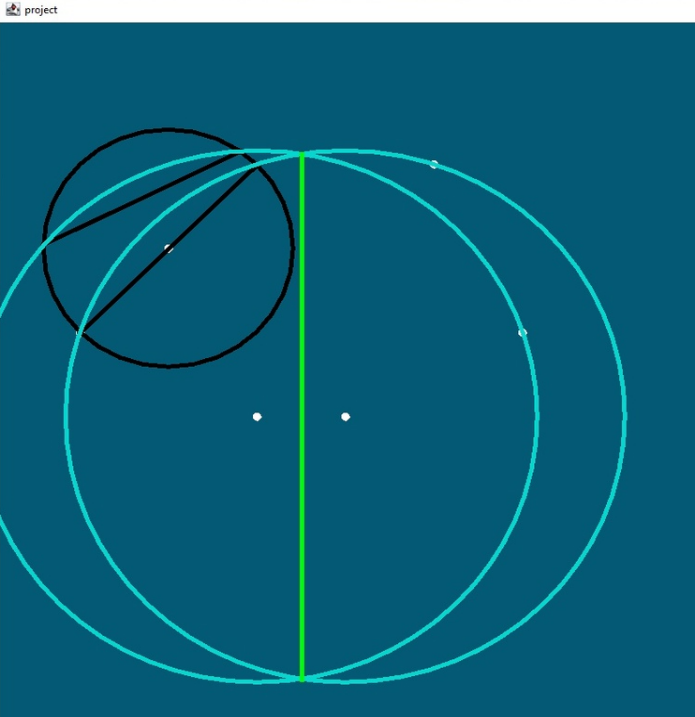
# 6. Проверка

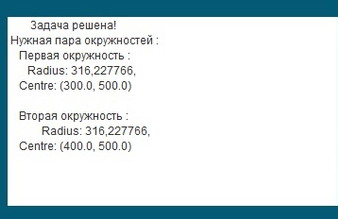
Для проверки правильности решённой задачи были проведены опыты для 3х решений задачи. Тесты: (проверка приведена в заключении)

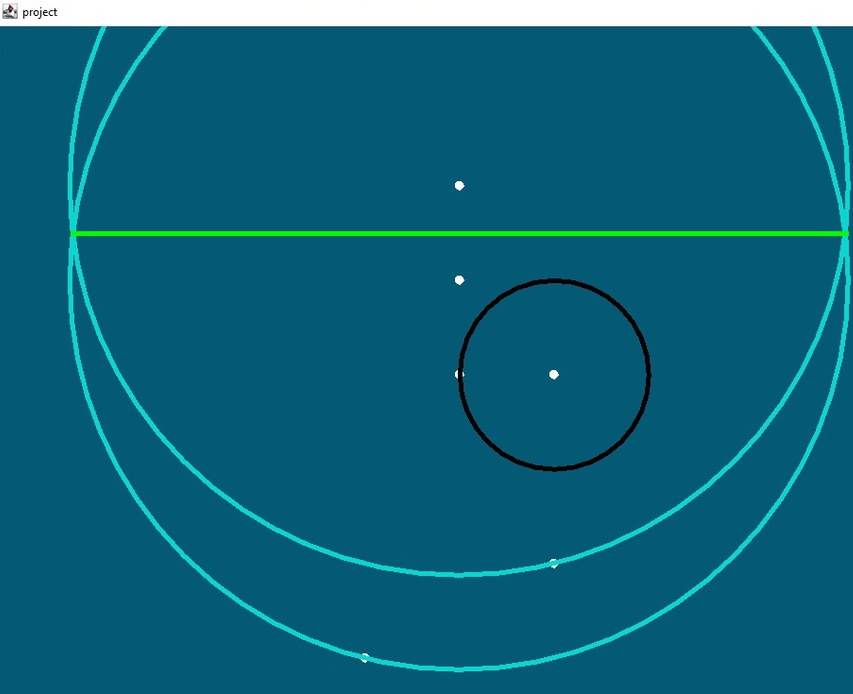
Тест 1

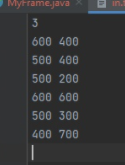


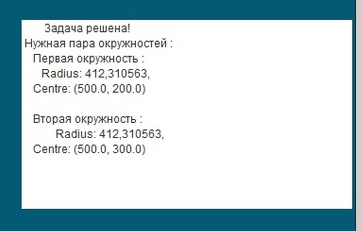


Тест 2



Тест 3



 7. Заключение

В рамках выполнения поставленной задачи было создано графическое приложение с необходимым по условию функционалом. Правильность решения задачи проверена вручную, в результате ошибок в вычислениях программы не выявлено.

# 

