

1. Это первый раздел. Первое предложение раздела, начинающегося со «@*», идёт в оглавление. Предложение должно заканчиваться точкой и быть не слишком длинным, иначе получите сообщения об ошибках. Разделы со звёздочкой всегда начинаются с новой страницы.

2. Если раздел без заголовка, то он начинается со знака «@», за которым следует пробел или возврат каретки.

Если Вам требуется в тексте символ «@», то в web-файле его нужно удваивать.

Если нужно в тексте отформатировать кусочек программного кода, но при этом он не должен идти в программу, то его заключают между двумя вертикальными линиями. Например, этот кусочек « $x:=y \operatorname{div} 2$;|» после форматирования будет выглядеть так « $x \leftarrow y \operatorname{div} 2$;».

Программа начинается с команды «@p» (безымянного модуля). Безымянных модулей может быть несколько, они последовательно объединяются в один большой кусок.

⟨Директивы компилирования 7⟩

program *web_exam*;

var *i*: *integer*; { Определим какой-нибудь индекс }

⟨Прочие переменные 3⟩

⟨Процедуры 4⟩

3. Здесь в безымянном модуле между скобками «@<...@>» пишутся названия именованных модулей. Всё содержимое безымянного модуля подставляется в программу при каждом его упоминании. Название может занимать несколько строк (но не очень много).

Если записывать имена модулей в программе и не ставить между ними точку с запятой, то они будут записываться подряд, что на печати выглядит не очень хорошо. Поэтому, целесообразно их разделять командой «@;», как показано выше.

Один модуль может описываться в нескольких разделах программы, и каждое такое описание начинается с «@<...@>». Например, так

⟨Прочие переменные 3⟩ ≡

j: *integer*; { ещё один индекс }

См. также раздел 5.

Этот код используется в разделах 2 и 4.

4. Имя модуля можно не прописывать до конца полностью, т. е. вместо «@<Процедуры@>» записать «@<Проц. . .@>». Урезанное имя должно быть достаточной длины, чтобы подходить только к одному модулю. После обработки weave все сокращённые имена будут представлены полными. Отметим, что имя модуля не должно быть начальной частью имени другого модуля, иначе будет выдано сообщение об ошибке.

⟨Процедуры 4⟩ ≡

procedure *Печатай_хвост*(*s*: *string*; *om_noz*: *integer*); { Печатает хвост строки *om_noz* до конца }

var ⟨Прочие переменные 3⟩ { Так мы определим нашу переменную *j* }

begin for *j* ← *om_noz* **to** *length*(*s*) **do**

begin *write*(*s*[*j*]);

end;

end;

См. также раздел 8.

Этот код используется в разделе 2.

5. ⟨Прочие переменные 3⟩ +≡

k: *integer*;

6. Когда нужно определить несколько именованных модулей подряд без текстовых описаний, нам требуется просто записать @ @<...@>, как это только что показано выше.

7. Вообще, из программы комментарии удаляются, поскольку они не важны для компилятора. Однако в паскале есть особый вид комментариев, устанавливающий директивы компиляции. Для них в web используются особые скобки: «@{...@}».

⟨Директивы компиличования 7⟩ ≡
 @{\$coperators +@}

Этот код используется в разделе 2.

8. В web есть скромные средства поддержки макросов. Макросы объявляются одним из следующих способов

«@d имя=константа »
 «@d имя==команды »
 «@d имя(#)=команды с параметром »

Имя макроса должно быть длиной не меньше двух символов. В именах различаются прописные и строчные буквы. Константы отделяются одним знаком «=», макроопределения двумя знаками «==». Макросы могут содержать один параметр, задаваемый в скобках знаком #. Можно задавать форматирование макроса командой «@f»:

«@f имя==имя команды, по образцу которой форматировать »

Сейчас эти макросы опишем и добавим ещё какую-нибудь функцию

```
define десять = 10 { Пример константы, только один знак «=» }
define пиши_десять ≡ write('десять_суть_',десять); { должно быть два знака равенства }
define прибавь_десять(#) ≡ # ← # + 10; { прибавляет к переменной 10 }
define доли_в_процентах ≡ *100/
format доли_в_процентах ≡ div { форматирование такое же, как и у div }
```

⟨Процедуры 4⟩ +≡

```
procedure ещё_процедура(x: real);
begin прибавь_десять(x); x ← 3 доли_в_процентах 4;
end;
```

9. Иногда требуется дословно передать в программу часть текста без изменений, тогда он заключается между командами «@=...@>».

Чтобы вставить разрыв в программу на паскале используется команда «@\», а в текст программы на TEX — «@/»

```
begin [Текст_без_изменений] i ← 2;
j ← 3; @\k ← i - j; Печатай_хвост('Одна_строка', 5); пиши_десять; readln;
end.
```

10. Конечно, полученная нами программа абсолютно бесполезна, но цель данного файла — показать основные средства для написания программы на языке web, а не получить какой-то полезный продукт. Поэтому читателю стоит не ограничиваться просмотром готового web_exam.dvi (web_exam.pdf, web_exam.ps, или во что он его превратил), но и разобрать сам исходник web_exam.web.

Приведённые выше команды охватывают необходимый минимум для разработки программ в системе web, но ими возможности этой системы не ограничиваются. Если читатель найдёт для себя полезной данную методику написания программ, то ему стоит обратиться к более полным руководствам по системе литературного программирования.

Мы же на этом закончим. Нам осталось только вывести указатели, что и будет сделано в следующем разделе со звёздочкой.

к. т. н., Жуков И. Б.
Ленинград, 23 марта 2013 года

<http://ibzh.eko3.ru>,
ibzh@yandex.ru

11. Указатели. Если вы не объявите отдельный раздел под указатели, то они с новой страницы не начнутся, а пойдут продолжением предыдущего текста. Средства литературного программирования сами о заголовках для указателей не позаботятся.

десять: 8.

доли_в_процентах: 8.

ещё_процедура: 8.

от_поз: 4.

Печатай_хвост: 4, 9.

пиши_десять: 8, 9.

прибавь_десять: 8.

operators: 7.

div: 8.

i: 2.

integer: 2, 3, 4, 5.

length: 4.

readln: 9.

real: 8.

string: 4.

web_exam: 2.

write: 4, 8.

⟨ Директивы компиляции 7 ⟩ Используется в разделе 2.

⟨ Процедуры 4, 8 ⟩ Используется в разделе 2.

⟨ Прочие переменные 3, 5 ⟩ Используется в разделах 2 и 4.

	Раздел	Стр.
Это первый раздел	1	1
Указатели	11	4