05.03.14-----------------------------------------------------------------------

Алгоритм Брезенхема:

1. Начало
2. Ввод хн, ун, хк, ук
3. Проверка отрезка на вырожденность: если вырожденный то высветить точку хн,ун и переход на конец
4. Sx=sign(dx), sy=sign(dy)
5. Dx=|dx|, dy=|dy|
6. Если dx>dy то обмен=0, иначе обмен=1 {t=dy; dy=dx; dx=t}
7. . //тангенс угла наклона
8. Цикл построения отрезка (по i=1 to dx+1)
   1. Высвечивание точки (xt,yt)
   2. Если (e>=0) { если (обмен=0) {yt=yt+sy} иначе {xt=xt+sx}; e=e-1 **(2)**} иначе { если (обмен=0) {xt=xt+sx} иначе {уt=уt+sy}; e=e+m **(3)**}
9. Конец

Недостатки: не все переменные являются переменными целого типа (e,m - действительные).

Чтобы перейти к алгоритму, работающему ТОЛЬКО с целыми:

; (1). Тогда в цикле в (2) будет e=e-2dx; в (3) e=e+2dy; не нужно вычислять m в 9м шаге.

Брезенхем предложил простейший алгоритм сглаживания; алгоритм брезенхема с устранением ступенчатости.

Используется при отображении ребёр многоугольника, который закрашивается. Идея состоит в сглаживании резких переходов от ступени к ступени. Сглаживание основывается на том, что каждый пиксель высвечивается со своим уровнем интенсивности. Уровень выбирается пропорционально площади части пикселя. 1 пиксель – квадрат с единичной стороной, а не математическая точка.

Так как интенсивность I~Si площади, то

1) отрезок связан (покрывает) на i шаге с одним пикселем. Обозначим Yi расстояние по вертикали от точки пересечения отрезка с пикселем, до левой нижней границы пикселя. Обозначим тангенс угла наклона отрезка через m, тогда Si=Sпр+Sтр = Yi\*1 + 1\*m/2 = Yi + m/2

2) отрезок покрывает на I шаге два пикселя. Yi – расстояние от нижней границы до пересечения с отрезком. Площадь нижней части S1 = Sпикс – Sтр1 = . Площадь части второго пикселя S2 . Складывая площади, S1+S2 = Yi + m/2

1) На очередном шаге Si = Yi + m/2; S(i+1)=Sпр1 + Sпр2 + Sтр = Si + m

2) S(i+1) = Si + m, но высвечивается верхний пиксель, нижний – не высвечивается, и его площадь не учитывается. S = S+m; если , то выражение корректируется, если , то S=S-1 (вычитаем площадь нижнего пикселя)

В данном случае в качестве ошибки можно рассматривать ei=Si (интенсивность пропорциональна ошибке) – однако её нельзя будет скорректировать через -0.5. За пороговый уровень можно взять другое значение. Обозначим w=1-m, e = e + w = m-0.5 + 1-m = ½. I=Imax/2 – начальный пиксель всегда высвечивается в половинной интенсивности.

W – пороговое значение. Если e>=w, то (e=e-w), иначе Y(i+1)=Yi.  
Чтобы внутри цикла не приходилось постоянно умножать на интенсивность, сделать это можно один раз в начале работы алгоритма (m=Imax\*m, e=Imax\*e, w=Imax-m).

При реализации алгоритма без сглаживания исходные данные – начальные и конечные координаты. Здесь же добавится количество уровней интенсивности, либо же максимальный уровень.

**Алгоритм Брезенхема для построения окружности**

Будем считать, что центр находится в начале координат, хс=ус=0. У направлена вверх, Х вправо.

Окружность – симметричная фигура, можно построить половину или четверть и отражать и поворачивать. Y=A(x) (отрисовка четвертькруга от (0,R) до (R,0)) – в первой четверти монотонно убывающая. На очередном шаге возможен переход из (xi,yi) в (xi+1,yi), (xi,yi-1), (xi+1,yi-1).

Критерий – местный аналог ошибки. Разность квадратов (расстояния от центра окружности до диагонального пикселя) и (расстояния от центра окружности до самой окружности).  
Если >0, то диагональный пиксель лежит вне окружности. Выбирается диагональный или вертикальный пиксель для отрисовки.  
Если =0, то диагональный пиксель лежит ровно на окружности. Выбирается диагональный.  
Если <0, то диагональный пиксель лежит внутри окружности. Выбирается диагональный или горизонтальный.