

МЕТОД НИСХОДЯЩЕГО ПРОЕКТИРОВАНИЯ*БНТУ, Минск**Научный руководитель Дробыш А.А.*

Проектирование является одной из основных фаз жизненного цикла программного обеспечения. Задачей этапа проектирования является исследование структуры системы и логических взаимосвязей ее элементов. На этапе проектирования создается структура будущей программы.

Современный подход к проектированию программ основан на декомпозиции задачи, которая в свою очередь основана на использовании абстракций. Целью при декомпозиции является создание модулей, которые представляют собой небольшие, относительно самостоятельные программы, взаимодействующие друг с другом.

В идеальной модульной программе любую часть логической структуры можно изменить, не вызывая изменений в ее других частях. Идеальная модульная программа состоит из независимых модулей, имеющих один вход и один выход.

Существует три группы классических методов проектирования модульных ПС: методы нисходящего проектирования; методы расширения ядра; методы восходящего проектирования.

Метод нисходящего проектирования позволяет обнаружить и исправить ошибки взаимосвязи блоков и логические ошибки на более ранних этапах программирования, когда внесение изменений еще не приводит к коренной перестройке всей программы. Основная идея метода нисходящего проектирования – не пытаться программировать сразу. Пошаговая детализация автоматически заставляет программиста формировать понятную ему же структуру программы. Нисходящее проектирование служит средством разбиения большой задачи на меньшие подзадачи так, чтобы каждую подзадачу можно было рассматривать независимо.

Суть метода нисходящего проектирования заключается в следующем. На начальном шаге в соответствии с общими функциональными требованиями к программному средству

разрабатывается его укрупненная структура без детальной проработки его отдельных частей. Затем выделяются функциональные требования более низкого уровня и в соответствии с ними разрабатываются отдельные компоненты программного средства, не детализированные на предыдущем шаге. Эти действия являются рекурсивными, то есть каждый из компонентов детализируется до тех пор, пока его составные части не будут окончательно уточнены.

Основными классическими стратегиями, на которых основана реализация метода нисходящего проектирования, являются: пошаговое уточнение; данная стратегия разработана Э. Дейкстрой; анализ сообщений; данная стратегия базируется на работах группы авторов (Йодана, Константайна, Мейерса).

Строгое следование принципам метода нисходящего проектирования может привести к тому, что процесс разработки всей программы продвинется весьма далеко, прежде чем выяснится, что компоненты нижних уровней не могут выполнить возлагаемые на них функции. В этом случае необходимо будет вернуться к верхним уровням и повторить проектирование и структурирование. Поскольку модули могут проектироваться независимо различными программистами, возможны ситуации, когда внутри двух или большего числа различных модулей возникнут потребности в одинаковых вычислениях, которые могли бы реализоваться одной и той же подпрограммой. Однако эта возможность может быть не использована, прежде всего, из-за того, что программисты могут этого не знать либо узнают слишком поздно, так что могут потребоваться слишком большие переделки. Этот недостаток в значительной степени преодолевается при правильном применении сквозного структурного контроля.

Подводя итоги можно сказать, что достоинство нисходящего проектирования состоит в том, что оно позволяет разработчикам сосредоточиться на основных проблемах и отложить принятие всех тех решений, которые не должны приниматься на данном этапе проектирования. Нисходящее проектирование требует с самого начала ставить и решать наиболее фундаментальные задачи, откладывая частные вопросы для последующего рассмотрения.