



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ
И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Кафедра «Естественные науки»

ИНФОРМАТИКА

РАБОТА С ТЕКСТАМИ В TURBO PASCAL

Методические указания к лабораторно-практическим работам
для иностранных слушателей дополнительных
общеобразовательных программ

Авторы

Моренко Б.Н.
Бабакова Л.Д.
Воскерчян О.М.

Ростов-на-Дону, 2017



ОГЛАВЛЕНИЕ

	с.
Аннотация	3
Методические указания	4
Задача 1	6
Задача 2	7
Задача 3	8
Задача 4	9
Задача 5	9
Задача 6	10
Задача 7	11
Задача 8	11
Задача 9	12
Задача 10	12
Задача 11	14
Задача 12	15
Задача 13	16
Задача 14	16
Задачи для самостоятельного решения	17
Литература и Интернет-источники	18





АННОТАЦИЯ

Методические указания к лабораторно-практическим работам по информатике предназначены для иностранных слушателей дополнительных общеобразовательных программ инженерно-технической и технологической направленности обучения. Содержат теоретические сведения, практические рекомендации и типовые задачи по работе с текстами в Turbo Pascal, а также дополнительные задания для самостоятельной работы.

Рекомендуются для лабораторно-практических занятий и самостоятельной работы иностранных слушателей по информатике.

Рецензент преп. Т.А. Тюрина

Авторы:

Моренко Б.Н. – к.т.н., доцент кафедры
«Естественные науки»

Бабакова Л.Д. – доцент кафедры
«Русский язык как иностранный»

Воскерчян О.М. – к.п.н., доцент кафедры
«Русский язык как иностранный»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

В текстовом режиме работы монитора управление осуществляется не каждой отдельной точкой экрана, как в графическом режиме работы, а сразу целой группой точек, образующих прямоугольную матрицу. В такой матрице некоторые точки имеют цвет фона, и они не участвуют в формировании изображения. Другие точки имеют иной цвет, называемый цветом символа, и из них формируется изображение символов (букв, цифр, знаков и т.д.). Местоположение прямоугольной матрицы на экране называется позицией символа. В текстовом режиме весь экран разделяется на позиции, сгруппированные в горизонтальные строки. Для каждой позиции следует указывать: вид отображаемого символа, цвет символа и цвет фона.

Изображаемый символ указывается его номером по кодовой таблице. Для этого требуется один байт памяти. Цвет символа и цвет фона располагаются в другом байте памяти, и называются атрибутами символа. Таким образом, для каждого символа, отображаемого на экране монитора необходимо иметь два байта памяти.

Каждая позиция экрана монитора нумеруется в пределах строки слева направо, начиная с 1. Строки нумеруются сверху вниз, также начиная с 1. Наиболее часто применяют матричные шрифты с размерами матрицы 14x8 точек (основной шрифт). Первое число в обозначении шрифта означает количество точек в матрице по вертикали, второе число - по горизонтали. Каждая строка содержит 80 позиций, т.е. на строке может быть отображено 80 символов. Количество строк на экране монитора - 25.

Структура экрана монитора в текстовом режиме 80x25 символов показана на Рис. 1. В позиции с координатами 1,1 изображен символ **D**, в позиции 80,1 - символ **N**, в позиции 1,25 - символ **G**. Символ **F** выведен в позиции 40,13.

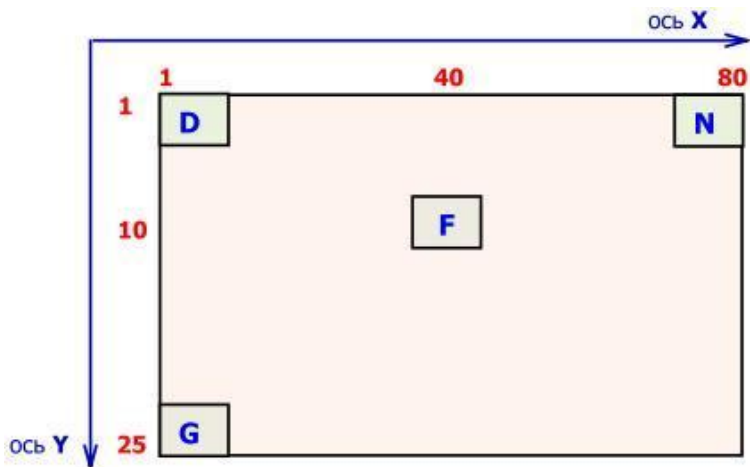


Рис. 1. Структура экрана в текстовом режиме 80x25

Управление или использование возможностей различных устройств компьютера в системе Turbo Pascal осуществляется стандартными подпрограммами, которые собраны в библиотеки, которые называются модулями.

Модуль *CRT* (*Console Routine*) обеспечивает управление вводом информации и ее выводом на экран дисплея в текстовом режиме (установку цвета, определение позиции для выводимой информации, взаимодействие с таймером, воспроизведение звука встроенным динамиком и т.д.).

Одной из процедур модуля *CRT* является *Procedure TextMode (Mode:Word)*, которая устанавливает заданный текстовый режим, очищает экран монитора и устанавливает курсор в левый верхний угол, имеющий координаты 1,1. Задаваемое значение параметра *Mode* определяет режим работы экрана монитора. Для того, чтобы стали доступными функции и процедуры модуля *CRT*, необходимо указать модуль *CRT* командой *Uses CRT*.

Имя константы	Значение константы	Описание режима
BW40	0	40 символов x 25 строк, черно-белый режим
Co40	1	40 символов x 25 строк, цветной режим
BW80	2	80 символов x 25 строк, черно-белый режим
Co80	3	80 символов x 25 строк, цветной режим
Font8x8	256	43 строки для EGA, 50 строк для VGA

Табл. 1. Режимы вывода текста на экран монитора

Программа *UNIT_01.pas* демонстрирует возможности использования модуля *CRT* по управлению режимами работы экрана монитора. С помощью процедуры *TextMode (Co40)* устанавливается цветной режим работы монитора и обеспечивается вывод 25 строк по 40 символов в каждой. Установив, например, процедуру *Co80* мы обеспечим вывод 25 строк по 80 символов в каждой.

```

program Demo_CRT; {Listing_1.pas}
uses Crt;
begin
    ClrScr;
    TextMode (Co40);
    Write ('Текстовый режим работы монитора');
    ReadLn;
end.

```

Изменить режим вывода текста *TextMode*, можно, используя значения констант из Табл. 1. Значение константы задается её числовым значением или именем, например, *Co40*, *BW80* и т. д.

Процедура *TextColor (Color:Byte)* определяет цвет символов, выводимых на экран монитора. Константа цвета *Color* изменяется в пределах от 0 до 15 (Табл. 2). Каждой константе соответствует свой цвет.

Цвет символов	Описание цвета	Номер цвета
Черный	Black	0
Синий	Blue	1
Зеленый	Green	2
Бирюзовый	Cyan	3
Красный	Red	4
Розовый	Magenta	5
Коричневый	Brown	6
Светло-серый	LightGray	7
Темно-серый	DarkGray	8
Светло-синий	LightBlue	9
Светло-зеленый	LightGreen	10
Светло-бирюзовый	LightCyan	11
Светло-красный	LightRed	12
Светло-розовый	LightMagenta	13
Желтый	Yellow	14
Белый	White	15
Мерцание символов	Blink	128

Табл. 2. Константы цветов для текстового режима

Для вывода на экран монитора строки «Текстовый режим работы монитора», состоящей из букв жёлтого цвета, внесём в программу *Listing_1.pas* соответствующие изменения, включив в неё дополнительно процедуру *TextColor* с указанием цвета выводимых символов.

```

program Demo_Crt;      {Listing_2.pas}
uses Crt;
begin
    ClrScr;
    
```

```

TextMode (Co40);
TextColor (Yellow);
Write (' Текстовый режим работы монитора');
ReadLn;
end.
    
```

Для того, чтобы вызвать мерцание надписи «Текстовый режим работы монитора» строку *TextColor (Red)* в программе Listing_2.pas следует записать в виде: *TextColor (Red + Blink)*.

```

program Demo_TextMode;    {Listing_3.pas}
uses Crt;
begin
    ClrScr;
    TextMode (Co40);
    TextColor (Red + Blink);  {Мерцание строк}
    Write (' Текстовый режим работы монитора');
    ReadLn;
end.
    
```

Процедура *TextBackGround (Color:Byte)* позволяет устанавливать различные цвета экрана монитора или его выделенной части – текстового окна. Допустимые значения переменной *Color* колеблются (Табл. 3) от 0 (чёрный цвет) до 7 (белый цвет). По умолчанию цвет экрана монитора – чёрный.

Табл. 3

Цвет монитора	Описание цвета	Номер цвета
Чёрный	Black	0
Синий	Blue	1
Зелёный	Green	2
Бирюзовый	Cyan	3
Красный	Red	4
Розовый	Magenta	5
Коричневый	Brown	6
Светло-серый	LightGray	7

Табл. 3. Цвета экрана монитора



Программа Listing_4.pas позволяет вывести строку «Текстовый режим работы монитора» красного цвета на голубом фоне.

```
program Demo_Crt;  {Listing_4.pas}
uses Crt;
begin
  ClrScr;
  TextMode (Co40);
  TextBackGround (Blue);    {Голубой фон}
  TextColor (Red);
  Write ('Текстовый режим работы монитора');
  ReadLn;
end.
```

Программа Listing_5.pas позволяет изменять палитру цветов экрана монитора.

```
program Demo_Crt;  {Listing_5.pas}
uses Crt;
var i: integer;
begin
  ClrScr;
  For i:=0 to 7 do
  begin
    TextBackGround (i);
    ClrScr;
    Delay (2000); {Пауза при демонстрации каждого цвета}
  end;
end.
```

В предыдущих программах строка «Текстовый режим работы монитора» выводилась в левом верхнем углу экрана монитора. Для вывода строки в любом другом месте экрана служит процедура *GotoXY* ($x,y:Byte$). Указанная процедура перемещает курсор и, соответственно, начало строки в точку экрана монитора с координатами (x,y) .



Для вывода строки «Текстовый режим работы монитора» в точке с координатами $x=6$, $y=10$ нужно выполнить следующую программу:

```
program Demo_Crt;      {Listing_6.pas}
uses Crt;
begin
  ClrScr;
  TextMode (Co40);
  TextColor (Red);
  TextBackGround (Green);
  GotoXY (6,10);      {Вывод текста в заданном месте}
  Write ('Текстовый режим работы монитора');
  ReadLn;
end.
```

В предыдущих программах при организации паузы в работе персонального компьютера при выводе результатов работы программы на экран монитора мы использовали функцию *ReadLn*, которая стоит в конце программы перед *End*. Возвращение в среду программирования осуществлялось при нажатии клавиши «Enter». Вместе с тем, в языке Turbo Pascal имеется процедура *Delay (MS:Word)*, которая позволяет на каждом шаге выполнения программы делать паузу продолжительностью *MS* миллисекунд. Величина паузы задается как числовая константа. Её значение подбирается экспериментально в зависимости от требуемого времени демонстрации изображения.

Ниже приведена программа, которая позволяет наблюдать текст «Текстовый режим работы монитора» на экране в течение нескольких секунд с последующим возвращением в среду программирования.

```
program Demo_Crt;    {Listing_7.pas}
uses Crt;
begin
    ClrScr;
    TextMode (Co40);
    TextColor (Red);
    TextBackGround (Green);
    GotoXY (6,10);
    Write ('Текстовый режим работы монитора');
    Delay (2000); {Задержка времени выхода из программы}
end.
```

В процессе разработки программ возникает необходимость выделения на экране монитора текстового окна и работы с текстом внутри него. Для этого в Turbo Pascal имеется процедура *Window* ($X1, Y1, X2, Y2:Byte$), где $X1, Y1, X2, Y2$ - координаты вершин текстового окна. Окно может быть окрашено в цвет, отличный от цвета экрана монитора. Эту возможность языка Turbo Pascal демонстрирует следующая программа.

```
program Demo_Crt;    {Listing_8.pas}
uses Crt;
begin
    TextBackGround (Black);
    ClrScr;
    Window (1,1,80,7);    {Координаты окна}
    TextBackGround (Cyan);    {Цвет окна}
    ClrScr;
    TextColor (Red);
    GotoXY (25,4);
    Write ('Текстовый режим работы монитора');
    ReadLn;
end.
```

В языке Turbo Pascal имеется возможность изменения яркости символов. Процедура *HingVideo* устанавливает повышенную яркость символов, *NormVideo* – нормальную яркость и *LowVideo* – пониженную яркость. Программа Listing_9.pas



демонстрирует возможность управления яркостью символов. Малая яркость символов соответствует цветам от 0 до 7 (тёмные цвета), высокая яркость – цветам от 8 до 15 (светлые цвета). Нормальная яркость цвета устанавливается по умолчанию операционной системой.

```
program Demo_Crt;      {Listing_9.pas}
uses Crt;
begin
  ClrScr;
  TextMode (Co80);
  TextColor (Green);
  HighVideo; {Повышенная яркость символов}
  GotoXY (25,4);
  Write ('Текстовый режим работы монитора');
  ReadLn;
end.
```

Рассмотрим несколько программ, которые позволяют использовать возможности языка программирования Turbo Pascal для обработки числовой и текстовой информации.

Задача 10. Составить программу, которая позволяет последовательно запрашивать и выводить на экран монитора названия дней недели по их порядковым номерам.

```
program Listing_10;
uses crt;
label M1;              {Определение метки}
var
  day : shortint;
begin
  ClrScr;
  M1:                   {Метка M1}
  WriteLn;
  Write ('Введите порядковый номер дня недели от 1 до 7. => ');
  ReadLn (day);
  Write (day);        {Вывод на экран монитора номера дня недели}
  case day of
```

```
1: WriteLn (' - Понедельник');
2: WriteLn (' - Вторник');
3: WriteLn (' - Среда');
4: WriteLn (' - Четверг');
5: WriteLn (' - Пятница');
6: WriteLn (' - Суббота');
7: WriteLn (' - Воскресенье');
0:      {Для выхода из программы введите цифру "0"}
      begin
          halt;
      end
      else
WriteLn ('Такого дня недели не существует! Повторите ввод!');
      end;      {Конец выполнения оператора case...of}
goto M1      {Переход на метку M1}
end.
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят выводить на экран монитора результаты выполнения программы в виде текста: «Первый день недели – понедельник», «Второй день недели – вторник» и т.д.
2. Внесите изменения в программу, которые позволят выводить на экран монитора название месяца (Январь и т.д.) по его порядковому номеру.
3. Внесите изменения в программу, которые позволят выводить на экран монитора названия времён года по порядковому номеру месяца, например, «4 – Весна».
4. Внесите изменения в программу, которые позволят выводить на экран монитора цифры от 0 до 10 и писать их названия на русском и английском языках, например, «1 – один, one»; «2 – два, two» и т.д.

Задача 11. Составить программу, которая дописывает на кассовом чеке после суммы покупки слово «рубль» в нужном падеже.

```
program Listing_11;
var
  n, r : integer;
begin
  WriteLn ('Введите сумму покупки, n=');
  ReadLn (n);
  if n>100
  then n:=n mod 100;
    if (n>= 11) and (n<=14)
    then
      WriteLn (' рублей')
    else
      begin
        r:=n mod 10;
        case r of
          0, 5..9 : WriteLn (n, ' рублей');
          1 : WriteLn (n, ' рубль');
          2..4 : WriteLn (n, ' рубля');
        end;
        {Конец выполнения оператора case...of}
      end;
  ReadLn;
end.
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят дописывать на чеке после суммы покупки слово «доллар» в нужном падеже.
2. Внесите изменения в программу, которые позволят выводить на экран монитора результаты работы программы в следующем виде: «Сумма вашей покупки составляет 51 рубль».

Задача 12. Составить программу, которая по введённому с клавиатуры символу определяет, что введено: цифра, прописная или строчная буква латинского или русского алфавита (кириллица).

```

program Listing_12;
uses crt;
var
    N : char;    {Вводимый символ}
begin
    ClrScr;
    Write ('Введите символ N=> ');
    ReadLn (N);
    case N of
        '0'..'9' : WriteLn (N, ' - это цифра');
        'A'..'Z' : WriteLn (N, ' - это прописная буква латинского алфавита');
        'a'..'z' : WriteLn (N, ' - это строчная буква латинского алфавита');
        'А'..'Я' : WriteLn (N, ' - это прописная буква русского алфавита');
        'а'..'я' : WriteLn (N, ' - это строчная буква русского алфавита');
    end;        {Конец выполнения оператора case...of}
    WriteLn ('Для выхода в рабочее окно нажмите клавишу Enter');
    ReadLn
end.
    
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят выводить на экран монитора надпись: «Это не цифра».
2. Внесите изменения в программу, которые позволят выводить на экран монитора надпись: «Это не буква».
3. Внесите изменения в программу, которые позволят выводить на экран монитора надпись: «Это не буква и не цифра. Это другой символ» при вводе неизвестных символов.
4. Внесите изменения в программу, которые позволят дополнительно выводить на экран монитора названия введённых символов:
 - - это точка, , - это запятая, ' - это апостроф,
 - : - это двоеточие, ; - это точка с запятой, / - это слеш,

Задача 13. Составить программу, которая по введённым с клавиатуры латинским буквам «Y» или «y» и «N» или «n» выводит на экран монитора фразы: «Этот символ означает Да» или «Этот символ означает Нет».

```
program Listing_13;
var
  ch : char ;
begin
  Write ('Введите символы «Y», «y» или «N», «n» =>');
  ReadLn (ch);
  case ch of
    'Y', 'y' : WriteLn ('Этот символ означает <Да>');
    'N', 'n' : WriteLn ('Этот символ означает <Нет>');
  end
  {Конец выполнения оператора case...of}
end.
```

Задача 14. Составить программу, которая по введённым с клавиатуры цифрам пишет их названия на русском языке, например, «0 – это цифра ноль».

```
program Listing_14;
var
  x : shortint;
begin
  Write ('Введите любую цифру от 0 до 9. => ');
  ReadLn (x);
  case x of
    0: WriteLn (x, ' – это цифра ноль');
    1: WriteLn (x, ' – это цифра один');
    2: WriteLn (x, ' – это цифра два');
    3: WriteLn (x, ' – это цифра три');
    4: WriteLn (x, ' – это цифра четыре');
    5: WriteLn (x, ' – это цифра пять');
    6: WriteLn (x, ' – это цифра шесть');
    7: WriteLn (x, ' – это цифра семь');
```




```
      8: WriteLn (x, ' – это цифра восемь');  
      9: WriteLn (x, ' – это цифра девять');  
    else  
      Write ('Это не цифра! Это число');  
    end;  
  ReadLn  
end.
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят выводить на экран монитора названия чисел от 20 до 29 на русском языке, например, «**24 – это число двадцать четыре**».
2. Внесите изменения в программу, которые позволят выводить на экран монитора названия чисел от 145 до 154 на русском языке, например, «**148 – это число сто сорок восемь**».
3. Внесите изменения в программу, которые позволят выводить на экран монитора названия чисел от 1250 до 1259 на русском языке, например, «**1256 – это одна тысяча двести пятьдесят шесть**».

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

- Задача 1. Составить программу, которая по заданному номеру месяца (1 – январь, 2 – февраль, 3 – март и т.д.) выводит название соответствующего времени года («Зима», «Весна», «Лето», «Осень»).
- Задача 2. Составить программу, которая по заданному номеру месяца (1 – январь, 2 – февраль, 3 – март и т.д.) выводит количество дней в указанном месяце.
- Задача 3. Составить программу, которая выводит словесное описание оценки, полученной студентом:
2 – «неудовлетворительно», 3 – «удовлетворительно», 4 – «хорошо», 5 – «отлично».

- Задача 4. Дано целое число в диапазоне от 1 до 99, которое соответствует возрасту человека в годах. Составить программу, которая по заданному возрасту выводит его описание на русском языке в нужном падеже, например, «Мне 41 год», «Мне 23 года», «Мне 18 лет».
- Задача 5. Составить программу, которая после введённой с клавиатуры денежной единицы от 1 до 99, дописывает слово «копейка» в нужном падеже, например, 1 копейка, 43 копейки, 65 копеек и т. д.

ЛИТЕРАТУРА И ИНТЕРНЕТ-ИСТОЧНИКИ

1. Культин Н.Б. Turbo Pascal в задачах и примерах / Н.Б. Культин. – СПб.: БХВ - Санкт-Петербург, 2000. – 256 с.: ил.
2. Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi / Второе издание, переработанное и дополненное. – СПб.,: БХВ – Санкт-Петербург, 1999. – 416 с., ил.
3. Меженный О.А. Turbo Pascal: учитесь программировать / О.А. Меженный – М.: Издательский дом «Вильямс», 2001. – 448 с.: ил.
4. Моренко Б.Н. Оператор выбора. Методические указания для практических работ по языку программирования Turbo Pascal для иностранных студентов предвузовской подготовки. – Ростов н/Д: Издательский центр ДГТУ, 2010. – 11 с.
5. Оператор выбора. Методические указания для практических работ по языку программирования Turbo Pascal для иностранных студентов предвузовской подготовки [Электронный ресурс] / ДГТУ. Каф. «Естественные науки»; сост.: Б.Н. Моренко. – Режим доступа:
<http://de.donstu.ru/CDOCourses/1cd291ff-1d0b-463e-b1ee-143fdd5033fe/956/634/index.html>



6. Рудаков П.И., Федотов М.А. Основы языка Pascal. – М.: Радио и связь, Горячая линия – Телеком, 2000. – 208 с.: ил. 2-ое издание, исправленное.
7. Соболев Б.В. Информатика: Учебник / Б.В. Соболев А.Б. Галин, Ю.В. Панов, Е.В. Рашидова, Н.Н. Садовой. – Ростов н/Д: Феникс, 2005. – 448 с.
8. Шауцукова Л.З. Информатика. Теория (с задачами и решениями). Режим доступа: <http://book.kbsu.ru/theory/index.html>
9. Шпак Ю.А. Turbo Pascal. Просто как дважды два / Ю.А. Шпак. – М.: Эксмо, 2008. – 400 с.