

## ЛАБОРАТОРНАЯ РАБОТА №13. МОДУЛЬ CRT

Практически любая программа, разработанная для ПЭВМ, использует дисплей для отображения вводимой и выводимой информации. В зависимости от типа используемого ПЭВМ адаптера (ПЭВМ комплектуется монохромным или цветным адаптером) всю выводимую на экран информацию подразделяют на текстовую или графическую. Соответственно выделяют текстовый или графический режимы. Текстовый режим поддерживается стандартным модулем CRT. В разделе интерфейса этого модуля определяют ряд констант и переменных.

Для задания режима экрана используются следующие константы (таблица 1).

Таблица 1. Константы режимов экрана.

Наименование константы	Значение константы	Размер экрана	Цвет выводимых символов	Вид адаптера
1	2	3	4	5
BW 40	0	40 * 25	Ч/Б	Цветной
Co 40	1	40 * 25	Цветной	Цветной
BW 80	2	80 * 25	Ч/Б	Цветной
Co 80	3	80 * 25	Цветной	Цветной
Mono	7	80 * 25	Ч/Б	Монохромный
Font 8*8	+ 256	80*43 (EGA) 80*50 (VGA)	Цветной Цветной	Цветной Цветной

При задании цвета и фона выводимых на экран символов предусмотрены следующие константы (таблица 2).

Таблица 2. Константы цвета и фона символов.

Цвет	Наименование константы цвета символов	Значение константы
Черный	Black	0
Синий	Blue	1
Зеленый	Green	2
Бирюзовый	Cyan	3
Красный	Red	4
Малиновый	Magenta	5
Коричневый	Brown	6
Светло-серый	LightGray	7
Темно-серый	DarkGray	8
Светло-голубой	LightBlue	9
Светло-зеленый	LightGreen	10
Светло-бирюзовый	LightCyan	11
Светло-красный	LightRed	12
Светло-малиновый	LightMagenta	13
Желтый	Yellow	14
Белый	White	15
Мерцающее изображение	Blik	+ 128

Кроме перечисленных констант в разделе интерфейса модуля CRT объявлены следующие переменные:

- DirectVideo: boolean - для организации прямой адресации видео памяти;
- CheckSnow: boolean - для регулировки "снежного" эффекта при прямом доступе к видеопамяти;
- LastMode: word - для хранения текущего текстового режима;
- TextAttr: byte - для хранения текущего цвета выводимых на экран дисплея символов;
- WindMin: word - для хранения координат верхнего левого угла текущего окна;
- WindMax: word - для хранения координат нижнего правого угла текущего окна.

### **Управление выводом текстовой информации**

При выводе на экран текстовой информации Turbo Pascal, в зависимости от выбранного режима, позволяет управлять следующими параметрами: интенсивностью свечения, фоном, цветом.

Для этого используются следующие стандартные процедуры (данные процедуры реализованы в модуле CRT, поэтому необходимо в начале программы подсоединить этот модуль командой Uses Crt):

- Low Video - устанавливает режим минимальной яркости свечения выводимых на экран символов;
- Norm Video - устанавливает нормальной яркости свечения выводимых на экран символов;
- High Video - устанавливает наибольшей яркости свечения выводимых на экран символов;
- TextBackGround (Color: byte) - устанавливает цвет фона (т.е. цвет области, которая окружает выводимый символ). Здесь Color - выражение целого типа в диапазоне от 0 до 7, соответствует одной из восьми констант, определенных в модуле CRT.
- TextColor (Color: byte) - устанавливает цвет выводимых символов. Здесь Color - выражение целого типа в диапазоне от 0 до 15, соответствует одной из констант, определенных в модуле CRT и приведенных в таблице 2.

### **Процедуры и Функции**

#### **Функции**

WhereX: Integer - функция, которая возвращает X-координату текущей позиции курсора (относительно активного окна);

WhereY: Integer - функция, которая возвращает Y-координату текущей позиции курсора (относительно активного окна);

KeyPressed: Boolean - функция которая возвращает значение True, если на клавиатуре была нажата какая-либо клавиша, и False в противном случае.

Repeat

WriteLn('Печатать пока не будет нажата любая клавиша');

Until KeyPressed;

Функция KeyPressed не распознает такие клавиши, как <ДОП>, <Shift>, <ЦИФ> и т.д.

ReadKey: char - функция, которая считывает код символа с клавиатуры. Специальные клавиши (функциональные клавиши, клавиши управления курсором, ДОП, КОН, и т.д.) генерируют расширенные коды клавиш, состоящие из двух значений, причем первое всегда равно #0. При нажатии специальной клавиши функция ReadKey возвращает сначала нулевой символ (#0), а затем вторую (расширенную) часть кода:

```
Var Kod: Char;  
Begin Kod:= ReadKey;  
If Kod = #0 Then Begin  
Kod:= ReadKey;  
If Kod= #79 Then WriteLn('Нажата КОН');  
End  
End.
```

## Процедуры

Window (X1, Y1, X2, Y2: byte) - процедура, которая определяет на экране новое активное текстовое окно. Окно - это ограниченная область экрана, выполняющая те же функции, что и полный экран. Однако следует учитывать, что после определения окна все координаты задаются относительно него (начиная с 1 от его левого верхнего угла), а не полного экрана. При определении окна X1, Y1 являются координатами левого верхнего угла окна, а X2 и Y2 - координатами нижнего правого угла с учетом того, что левый верхний угол полного экрана имеет координаты (1,1), а минимальное окна должно включать один столбец и одну строку. При неправильном задании параметров X1, Y1 или X2, Y2 (например, когда X1>80 или Y2=30) процедура Window игнорируется;

ClrScr - процедура, которая очищает активное окно и устанавливает курсор в левый верхний угол;

ClrEol - процедура, которая очищает строку активного окна от текущей позиции курсора до конца строки без изменения позиции курсора;

GotoXY (X, Y: byte) - процедура, которая перемещает курсор в позицию с координатами X, Y в рамках активного окна.

Delay (Time: word) - процедура, которая вызывает задержку выполнения программы на заданное число миллисекунд Time (0,001с);

Sound (Hz: word) - процедура, которая включает внутренний динамик. Hz задает частоту генерируемого динамиком сигнала в герцах. Звуковой сигнал звучит до тех пор, пока не будет выключен с помощью процедуры NoSound;

NoSound - процедура, которая отключает внутренний динамик;

DelLine - удаляет строку, на которой находится курсор, и смещает на позицию вверх все строки, расположенные ниже. В нижнюю часть активного окна добавляется новая строка;

InsLine - в рамках текущего окна вставляет пустую строку, начиная с позиции, где находится курсор;

AssignCrt (Var F: text) - используется для задания текстового файла F, связываемого с дисплеем, для последующего его быстрого ввода - вывода.

### Программирование с использованием модуля Crt.

Пример 1. Задача: курсор от левого верхнего угла экрана движется вниз и вправо уступом с задержкой на 0.5 с на каждой позиции; при достижении строки 12 выдается сообщение "Строка 12" и следует останов программы.

```
Program primer;  
Uses CRT;  
Var i : integer;  
Begin i:=1; CLRSCR;  
  While true do  
    Begin if (whereX =12) and (whereY = 12) then  
      Begin  
        Write ('строка 12');  
        HALT;  
        End;  
      GotoXY(i, i); delay (500); i:=i+1;  
    End;  
  End.  
End.
```

В этой программе для останова и выхода из бесконечного цикла "while true do" использована процедура HALT. В Паскале есть еще одна похожая процедура - выхода из блока "EXIT".

Пример 2. Требуется организовать 3 окна, в каждом из них напечатать текст. Экран выдачи данных остается до нажатия любой клавиши. Прежде чем привести текст программы, отметим основные правила, которые надо соблюдать при программировании с использованием окон.

- После образования окна необходима процедура очистки экрана. Если этого не сделать, то установленный Вами цвет фона будет действовать не на все окно, а только на позиции с выведенными символами.
- В конце программы необходимо вернуться к исходному режиму работы дисплея – задать стандартные цвета фона и символов, или использовать процедуру Textmode. Если не сделать этого, то при следующем прогоне программы весь экран будет цвета последнего окна, т.е. Вы не увидите работу Вашей программы, как она была задумана.

```
Program primer2;  
Uses CRT;  
Begin  
  CLRSCR; { Очистка экрана }  
  
  { Организуем первое окно, синие буквы по оранжевому фону }  
  Textbackground (6); Textcolor (1); Window (2, 2, 40, 10 );  
  CLRSCR; { Очистка окна }  
  GotoXY (10 , 3 ); writeln ('Томский' ); { установка курсора и печать текста }
```

```

{ Организуем второе окно, красные буквы по зеленому фону }
Textbackground (2); Textcolor (4); Window (20, 8, 60, 17 );
CLRSCR; { Очистка окна }
GotoXY (4 , 4 ); writeln ('политехнический') ; {установка курсора и печать
текста}

```

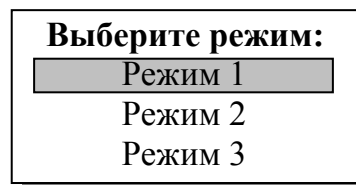
```

{ Организуем третье окно, желтые буквы по голубому фону }
Textbackground (3); Textcolor (14); Window (55, 20, 78, 24 );
CLRSCR; { Очистка окна }
writeln ('университет') ; {в этом окне печать текста начинается сразу от
левого верхнего угла окна, т.е. с позиции 1, 1}
while not keypressed do; {Цикл – пустой, ожидание нажатия любой клавиши }
Textmode (3); {Установка исходного режима}
End.

```

В этой программе второе окно частично закрывает первое.

Пример 3. Рассмотрим несложную программу, выполняющую вывод на экран дисплея меню следующего вида:



Выбор конкретного режима из этого меню должен осуществляться с помощью клавиш <↑> и <↓>. При нажатии этих клавиш должен генерироваться звук продолжительностью 0.5 сек. При этом, строка на которую подает выбор, должна выделяться цветом и фоном. Выбор фиксируется с помощью клавиши Enter (Ввод). После этого должно быть выдано сообщение о выбранном режиме.

Данная программа выглядит следующим образом:

```

Program Prim;
Uses Crt; {Подсоединение модуля Crt}
Type Mas = array[1..3] of string[20];
Const {Описание массива с изображением меню}
      Stor : Mas = ( ' Режим 1 ',
                    ' Режим 2 ',
                    ' Режим 3 ');
Var
  I, K: byte;
  Kod: char;
Begin
  TextMode(CO80); {Установка текстового режима 80*25}
  ClrScr;
  Window(31,11,50,16); {Организация окна}
  TextBackGround(7); {Задание фона}
  TextColor(0); {Задание цвета выводимых символов}
  ClrScr;

```

```

К := 1; {К- номер режима}
GotoXY(3,2); {Вывод заглавия меню}
Write('Выберите режим : ');
{Выделение выбранной строки фоном и цветом}
Kod:= ' ';
While Kod<>#13 Do
  Begin
    For i:=1 to 3 Do
      Begin
        If I=K then Begin
          TextBackGround(0);
          TextColor(7);
          End;
        Else Begin
          TextBackGround(7);
          TextColor(0);
          End;
        GotoXY(1,I+2);
        Write(Stor[I]); {Вывод строк меню}
        End;
      {Отображение выбора с помощью клавиш <↑> и <↓>}
      Kod:= RadKey; {Считывание символа}
      Sound(700); {Генерация звука}
      Delay(500); {Задержка 0.5 сек}
      NoSound; {Прекращение генерации звука}
      {Если нажата функциональная клавиша}
      If Kod=#0 Then
        Begin
          Kod:= Readkey; {Считывание второго байта}
          If Kod=#72 Then {Если нажата клавиш <↑>}
            If K>1 Then K:=K-1 Else K:=3;
          If Kod=#80 Then {Если нажата клавиш <↓>}
            If K>1 Then K:=K-1 Else K:=3;
          End;
        End;
      End;
    Window(1,1,80,25); {Переход к полному экрану}
    ClrScr;
    GotoXY(33,15);
    Write('Выбран режим - ',K:1);
  End.

```

Рассмотрим работу этой программы. Для вывода на экран изображения меню выделяется соответствующая типизированная константа - массив Stor. Каждый элемент этого массива содержит соответствующую строку из меню.

Сначала устанавливается текстовый режим с размером окна 5\*20. Задается цвет и фон выводимых символов (TextBackGround, TextColor).

Переменная K принимает значения, равное номеру выбранного режима (предполагается, что вначале K=1). Затем выводится заглавие меню с приглашением выбрать нужный режим.

Индикация строки, соответствующей выбранному из меню варианту, обеспечивается в цикле for .., в котором при условии I=K (где I - управляющая переменная цикла) изменяется цвет и фон выводимых символов.

Затем ожидается нажатие клавиши <↑> и <↓> на клавиатуре (ReadKey). Каждое нажатие клавиши будет сопровождаться звуковым сигналом, продолжительностью 0,5 сек. (Sound, Delay, NoSound). Если будет нажата клавиша <↑> и выделенной на этот момент будет первая строка, то произойдет переход к последней строке из меню и т.п.

Как только будет нажата клавиша Enter (Kod=#13), произойдет очистка экрана и будет выдано сообщение о номере выбранного варианта.

## ВАРИАНТЫ ЗАДАНИЙ

1. Ввести массив из 5 чисел. Упорядочить массив и вывести данные в окно. После нажатия клавиши < пробел > перейти в режим 25\*40 и снова вывести упорядоченный массив в окно.

Выход из программы - нажатием клавиши < Enter >.

При появлении окон должен звучать звуковой сигнал.

2. Ввести массив из 10 чисел. Все четные числа вывести в одном окне, в строку; все нечетные числа - во втором окне, в столбик. Второе окно должно появиться через 2 сек после первого. При появлении второго окна звучит звуковой сигнал.

Выход из программы - нажатием клавиши Esc.

3. Ввести массив из 7 чисел. Организовать окно и вывести в него массив по следующему образцу:

	4
	3 5
	2 6
1	7

(Был введен массив чисел от 1 до 7).

Через 2 сек должно появиться второе окно, в котором выведены max и min числа массива.

В промежутке между появлением первого и второго окна звучит звуковой сигнал.

Выход из программы - нажатием клавиши Enter.

4. Ввести массив из 10 чисел. Вывести 3 наибольших числа массива в трех разных окнах.

Окна появляются друг за другом через 2 сек, в течении которых звучит звуковой сигнал.

При решении задачи используется текстовый режим C40.

Выход из программы - нажатием любой клавиши ↑, ↓, ←, →.

5. Ввести массив из 10 чисел. Упорядочить массив. Первую его половину вывести в одном окне, вторую - в другом окне. Выводимые данные должны мерцать. Второе окно появляется через 2 сек после первого.

По окончании работы программы звучит приятный звуковой сигнал. Выход из программы - нажатием клавиши <Enter>.

6. Ввести массив из 10 чисел. Найти минимальное, максимальное и среднее значение. Эти значения выдаются в различных окнах, окна появляются случайным образом в различных местах экрана через 2 сек. При появлении каждого из окон звучит звуковой сигнал разной тональности.

Выход из программы - нажатием любой буквенной клавиши.

7. Ввести массив из 7 символов. Организовать 3 окна, в одном - цифры, в другом - буквы, в третьем - знаки, которые содержались во введенном массиве.

Окна появляются поочередно через 3 сек., каждое со своим звуковым сигналом.

Выход из программы - нажатием клавиши Esc.

8. Ввести массив из 7 чисел.

Перейти в режим C40. Выдать на экран числа в порядке возрастания, каждое в своем окне.

Место появления и цвет фона окон задаются случайным образом.

Окна появляются через 2 сек. друг за другом.

По окончании работы программы звучит звуковой сигнал. Выход из программы - нажатием клавиши <пробел>.

9. Ввести строчку из букв.

Гласные буквы вывести в одном окне, по диагонали окна; согласные - в другом окне, также по диагонали.

Второе окно должно появиться через 3 сек. после первого.

По окончании работы программы звучит звуковой сигнал.

Выход из программы - нажатием клавиши Esc.

10. Ввести строку из букв. Упорядочить буквы по алфавиту. Выдать каждую букву на экран в своем небольшом окне. Окна появляются через 1 сек. каждое, место их появления случайное. После выдачи всех данных звучит звуковой сигнал. Выход из программы звуковой сигнал.

11. Ввести массив из 10 чисел. Упорядочить массив по возрастанию. Вывести результат в окне, расположенном в нижней части экрана. Каждое значение упорядоченного массива появляется через 2 сек. при этом звучит звуковой сигнал который при появлении каждого нового значения становится все выше.

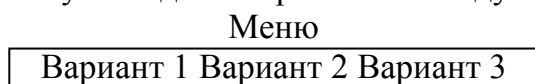
Выход из программы - нажатие клавиши <ESC>.



12. Ввести небольшой текст. Открыть окон в режиме 25\*40 и организовать вывод текста в режиме "бегущей строки". После вывода всего текста включить звуковой сигнал на 2 сек.

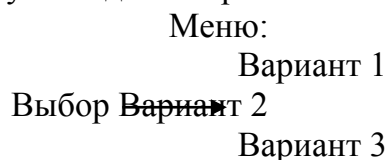
13. Разработать программу, которая при нажатии клавиши <F9> выводит на экран текстовое окно. При нажатии клавиши <<-> сдвигается влево левая рамка окна, при нажатии <-> - вправо правая рамка окна, при нажатии <↓> - вниз нижняя рамка окна, при нажатии <↑> вверх верхняя рамка окна.

14. Разработать программу вывода на экран меню следующего вида:



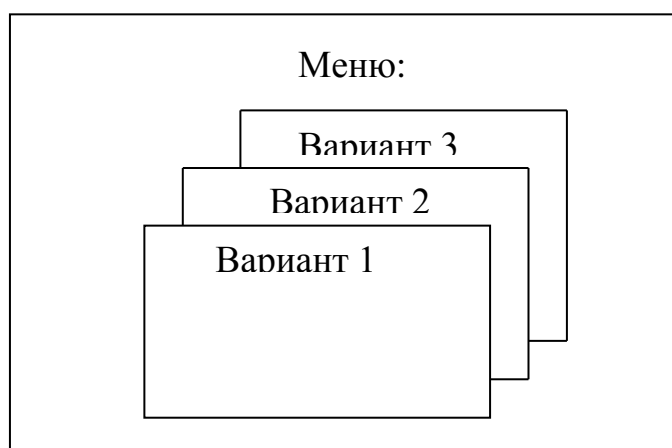
Отображающего процесс выбора любого из перечисленных вариантов с помощью <<-> и <->. При этом вариант, на который падает выбор, должен выделяться повышенной яркостью или цветом. При нажатии клавиши <Enter> или <↓> на месте соответствующего варианта должно появляться другое окно.

15. Разработать программу вывода на экран меню следующего вида:



Отображающего процесс выбора одного из вариантов с помощью клавиш <↑> и <↓>. При этом к строке, на которую падает выбор, должна быть направлена стрелка от слова выбор.

16. Разработать программу вывода на экран меню следующего вида:



В этом меню приведен полный список вариантов. Название каждого варианта оформляется в отдельном окне. Полностью видно только одно окно, на который падает выбор. При нажатии определенных клавиш порядок окон меняется. Разработанная программа должна отображать этот процесс выбора варианта.

17. Ввести массив из 10 чисел. Найти три минимальных числа этого массива.

При нажатии клавиши 1 в новом окне выводится первый минимум и звучит звуковой сигнал. При нажатии клавиши 2 открывается новое окно и выводится второй минимум, при этом также звучит звуковой сигнал. Аналогичные действия произвести с клавишей 3. Открытие новых окон определяется случайным образом. Выход из программы - нажатием клавиши Esc.

18. Создать окно и ввести в него небольшой текст. Слова разделять пробелами.

Организовать новое окно, в котором будет выводиться этот текст следующим образом: слова появляются друг за другом с интервалом 3 сек. В течении этих трех секунд звучит звуковой сигнал.

19. Создать окно и ввести в него небольшой текст. Организовать редактирование текста с помощью клавиш Del, Home, End, Esc, ,обеспечивающих следующие функции:

- передвижение курсора по тексту влево и вправо соответственно;
- Home, End - курсор в начало и конец текста соответственно;
- Del - удаление символа над курсором;
- удаление символа перед курсором;
- Esc - выход из программы.

20. Составить программу, выводящую текстовое окно случайным образом. В окне должен выводиться номер группы. Окно должно исчезать через 2 сек. и появляться в новом случайном месте.

При появлении окна звучит звуковой сигнал. Выход из программы нажатием клавиши - Esc.