Тема 3:

Алгоритмы компьютерной графики

План:

3.1. Общая характеристика алгоритмов машинной графики

3.2. Преобразования координат

3.3. Методы растрирования в компьютерной графике

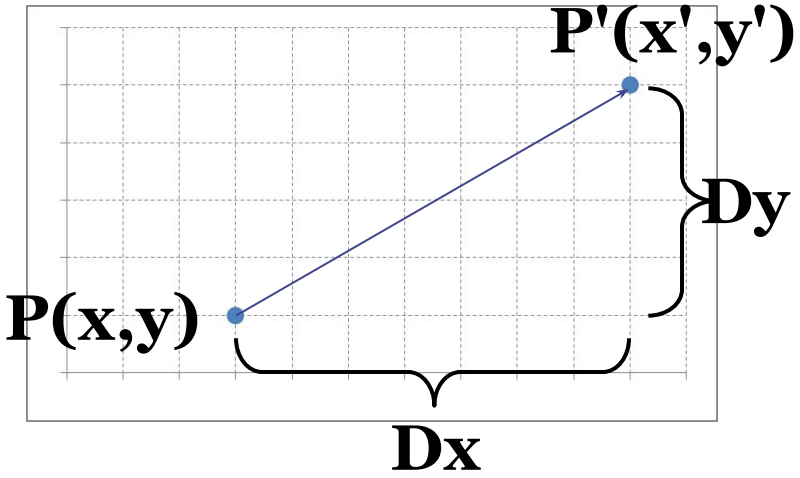
3.4. Закрашивание фигур

3.5. Удаление невидимых линий

3.6. Триангуляция

Перенос

$P(x, y) \rightarrow P'(x',y')$



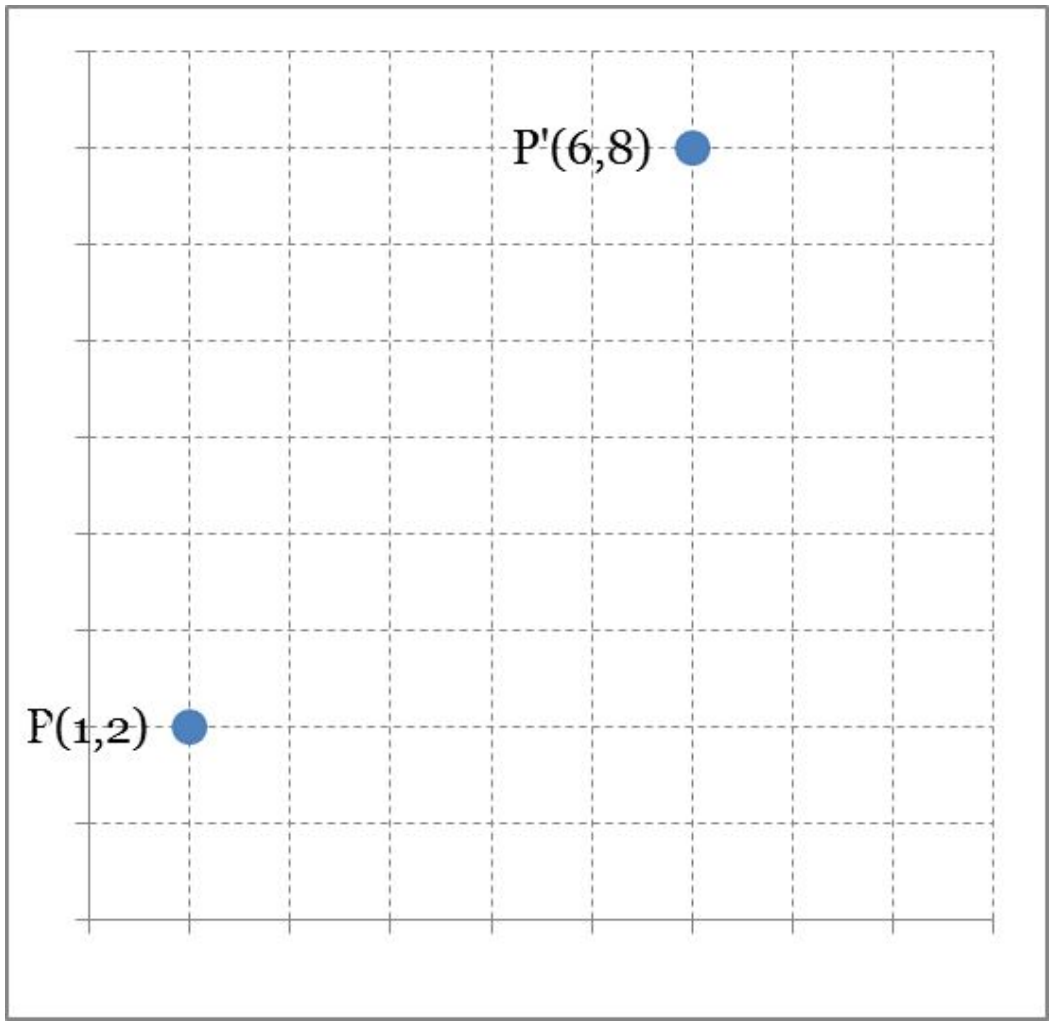
$x'=x+Dx$

$y'=y+Dy$

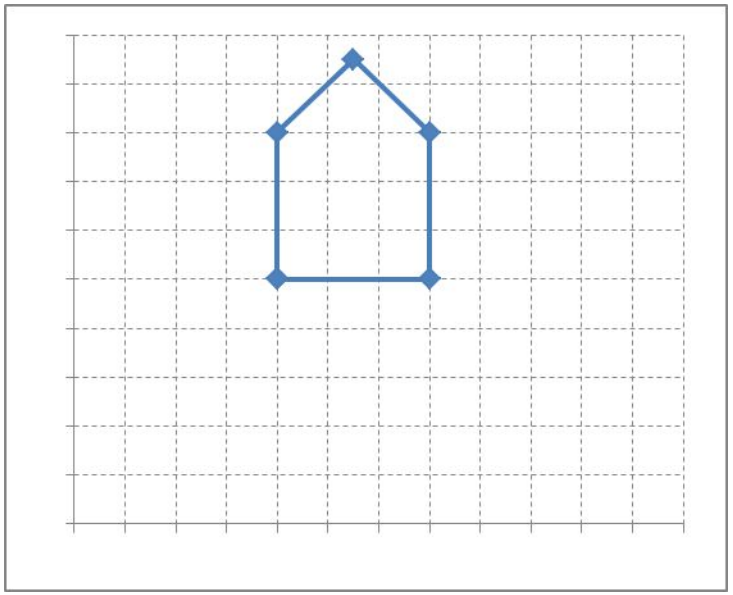
$[x'\ y']=[x\ y] + [Dx\ Dy]$

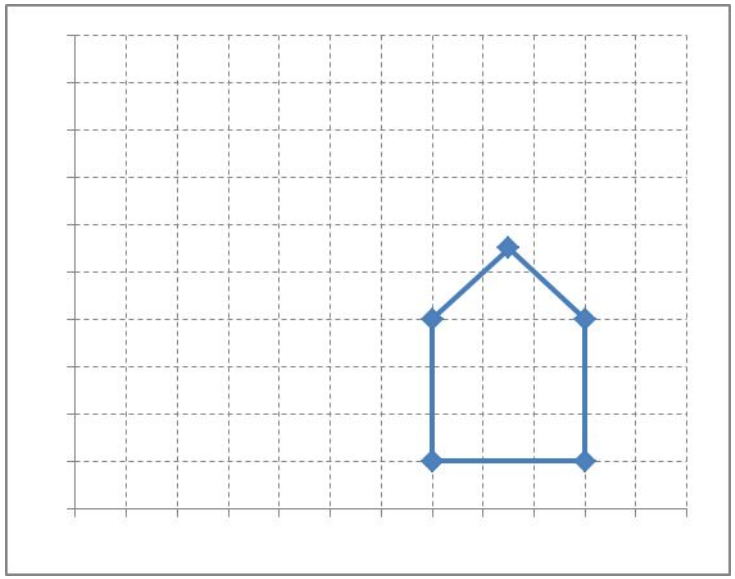
$P'=P+D$

Пример:



Пример:





Перенос контура домика на расстояние (3, -4)

Масштабирование

S_x раз вдоль оси Ox $x'=S_x \cdot x$

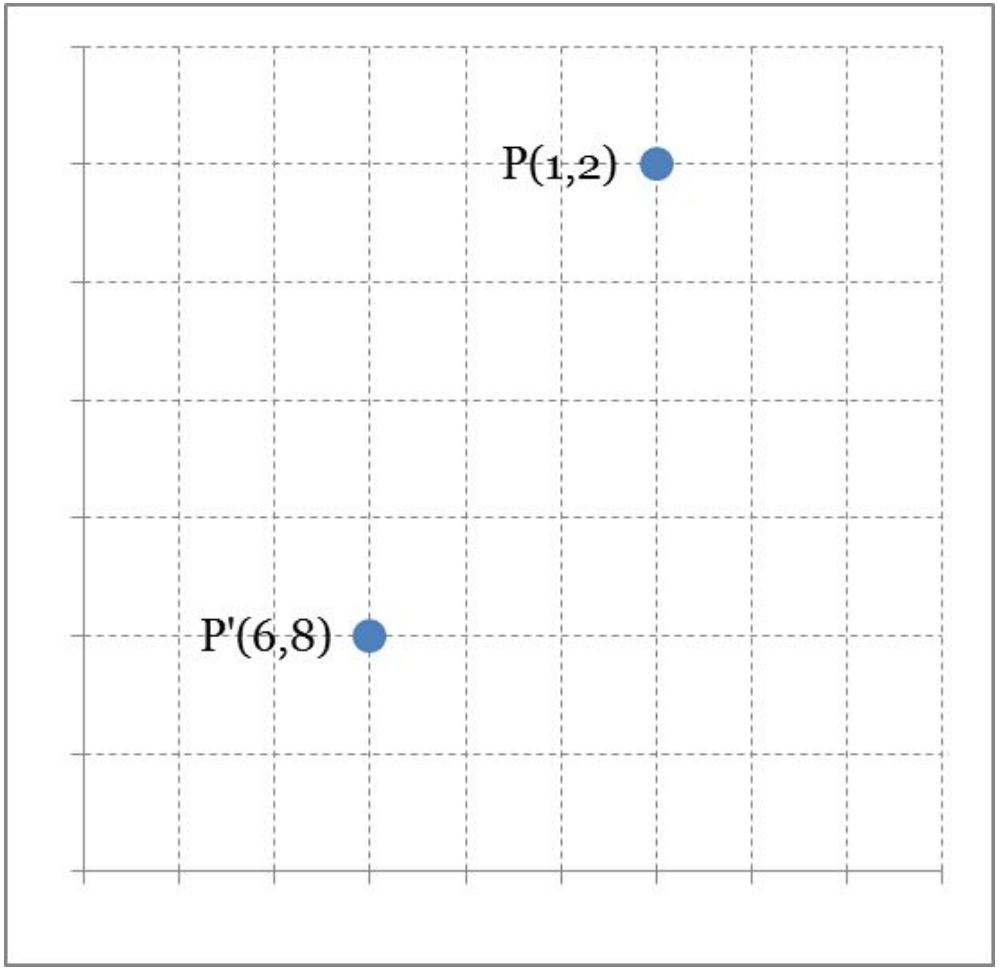
S_y раз вдоль оси Oy $y'=S_y \cdot y$

$$S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

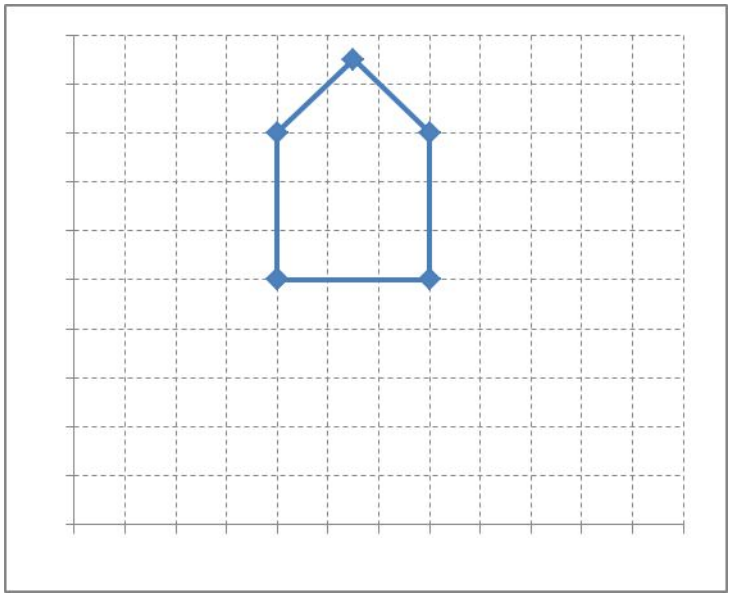
$$[x'y'] = [x\ y] \cdot \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

$$P'=P \cdot S$$

Пример:



Пример:



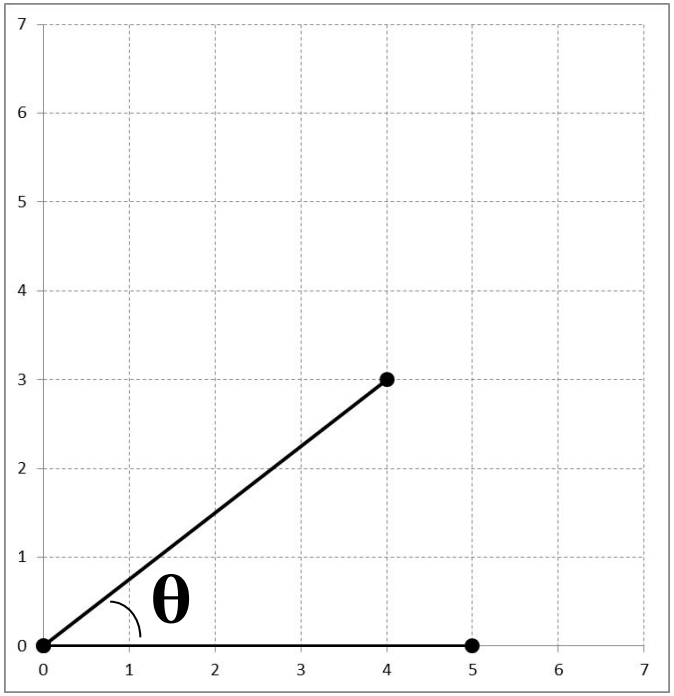
Масштабирование контура домика:

- **по оси Ox на (1/2);**
- **по оси Oy на (2/3).**

Поворот

$x' = x \cdot \cos\theta - y \cdot \sin\theta$

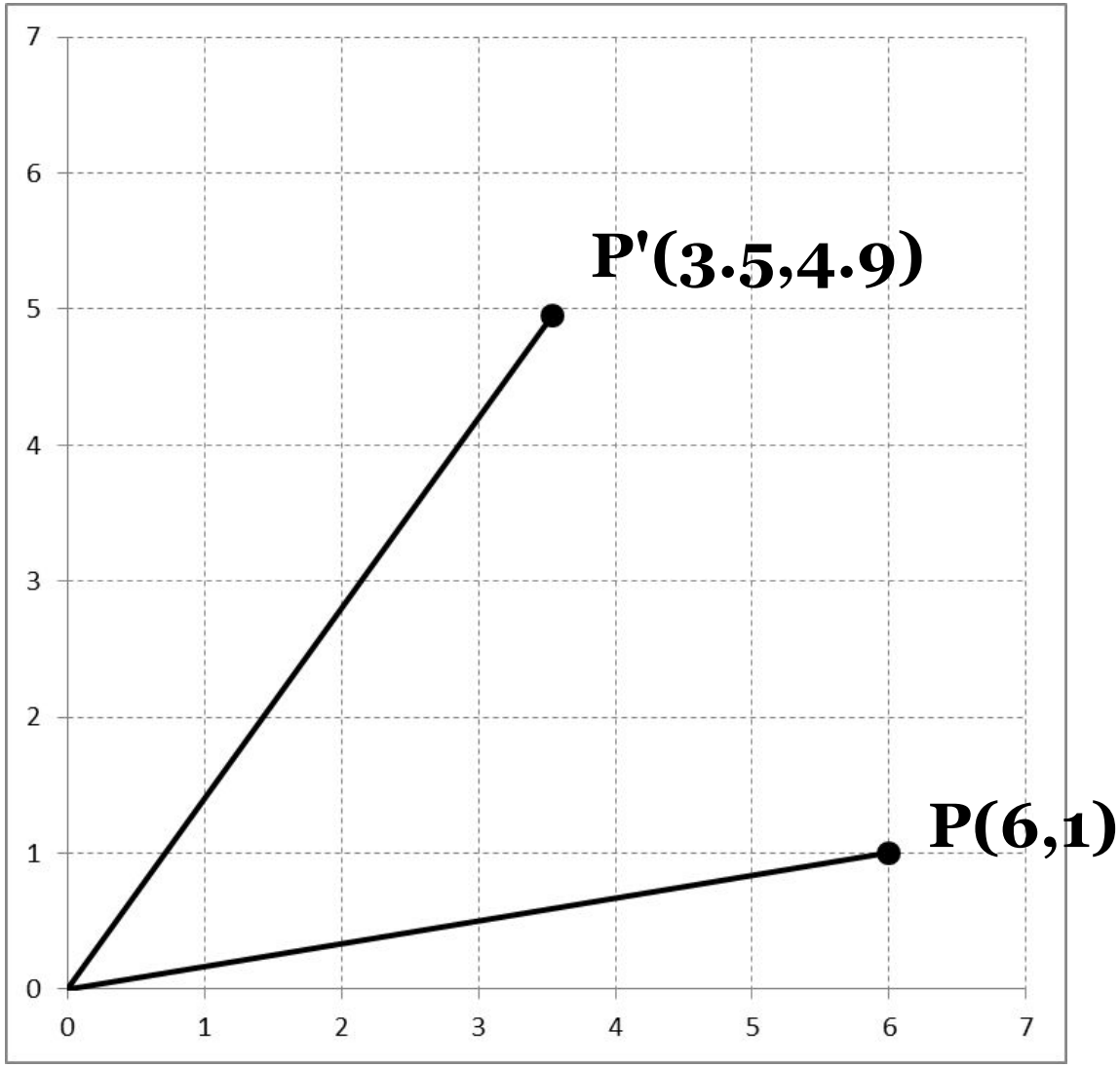
$y' = x \cdot \sin\theta + y \cdot \cos\theta$



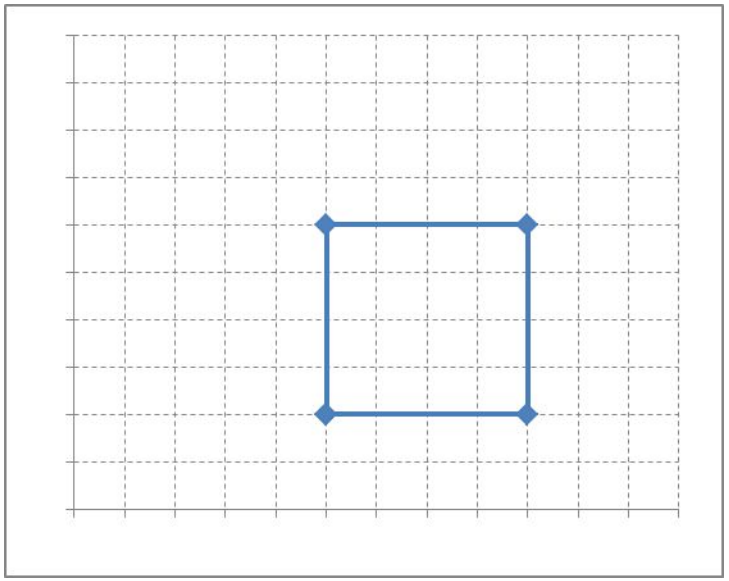
$$[x' \quad y'] = [x \quad y] \times \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \leftarrow \mathbf{R}$$

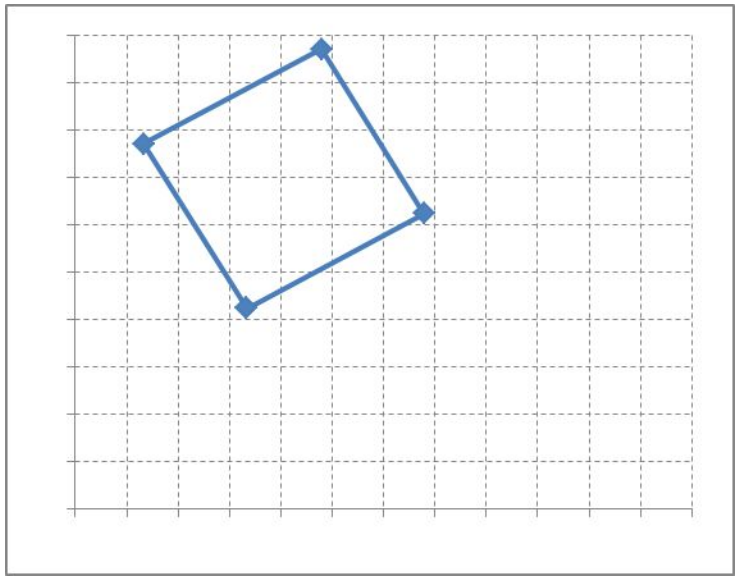
$$\boldsymbol{P}' = \boldsymbol{P} \cdot \boldsymbol{R}$$

Пример:



Пример:





Поворот квадрата на угол 30°

Преобразования в матричной форме:

$P' = P + D$

$P' = P \cdot S$

$P' = P \cdot R$

В однородных координатах

$P(x,\ y) \rightarrow P(W \cdot x, W \cdot y,\ W)$

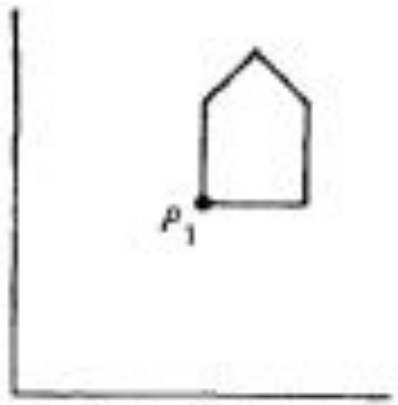
$P(X, Y, W) - x = X/W\ y = Y/W$

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dx & Dy & 1 \end{bmatrix}$$

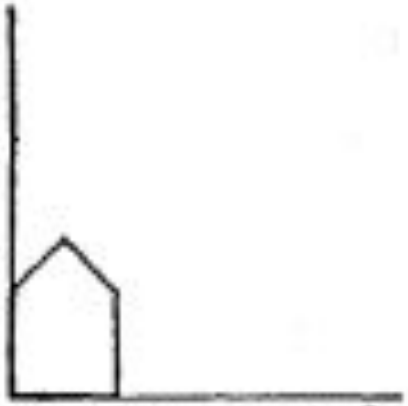
или

$$P' = P \times D$$

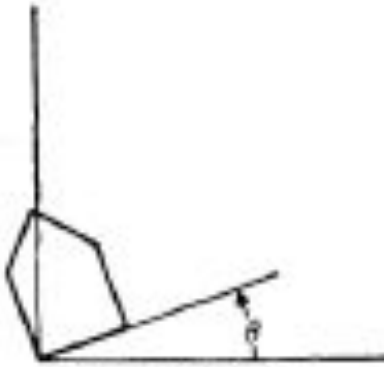
Пример композиции



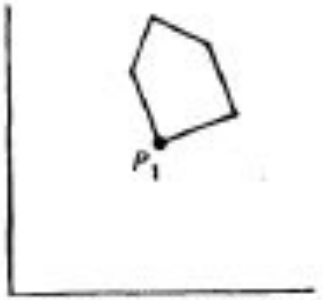
Первоначальный домик



После переноса точки P_1 в начало координат



После поворота



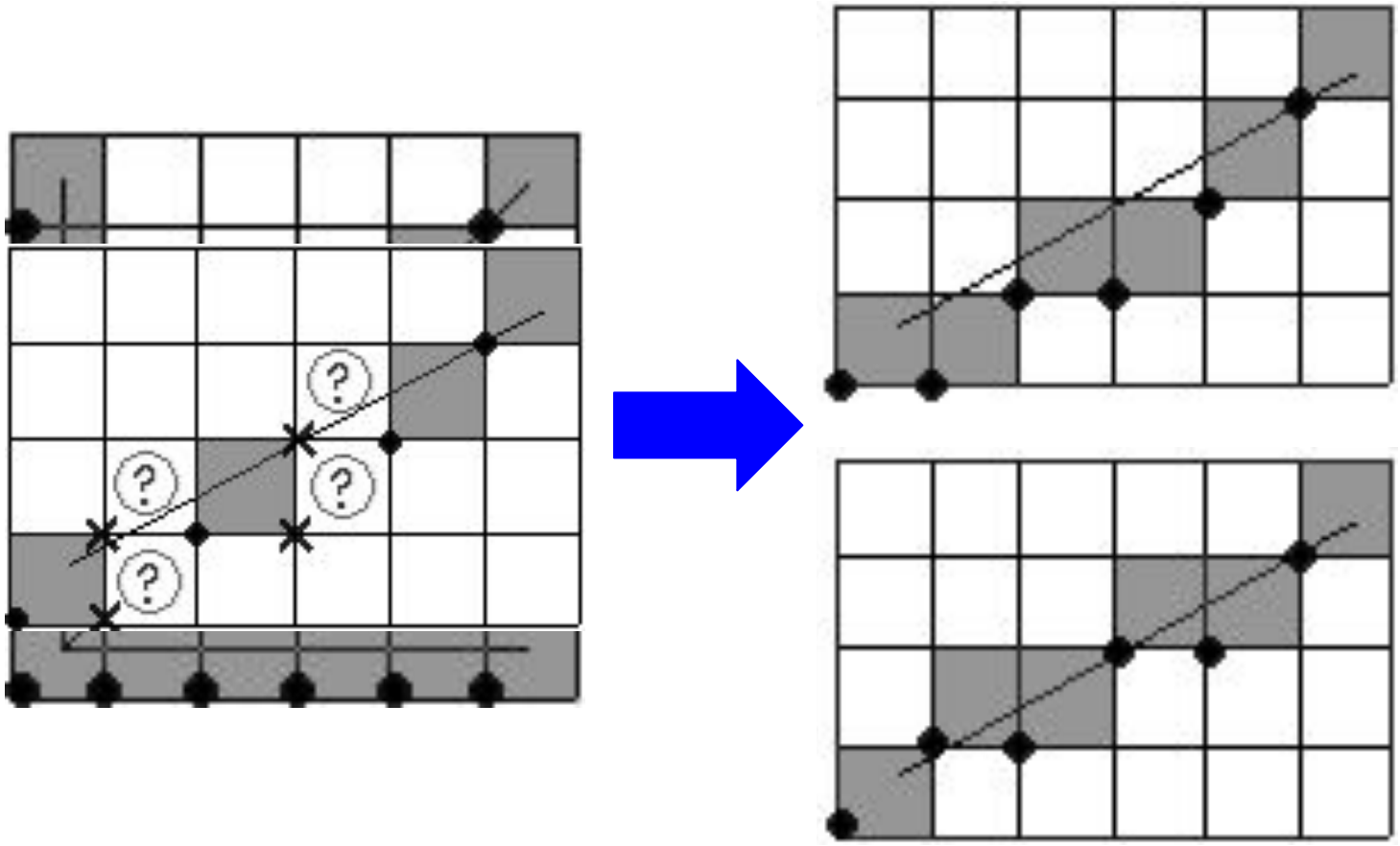
После переноса из начала координат в точку P_1

Пример:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_1 & -y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_1 & y_1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ x_1(1-cos\theta)+y_1 sin\theta & y_1(1-cos\theta)-x_1 sin\theta & 1 \end{bmatrix}$$

Растрирование прямых

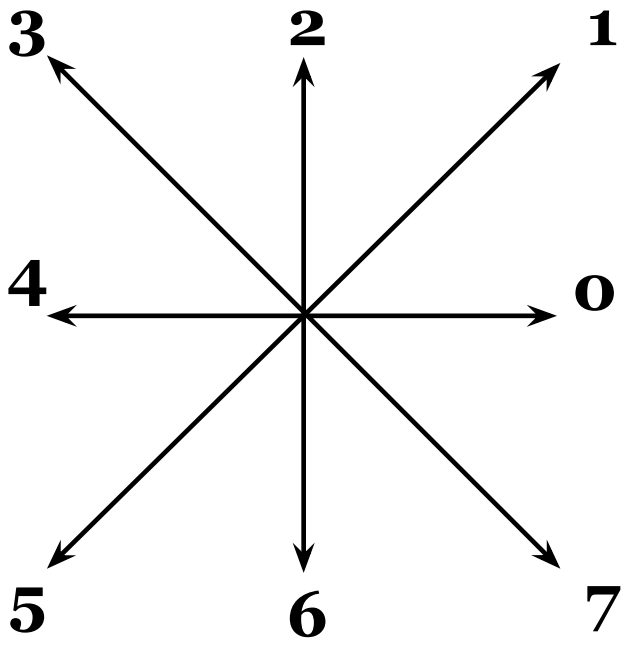


Алгоритмы растрирования прямой

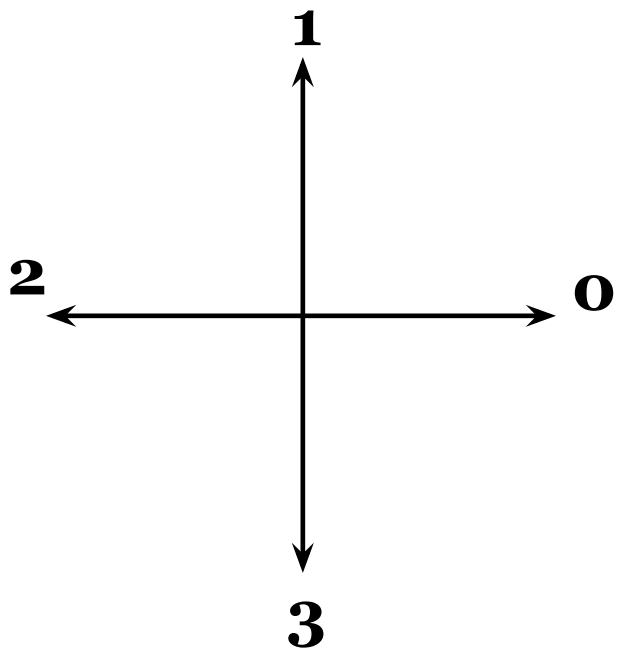
- алгоритм цифрового дифференциального анализатора (ЦДА);
- алгоритм Брезенхема.

Схемы цепочного кодирования:

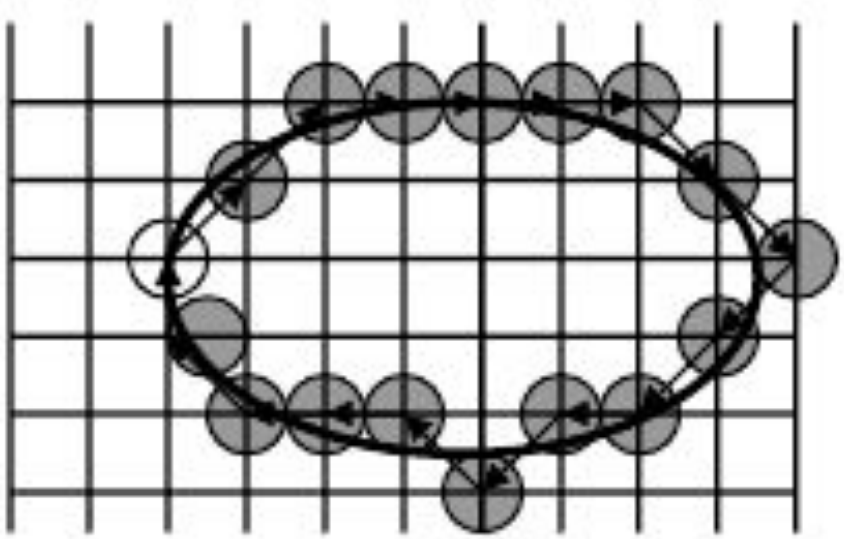
8-точечная



4-точечная



Растрирование эллипса



Цепочный код растрирования эллипса:

<1100007755453443 3> = $<1^2 0^4 7^2 5^2 4534^2 3^2>$

Алгоритм ЦДА

(DDA – Digital Differential Analyzer)

Алгоритм ЦДА

Основные расчетные формулы:

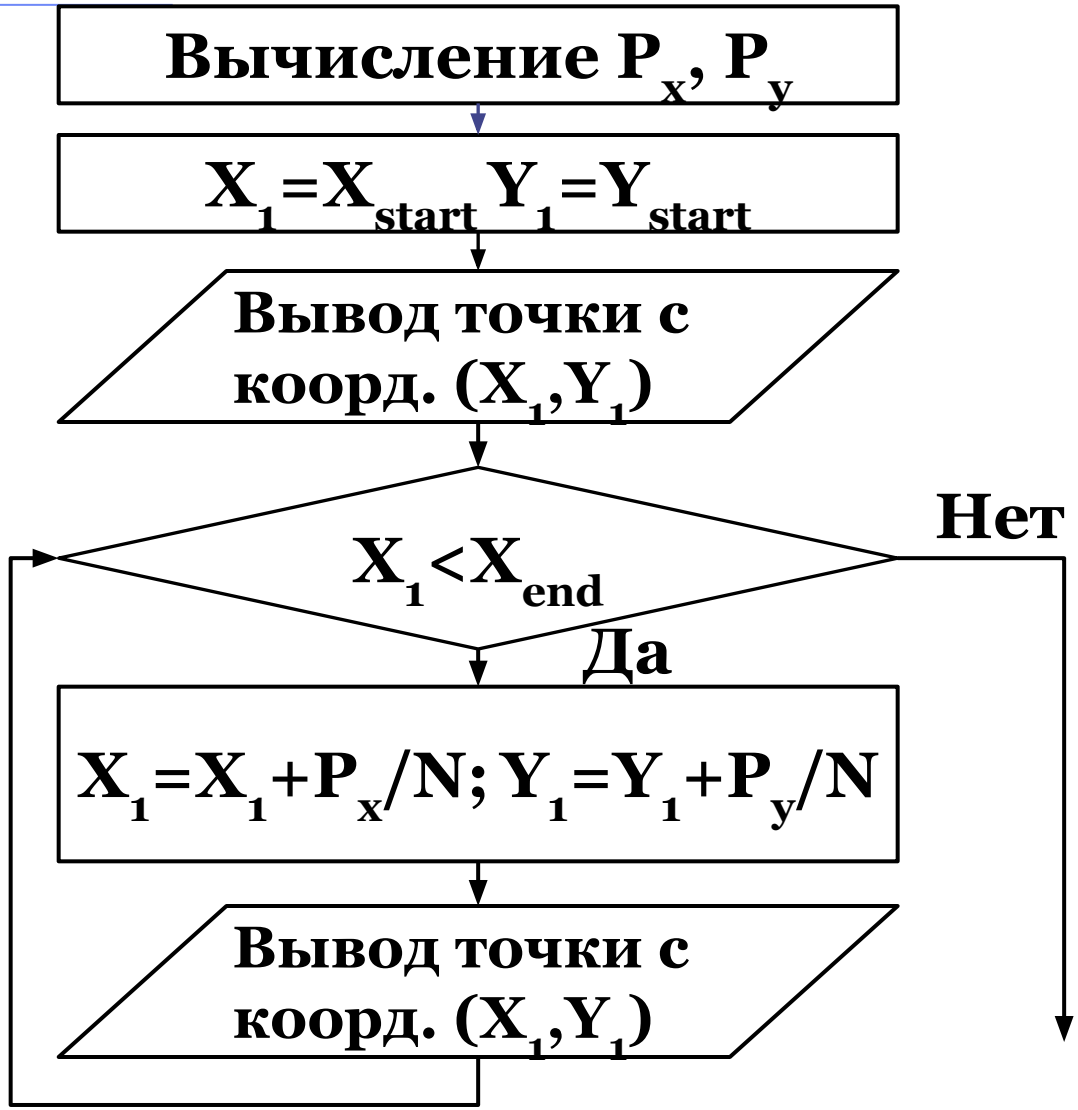
$$X_{i+1}=X_i+P_x$$

$$Y_{i+1}=Y_i+P_y$$

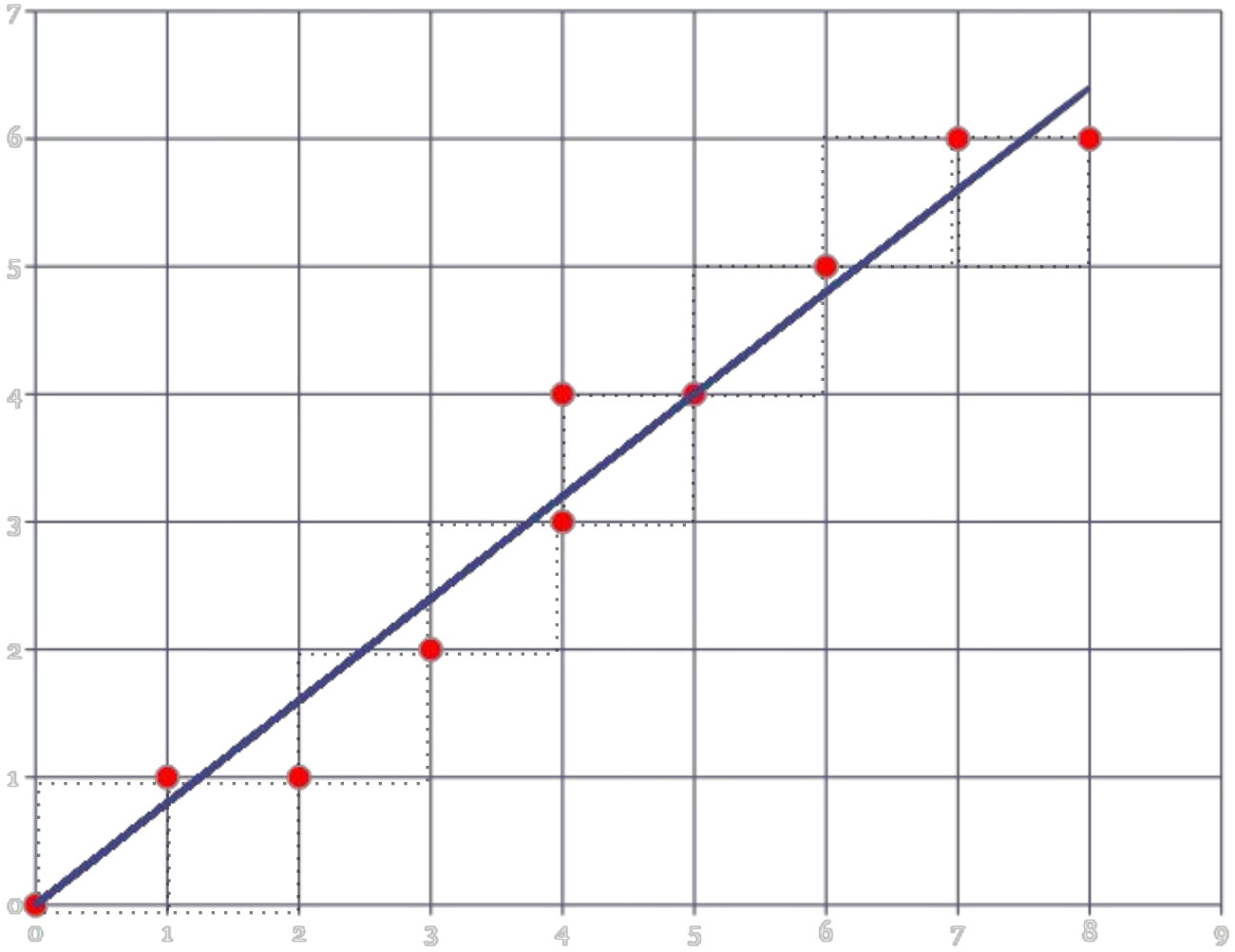
где $P_y = Y_{end}-Y_{start}$ – приращение координат отрезка по оси **Y**;

$P_x = X_{end}-X_{start}$ – приращение координат отрезка по оси **X**.

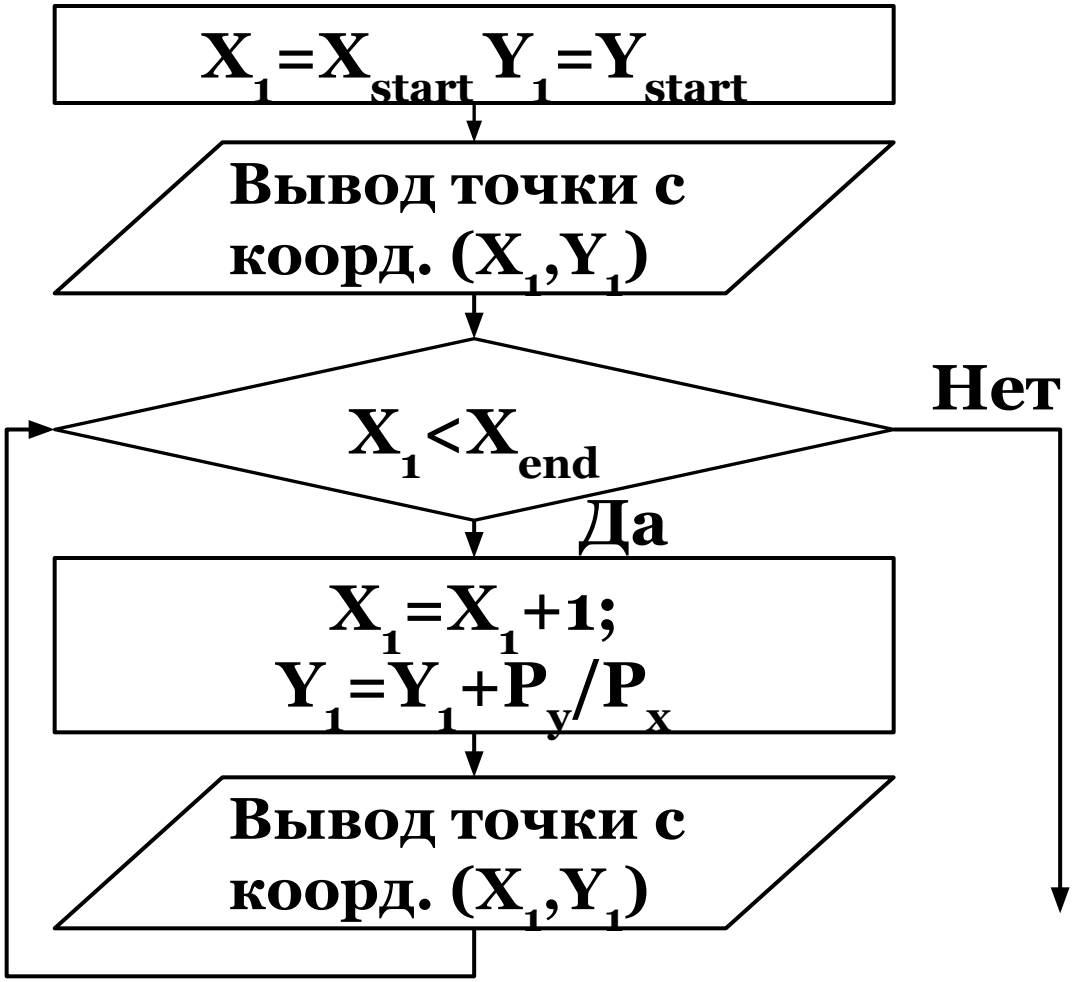
Алгоритм симметричного ЦДА



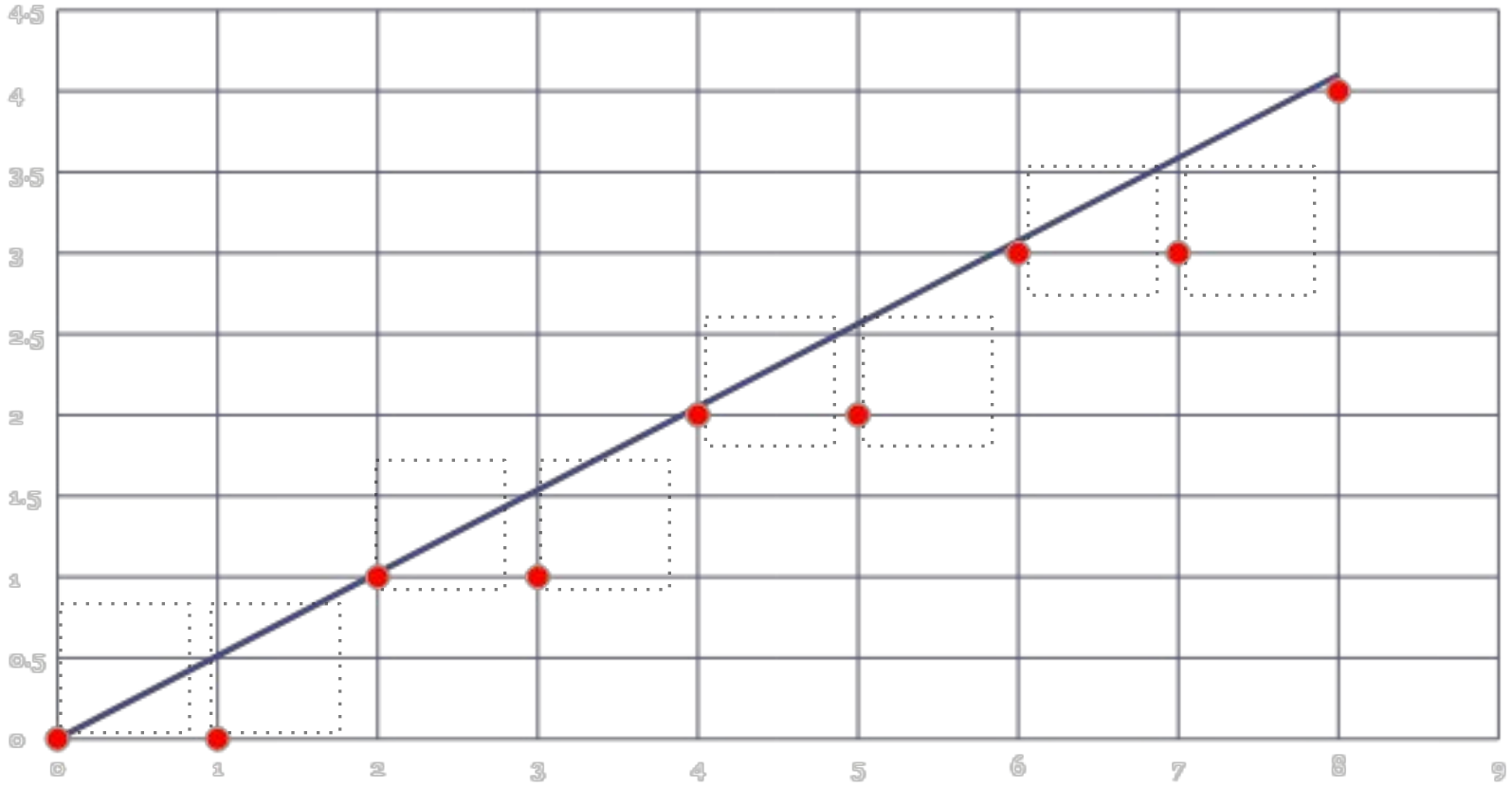
Пример генерации отрезка симметричным ЦДА



Алгоритм несимметричного ЦДА для $P_x > P_y$

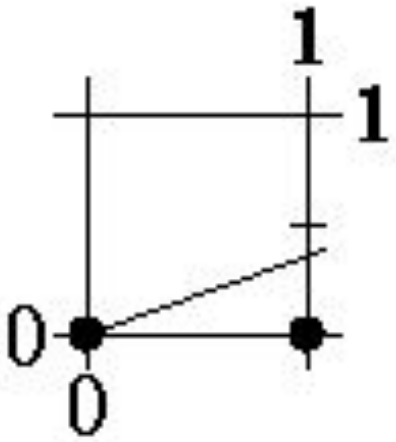


Пример генерации отрезка несимметричным ЦДА

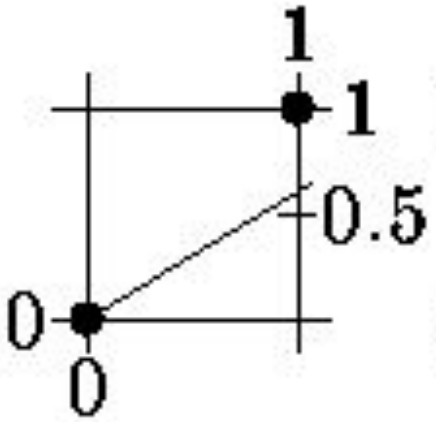


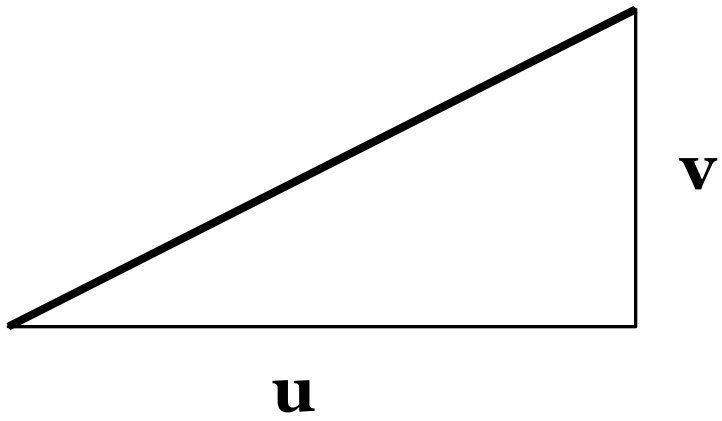
Алгоритм Брезенхема

k < 1/2

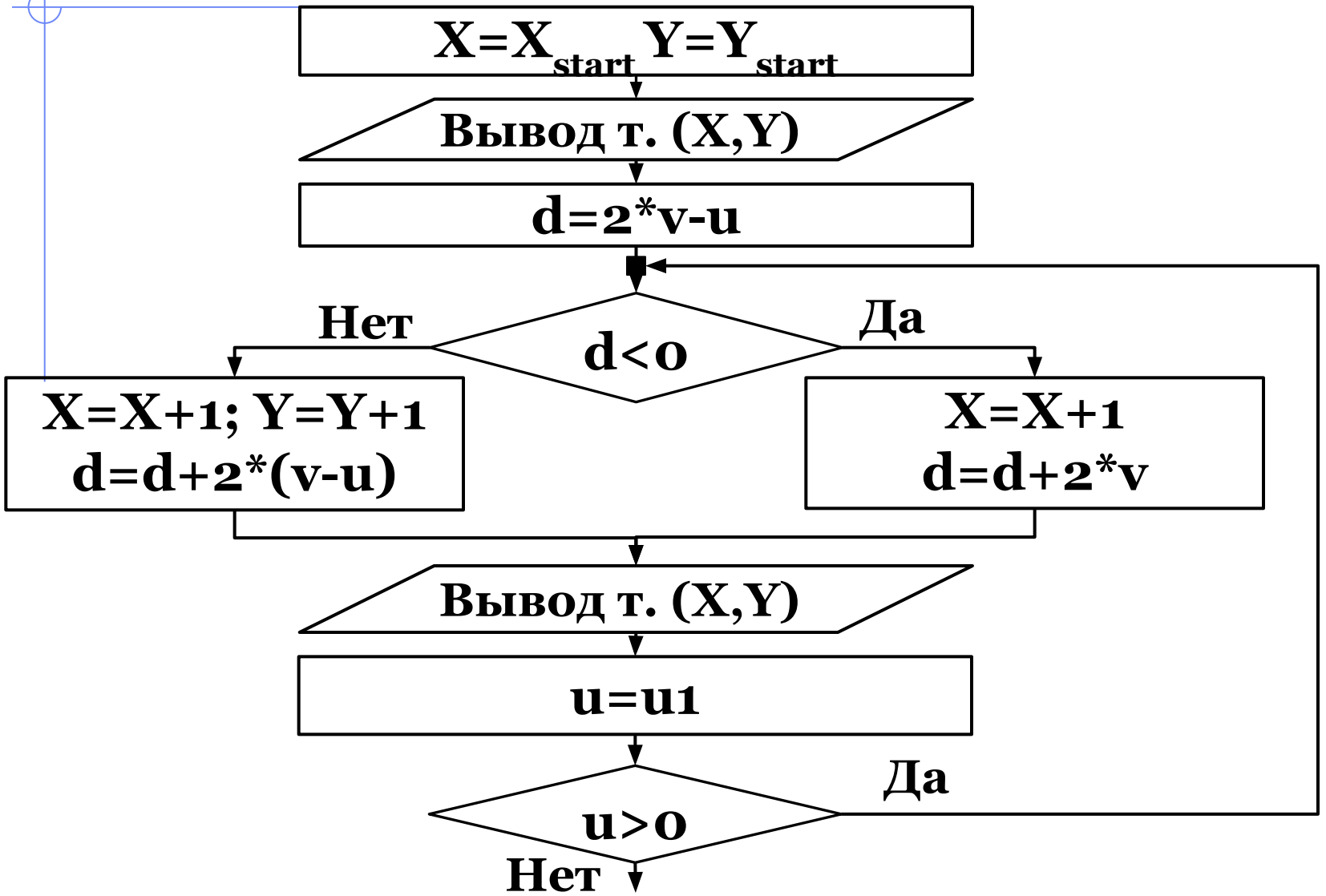


k > 1/2

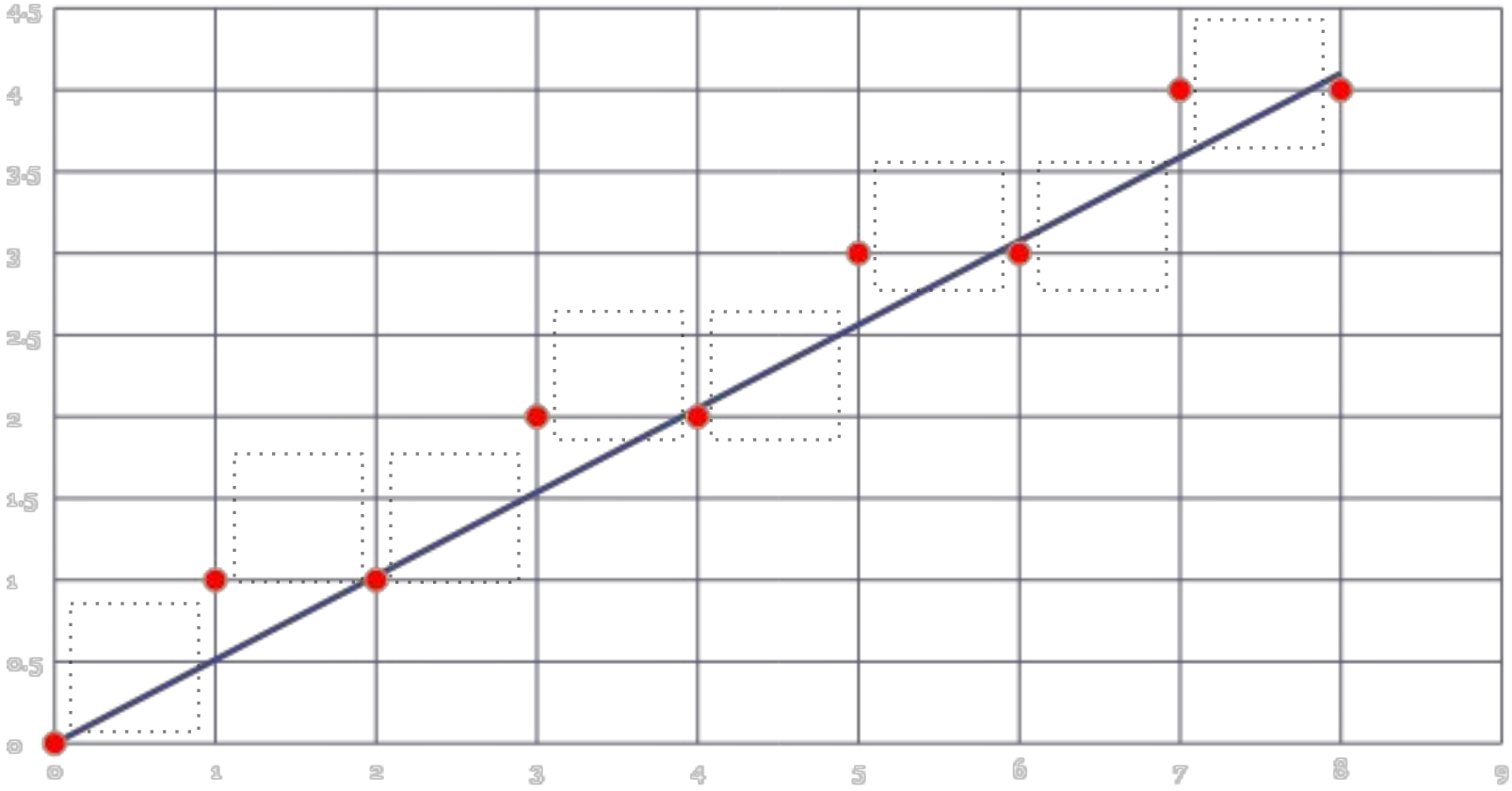




Алгоритм Брезенхема

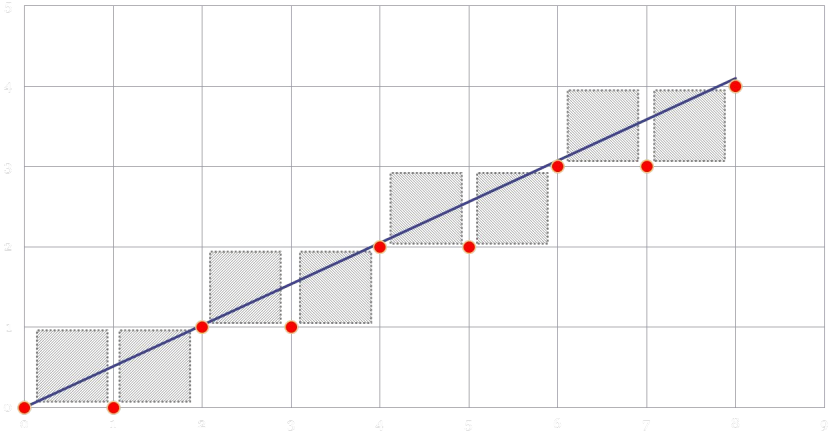


Пример генерации отрезка методом Брезенхема

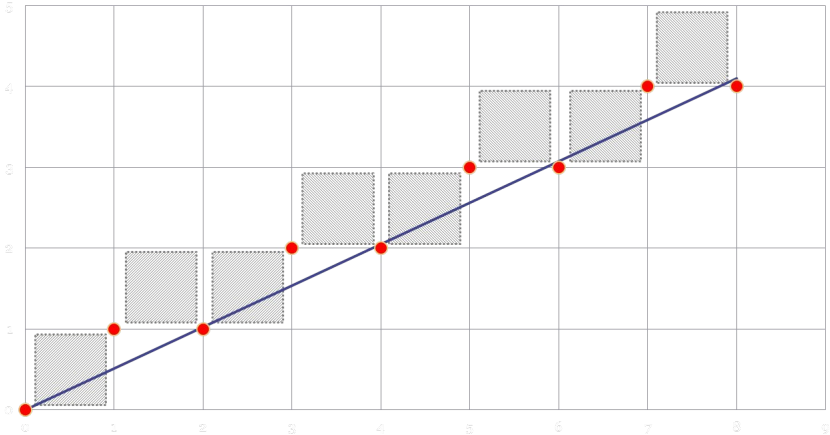


Сравнение примеров для 2-х методов

ЦДА

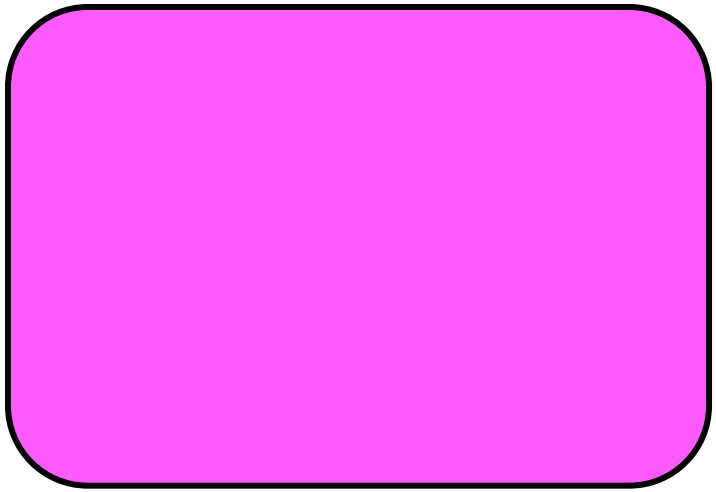


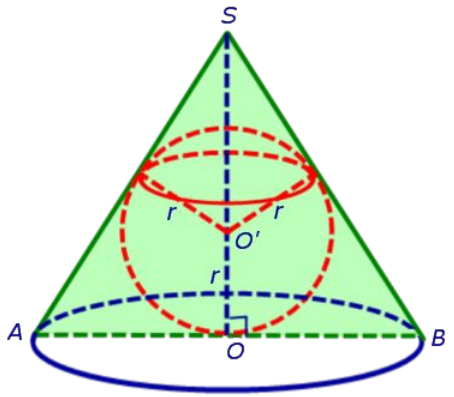
Брезенхема



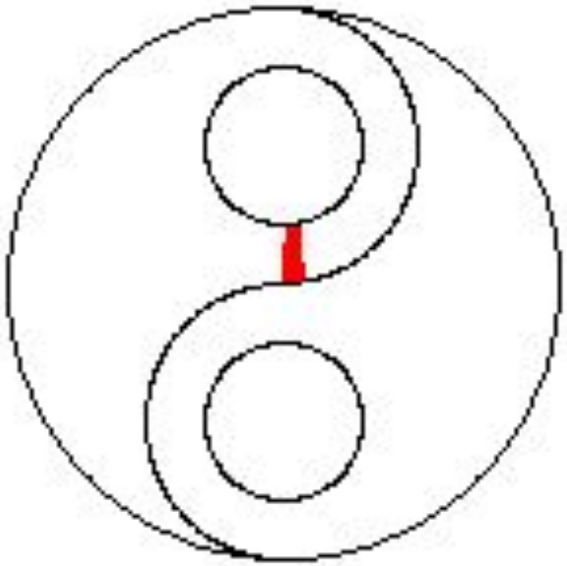
Методы заливки фигур:

- сканирование строк;
- затравочное заполнение



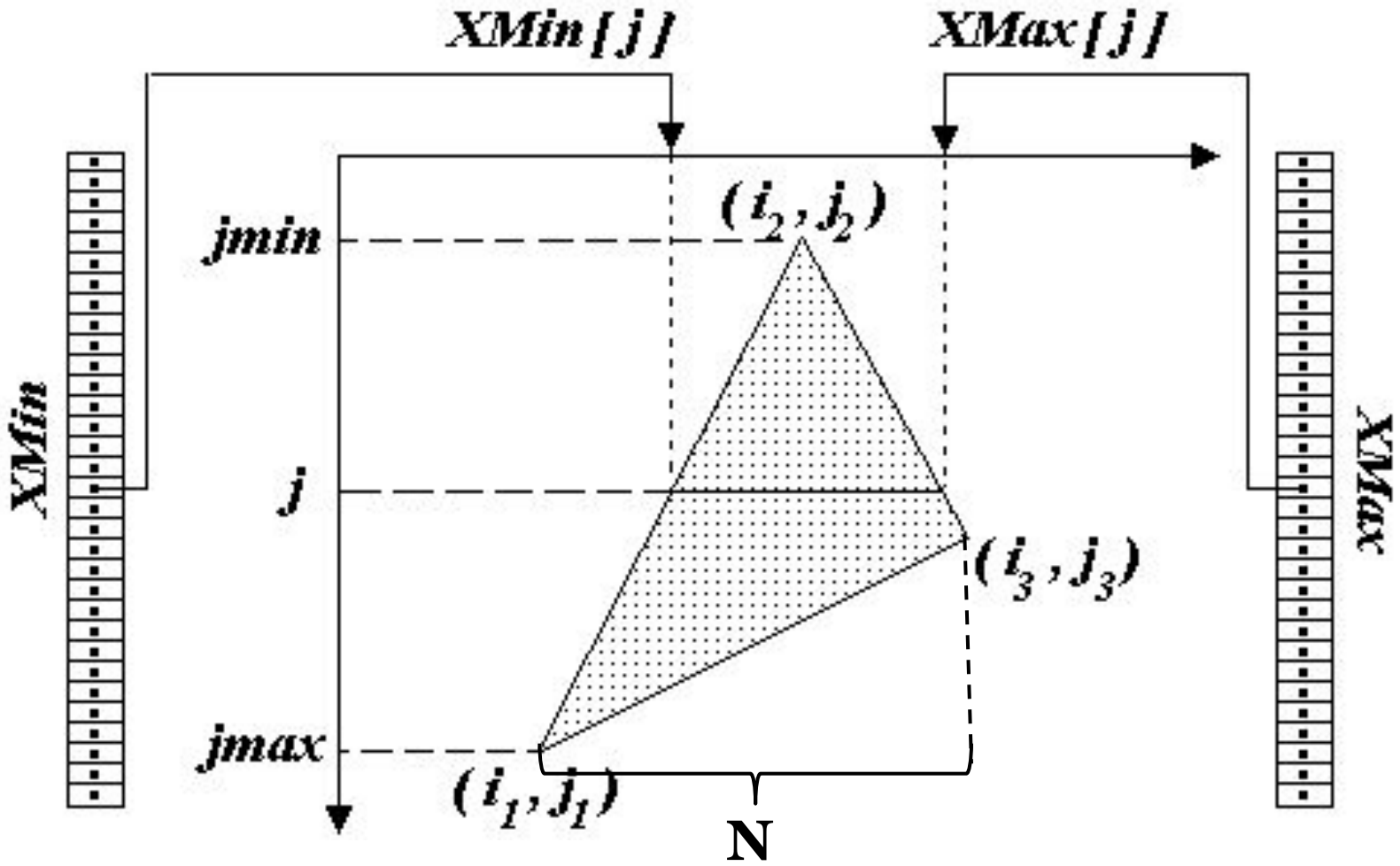


Метод сканирования строк

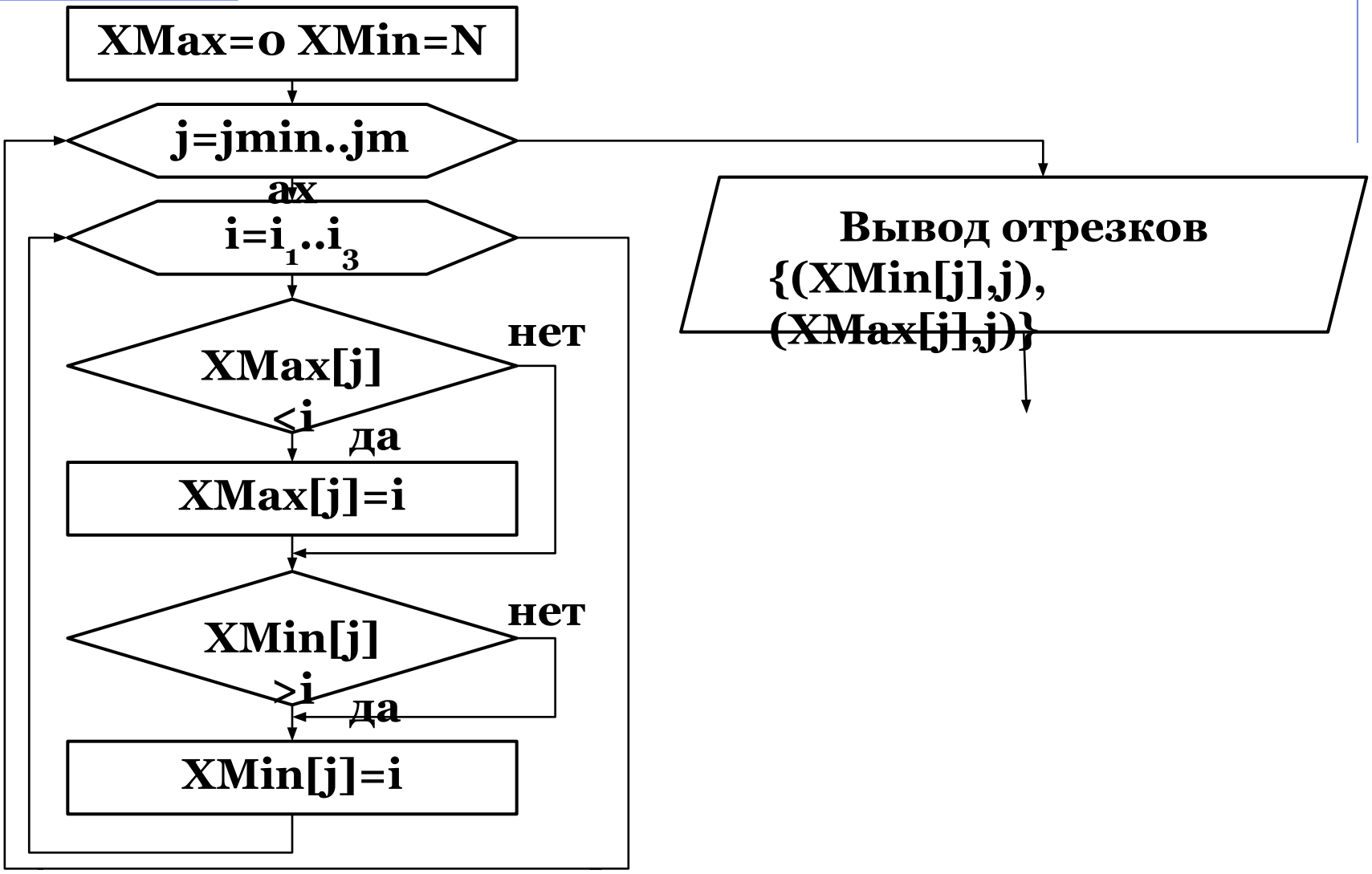


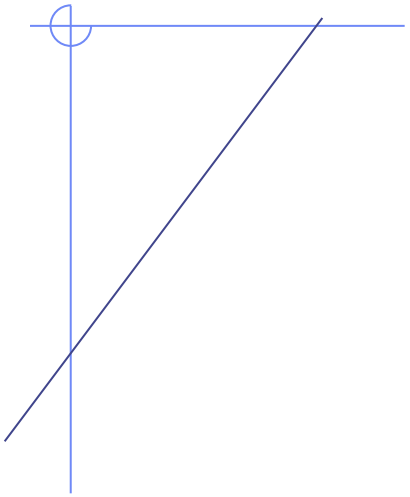
Метод сканирования строк

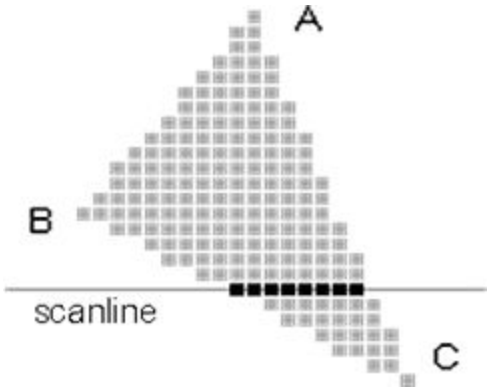
XMin, XMax

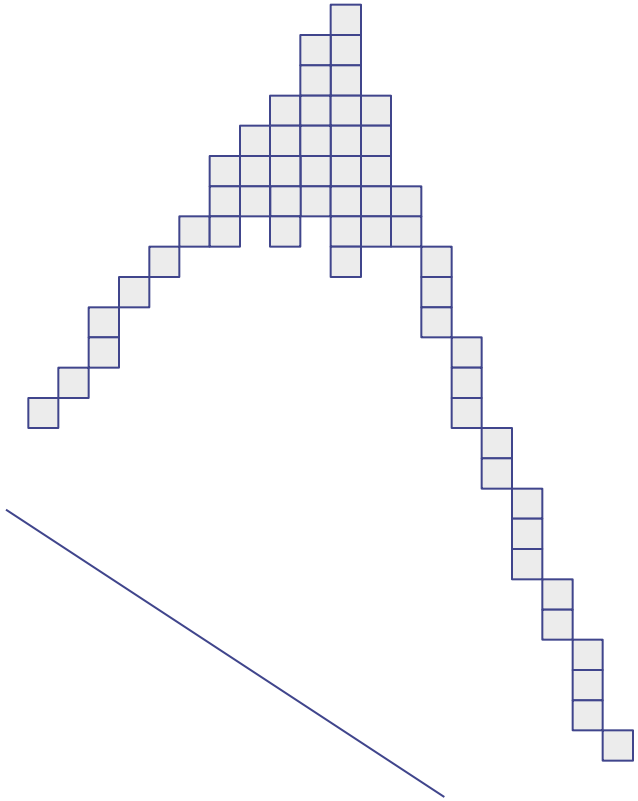


Алгоритм метода сканирования строк:

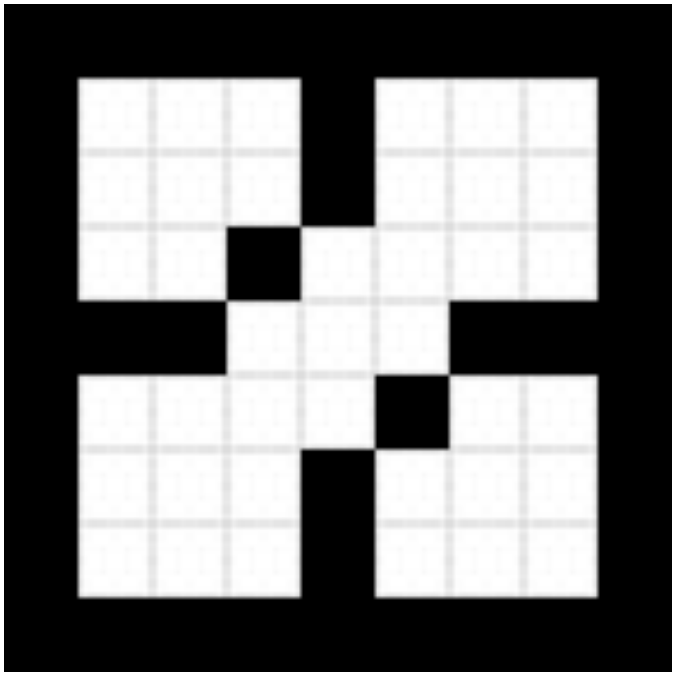




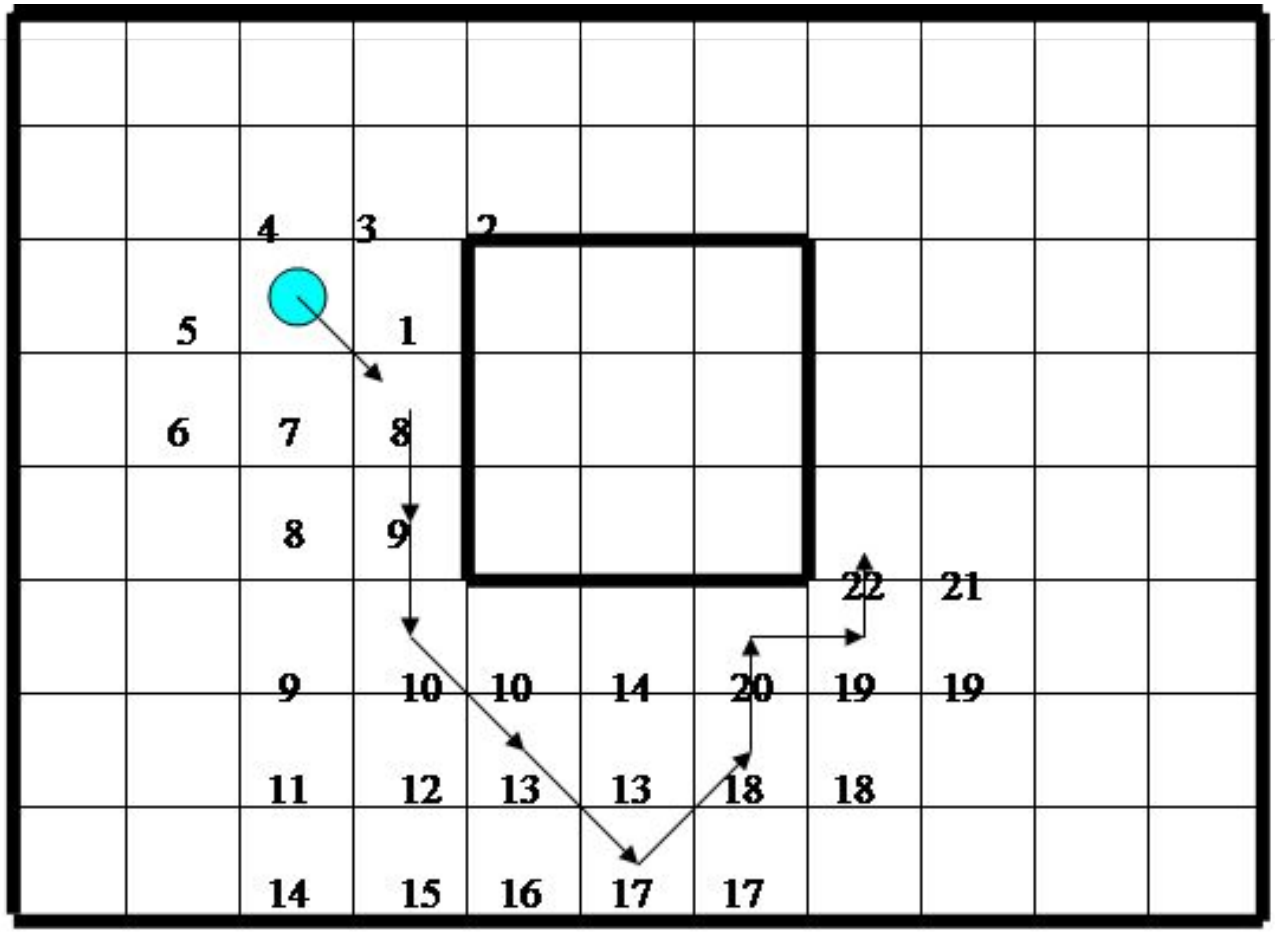

A
B
scanline
C



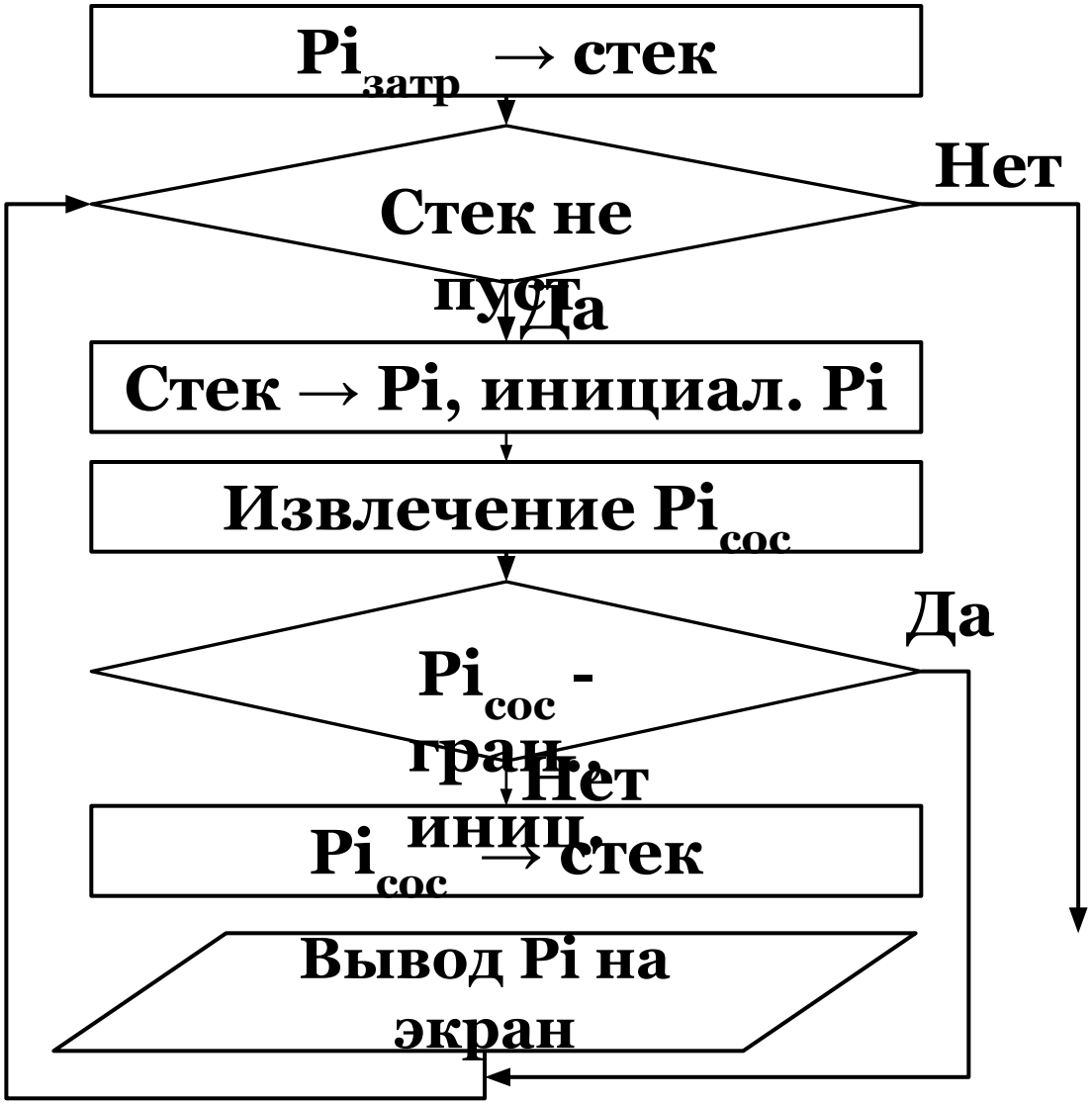
Метод затравочного заполнения



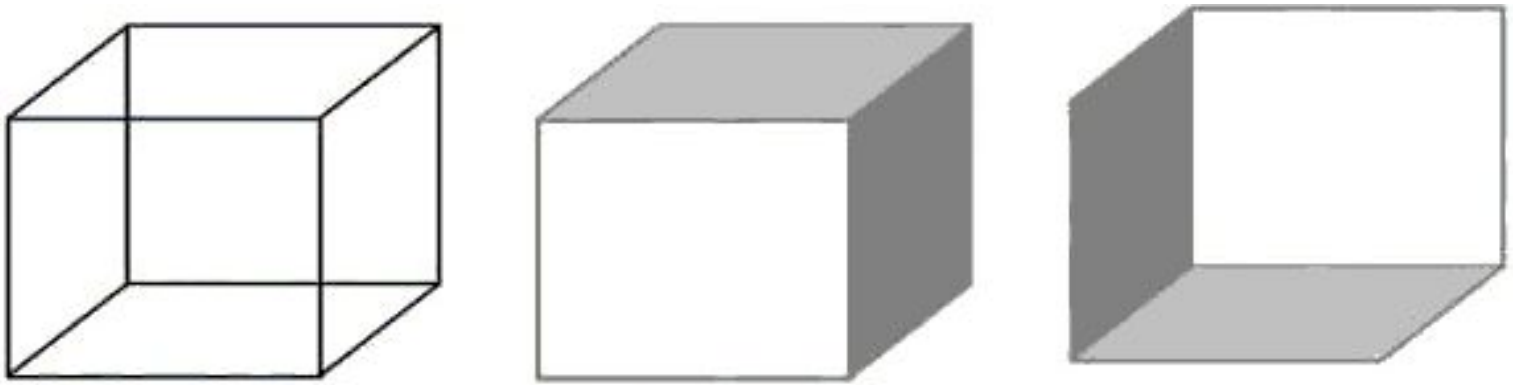
Метод затравочного заполнения:



Алгоритм затравочного заполнения



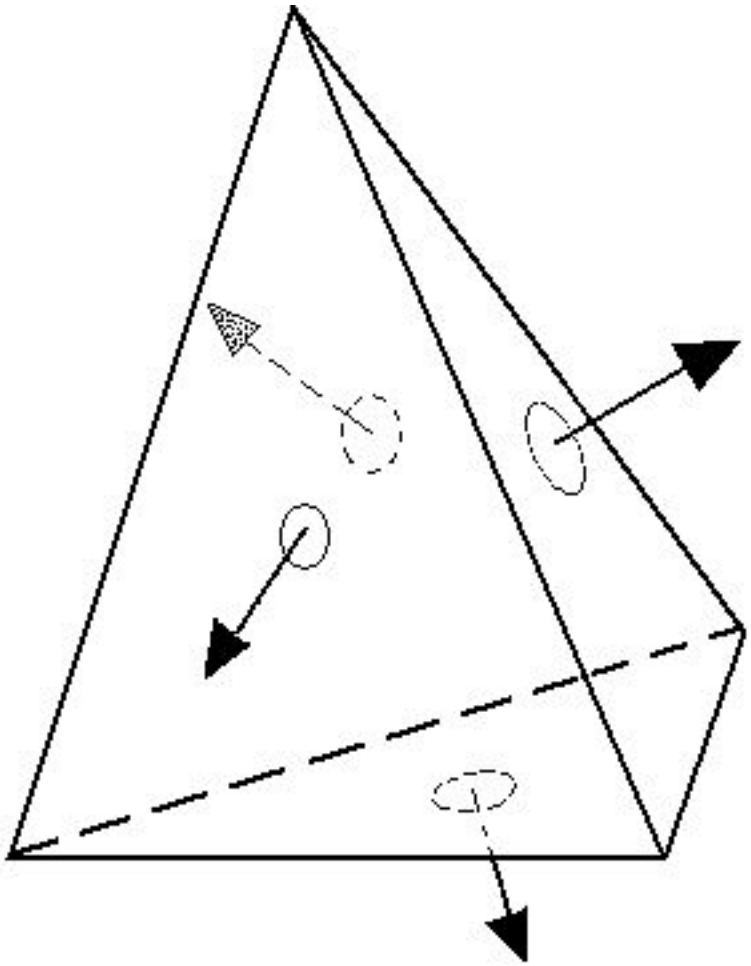
Удаление невидимых линий и поверхностей



Удаления невидимых линий:

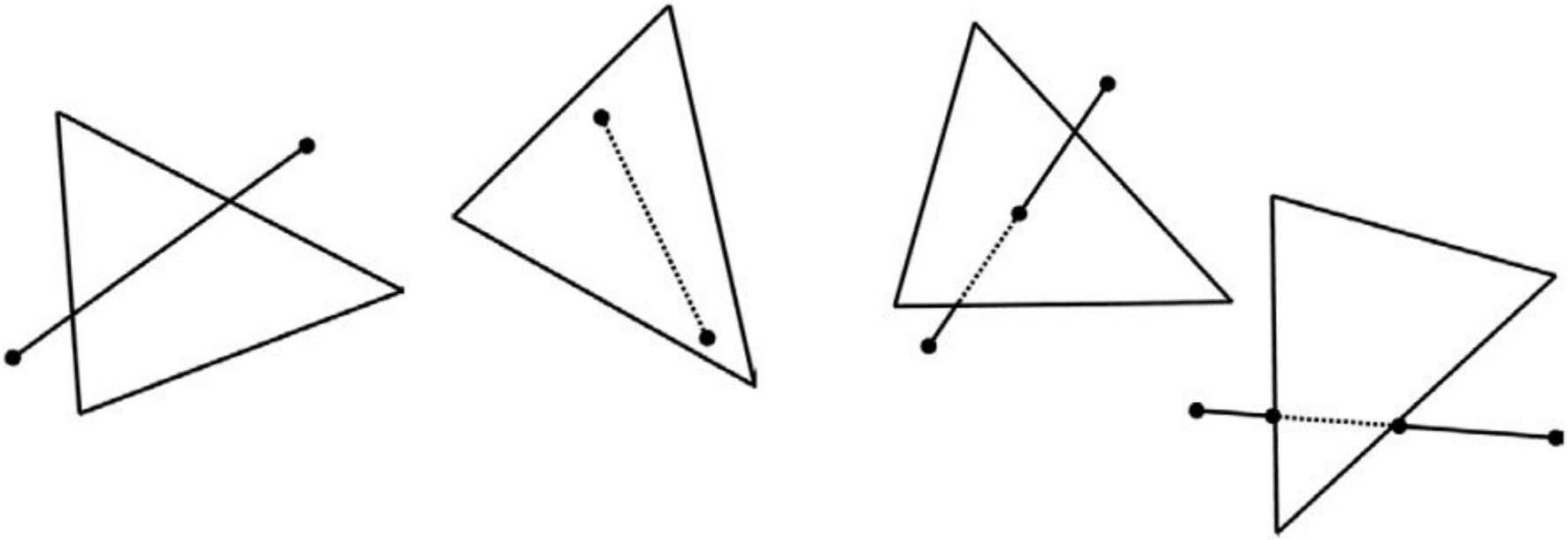
- Метод Робертса;
- Метод Аппеля;
- Метод Варнока;
- Метод Вейлера-Азертона;
- метод Z-буфера;
- метод построчного сканирования

Метод Робертса



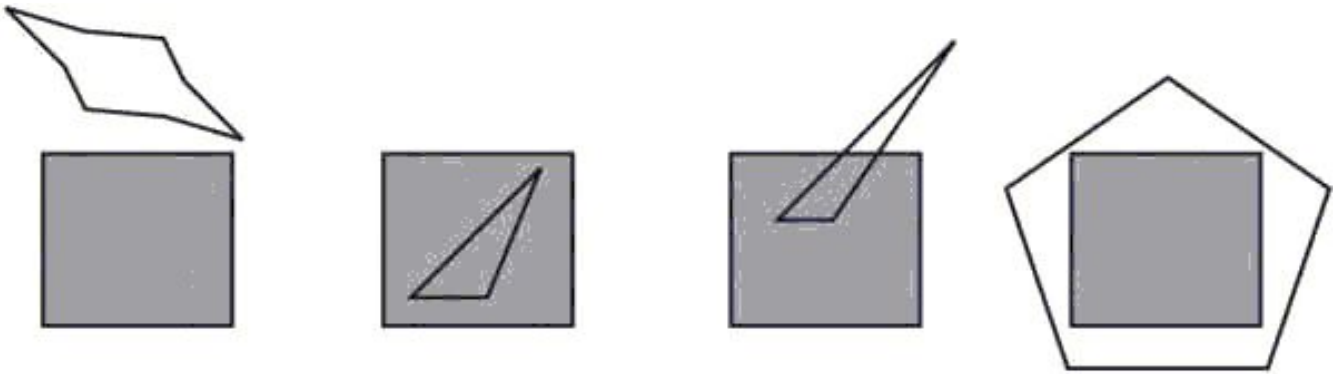
Алгоритм Робертса:

1. отбрасываются все ребра, обе образующие грани которых являются нелицевыми;
2. проверяется каждое из оставшихся ребер со всеми гранями многогранника на закрывание. При этом возможны три случая:



Метод Варнока

- **внешним**, если он целиком находится вне окна;
- **внутренним**, если он целиком расположен внутри окна;
- **пересекающим**, если он пересекает границу окна;
- **охватывающим**, если окно целиком расположено внутри него.



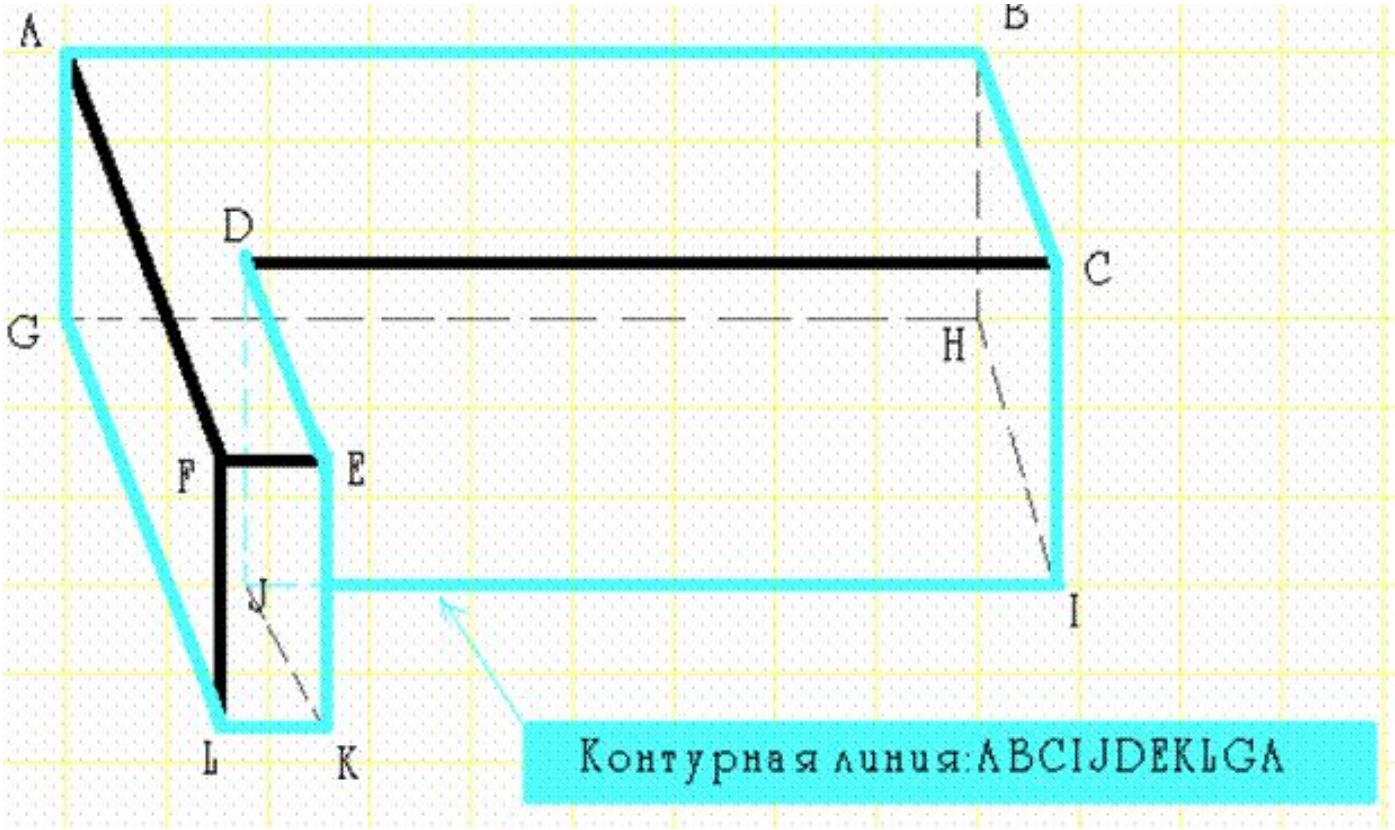
Алгоритм Варнока (продолжение)

1. Если все многоугольники сцены являются внешними по отношению к окну, то оно пусто и изображается фоновым цветом.
2. Если только один многоугольник сцены является по отношению к окну внутренним, то оно заполняется фоновым цветом, а многоугольник заполняется своим цветом.
3. Если только один многоугольник сцены имеет общие точки с окном и является по отношению к нему пересекающим, то окно заполняется фоновым цветом, а часть многоугольника, принадлежащая окну, заполняется цветом многоугольника.

Алгоритм Варнока (продолжение)

4. Если только один многоугольник охватывает окно и нет других многоугольников, имеющих общие точки с окном, то окно заполняется цветом этого многоугольника.
5. Если существует хотя бы один многоугольник, охватывающий окно, то среди всех таких многоугольников выбирается тот, который расположен ближе всех многоугольников к точке наблюдения, и окно заполняется цветом этого многоугольника.
6. В противном случае производится новое разбиение окна.

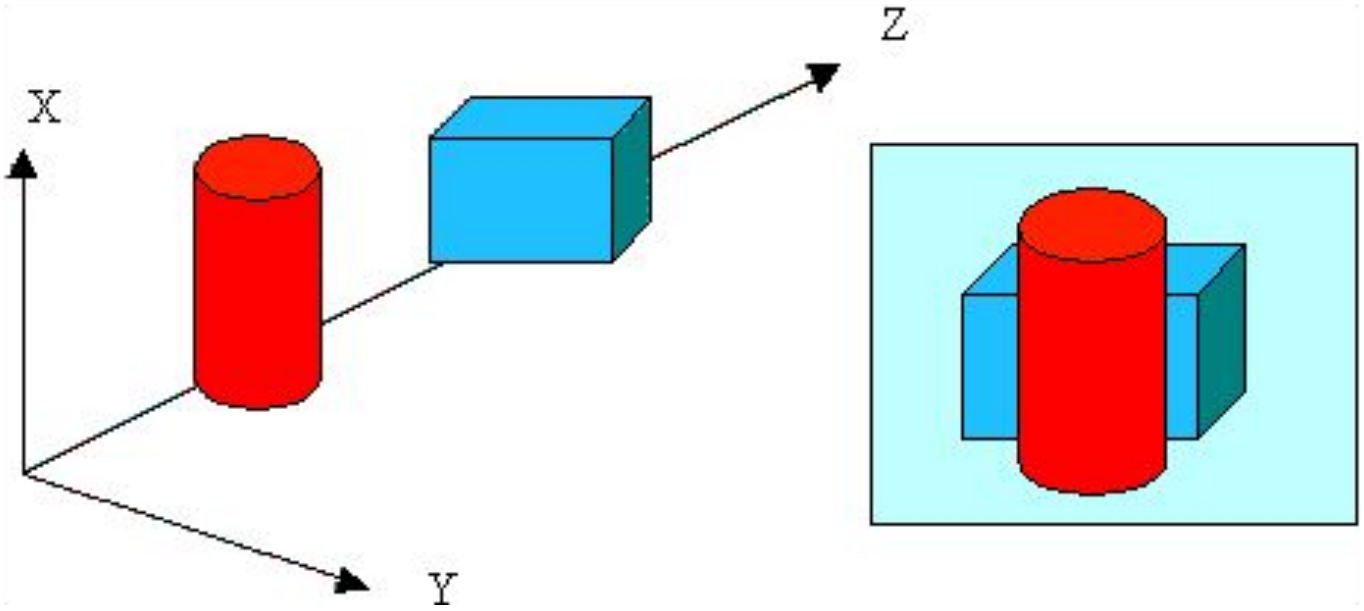
Метод Аппеля



Алгоритм Аппеля:

1. Определяется количественная невидимость вершины.
2. Просматривается изменение количественной невидимости вдоль каждого из ребер, выходящих из данной вершины.
3. Выполняется переход к следующей вершине и возврат к п. 1).
4. Если ребро выходит из вершины, принадлежащей контурной линии, проверяется, не закрывается ли это ребро одной из граней.

Метод z-буфера



Алгоритм Z-буфера:

1. Заполнить буфер кадра фоновым значением интенсивности или цвета.
2. Заполнить z-буфер минимальным значением z.
3. Преобразовать каждый многоугольник в растровую форму в произвольном порядке.
4. Для каждого *Пиксел*(x,y) в многоугольнике вычислить его глубину $z(x,y)$.
5. Сравнить глубину $z(x,y)$ со значением *Zбуфер*(x,y), хранящимся в z-буфере в этой же позиции.
6. Если $z(x,y) >$ *Zбуфер* (x,y), то записать атрибут этого многоугольника в буфер кадра и заменить *Zбуфер*(x,y) на $z(x,y)$. В противном случае никаких действий не производить.

Триангуляция

Методы триангуляции:

- Делоне;
- Форчуна

Алгоритм триангуляции

1. Берем три вершины A_1, A_2, A_3
2. Проверяем образуют ли векторы A_1A_3, A_1A_2 и их векторное произведение левую тройку векторов.
3. Проверяем нет ли внутри треугольника $A_1A_2A_3$ какой-либо из оставшихся вершин многоугольника.

Алгоритм триангуляции

4. Если оба условия выполняются, то строим треугольник $A_1A_2A_3$, а вершину A_2 исключаем из многоугольника, не трогая вершину A_1, сдвигаем вершины A_2 (A_2 на A_3), A_3 (A_3 на A_4)
5. Если хоть одно условие не выполняется, переходим к следующим трем вершинам.
6. Повторяем с 1 шага, пока не останется три вершины.

Практическая реализация