

ЛЕКЦИЯ 34-37

Темы:

34 Структурное и объектно-ориентированное программирование в проектировании программного обеспечения распределенных информационных систем.

35 Структурный подход к проектированию информационных систем подходов к проектированию программного обеспечения

36 Проектирование информационных систем на основе объектно-ориентированного подхода

37 Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов

Цель: изучить основные методы проектирования.

Методики, используемые при проектировании, сначала программ, а затем и систем в целом, формировались в течение длительного промежутка времени.

Необходимость таких методик проявлялась при разработке сложных программных систем в условиях дефицита времени на разработку.

Большинство методик сначала были внутренними стандартами больших корпорации, а потом перешли на международный стандарт.

В основе наиболее известных методик проектирования ИС лежат два подхода:

1. **структурный**
2. **объектно-ориентированный.**

1. **Структурные методы анализа и проектирования используют иерархические структуры для моделирования объекта исследования.**

Структурное проектирование основано на алгоритмической декомпозиции, особое внимание в которой уделяется порядку происходящих событий.

Эти методы предназначены, в основном, для построения функциональных моделей и моделей данных разного уровня.

Как уже рассказывалось, есть главная цель и есть ее под разделы. Мы разделяем все части на под части.

2. **Объектно-ориентированный подход основан на выделении агентов, которые являются либо субъектами действий, либо объектами действий. При объектно-ориентированной декомпозиции каждый объект обладает своим собственным поведением и каждый из них моделирует некоторый объект реального мира.**

Выделяются актеры, описываются их свойства, действия которые они выполняют и уже потом с их помощью моделируют взаимосвязь всех объектов.

вряд ли удастся спроектировать сложную систему одновременно двумя способами, но можно применить их последовательно.

СТРУКТУРНЫЙ ПОДХОД

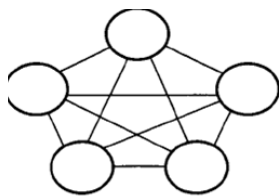
Структурный подход состоит в декомпозиции (разбиении) системы на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи, и т.д. Процесс разбиения продолжается вплоть до конкретных процедур.

Все наиболее распространенные структурные методы базируются на следующих принципах:

1. - принцип разбиения сложной проблемы на множество меньших независимых задач, легких для понимания и решения;

Задачу разбиваем на не зависимые подзадачи которые можно выполнять параллельно или они не взаимосвязаны с друг другом логической цепочкой .

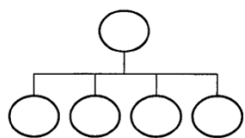
Вывести диалоговое окно приветствия, один разрабатывает структуру окна, другой



рисует окно.

2. - принцип организации составных частей в иерархические структуры.

Это когда есть разделение на модули не зависимые между собой, но они все идут от



главной задачи.

Модели используемые в структурном подходе::

1. SADT (Structured Analysis and Design Technique) – метод структурного анализа и проектирования

модели и соответствующие функциональные диаграммы, объединенные данным названием;

2. DFD (Data Flow Diagrams) – диаграммы потоков данных-

используются для описания структуры проектируемой системы

ERD (Entity-Relationship Diagrams) – диаграммы "сущность-связь"

описания модели данных логического и физического уровней.

3 Структурное и объектно-ориентированное программирование в проектировании программного обеспечения распределенных информационных систем

3.1 Проектирование программного обеспечения распределенных информационных систем

Современные крупные проекты информационных систем характеризуются, следующими особенностями:

1. сложность описания (достаточно большое количество функций, процессов, элементов данных и сложные взаимосвязи между ними), требующая тщательного моделирования и анализа данных и процессов;

2. наличие совокупности тесно взаимодействующих компонентов (подсистем), имеющих свои локальные задачи и цели функционирования (например, традиционных приложений, связанных с обработкой транзакций и решением регламентных задач, и приложений аналитической обработки (поддержки принятия решений), использующих нерегламентированные запросы к данным большого объема);

3. отсутствие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений и прикладных систем;

4. необходимость интеграции существующих и вновь разрабатываемых приложений;

5. функционирование в неоднородной среде на нескольких аппаратных платформах;

6. разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и сложившимся традициям использования тех или иных инструментальных средств;

7. существенная временная протяженность проекта, обусловленная, с одной стороны, ограниченными возможностями коллектива разработчиков, и, с другой стороны, масштабами организации-заказчика

Для успешной реализации проекта объект проектирования должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные и информационные модели информационной системы.

Однако до недавнего времени проектирование информационных систем выполнялось в основном на интуитивном уровне с применением неформализованных методов, основанных на искусстве, практическом опыте, экспертных оценках и дорогостоящих экспериментальных проверках качества функционирования информационных систем.

Кроме того, в процессе создания и функционирования информационных систем информационные потребности пользователей могут изменяться или уточняться, что еще более усложняет разработку и сопровождение таких систем.

Для целенаправленного выполнения проекта должен быть выполнен ряд работ, различных как по своему назначению, так и по квалификационным требованиям, предъявляемым к разработчикам. Иными словами, в ходе развития проекта командой разработчиков выполняются те или иные функции.

Функции, выполняемые разработчиками, — понятие неформализованное. В разных проектах оно может обретать свое содержание. 1) функции кодирования т.е. записи программы на алгоритмическом языке по имеющимся спецификациям, анализа требований, 2) тестирования 3) отладки.

В рамках деятельности менеджера любого проекта необходимо организовать распределение функций проекта между исполнителями.

считают эти действия одной из функций менеджера. В результате ее выполнения члены команды, выполняющей проект, начинают играть соответствующие роли.

Разработка современного программного комплекса представляет собой сложный и длительный процесс, который состоит из следующих этапов:

1. Предпроектные исследования, или анализ;
2. Проектирование и подготовка спецификаций;
3. Программирование, или кодирование;
4. Отладка;
5. Тестирование.

Во время предпроектного исследования собирается информация о предметной области, подбираются исходные данные и определяются функциональные требования к системе. Изучаются необходимые потребительские характеристики разрабатываемого программного обеспечения.

На этапе проектирования широко используется литература, стандарты и нормативные документы. Рассматриваются различные варианты реализации тех или иных функций и выбираются оптимальные. Здесь же разрабатывается архитектура, программный комплекс разбивается на подсистемы, определяются способы их взаимодействия. Результатом этого этапа является документация, с которой можно приступить к программированию, а именно: блок-схемы, SDL-диаграммы, функциональные схемы, описания алгоритмов.

На стадии программирования, или кодирования, происходит воплощение всех наработок, сделанных на предыдущем этапе в работающую систему. Все действия фиксируются как на уровне комментариев в исходных кодах программ, так и в проектной документации. Необходимо организовать процесс таким образом, чтобы в любой момент времени был возможен откат на несколько шагов назад.

Отладка тесно связана с программированием, эти этапы разделяются весьма условно. Отладка может производиться либо непосредственно на оборудовании, для которого предназначено разрабатываемое программное обеспечение, либо на различных эмуляторах. После программирования и отладки должна быть получена законченная рабочая система.

Тестирование представляет собой проверку всех функциональных возможностей разработанного программного обеспечения на том оборудовании, для которого оно разрабатывалось и в условиях, максимально приближенных к реальным.

На каждом этапе применяются специализированные средства разработки: во время разработки алгоритмов и архитектуры в основном используются различные редакторы; при программировании применяются компиляторы и линковщики, при отладке – отладчики, при тестировании может использоваться специальное тестовое оборудование. В самом простом случае, без какой-либо автоматизации, все перечисленные средства являются автономными и никак не связаны между собой.

3.2 Структурный подход к проектированию информационных систем

Сущность структурного подхода к разработке информационных систем заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее.

Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы «снизу-вверх» от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Структурный подход к программированию представляет собой методологию создания программ. Его внедрение обеспечивает:

- 1. повышение производительности труда программистов при написании и контроле программ;**

2. получение программ, которые более пригодны для сопровождения, так как состоят из отдельных модулей;

3. создание программ коллективом разработчиков;

4. окончание создания программ в заданный срок.

В структурированных программах обычно легко прослеживается основной алгоритм, они удобнее в отладке и менее чувствительны к ошибкам программирования. Эти свойства являются следствием важной особенности подпрограмм, каждая из которых представляет собой во многом самостоятельный фрагмент программы, связанный с основной программой лишь с помощью нескольких параметров.

Все наиболее распространенные **методологии структурного подхода базируются на ряде общих принципов**

1. **принцип «разделяй и властвуй»** - принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;

2. **принцип иерархического упорядочивания** - принцип организации составных частей проблемы 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 них являются следующие принципы:

5. **принцип абстрагирования** - заключается в выделении существенных аспектов системы и отвлечения от несущественных аспектов;

6. **принцип формализации** - заключается в необходимости строгого методического подхода к решению проблемы;

7. **принцип непротиворечивости** - заключается в обоснованности и согласованности элементов;

8. **принцип структурирования данных** - заключается в том, что данные должны быть структурированы и иерархически организованы.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых, являются следующие:

- SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы;
- DFD (Data Flow Diagrams) диаграммы потоков данных;
- ERD (Entity-Relationship Diagrams) диаграммы «сущность-связь».

На стадии проектирования информационной системы модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру программного обеспечения, структурные схемы программ и диаграммы экранных форм.

Структурное проектирование позволяет одновременно сосредотачиваться на меньшем количестве деталей.

Нисходящее проектирование хорошо работает , когда проблема имеет ясно выраженный иерархический характер.

3.3 Проектирование информационных систем на основе объектно-ориентированного подхода

Отличия структурного о объектно ориентированного подхода:

1.Первое отличие структурного подхода от объектно-ориентированного подхода заключается в принципах декомпозиции и структурной организации элементов (компонентов, модулей) системы. Согласно этим принципам система представляет собой структуру, состоящую из четко выраженных модулей, связанных между собой определенными отношениями.

Виды декомпозиции:

1) структурного подхода (первый вид декомпозиции) выполняется функциональная (процедурная, алгоритмическая) декомпозиция системы,

т. е. она представляется в виде иерархии (дерева) взаимосвязанных функций. На высшем уровне система представляется единым целым с наивысшей степенью абстракции и по мере детализации (добавления уровней) разбивается на функциональные компоненты с более конкретным содержанием.

2)объектно-ориентированный. В рамках этого подхода система разбивается на набор объектов, соответствующих объектам реального мира, взаимодействующих между собой путем посылки сообщений.

2.Вторым отличием является объединение в объекте как атрибутивных данных (характеристики, свойства), так и поведения (функции, методы). В функционально-ориентированных системах функции и данные хранятся (существуют) отдельно.

3. Третье отличие двух подходов заключается в структурной организации внутри модулей системы. В структурном подходе модуль состоит из функций, иерархически связанных между собой отношением композиции.

т. е. функция состоит из подфункций, подфункция из подподфункций и т.д.

В объектно-ориентированном подходе иерархия выстраивается с использованием двух отношений: композиции и наследования.

При этом в объектно-ориентированном подходе «объект-часть» может включаться сразу в несколько «объектов-целое».

Таким образом, модуль в структурном подходе представляется в виде дерева, а в объектно-ориентированном подходе – в виде ориентированного графа, т. е. с помощью более общей структуры.

Объектно-ориентированное проектирование – это конструирование программных систем как структурных коллекций, реализующих абстрактные типы данных.

Объектно-ориентированное проектирование и объектно-ориентированное программирование улучшают возможности нисходящего проектирования, концентрируя больше внимание на данных системы, а не на том, что система делает.

Это подход позволяет создавать системы, которые легче сопровождать, они более гибкие, более устойчивые и более приспособлены к многократному использованию, чем создаваемые при нисходящем структурном подходе.

Преимущества объектно-ориентированного метода:

1. - работают на более высоком уровне абстракции;
2. - нет «прыжков» между фазами;
3. - поддерживают данные, которые имеют тенденцию, к большей стабильности, чем функции;
4. - поощряют и поддерживают классические достоинства хорошего программирования и проектирования;
5. - сопровождаются инструментами, обеспечивающими поддержку повторного использование кода.

Объектно-ориентированный подход имеет два аспекта:

1. объектно-ориентированная разработка программного обеспечения;
2. объектно-ориентированная реализация программного обеспечения.

1. Объектно-ориентированная разработка программного обеспечения связана с применением объектно-ориентированных моделей при разработке программных систем и их компонентов. К объектно-ориентированной разработке относятся:

- объектно-ориентированные технологии разработки программных систем;
- инструментальные средства, поддерживающие эти технологии.

Можно выделить следующие объектно-ориентированные методологии разработки программного обеспечения:

- RUP (Rational Unified Process);
- OMT (Object Modeling Technique);
- SA/SD (Structured Analysis/Structured Design);
- JSD (Jackson Structured Development);
- OSA (Object-Oriented System Analysis)

2. Объектно-ориентированный подход в проектировании, предполагает декомпозицию информационных систем. объектно-ориентированном подходе декомпозиции подлежат объекты, которые характеризуются определенной структурой данных. Здесь декомпозиция идет от данных. В объектно-ориентированном подходе выделяют классы объектов. Каждый класс содержит однородные объекты. Объектам одного класса присуще одинаковое множество методов реагирования на внешние сообщения.

Иерархическая декомпозиция системы представляется в виде иерархии классов объектов, а функционирование системы – в виде взаимодействия объектов, обменивающихся сообщениями.

Среди свойств объектов в объектно-ориентированном подходе можно отметить:

- **инкапсуляция**, что означает скрытие информации. Смысл этого свойства в том, что состав и структура атрибутов объекта не зависит от сообщений, поступающих извне;

- **наследование** – это свойство, связанное с выделением иерархических классов объектов, то есть существуют родительские и дочерние классы. Оно проявляется в том, что, методы реагирования объекта, предусмотренные родительским классом, автоматически присваивают объектам дочерних классов, то есть родительские классы имеют общие методы, а дочерние – как общие, так и частные;

- **полиморфизм** – возможность выбора объектом в ответ на получаемые им сообщения какого-либо метода из множества методов в зависимости от того, какое пришло сообщение.

Наличие этих свойств у объекта позволяет в объектно-ориентированном подходе добиться параллельности и автономности разработки отдельных компонент системы,

т.е. возможно создание прототипов с дальнейшей интеграцией отдельных прототипов в единую систему и использование итерационного подхода к разработке информационных систем.

достоинство объектно-ориентированного подхода

1) состоит в упрощении накопления типовых проектных решений с тем, чтобы в дальнейших разработках новых информационных систем осуществить сбор новой системы из готовых компонент.

Эта особенность связана с тем, что классы объектов повторяются в определенной мере при переходе от одной информационной системы к другой, а для повторяющихся классов уже запрограммированы методы, разработаны и описаны структуры объектов данных.

2) является отсутствие строгой последовательности в выполнении стадий как в прямом, так и в обратном направлениях процесса проектирования по отдельным компонентам.

3) в упрощении проектирования информационных систем, при наличии типовых проектных решений по отдельным компонентам, а также легкости модификации, поскольку модификация касается лишь отдельных компонент.

3.4. Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов

Основой взаимосвязи между структурным и объектно-ориентированным подходами является общность ряда категорий и понятий обоих подходов (процесс и вариант использования, сущность и класс и др.). Эта взаимосвязь может проявляться в различных формах.

Взаимосвязь между структурным и объектно-ориентированным подходами достаточно четко просматривается в различных технологиях создания программного обеспечения.

Главный недостаток структурного подхода заключается в следующем: процессы и данные существуют отдельно друг от друга (как в модели деятельности организации, так и в модели программной системы), причем проектирование ведется от процессов к данным. Таким образом, помимо функциональной декомпозиции, существует также структура данных, находящаяся на втором плане.

В объектно-ориентированном подходе основная категория объектной модели - класс - объединяет в себе на элементарном уровне как данные, так и операции, которые над ними выполняются (методы). Именно с этой точки зрения изменения, связанные с переходом

от структурного к объектно-ориентированному подходу, являются наиболее заметными. Разделение процессов и данных преодолено, однако остается проблема преодоления сложности системы, которая решается путем использования механизма компонентов.

Данные по сравнению с процессами являются более стабильной и относительно редко изменяющейся частью системы. Отсюда следует главное достоинство объектно-ориентированного подхода: объектно-ориентированные системы более открыты и легче поддаются внесению изменений, поскольку их конструкция базируется на устойчивых формах.

объектно-ориентированная модель наиболее адекватно отражает реальный мир, представляющий собой совокупность взаимодействующих объектов. Но на практике в настоящий момент продолжается формирование UML, и количество CASE-средств, поддерживающих объектно-ориентированный подход, невелико по сравнению с поддерживаемыми структурный подход.

Кроме того, диаграммы, отражающие специфику объектного подхода (диаграммы классов и т.п.), гораздо менее наглядны и плохо понимаемы непрофессионалами. Поэтому одна из главных целей внедрения CASE-технологии, а именно снабжение всех участников проекта (в том числе и заказчика) общим языком «для передачи понимания», обеспечивается на сегодняшний день только структурными методами.

При переходе от структурного подхода к объектному, как при всякой смене технологии, необходимо вкладывать деньги в приобретение новых инструментальных средств.

Здесь следует учесть и расходы на обучение (овладение методом, инструментальными средствами и языком программирования). Для некоторых организаций эти обстоятельства могут стать серьезными препятствиями.

Несмотря на отдельные критические замечания в адрес ООП, в настоящее время именно эта парадигма используется в подавляющем большинстве промышленных проектов. Однако, нельзя считать, что ООП является наилучшей из методик программирования во всех случаях.

Процедурное программирование лучше подходит для случаев, когда важны быстрдействие и используемые программой ресурсы, но требует большего времени для разработки.

Объектное программирование подходит когда важна управляемость проекта и его модифицируемость, а также скорость разработки.

При классификации критических высказываний в адрес объектно-ориентированного программирования, можно выделить несколько аспектов критики данного подхода к программированию:

1. Критика рекламы объектно-ориентированного программирования.

2. Оспаривание эффективности разработки методами объектно-ориентированного программирования.

3. Производительность объектно-ориентированных программ.

4. Критика отдельных технологических решений в объектно-ориентированных-языках и библиотеках.

5.снижению производительности программ из-за использования объектно-ориентированных средств:

1.Динамическое связывание методов.

Обеспечение полиморфного поведения объектов приводит к необходимости связывать методы, вызываемые программой (то есть определять, какой конкретно метод будет вызываться) не на этапе компиляции, а в процессе исполнения программы, на что тратится дополнительное время. При этом реально динамическое связывание требуется не более чем для 20 % вызовов, но некоторые объектно-ориентированные-языки используют его постоянно.

2.Значительная глубина абстракции.

3.Наследование «размывает» код.

4. Инкапсуляция снижает скорость доступа к данным.

Запрет на прямой доступ к полям класса извне приводит к необходимости создания и использования методов доступа. И написание, и компиляция, и исполнение методов доступа сопряжено с дополнительными расходами.

5.Динамическое создание и уничтожение объектов.

Динамически создаваемые объекты, как правило, размещаются в куче, что менее эффективно, чем размещение их на стеке и, тем более, статическое выделение памяти под них на этапе компиляции.

Декомпозиция системы (в частности- программной системы - ПС)

- Функциональная – на основе потока данных с выделением обрабатывающих функций
- Объектная – на основе выделения сущностей, обладающих собственными наборами данных, состояниями и наборами операций

отличия =

- В первом случае внимание концентрируется на порядке происходящих событий (действиях)
- Во втором – на агентах, являющихся либо объектами, либо субъектами действий

Структурные единицы

Основной структурной единицей при функциональной декомпозиции является процедура как программная реализация алгоритма

Основной структурной единицей при объектно-ориентированной декомпозиции является объект как объединение данных и действий над ними

Сущность объектного подхода к разработке программных средств , состоит в том, что вся разработка системы основывает на выделении в объектов , их взаимосвязей, функций и свойств в той предметной области и среде исполнения , для работы в которых предназначается данное программное средство , а также в связи с теми задачами, которое данное ПС средство должно решать.

Контрольные вопросы:

1. Какие две методики проектирования вы знаете?
2. Что такое сущность ?
3. Что такое атрибут?
4. Какие сущности виды вы знаете?
5. Какие модели проектирования в структурном подходе вы знаете? Опишите их
6. Что представляет собой ER-диаграмма?
7. Какие типы связей вы знаете?
8. Какие графические примитивы вы знаете ?

Список

использованных

источников:

1. Технологии разработки программного обеспечения С.А. Орлов
2. Технологии разработки программного обеспечения В.В. Бахтизин, Л.А. Глухова
3. Project Management For Dummies / Управление проектами для "чайников"
4. Л. Н. Боронина З. В. Сенук основы управления проектами
5. https://studopedia.su/3_11546_metodi-strukturnogo-proektirovaniya.html
6. <https://studfiles.net/preview/3828360/page:7/>
7. <https://studfiles.net/preview/3828360/page:6/>