

Методы проектирования ПО

Модульное программирование

Разрабатываемые ПС, как правило, являются большими программными продуктами, учитывая рассмотренный ранее принцип Keep It Simple Stupid (KISS) имеет смысл применять меры для упрощения пс.

Одним из вариантов является разработка по частям, которые называются программными модулями.

Сам такой метод разработки программ называют ***модульным программированием***.

Программный модуль – это любой фрагмент описания процесса, оформляемый как самостоятельный программный продукт, пригодный для использования в описаниях процесса.

Это означает, что каждый программный модуль программируется, компилируется и отлаживается отдельно от других модулей программы, и тем самым, физически разделен с другими модулями программы.

Важным свойством программного модуля является возможность включаться в состав разных программ, если выполнены условия его использования, декларированные в документации по этому модулю.

Таким образом, программный модуль может рассматриваться и как средство борьбы со сложностью программ, и как средство борьбы с дублированием в программировании.

?Сформулируйте и вставьте в конспект не менее 6 целей, определяющих, для чего создаются модули

К сожалению не всегда удастся создать именно хороший модуль, который бы одинаково успешно решал и возложенные на него задачи и справлялся со сложностью программ. Выделить хороший с этой точки зрения модуль является серьезной творческой задачей.

Для оценки приемлемости выделенного модуля используются различные критерии критерии.

По критериям Хольта:

- хороший модуль снаружи проще, чем внутри;
- хороший модуль проще использовать, чем построить

Майерс предлагает для оценки приемлемости программного модуля использовать более конструктивные его характеристики:

- размер модуля
- прочность модуля
- сцепление с другими модулями
- рутинность модуля (независимость от предыстории обращений к нему).

Основные характеристики программного модуля

1. Размер модуля измеряется числом содержащихся в нем операторов или строк.

Модуль не должен быть слишком маленьким или слишком большим. Маленькие модули приводят к громоздкой модульной структуре программы и могут не окупать накладных расходов, связанных с их оформлением. Большие модули неудобны для изучения и изменений, они могут существенно увеличить суммарное время повторных трансляций программы при отладке программы. Обычно рекомендуются программные модули размером от нескольких десятков до нескольких сотен операторов.

2. Прочность модуля – это мера его внутренних связей. Чем выше прочность модуля, тем больше связей он может спрятать от внешней по отношению к нему части программы и, следовательно, тем больший вклад в упрощение программы он может внести. Модуль может быть выделен, например, при обнаружении в разных местах программы повторения одной и той же последовательности операторов, которая и оформляется в отдельный модуль.

Основные характеристики программного модуля

Для использования рекомендуются **два высших** по прочности класса модулей:

1) Функционально прочный модуль – это модуль, выполняющий (реализующий) одну какую-либо определенную функцию.

При реализации этой функции такой модуль может использовать и другие модули.

2) Информационно прочный модуль – это модуль, выполняющий (реализующий) несколько операций (функций) над одной и той же структурой данных (информационным объектом), которая считается неизвестной вне этого модуля. Для каждой из этих операций в таком модуле имеется свой вход со своей формой обращения к нему. Такой класс следует рассматривать как класс программных модулей с высшей степенью прочности. Информационно прочный модуль может реализовывать, например, абстрактный тип данных. (ФУНКЦИЯ, ПРОЦЕДУРА).

Основные характеристики программного модуля

3. Сцепление модуля – это мера его зависимости по данным от других модулей. Характеризуется способом передачи данных. Чем слабее сцепление модуля с другими модулями, тем сильнее его независимость от других модулей.

Для оценки степени сцепления Майерс предлагает упорядоченный набор из шести видов сцепления модулей.

Худшим видом сцепления модулей является сцепление по содержимому. Таким является сцепление двух модулей, когда один из них имеет прямые ссылки на содержимое другого модуля (например, на константу, содержащуюся в другом модуле). Такое сцепление модулей недопустимо. Не рекомендуется использовать также сцепление по общей области – это такое сцепление модулей, когда несколько модулей используют одну и ту же область памяти.

Основные характеристики программного модуля

4. Рутинность модуля – это его независимость от предыстории обращений к нему.

Модуль будем называть *рутинным*, если результат (эффект) обращения к нему зависит только от значений его параметров (и не зависит от предыстории обращений к нему).

Модуль будем называть *зависящим от предыстории*, если результат (эффект) обращения к нему зависит от внутреннего состояния этого модуля, изменяемого в результате предыдущих обращений к нему. Не рекомендуется использовать зависящие от предыстории (непредсказуемые) модули, так как они провоцируют появление в программах хитрых (неуловимых) ошибок.

Основные характеристики программного модуля

Однако такая рекомендация вступает в противоречие с принципом информационной прочности модуля, так как во многих случаях именно зависящий от предыстории модуль является наиболее информационно прочен.

В этом случае можно воспользоваться более осторожной рекомендацией:

- всегда следует использовать рутинный модуль, если это не приводит к плохим (не рекомендуемым) сцеплениям модулей;
- зависящие от предыстории модули следует использовать только в случае, когда это необходимо для обеспечения параметрического сцепления;
- в спецификации зависящего от предыстории модуля должна быть четко сформулирована эта зависимость таким образом, чтобы было возможно прогнозировать поведение (эффект выполнения) данного модуля при разных последующих обращениях к нему.

Основные характеристики программного модуля

В связи с последней рекомендацией может быть полезным определение внешнего представления (ориентированного на информирование человека) состояний зависящего от предыстории модуля.

В этом случае эффект выполнения каждой функции (операции), реализуемой этим модулем, следует описывать в терминах этого внешнего представления, что существенно упростит прогнозирование поведения данного модуля.