



**Модули  
ПРОДОЛЖЕНИЕ**

# СТРУКТУРА ПРОГРАММНОГО ПРОДУКТА

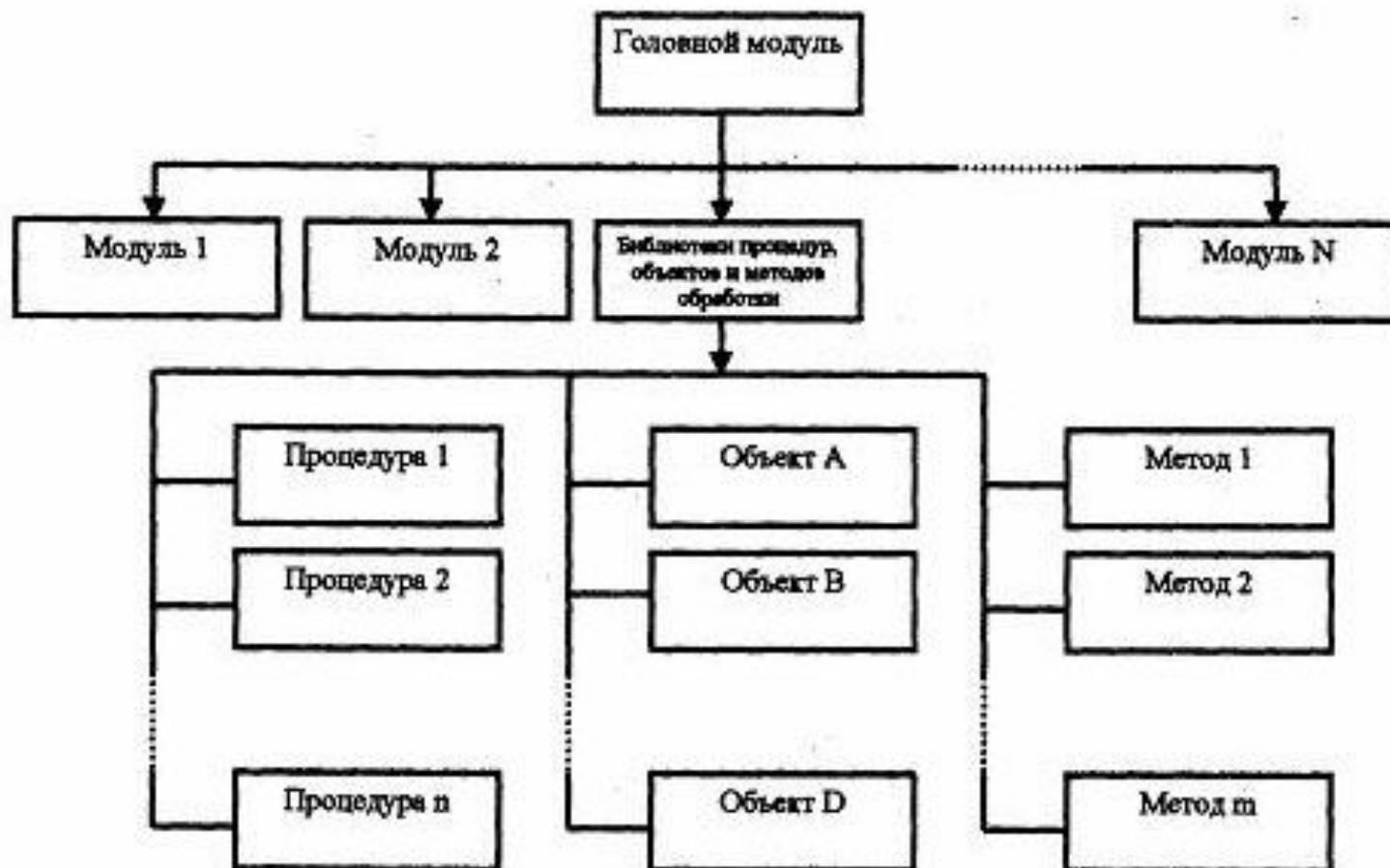
- Программные продукты имеют конструкцию (архитектуру) построения - состав и взаимосвязь программных модулей
- *Модуль — это самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки автономном от других программных модулей.*



- Структуризация программ выполняется для удобства разработки программирования, отладки и внесения изменений в программный, продукт.
- Некоторые программные продукты используют модули из готовых библиотек стандартных подпрограмм, процедур, функций, объектов, методов обработки данных



# - СТРУКТУРНАЯ СХЕМА ПРОГРАММНОГО ПРОДУКТА



## СРЕДИ МНОЖЕСТВА МОДУЛЕЙ РАЗЛИЧАЮТ:

- 
- - *головной модуль* - управляет запуском программного продукта (существует **в единственном числе**);
- - *управляющий модуль* — обеспечивает вызов других модулей на обработку;
- - *рабочие модули* — выполняют функции обработки;
- - *сервисные модули* и библиотеки, утилиты — осуществляют обслуживающие функции.



- В работе программного продукта активизируются необходимые программные модули. Управляющие модули задают последовательность вызова на выполнение очередного модуля. Информационная связь модулей обеспечивается за счет использования общей базы данных либо межмодульной передачи данных через переменные обмена.
- Каждый модуль может оформляться как самостоятельно хранимый файл; для функционирования программного продукта необходимо наличие программных модулей в полном составе.



# МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ

- Модульное программирование – это логически взаимосвязанная совокупность функциональных элементов, оформленных в виде отдельных программных модулей.
- Модуль характеризуют:
- Один вход и один выход, на входе модуль получает набор исходных данных выполняет обработку и возвращает один набор результатных данных т.е. реализует стандартную функцию Input-Process-Output.
- Функциональная завершенность – модуль выполняет перечень операций для реализации каждой отдельной функции с полным составе.



- Логическая независимость – результат работы программного модуля зависит от исходных данных и не зависит от работы других модулей.
- Слабые информационные связи с другими программными модулями – обмен информацией между модулями должен быть минимальным.
- Обозримый по размеру сложности программный элемент.
- Каждый модуль состоит из спецификации и тела. Спецификации определяют правила использования модуля, а тело – способ реализации процесса обработки.

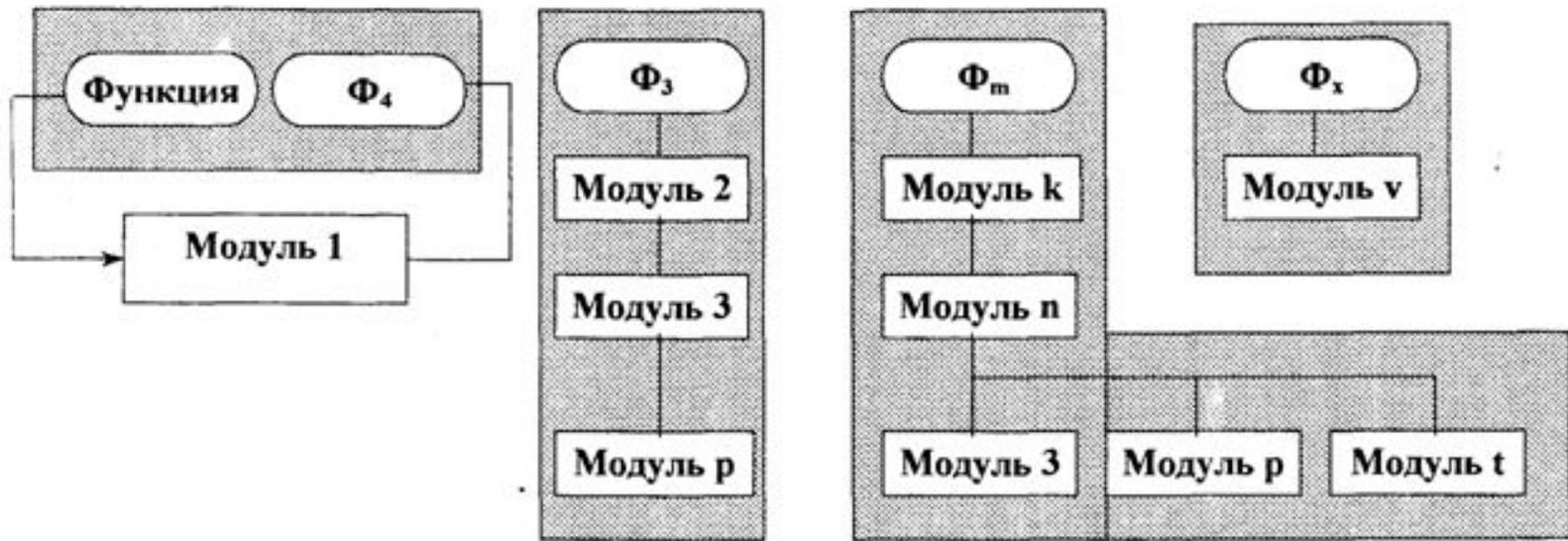


# ПРИНЦИПЫ МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ

- сначала определяется состав и подчиненность функции, а затем набор программных модулей реализующих эти функции.
- Функции верхнего уровня обеспечиваются главным модулем, он управляет выполнением нижестоящих функций, которым соответствуют подчиненные модули.
- При определении набора модулей необходимо учитывать:
  - каждый модуль вызывается на выполнение вышестоящим и закончив работу , возвращает управление вызвавшему его модулю;
  - принятие основных решений в алгоритме выносится на максимально высокий уровень по иерархии уровней.
  - для использования одной и той же функции в разных местах создается один модуль, который вызывается на выполнение по мере необходимости.
- В результате дальнейшей детализации алгоритма создается функциональная модульная схема (ФМС), которая является основой для программирования.



# ФУНКЦИОНАЛЬНО-МОДУЛЬНАЯ СТРУКТУРА ПРИЛОЖЕНИЯ



- Некоторые функции могут выполняться с помощью одного и того же программного модуля (например, функции  $\Phi_1$  и  $\Phi_2$ ).
- \* Функция  $\Phi_3$  реализуется в виде последовательности выполнения программных модулей.
- \* Функция  $\Phi_T$  реализуется с помощью иерархии связанных модулей.
- \* Модуль  $\Pi$  управляет выбором на выполнение подчиненных модулей.
- \* Функция  $\Phi_x$  реализуется одним программным модулем.

