Лекция 18. Структура программных модулей ИС

Учебные вопросы:

1. **Понятие модульной структуры программы.**
2. **Основные характеристики программного модуля.**
3. **Методы разработки структуры программы.**
4. **Контроль структуры программы.**

Литература: **[9], [23], [28].**

Понятие модульной структуры программы

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

Модульная структура программы – это древовидная структура, включая деревья со сросшимися ветвями. В узлах такого дерева размещаются программные модули, а направленные дуги показывают, что в тексте модуля, из которого они исходят, имеется ссылка на модули, в который они входят.

Основные характеристики программного модуля

- **Размер модуля** – число содержащихся в нем операторов (строк). Модуль не должен быть слишком маленьким или слишком большим. Маленькие модули приводят к громоздкой модульной структуре программы. Большие модули неудобны для изучения и изменений. Обычно рекомендуются программные модули размером от нескольких десятков до нескольких сотен операторов.
- **Связность модуля** – это мера зависимости его частей, внутренняя характеристика модуля. Чем выше связность модуля, тем лучше результат проектирования, то есть тем «черней» его ящик. Для измерения связности используют силу связности (СС). Существует 7 типов связности.
- **Сцепление модуля** – это мера его зависимости по данным от других модулей. Характеризуется способом передачи данных. Это внешняя характеристика модуля, которую желательно уменьшать. Чем слабее сцепление модуля с другими модулями, тем сильнее его независимость от других модулей. Количественно сцепление измеряется степенью сцепления (СЦ). Выделяют 6 типов сцепления.
- **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, хранящего следы предыдущих обращений к нему.

Методы разработки структуры программы

Методы разработки структуры программы

Метод восходящей разработки – сначала строится модульная структура программы в виде дерева. Затем поочередно программируются модули программы, начиная с модулей самого нижнего уровня, в таком порядке, чтобы для каждого программируемого модуля были уже запрограммированы все модули, к которым он может обращаться. После этого производится их поочередное тестирование и отладка в принципе в таком же (восходящем) порядке.

Метод нисходящей разработки – сначала строится модульная структура программы в виде дерева. Затем поочередно программируются модули программы, начиная с модуля самого верхнего уровня (головного), переходя к программированию какого-либо другого модуля только в том случае, если уже запрограммирован модуль, который к нему обращается. После того, как все модули программы запрограммированы, производится их поочередное тестирование и отладка в таком же (нисходящем) порядке.

Методы разработки структуры программы

Конструктивный подход – модификация нисходящей разработки, модульная древовидная структура программы формируется в процессе программирования модуля. Сначала программируется головной модуль, исходя из спецификации программы в целом. Причем спецификация программы является одновременно и спецификацией ее головного модуля.

Архитектурный подход – модификация восходящей разработки, модульная структура программы формируется в процессе программирования модуля. При этом повышается уровень используемого языка программирования, а не разработка конкретной программы. Это означает, что для заданной предметной области выделяются типичные функции, каждая из которых может использоваться при решении разных задач в этой области, и специфицируются, а затем и программируются отдельные программные модули, выполняющие эти функции.

Целенаправленная конструктивная реализация – на достаточно ранней стадии создается работающий вариант разрабатываемой программы.

Контроль структуры программы

Статический контроль – это оценка структуры программы с точки зрения хорошо ли программа разбита на модули с учетом значений рассмотренных выше основных характеристик модуля.

Смежный контроль сверху – это контроль со стороны разработчиков архитектуры и внешнего описания программной системы.

Смежный контроль снизу – это контроль спецификации модулей со стороны разработчиков этих модулей.

Сквозной контроль – это мысленное прокручивание (проверка) структуры программы при выполнении заранее разработанных тестов.