

Команды модуля CRT.

Во многих случаях стандартные для **Turbo Pascal** возможности ввода/вывода данных с помощью *стандартных* процедур оказываются недостаточными для разработки удобных в использовании диалоговых программ.

В библиотеке (модуле) **CRT**, предусмотрены несколько подпрограмм, существенно *увеличивающих* возможности текстового ввода/вывода.

Оператор	Модуль	Назначение	Пример
WINDOW	CRT	Процедура. Установить границы текстового окна, относительно левого верхнего угла экрана. Форма записи: WINDOW (X1 , Y1 , X2 , Y2) X1 , Y1 – координаты левого верхнего угла; X2 , Y2 – координаты правого нижнего угла.	WINDOW (1 , 1 , 10 , 5)
CLRSCR	CRT	Процедура. Очистить экран или текущее текстовое окно, помещая курсор в левый верхний угол (координаты 1,1) и закрасить текущим цветом фона.	CLRSCR
CLREOL	CRT	Процедура. Стирает все символы до конца строки, начиная с позиции курсора.	CLREOL
INSLINE	CRT	Процедура. Вставить пустую строку, все нижестоящие строки перемещаются на одну позицию вниз.	INSLINE
DELLINE	CRT	Процедура. Стереть строку, на которой находится курсор, все нижестоящие строки перемещаются на одну позицию вверх.	DELLINE
TEXTCOLOR	CRT	Процедура. Установить цвет выводимых символов. Форма записи: TEXTCOLOR (COLOR) COLOR – цвет выводимых символов (диапазон 0 – 15).	TEXTCOLOR (BLUE)
TEXTBACKGROUND	CRT	Процедура. Установить цвет фона выводимых символов. Форма записи: TEXTBACKGROUND (COLOR) COLOR – цвет фона (диапазон 0 – 7).	TEXTBACKGROUND (12)
GOTOXY	CRT	Процедура. Установить позицию курсора, относительно левого верхнего угла активного текстового окна. Форма записи: GOTOXY (X , Y) X , Y – координаты.	GOTOXY (12 , 3)
WHEREX	CRT	Функция целого типа. Узнать позицию курсора по X .	X :=WHEREX
WHEREY	CRT	Функция целого типа. Узнать позицию курсора по Y .	Y :=WHEREY
KEYPRESSED	CRT	Функция логического типа. Возвращает значение типа BOOLEAN , указывающее состояние буфера клавиатуры: FALSE означает, что буфер пуст, а TRUE – что в буфере есть хотя бы один символ, ещё не прочитанный программой. Обращение к функции <i>не задерживает</i> исполнение программы.	REPEAT WRITE ('SLOVO') UNTIL KEYPRESSED
READKEY	CRT	Функция символьного типа, возвращает значение типа CHAR . <i>Останавливает</i> выполнение программы до нажатия любой клавиши и возвращает её код.	A :=READKEY
DELAY	CRT	Процедура. Установить задержку работы программы на заданный интервал времени. Форма записи: DELAY (I) I – выражение типа WORD , определяющее интервал времени.	DELAY (1000)
SOUND	CRT	Процедура. Выдать звук заданной частоты. Форма записи: SOUND (I) I – выражение типа WORD , частота звука в герцах.	SOUND (500)
NOSOUND	CRT	Процедура. Выключить звук.	NOSOUND

Пример программы.

```

Program CRT_Demo;
Uses CRT;
Var
  I : Integer;
BEGIN
Window(1,1,80,25);
TextBackGround(1);
ClrScr;
For I:=1 To 10 Do
  Begin
    Window(40-I,12,40+I,14);
    TextBackGround(7);
    ClrScr;
    Delay(1000);
  End;
TextColor(0);
GotoXY(7,2);
Writeln('Модуль CRT');
Window(1,1,80,25);
GotoXY(23,24);
TextColor(15);
TextBackGround(0);
Write('Для продолжения нажмите любую клавишу. ');
ReadKey;
END.

```

Коды цвета.

Black	= 0	Чёрный
Blue	= 1	Тёмно-синий
Green	= 2	Тёмно-зелёный
Cyan	= 3	Бирюзовый
Red	= 4	Красный
Magenta	= 5	Фиолетовый
Brown	= 6	Коричневый
LightGray	= 7	Светло-серый
DarkGray	= 8	Тёмно-серый
LightBlue	= 9	Синий
LightGreen	= 10	Светло-зелёный
LightCyan	= 11	Светло-бирюзовый
LightRed	= 12	Розовый
LightMagenta	= 13	Малиновый
Yellow	= 14	Жёлтый
White	= 15	Белый
Blink	= 128	Мерцание символа

В место кода можно использовать названия цветов, и исходя из этого операторы `TextColor(15)` и `TextColor(White)` аналогичны по своему действию.

Границы экрана.

Максимальное количество столбцов – 80, а строк – 25.

При превышении координат им присваивается, как правило, значение 1.

