Модули. Структура модулей. Стандартные модули ТР

Модуль - это подключаемая к программе библиотека ресурсов. Он может содержать описания типов, констант, переменных и подпрограмм.

Модули применяются либо как **библиотеки**, которые могут использоваться различными программами, либо **для разбиения сложной программы на составные части**.

Модули можно разделить на:

1. **Стандартные**, которые входят в состав системы программирования
2. **Пользовательские**, то есть создаваемые программистом.

Чтобы подключить модуль к программе, его требуется предварительно скомпилировать. Результат компиляции каждого модуля хранится на диске в отдельном файле с расширением **.tpu.**

Описание модулей

Исходный текст каждого модуля хранится в отдельном файле с расширением .pas.

Общая структура модуля:

```
unit имя;       { заголовок модуля }
interface       { интерфейсная секция модуля }
{ описание глобальных элементов модуля (видимых извне) }
implementation { секция реализации модуля }
{ описание локальных (внутренних) элементов модуля }
    begin          { секция инициализации }
                { может отсутствовать }
    end.
```

ВНИМАНИЕ!

Имя файла, в котором хранится модуль, должно совпадать с именем, заданным после ключевого слова unit.

Модуль может использовать другие модули, для этого их надо перечислить в операторе **uses**, который может находиться только непосредственно после ключевых слов **interface** или **implementation**.

Для сохранения скомпилированного модуля на диске требуется установить значение пункта **Destination** меню **Compile** в значение Disk. Компилятор создаст файл с расширением **.tpu**.

- **В интерфейсной секции модуля** определяют константы, типы данных, переменные, а также заголовки процедур и функций.

- **В секции реализации** описываются подпрограммы, заголовки которых приведены в интерфейсной части. Заголовок подпрограммы должен быть или идентичным указанному в секции интерфейса, или состоять только из ключевого слова procedure или function и имени подпрограммы. Для функции также указывается ее тип. Кроме того, в этой секции можно определять константы, типы данных, переменные и внутренние подпрограммы.

- **Секция инициализации** предназначена для присваивания начальных значений переменным, которые используются в модуле. Операторы, расположенные в секции инициализации модуля, выполняются перед операторами основной программы.

Пример №1

Оформим в виде модуля подпрограмму вычисления среднего арифметического значения элементов массива.

```
unit Average;
interface
    const n = 10;
    type mas = array[1 .. n] of real;
    procedure average(x : mas; var av : real);
implementation
    procedure average(x : mas; var av : real);
    var i : integer;
        begin
            av := 0;
            for i := 1 to n do av := av + x[i];
            av := av / n;
        end;
    end.
```

Использование модулей

Для использования в программе величин, описанных в интерфейсной части модуля, имя модуля указывается в разделе uses. Можно записать несколько имен модулей через запятую, например:

```
program example;
uses Average, Graph, Crt;
```

Поиск модулей выполняется сначала в библиотеке исполняющей системы, затем в текущем каталоге, а после этого - в каталогах, заданных в диалоговом окне **Options/Directories.**

Если в программе описана величина с тем же именем, что и в модуле, для обращения к величине из модуля требуется перед ее именем указать через точку имя модуля.

Стандартные модули Паскаля

В Паскале имеется ряд стандартных модулей, в которых описано большое количество встроенных констант, типов, переменных и подпрограмм. Каждый модуль содержит связанные между собой ресурсы.

Модуль System

Модуль содержит базовые средства языка, которые поддерживают ввод-вывод, работу со строками, операции с плавающей точкой и динамическое распределение памяти. Этот модуль автоматически используется во всех программах, и его не требуется указывать в операторе uses. Он содержит все стандартные и встроенные процедуры, функции, константы и переменные Паскаля.

Модуль Crt

Модуль предназначен для организации эффективной работы с экраном, клавиатурой и встроенным динамиком. При подключении модуля Crt выводимая информация посылается в базовую систему ввода-вывода (BIOS) или непосредственно в видеопамять.

Модуль Crt позволяет:

- выполнять вывод в заданное место экрана заданным цветом символа и фона;
- открывать на экране окна прямоугольной формы и выполнять вывод в пределах этих окон;
- очищать экран, окно, строку и ее часть;
- обрабатывать ввод с клавиатуры;
- управлять встроенным динамиком.

Модули Dos и WinDos

Модули Dos и WinDos содержат подпрограммы, реализующие возможности операционной системы MS-DOS - например, переименование, поиск и удаление файлов, получение и установку системного времени, выполнение программных прерываний и так далее. Эти подпрограммы в стандартном Паскале не определены. Для поддержки подпрограмм в модулях определены константы и типы данных.

Модуль Dos использует строки Паскаля, а WinDos - строки с завершающим нулем.

Модуль Graph

Модуль обеспечивает работу с экраном в графическом режиме.

Модуль Graph обеспечивает:

- вывод линий и геометрических фигур заданным цветом и стилем;
- закрашивание областей заданным цветом и шаблоном;
- вывод текста различным шрифтом, заданного размера и направления;
- определение окон и отсечение по их границе;
- использование графических спрайтов и работу с графическими страницами.

Модуль Strings

Модуль предназначен для работы со строками, заканчивающимися нуль-символом, то есть символом с кодом 0 (их часто называют ASCIIZ-строки). Модуль содержит функции копирования, сравнения, слияния строк, преобразования их в строки типа string, поиска подстрок и символов.

Для хранения ASCIIZ-строк используются массивы символов с нулевой базой, например:

```
var  str : array[0 .. 4000] of char;
     p: pChar;
```

Массивы символов с нулевой базой и указатели на символы совместимы.

Printer

Модуль, служащий для программного вывода на принтер.

Чтобы использовать любой из этих модулей, его достаточно подключить оператором uses.

Домашнее задание

РТ стр 50 – 54