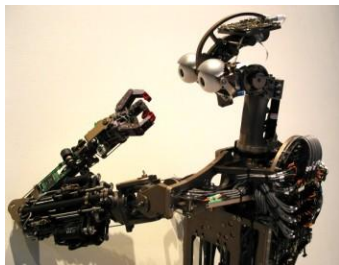


# Введение в компьютерную графику

# COMPUTER VISION

распознавание образов



# IMAGE PROCESSING

обработка изображений



# COMPUTER GRAPHICS

машинная графика

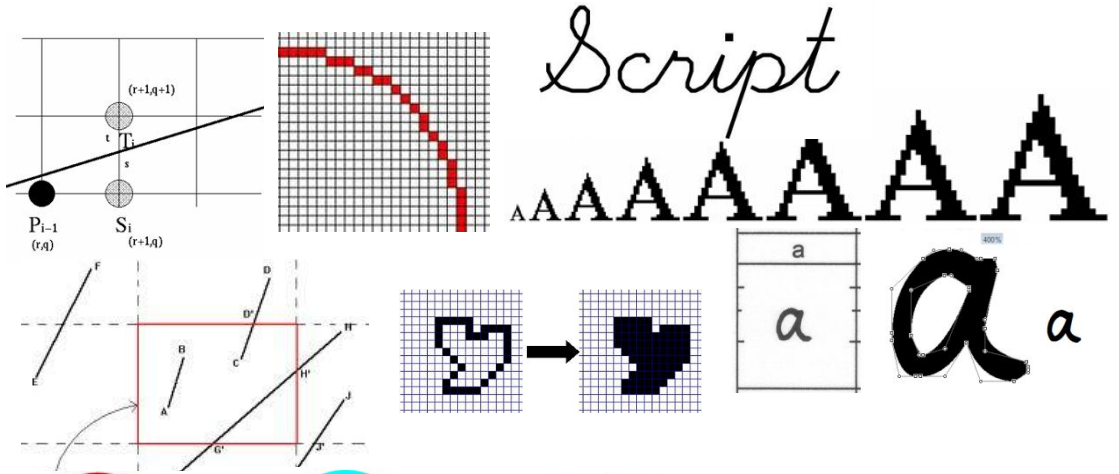


# • План курса

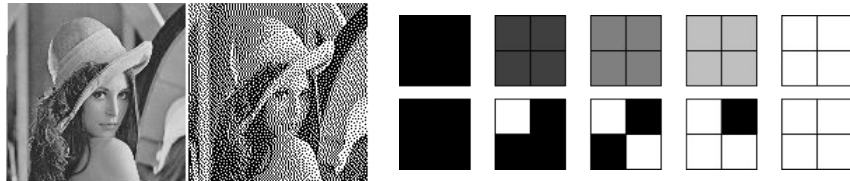
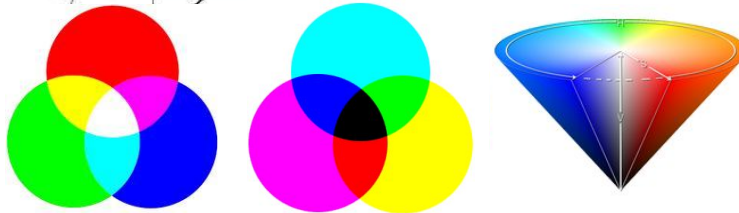
Введение



Растровая графика



Цвет



# План курса

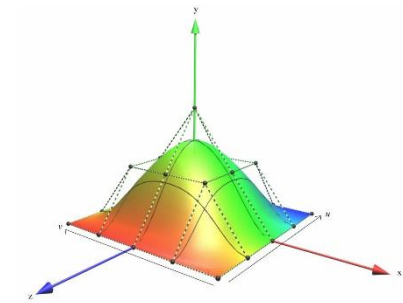
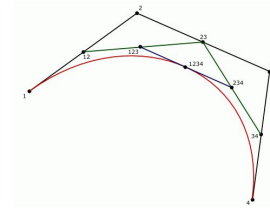
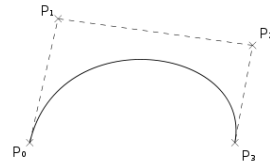
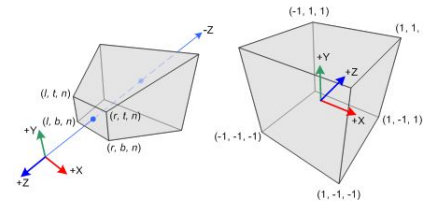
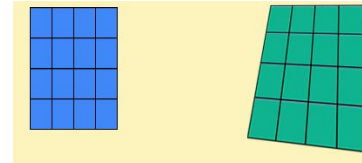
## Математика для КГ

$$\text{trans}(\Delta x, \Delta y, \Delta z) = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

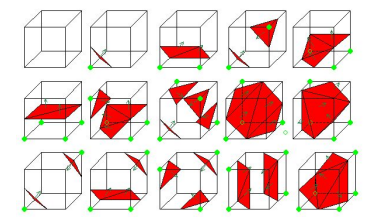
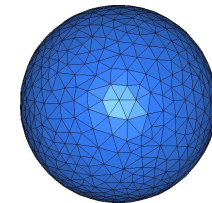
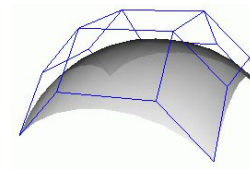
$$\text{rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rot}(z, \theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

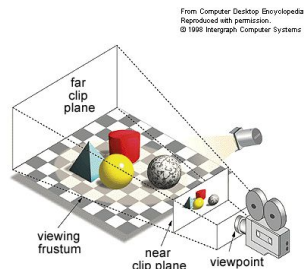


## Кривые и поверхности



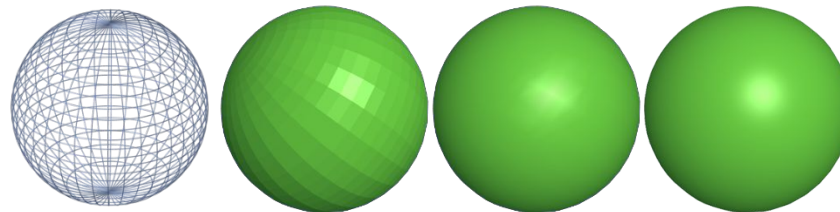
# План курса

Визуализация

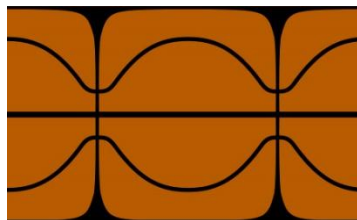


Удаление невидимых линий и поверхностей

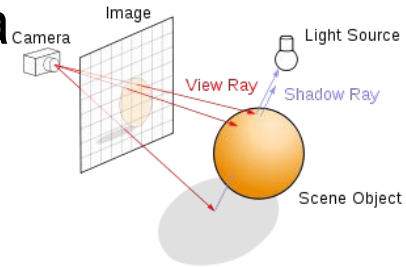
Освещение и тонирование



Текстурирование



# План курса



## Алгоритм трассировки лучей

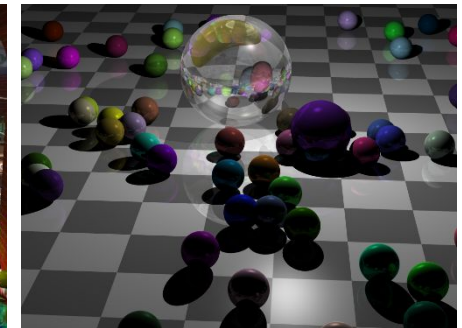
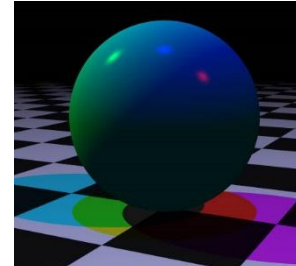
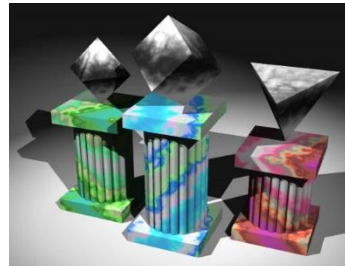
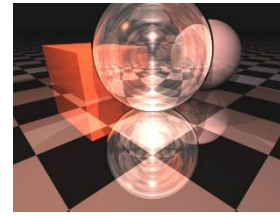
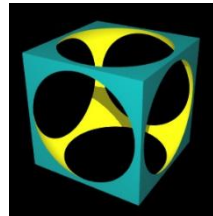
Базовый алгоритм

Модели освещения

Методы оптимизации

Шум

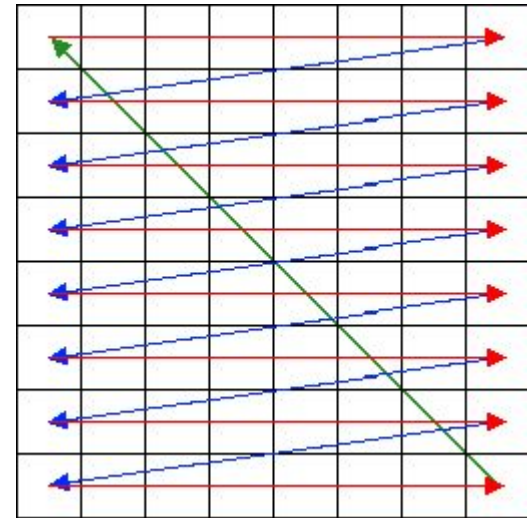
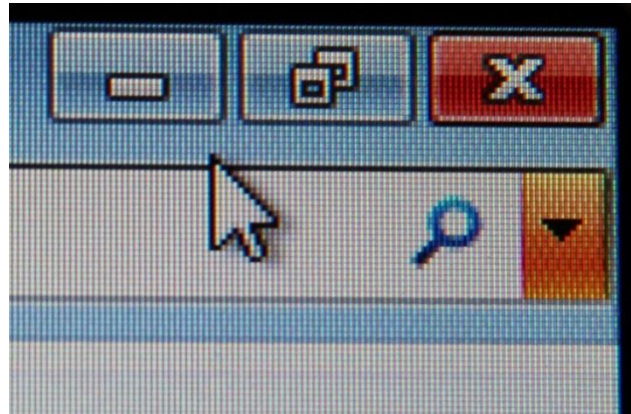
Распределенная трассировка лучей



# Аппаратные аспекты



Vector vs. Raster



# Видеопамять

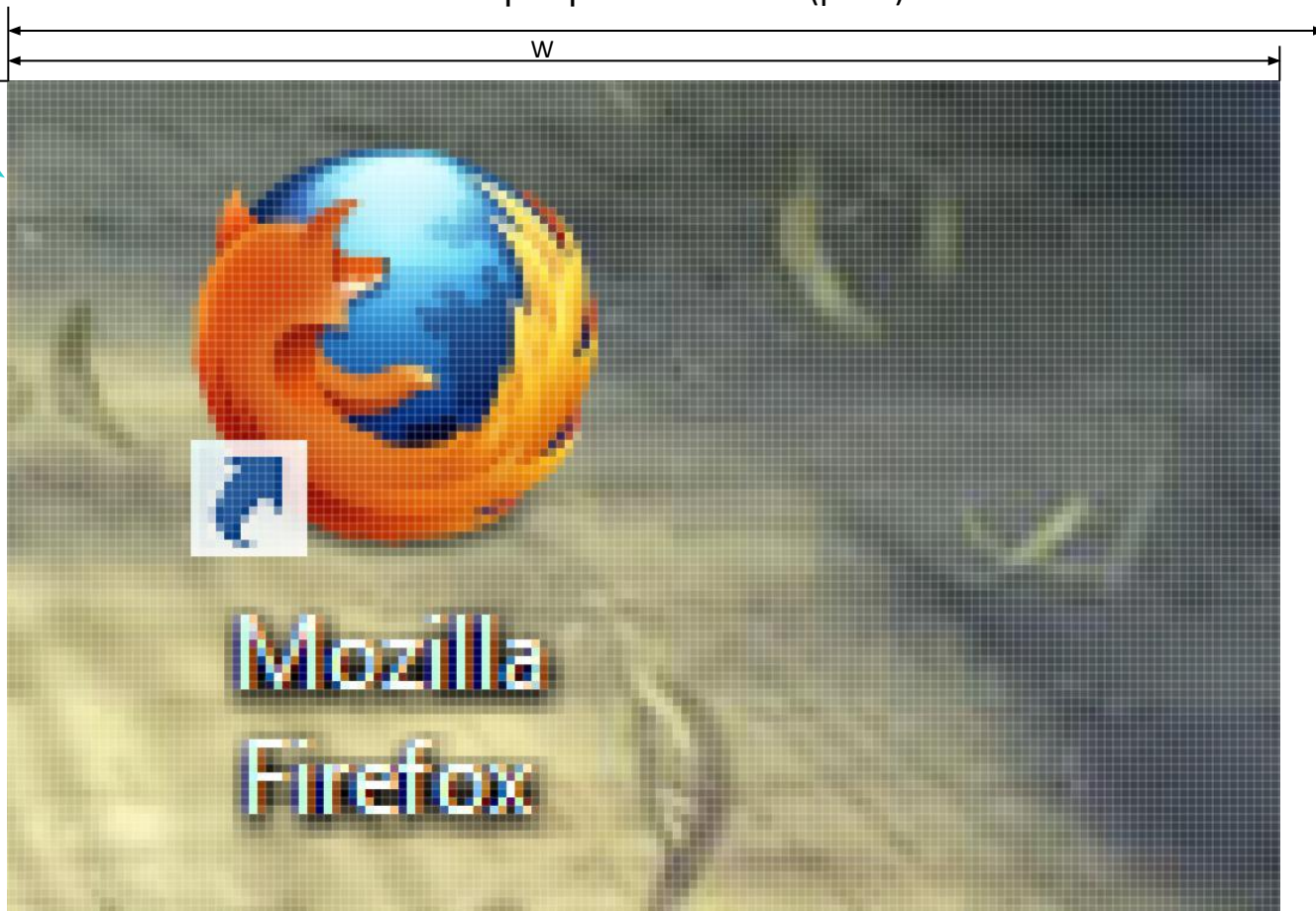


Размер строки в байтах (pitch)

w

Видео  
память

h



Внеэкранный память



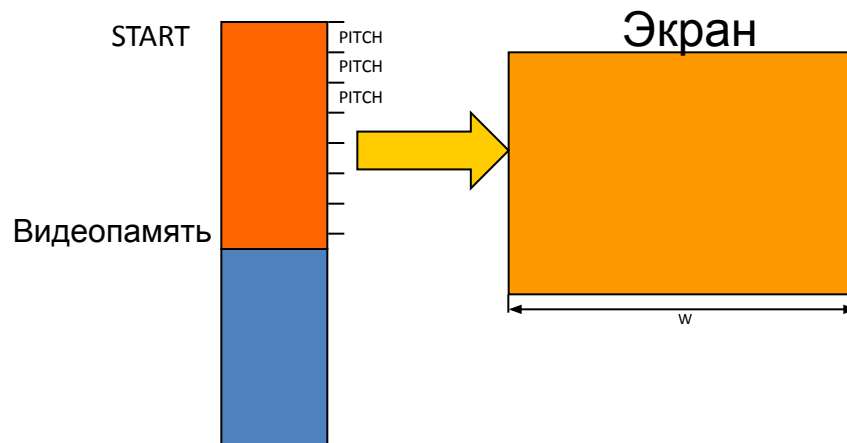
# Видеопамять

Адресация точек:

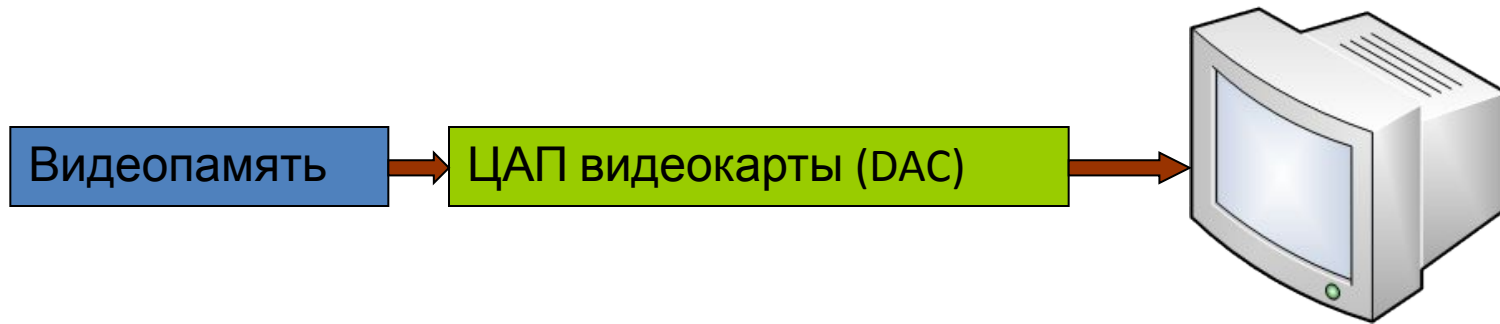
Стартовый адрес + Y \* Pitch + X

```
#define START ?????
#define PITCH ?????

void DrawPixel( int X, int Y, unsigned long ColorRGB )
{
    *(unsigned long *) (START + Y * PITCH + X) = ColorRGB;
}
```



# Цвет: палитра



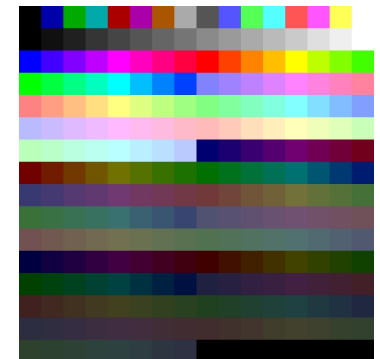
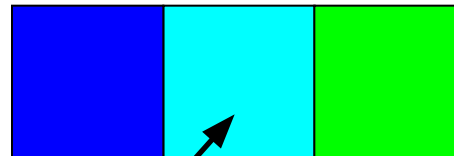
Номера цветов в видеопамяти

1	3	2
---	---	---

Палитра: таблица соответствия цветов

	Red	Green	Blue	
0	0	0	0	Black
1	0	0	255	Blue
2	0	255	0	Green
3	0	255	255	Cyan
...	...	...	...	
255	127	111	55	Brown

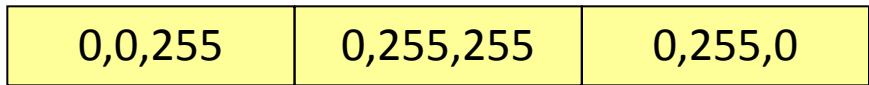
Точки (пиксели) на экране



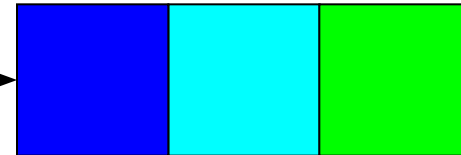
Стандартная палитра VGA

# Цвет: True Color

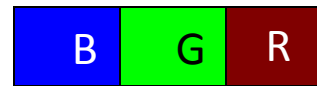
Триады RGB цветов в видеопамяти



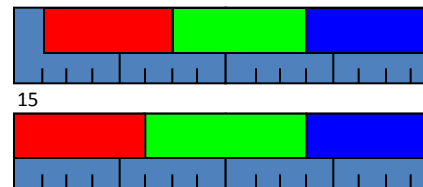
Точки (пиксели) на экране



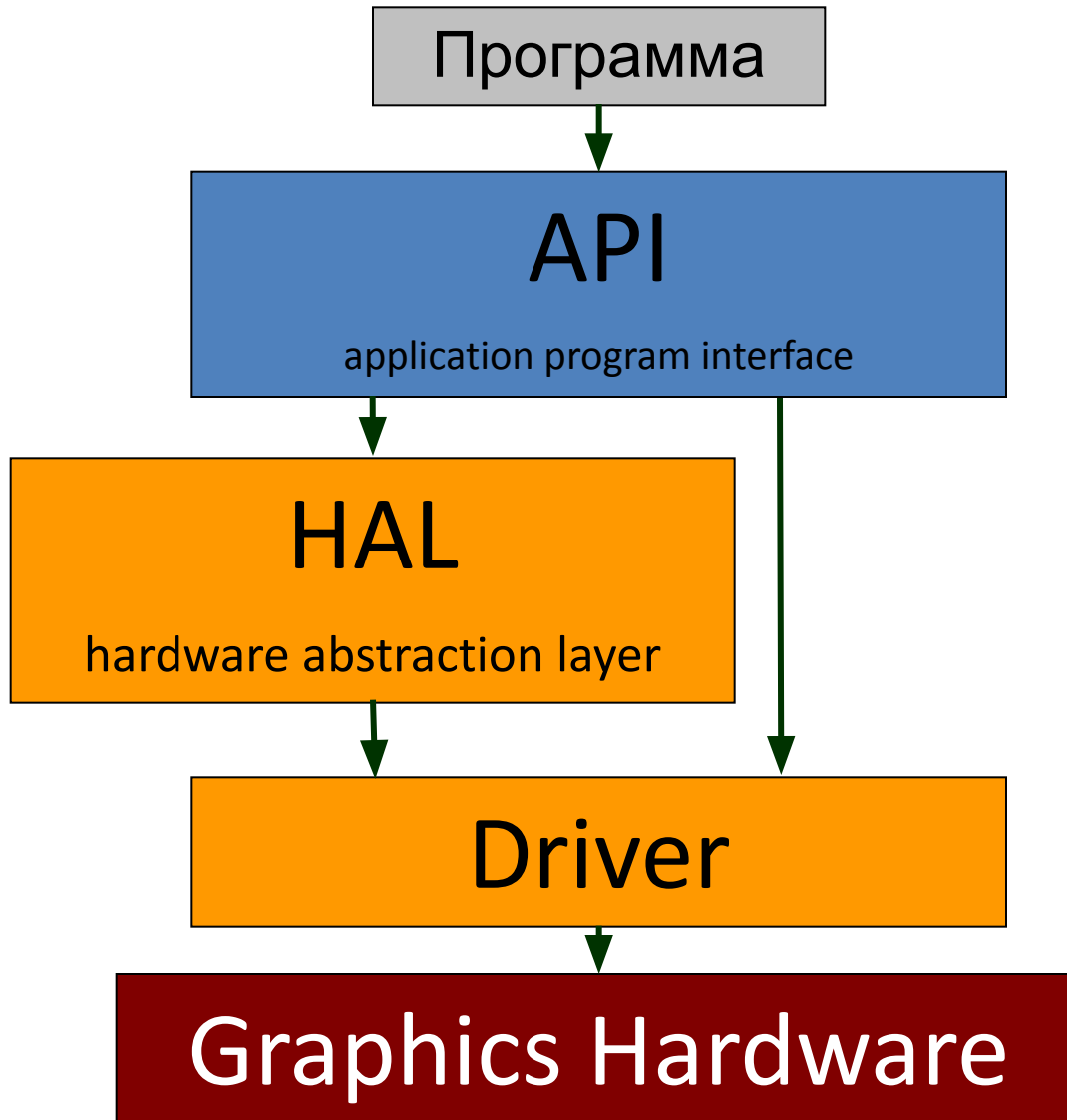
TrueColor: 24/32 бита (8r 8g 8b)



HiColor: 15/16 бит (5r 5g 5b / 5r 6g 5b)



# Структура библиотек



## Структура библиотек

- Функции рисования линейных объектов (и точек)
  - `Line(x1,y1,x2,y2)` `MoveTo(x,y)` `LineTo(x,y)` `SetPixel(x,y,c)`
- Функции рисования площадных объектов
  - `Rectangle(x1,y1,x2,y2)` `Circle(x,y,r)` `Polygon(points,n)`
- Функции вывода текста
  - `DrawText(x,y,string)`
- Функции задания атрибутов рисования
  - цвета и формы «перьев» для линейных объектов, шаблоны «кистей» для площадных, параметры вывода шрифтов (текста), код логической операции при выводе (`REPLACE`, `OR`, `AND`, `XOR`);
  - управление областями отсечения
- Функции задания преобразований системы координат
- Функции управления буферизацией
  - создание, копирование, уничтожение буферов изображений (bitmaps)

## Пример в DOS (прямая запись в память)

```
#define VGA256_MODE 0x13
#define TEXT_MODE 3

void SetMode( int mode )
{
    __asm {
        mov  AX, mode
        int  0x10
    }
}

void PutPixel( int x, int y, int color )
{
    *(unsigned char *) (0xA0000000UL + y * 320U + x) = color;
}

int main( void )
{
    /* Инициализация */
    SetMode(VGA256_MODE);

    /* Отрисовка зеленой точки */
    PutPixel(10, 10, 2);

    sleep(5);
    /* Выход из графического режима */
    SetMode(VGA256_MODE);
    return 0;
}
```