Физико-математический лицей N239

Нахождение пары «широкий луч»-окружность с максимальной площадью пересечения

Отчёт о проекте по ИКТ

Работу выполнил

Ученик 10-1 класса

Алексеев Григорий

Санкт-Петербург

2021

1. Постановка задачи

На плоскости задано множество «широких лучей» и множество окружностей. Найти такую пару «широкий луч»-окружность, что фигура, находящаяся внутри «широкого луча» и окружности, имеет максимальную площадь.

2. Уточнение исходных и выходных данных и ограничения на них

2.1 Исходные данные

Окружности и «широкие лучи» генерируются случайно или задаются пользователем. Информацию о них программа получает в виде координат. Они хранятся соответственно в файлах circles.txt и widebeams.txt. Каждая фигура задаётся 3 координатами.

2.2 Выходные данные

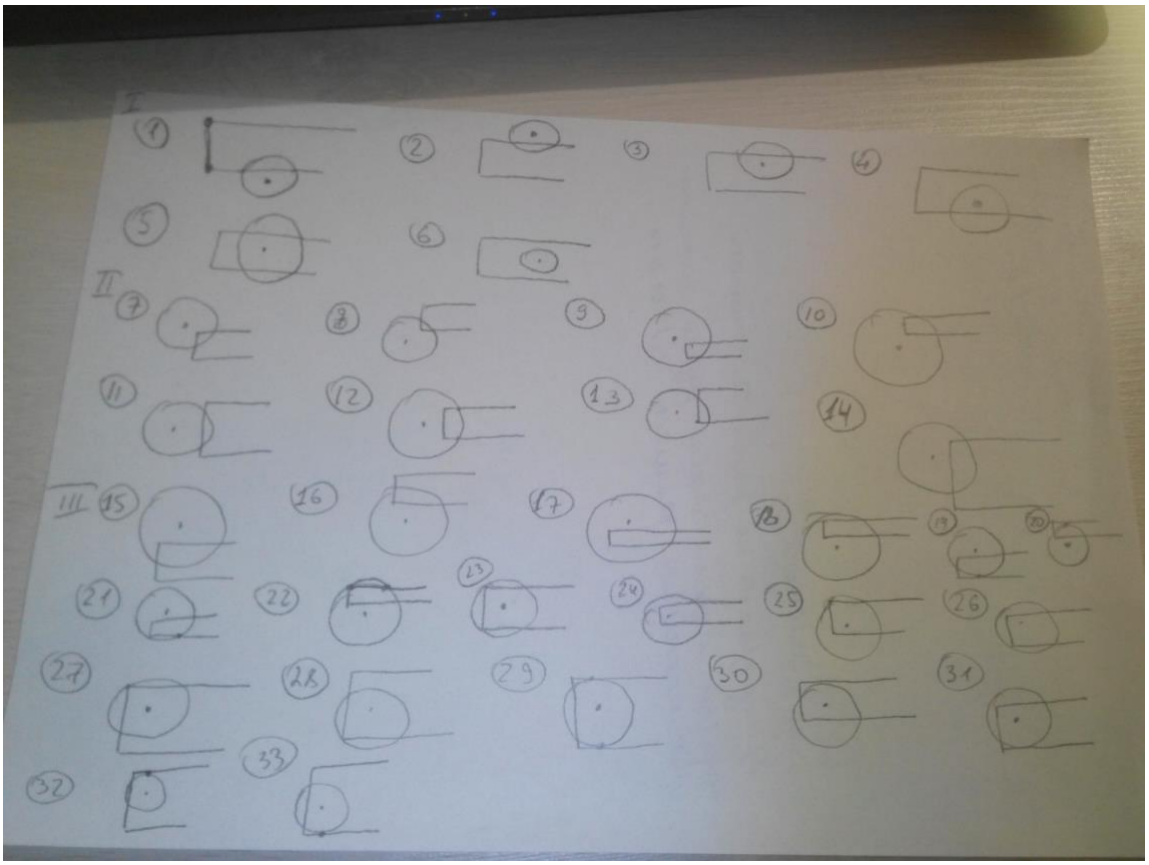
Необходимо изобразить окружности и «широкие лучи» на экране, выделить нужную пару по контуру, выделить получившуюся фигуру.

3. Выбор метода решения

3.1 Анализ исходных данных и выбор используемой структуры данных

В задаче анализируются координаты в виде вещественных чисел, так как я использую тригонометрические формулы и другие сложные математические операции, а также работаю на малой координатной плоскости. Данные хранятся в массивах, для окружностей и «широких лучей» соответственно. Каждой фигуре соответствует ячейка памяти

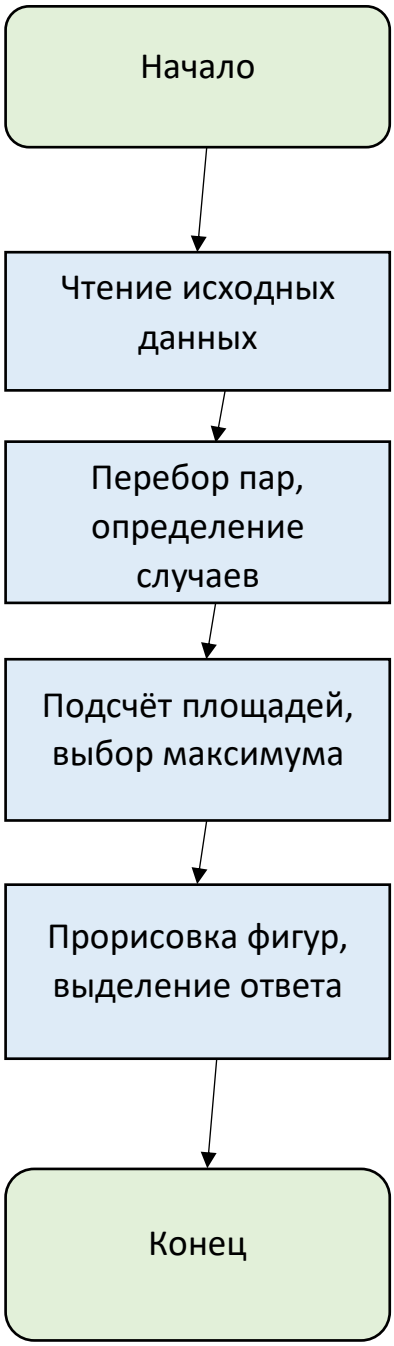
3.2 Выбор метода решения



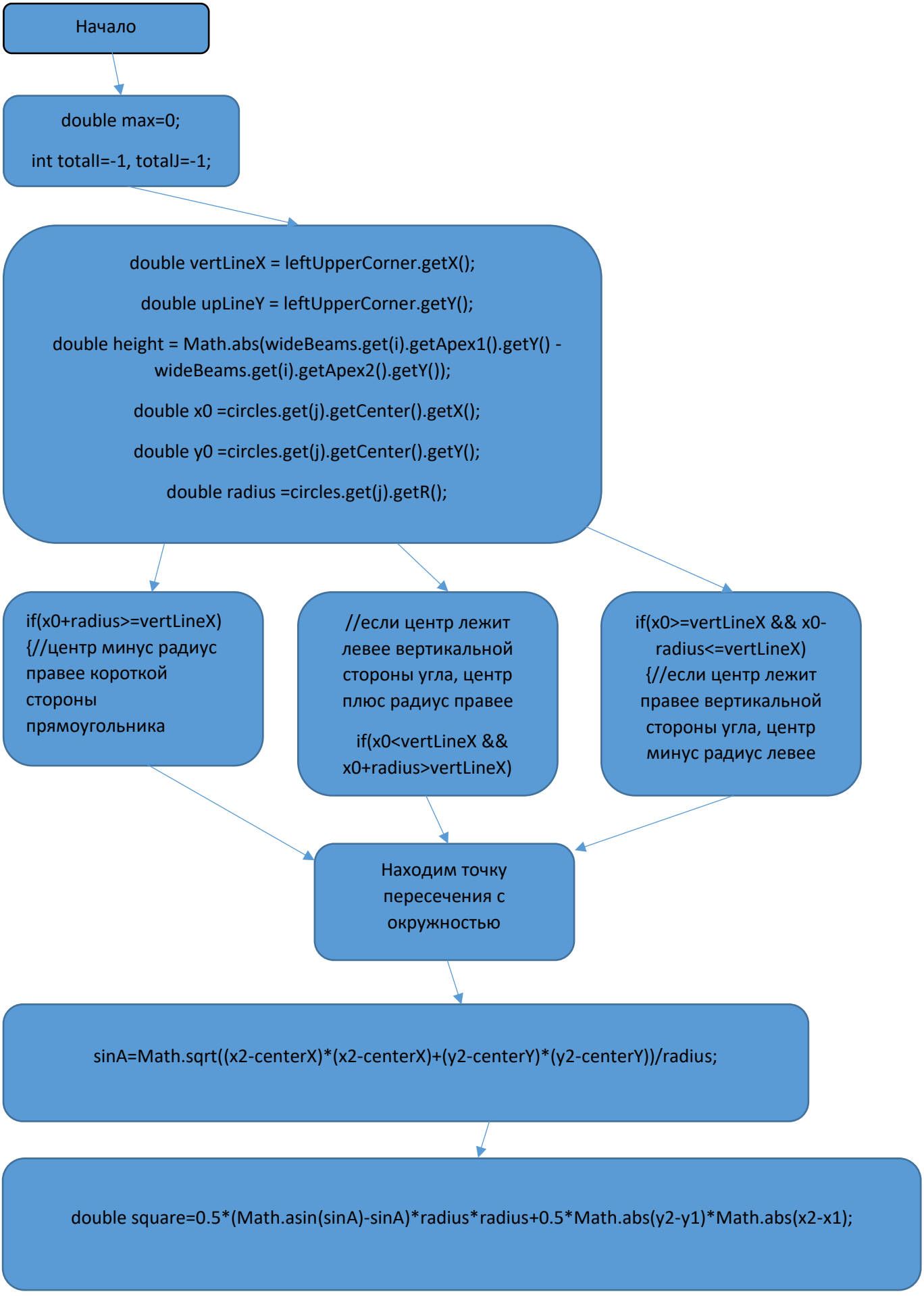
Всего есть 33 варианта взаимного расположения окружности и «широкого луча». Мы считываем координаты, определяем для каждой пары нужный нам случай, получившуюся фигуру делим на простые части, площади которых умеем считать, считаем их. Из этих площадей выбираем наибольшую, рисуем окружности и «широкие лучи», выделяем то, что получилось.

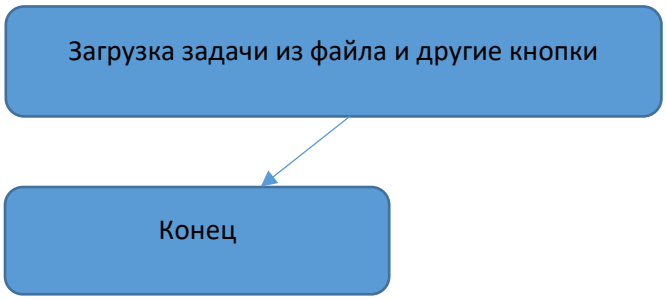
4. Составление алгоритма

4.1 Обобщенная блок-схема алгоритма



4.2 Блок-схема алгоритма




Загрузка задачи из файла и другие кнопки
Конец

5. Листинг программы

```
public static final String PROBLEM_CAPTION = "Итоговый проект
ученика 10-1 Алексеева Григория";

//Заголовок окна

private static final String FILE_NAME_CIRCLE = "circles.txt";

 private static final String FILE_NAME_WIDEBEAM =
"widebeams.txt";

//Путь к файлам

ArrayList<Circle> circles;

 ArrayList<WideBeam> wideBeams;


 public Problem() {

  circles=new ArrayList<>();

  wideBeams = new ArrayList<>();

 }

//Список точек


if(x0+radius>=vertLineX) {//центр минус радиус правее короткой
стороны прямоугольника+


if(x0>=vertLineX && x0-radius<=vertLineX) {//если центр лежит
правее вертикальной стороны угла, центр минус радиус левее


// центр лежит левее вертикальной стороны угла, центр плюс
радиус правее

  if(x0<vertLineX && x0+radius>vertLineX)


//находим точки персечения с окружностью

  if(x1>x0 && y1<y0) {

   x2=x0+Math.sqrt(radius*radius-(y1-y0)*(y1-y0));

   y2=y0+Math.sqrt(radius*radius-(x1-x0)*(x1-x0));

  }
```

```
  if(x1<x0 && y1<y0) {
   x2=x0-Math.sqrt(radius*radius-(y1-y0)*(y1-y0));
   y2=y0+Math.sqrt(radius*radius-(x1-x0)*(x1-x0));
  }

  if(x1<x0 && y1>y0 ) {
   x2=x0-Math.sqrt(radius*radius-(y1-y0)*(y1-y0));
   y2=y0-Math.sqrt(radius*radius-(x1-x0)*(x1-x0));
  }

  if(x1>x0 && y1>y0) {
   x2=x0+Math.sqrt(radius*radius-(y1-y0)*(y1-y0));
   y2=y0-Math.sqrt(radius*radius-(x1-x0)*(x1-x0));
  }
  double centerX=(x2+x1)/2;
  double centerY=(y2+y1)/2;
  double sinA=0;
  sinA=Math.sqrt((x2-centerX)*(x2-centerX)+(y2-centerY)*(y2-
centerY))/radius;

//площадь сектора будет площадь сегмента плюс площадь
треугольника
  double square=0.5*(Math.asin(sinA)-
sinA)*radius*radius+0.5*Math.abs(y2-y1)*Math.abs(x2-x1);
  return square;
 }

//находим площадь сегмента
  public double squareSegment(/*WideBeam wideBeam,*/ double
radius, double x0, double y0, double rectangleSideCoord, boolean
horizontal ) {

//находим точки пересечения стороны прямоугольника и круга
   double[] array=new double[2];
```

```
  if(horizontal) {
   array[0]= x0+Math.sqrt(radius*radius-(rectangleSideCoord-
y0)*(rectangleSideCoord-y0));
   array[1] = x0-Math.sqrt(radius*radius-(rectangleSideCoord-
y0)*(rectangleSideCoord-y0));
  }
  else {
  array[0]= y0+Math.sqrt(radius*radius-(rectangleSideCoord-
x0)*(rectangleSideCoord-x0));
  array[1]= y0-Math.sqrt(radius*radius-(rectangleSideCoord-
x0)*(rectangleSideCoord-x0));
  }

  double center = (array[0]+array[1])/2;

  double sinA=(array[0]-center)/radius;

  double square=0.5*(Math.asin(sinA)-sinA)*radius*radius;

  return square;
}
```

6. Пример работы программы

6.1 Исходные данные

6.2 Выходные данные

Окружности:

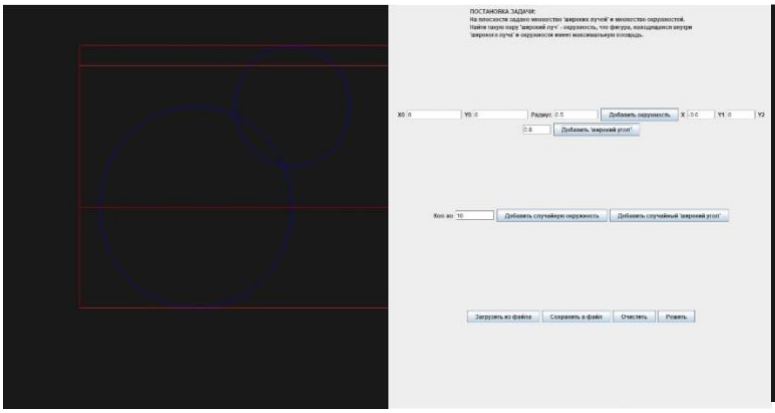
0.0, 0.0, 0.5

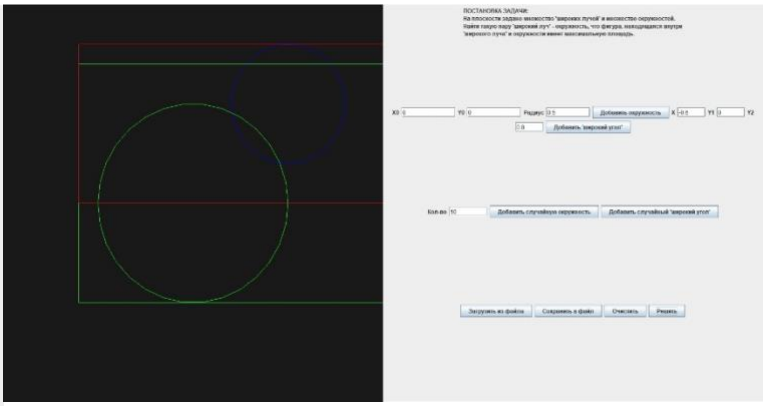
0.5, 0.5, 0.3

Лучи:

-0.6, 0.0, 0.8

-0.6, -0.5, 0.7


ПОСТАНОВКА ЗАДАЧИ:
На плоскости задано множество "вирожих лучей" и множество окружностей.
Найти пакую пару "вирожий луч" - окружность, что фигура, находящаяся внутри
"вирожего луча" и окружности имеет максимальную площадь.
X0 0
Y0 0
Радиус 0.5
Добавить окружность
X 0.0
Y1 0
Y2
0.0
Добавить "вирожий угол"
Кол-во 10
Добавить случайную окружность
Добавить случайный "вирожий угол"
Загрузить из файла
Сохранить в файл
Очистить
Решить


ПОСТАНОВКА ЗАДАЧИ:
На плоскости задано множество "вирожих лучей" и множество окружностей.
Найти пакую пару "вирожий луч" - окружность, что фигура, находящаяся внутри
"вирожего луча" и окружности имеет максимальную площадь.
X0 0
Y0 0
Радиус 0.5
Добавить окружность
X 0.0
Y1 0
Y2
0.0
Добавить "вирожий угол"
Кол-во 10
Добавить случайную окружность
Добавить случайный "вирожий угол"
Загрузить из файла
Сохранить в файл
Очистить
Решить