**СТРУКТУРА МОДУЛЕЙ И ОСОБЕННОСТИ РАБОТЫ С НИМИ.**

[ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 1](#_Toc252111243)

[КОНТРОЛЬНЫЕ ВОПРОСЫ 3](#_Toc252111244)

[ВАРИАНТЫ ЗАДАНИЙ 3](#_Toc252111245)

# ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Наличие модулей в TURBO PASCAL позволяет программировать и отлаживать программу по частям, создавая библиотеки подпрограмм и данных, воспользоваться возможностями стандартных модулей, практически неограниченно увеличивать кодовую (содержащую коды команд) часть команд.

Модуль состоит из следующих частей:

* + заголовок модуля,
  + интерфейс модуля,
  + исполнительная часть модуля,
  + секция инициализации.

Заголовок модуля состоит из зарезервированного слова unit и идентификатора. Идентификатор модуля должен быть уникальным.

Пример заголовка: Unit MyModule.

Модуль должен быть помещен в файл, имя которого совпадает с именем модуля, а его расширение должно быть .PAS

Через интерфейс модуля осуществляется взаимодействие основной программы с модулем (модуля с модулем). В интерфейсе указываются константы, типы, переменные, процедуры и функции, которые могут быть использованы основной программой (модулем) при вызове этого модуля.

Интерфейс начинается словом interface. Далее после слова uses указываются имена модулей, которые используются данным модулем (необязательная часть). После этого могут быть: раздел объявления констант, раздел объявления типов, раздел объявления переменных, раздел объявления процедур и функций. Объявление процедуры может содержать директиву inline. В разделе объявления процедур и функций указываются лишь заголовки подпрограмм (за исключением тех подпрограмм, которые используют директиву inline). Сами подпрограммы приводятся в исполнительной части.

В интерфейсах различных модулей недопустимо цикличное обращение друг к другу, т.к. компилятор в этом случае не может установить связь.

Исполнительная часть включает в себя все подпрограммы модуля. Она может также включать локальные метки, константы, типы и переменные.

Исполнительная часть начинается словом implementation, затем после слова uses указываются имена модулей, которые используются подпрограммами данной исполнительной части (этот раздел необязательный). Если какой-то модуль уже указан в интерфейсе модуля, то в исполнительной части его повторять не следует. Далее могут быть: раздел объявления меток, раздел объявления локальных констант, раздел объявления локальных типов, раздел объявления локальных переменных. Затем идут описания подпрограмм модуля. При описании подпрограмм допустимо использовать сокращенные заголовки, как и в случае использования директивы forward.

В отличие от интерфейсов модулей в исполнительных частях допустимо циклическое обращение друг к другу, т.к. все равно взаимодействие осуществляется через интерфейсы, и здесь не возникает проблемы с установлением необходимых связей.

В некоторых случаях перед обращением к модулю следует провести его инициализацию (например, если необходимо установить связь с теми или иными файлами с помощью процедуры Assign, инициализировать какие-то переменные и т.д., то необходимые действия можно выполнить в секции инициализации модуля). Эта секция начинается словом begin, после которого идут исполняемые операторы, а затем помещается слово end (с точкой), например:

Begin

Assign(f, ‘file1.dat’);

End;

Следует иметь в виду, что операторы секции инициализации выполняются единственный раз в момент запуска программы.

Исходный текст модуля должен быть откомпилирован с помощью директивы Make подменю Compile и записан на диск. Результатом компиляции модуля является файл с расширением .TPU (Turbo Pascal Unit). Для подключения модуля к программе необходимо указать его имя в разделе описания модулей, например:

uses CRT, Graph;

В том случае, если имена переменных в интерфейсной части модуля и в программе, использующей этот модуль, совпадают, обращение будет происходить к переменной, описанной в программе. Для обращения к переменной, описанной в модуле, необходимо применить составное имя, состоящее из имени модуля и имени переменной, разделенных точкой.

Например, пусть имеется модуль, в котором описана переменная К:

unit M;

interface

var K: Integer;

implementation

.................

end.

Пусть программа, использующая этот модуль, также содержит переменную К:

Program P;

uses M;

var K: Char;

begin

.............

end.

Для того, чтобы в программе P иметь доступ к переменной K из модуля M, необходимо задать составное имя M.K.

Использование составных имен применяется не только к именам переменных, а ко всем именам, описанным в интерфейсной части модуля.

Рекурсивное использование модулей запрещено.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Из каких частей состоит модуль? Какие из них являются обязательными?

2. С какого слова начинается заголовок модуля?

3. Для чего нужен интерфейс?

4. В чем отличие между интерфейсом и исполнительной частью модуля?

5. Как организовать подключение того или иного модуля?

# ВАРИАНТЫ ЗАДАНИЙ

Программа, использующая модуль, который:

1) подсчитывает, сколько раз введенное с клавиатуры число встречается в массиве;

2) вычисляет сумму элементов матрицы, распложенных на главной диагонали;

3) проверяет, есть ли в матрице элементы с одинаковым значением;

4) считает среднее геометрическое элементов массива по столбцам;

5) вычисляет среднее арифметическое ненулевых элементов массива по строкам.