

Контрольная работа – 2 семестр

Виды проектирования

Функционально-ориентированное или структурное проектирование (Function-Oriented – Structured Design).

Это один из классических методов проектирования, в котором декомпозиция сфокусирована на идентификации основных программных функций и, затем, детальной разработке и уточнении этих функций “сверху-вниз”. Структурное проектирование, обычно, используется после проведения структурного анализа с применением диаграмм потоков данных и связанным описанием процессов. Исследователи предлагают различные стратегии и метафоры или подходы для трансформации DFD в программную архитектуру, представляемую в форме структурных схем. Например, сравнивая управление и поведение с получаемым эффектом.

Объектно-ориентированное проектирование (Object-Oriented Design).

Представляет собой множество методов проектирования, базирующихся на концепции объектов. Данная область активно эволюционирует с середины 80-х годов, основываясь на понятиях объекта (сущности), метода (действия) и атрибута (характеристики). Здесь главную роль играют полиморфизм и инкапсуляция, в то время, как в компонентно-ориентированном подходе большее значение придается мета-информации, например, с применением технологии отражения (reflection). Хотя корни объектно-ориентированного проектирования лежат в абстракции данных (к которым добавлены поведенческие характеристики), так называемый responsibility-driven design или проектирование на основе функциональной ответственности по SWEBOK может рассматриваться как альтернатива объектно-ориентированному проектированию.

Проектирование на основе структур данных (Data-Structure-Centered Design). В данном подходе фокус сконцентрирован в большей степени на структурах данных, которыми управляет система, чем на функциях системы. Инженеры по программному обеспечению часто вначале описывают структуры данных входов (inputs) и выходов (outputs), а, затем, разрабатывают структуру управления этими данными (или, например, их трансформации).

Компонентное проектирование (Component-Based Design).

Программные компоненты являются независимыми единицами, которые обладают однозначно-определенными (well-defined) интерфейсами и зависимостями (связями) и могут собираться и развертываться независимо друг от друга. Данный подход призван решить задачи использования, разработки и интеграции таких компонент с целью повышения повторного использования активов (как архитектурных, так и в форме кода).

Компонентно-ориентированное проектирование является одной из наиболее динамично развивающихся концепций проектирования и может рассматриваться как предвестник и основа сервисно-ориентированного подхода (Service-Oriented Architecture, SOA) в проектировании, не рассматриваемого, к сожалению, в SWEBOK, но все более активно используемого в индустрии и смещающего акценты с аспектов организации связи интерфейс-реализация к обмену информацией на уровне интерфейс-интерфейс (то есть – межкомпонентному взаимодействию). По мнению автора книги, уже наступил тот момент, когда необходимо вводить отдельную тему, посвященную сервисно-ориентированному подходу в проектировании и сервисно-ориентированным архитектурам, как моделям. В частности, нотация UML 2.0 уже позволяет решать ряд вопросов, связанных с визуальным представлением соответствующих архитектурных решений, где сервисы (службы) могут рассматриваться как публикуемая функциональность одиночных компонентов и групп компонентов, объединенных в более «крупные» блоки, обеспечивающие предоставление соответствующей сервисной функциональности.

Другие интересные, но менее распространенные подходы, в основном, представляют собой формальные и точные (строгие) методы, а также, методы трансформации.

[◀ Лекции - 2 семестр](#)

Перейти на...

[Идентификация личности ▶](#)