## Практическая работа № 2.

## Структурный подход к проектированию программного обеспечения.

Прочитав эту главу, вы узнаете:

· *Что представляет собой структурный подход к проектированию ПО.*

· *В чем заключается метод функционального моделирования SADT.*

· *Как строятся диаграммы потоков данных и "сущность- связь".*

### СУЩНОСТЬ СТРУКТУРНОГО ПОДХОДА

**1.1. ПРОБЛЕМА СЛОЖНОСТИ БОЛЬШИХ СИСТЕМ**

Проблема сложности является главной проблемой, которую приходится решать при создании больших и сложных систем любой природы, в том числе и ЭИС. Ни один разработчик не в состоянии выйти за пределы человеческих возможностей и понять всю систему в целом. Единственный эффективный подход к решению этой проблемы, который выработало человечество за всю свою историю, заключается в построении сложной системы из небольшого количества крупных частей, каждая из которых, в свою очередь, строится из частей меньшего размера и т.д., до тех пор, пока самые небольшие части можно будет строить из имеющегося материала. Этот подход известен под самыми разными названиями, среди них такие, как "разделяй и властвуй" *(dividi et impera),*иерархическая декомпозиция и др. По отношению к проектированию сложной программной системы это означает, что ее необходимо разделять (декомпозировать) на небольшие подсистемы, каждую из которых можно разрабатывать независимо от других. Это позволяет при разработке подсистемы любого уровня держать в уме информацию только о ней, а не обо всех остальных частях системы. Правильная декомпозиция является главным способом преодоления сложности разработки больших систем ПО. Понятие "правильная" по отношению к декомпозиции означает следующее:

· количество связей между отдельными подсистемами должно быть минимальным;

· связность отдельных частей внутри каждой подсистемы должна быть максимальной.

Структура системы должна быть таковой, чтобы все взаимодействия между ее подсистемами укладывались в ограниченные, стандартные рамки:

· каждая подсистема должна инкапсулировать свое содержимое (скрывать его от других подсистем);

· каждая подсистема должна иметь четко определенный интерфейс с другими подсистемами.

Инкапсуляция позволяет рассматривать структуру каждой подсистемы независимо от других подсистем. Интерфейсы позволяют строить систему более высокого уровня, рассматривая каждую подсистему как единое целое и игнорируя ее внутреннее устройство.

**1.1.2. СТРУКТУРНЫЙ ПОДХОД К РАЗРАБОТКЕ ПО**

На сегодняшний день в программной инженерии существуют два основных подхода к разработке ПО ЭИС, принципиальное различие между которыми обусловлено разными способами декомпозиции систем. Первый подход называют *функционально-модульным или структурным.*В его основу положен принцип функциональной декомпозиции, при которой структура системы описывается в терминах иерархии ее **функций** и передачи информации между отдельными функциональными элементами. Второй, *объектно-ориентированный*подход использует объектную декомпозицию. При этом структура системы описывается в терминах **объектов** и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами.

Итак, сущность структурного подхода к разработке ПО ЭИС заключается в его декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, те — на задачи и так далее до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. При разработке системы "снизу вверх", от отдельных задач ко всей системе, целостность теряется, возникают проблемы при описании информационного взаимодействия отдельных компонентов.

Все наиболее распространенные методы структурного подхода базируются на ряде общих принципов (Калянов Г.Н. Консалтинг при автоматизации предприятий: Науч.-практич. изд. Сер. Информатизация России на пороге XX века. — М.: СИНТЕГ, 1997). Базовыми принципами являются:

· *принцип "разделяй и властвуй"*(см. подразд. 2.1.1);

· *принцип иерархического упорядочения —*принцип организации составных частей системы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Выделение двух базовых принципов не означает, что остальные принципы являются второстепенными, поскольку игнорирование любого из них может привести к непредсказуемым последствиям (в том числе и к провалу всего проекта). Основными из этих принципов являются:

· *принцип абстрагирования*— выделение существенных аспектов системы и отвлечение от несущественных;

· *принцип непротиворечивости —*обоснованность и согласованность элементов системы;

· *принцип структурирования данных —*данные должны быть структурированы и иерархически организованы.

В структурном подходе используются в основном две группы средств, описывающих функциональную структуру системы и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются:

· *DFD (Data Flow Diagrams) —*диаграммы потоков данных;

· *SADT (Structured Analusis and Design Technique —*метод структурного анализа и проектирования,) — модели и соответствующие функциональные диаграммы;

· *ERD (Entity-Relationship Dialodrams) –*диаграммы "сущность-связь".

Диаграммы потоков данных и диаграммы "сущность-связь" — наиболее часто используемые в CASE-средствах виды моделей.

Конкретный вид перечисленных диаграмм и интерпретация их конструкций зависят от стадии ЖЦ ПО.

**На стадии формирования требований к ПО**SADT-модели и DFD используются для построения модели "AS-IS" и модели "TO-BE", отражая, таким образом, существующую и предлагаемую структуру бизнес-процессов организации и взаимодействие между ними (использование SADT-моделей, как правило, ограничивается только данной стадией, поскольку они изначально не предназначались для проектирования ПО). С помощью Е1Ш выполняется описание используемых в организации данных на концептуальном уровне, не зависимом от средств реализации базы данных (СУБД).

**На стадии проектирования**DFD используются для описания структуры проектируемой системы ПО, при этом они могут уточняться, расширяться и дополняться новыми конструкциями. Аналогично ERD уточняются и дополняются новыми конструкциями, описывающими представление данных на логическом уровне, пригодном для последующей генерации схемы базы данных. Данные модели могут дополняться диаграммами, отражающими системную архитектуру ПО, структурные схемы программ, иерархию экранных форм и меню и др.

Перечисленные модели в совокупности дают полное описание ПО ЭИС независимо оттого, является ли система существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от сложности системы и необходимой полноты ее описания.

Предметной областью для большинства примеров диаграмм, приведенных в данной главе, является налоговая система РФ, наиболее полное описание которой содержится в Налоговом кодексе РФ. Информационные технологии, применяемые в налоговой системе РФ, имеют определенные особенности (Черник Д.Г., Починок А.П , Морозов В.П. Основы налоговой системы: Учебное пособие для вузов / Под ред. Д.Г. Черника. – М.: Финансы, ЮНИТИ, 1999.).