The background is a dark blue gradient with a subtle, repeating pattern of light blue dots connected by thin lines, creating a grid-like or mesh effect that recedes into the distance.

# Методика изучения линейных алгоритмов на основе графических операторов языка программирования Pascal.

Прежде чем приступить к созданию графических программ на Turbo Pascal, необходимо ознакомиться с богатыми графическими возможностями этого языка, сосредоточенными в стандартных библиотеках (модулях) **GRAPH** и **CRT**. Эти библиотеки (модули) содержат разнообразные константы, переменные, процедуры и функции, рассчитанные на работу с монитором в текстовом и графическом режимах.

# Текстовый режим

При работе в текстовом режиме экран монитора разбивается на строки, строки — на позиции, в каждую из которых можно поместить один символ. Положение символа на экране задается двумя координатами — вертикальной  $Y$  и горизонтальной  $X$ , где  $X$  - это номер позиции в строке, а  $Y$  — номер строки. Чаще всего на экране в текстовом режиме размещается 25 строк по 80 позиций.

# Текстовый режим

Х позиций

У  
с  
т  
р  
о  
к

25 строк по 80 позиций

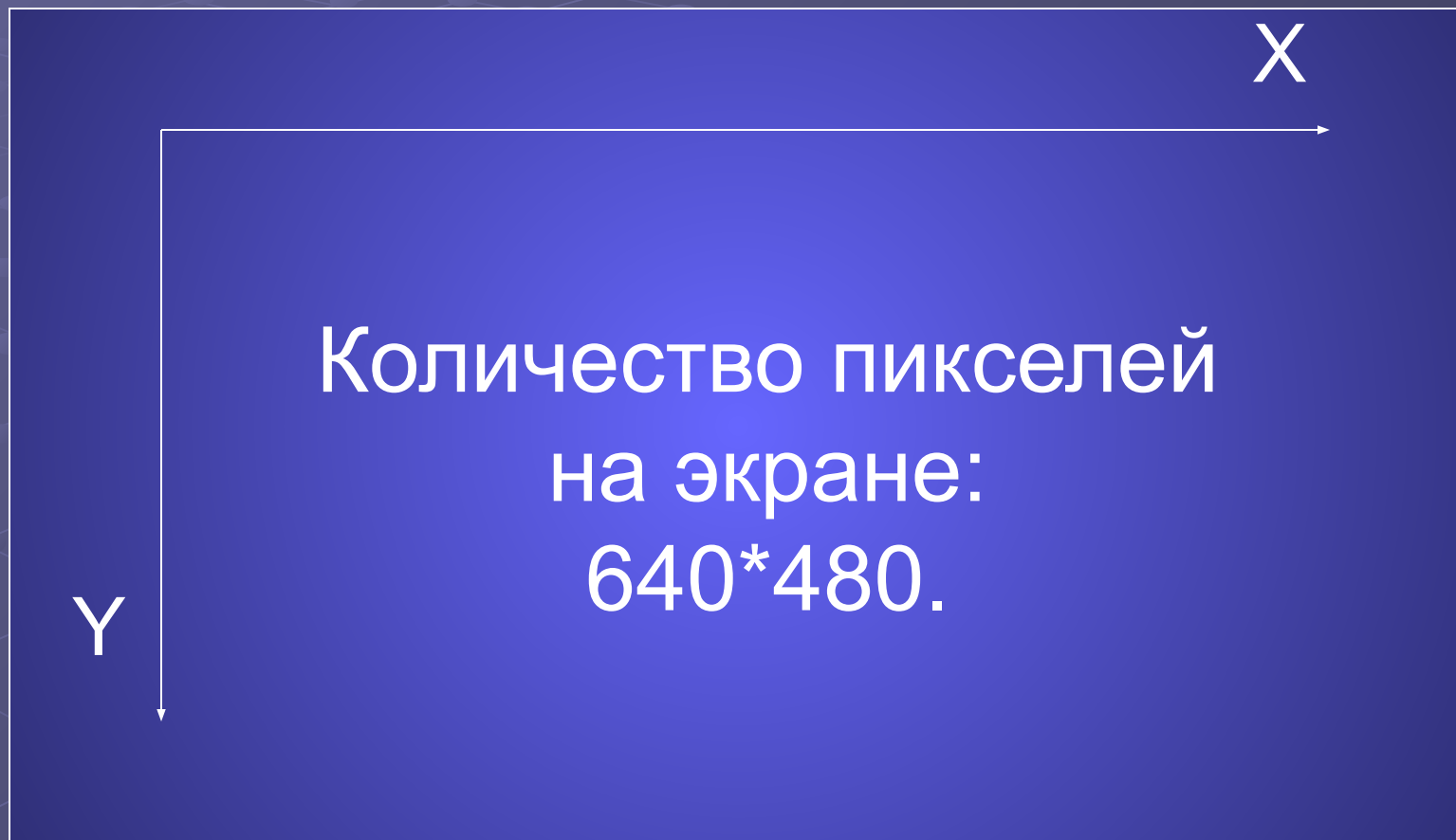
# Графический режим

При работе в **графическом режиме** весь экран разбивается на отдельные точки - пикселы.

Положение пикселя также задается двумя координатами - X и Y. Координата X увеличивается слева направо, а координата Y — сверху вниз.

Количество пикселов на экране зависит от типа графического адаптера и для распространенного адаптера VGA составляет 640x480.

# Графический режим





# Правила создания графической программы (1)

Подключение моделей CRT и GRAPH к программе осуществляется с помощью ключевого слова Uses (англ. uses - использует):

**Uses Crt, Graph;**

# Правила создания графической программы (2)

Программа при переключении в графический режим должна определить тип видеоадаптера.

При автоматическом распознавании видеоадаптера в правой части оператора присваивания используется константа Detect (или нулевое значение).

```
GrDriver:=Detect;
```



# Правила создания графической программы (3)

Второе, что должна сделать программа, — задать определенный графический режим.

```
InitGraph(GrDriver, GrMode, ' ');
```

# Правила создания графической программы (4)

Переключение в графический режим (*инициализация графического режима*) обычно сопровождается обработкой возможных ошибок инициализации. Эти ошибки могут быть связаны с отсутствием графического драйвера или неправильными значениями параметров. При наличии ошибки процедура `InitGraph` вырабатывает соответствующий, отличный от нуля результат — *код ошибки*. Этот код можно получить с помощью функции `GraphResult`.

# Правила создания графической программы (5)

Завершение работы в графическом режиме производится с помощью процедуры `CloseGraph`, которая выгружает драйвер из памяти и восстанавливает предыдущий видеорежим.

# Модуль Graph Turbo Pascal

Модуль Graph Turbo Pascal содержит около пятидесяти различных процедур и функций, предназначенных для работы с графическим экраном. В этом же модуле описаны некоторые встроенные константы и переменные, которые могут быть использованы в графических программах.

# Модуль Graph Turbo Pascal

Основную часть модуля составляют процедуры вывода базовых графических элементов, таких как точки, отрезки прямых линий, дуги и целые окружности и т.д. Такие элементы называются *графическими примитивами*. Другая группа процедур предназначена для управления графическим режимом.



# МОДУЛЬ CRT

Библиотека (модуль) CRT  
содержит константы, переменные,  
процедуры и функции,  
обеспечивающие управление  
текстовым режимом работы  
монитора и звуковым генератором.



# МОДУЛЬ GRAPH

Библиотека (модуль) Graph  
содержит константы, процедуры и  
функции для управления  
графическим режимом работы  
монитора.

# Константы цвета

- Black = 0; {Черный}
- Blue = 1; {Синий}
- Green = 2; {Зеленый}
- Cyan = 3; {Голубой}
- Red = 4; {Красный}
- Magenta = 5; {Фиолетовый}
- Brown = 6; {Коричневый}
- LightGray = 7; {Светлосерый}
- DarkGray = 8; {Темносерый}
- LightBlue = 9; {Яркосиний}
- LightGreen = 10; {Яркозеленый}
- LightCyan = 11; {Яркоголубой}
- LightRed = 12; {Розовый}
- LightMagenta = 13; {Малиновый}
- Yellow = 14; {Желтый}
- White = 15; {Белый}

# Константы типов и толщины линий

- SolidLn = 0;  
{Сплошная}
- DottedLn = 1;  
{Точечная}
- CenterLn = 2;  
{Штрихпунктирная}
- DashedLn = 3;  
{Пунктирная}
- NormWidth=1;  
{Нормальная  
толщина}
- ThickWidth = 3;  
{Тройная толщина}

# Константы шаблона штриховки

