

Лаборатория информационных технологий (ИТЛаб)
При поддержке фирмы Intel

Учебно-исследовательский проект

Инструментальные средства поддержки
жизненного цикла программного обеспечения

Куратор проекта:

Сысоев А.В.

Составители

Стариков В.

Зеленцова М.

Нижний Новгород
2004г.

Раздел2: Методологии и технологии проектирования ИС

Основные вопросы:

- ◆ Общие требования к методологии и технологии
- ◆ Методология RAD
- ◆ Структурный подход
 - ◆ Методология функционального моделирования SADT
 - ◆ Моделирование потоков данных (DFD)
 - ◆ Case-метод Баркера (ERD)

Общие требования к методологии и технологии

Методологии, технологии и инструментальные средства проектирования (CASE-средства) составляют основу проекта любой ИС. Методология реализуется через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства, которые обеспечивают выполнение процессов ЖЦ. Технология проектирования определяется как совокупность трех составляющих:

- ◆ пошаговой процедуры, определяющей последовательность технологических операций проектирования
- ◆ критериев и правил, используемых для оценки результатов выполнения технологических операций
- ◆ нотаций (графических и текстовых средств), используемых для описания проектируемой системы

Общие требования к методологии и технологии

Общие требования к технологии проектирования, разработки и сопровождения:

- ◆ технология должна поддерживать полный ЖЦ ПО
- ◆ технология должна обеспечивать гарантированное достижение целей разработки ИС с заданным качеством и в установленное время
- ◆ технология должна обеспечивать возможность выполнения крупных проектов в виде подсистем (т.е. возможность декомпозиции проекта на составные части, разрабатываемые группами исполнителей ограниченной численности с последующей интеграцией составных частей)
- ◆ технология должна обеспечивать возможность ведения работ по проектированию отдельных подсистем небольшими группами (3-7 человек)

Общие требования к методологии и технологии

Общие требования к технологии проектирования, разработки и сопровождения:

- ◆ технология должна обеспечивать минимальное время получения работоспособной ИС
- ◆ технология должна предусматривать возможность управления конфигурацией проекта, ведения версий проекта и его составляющих, возможность автоматического выпуска проектной документации и синхронизацию ее версий с версиями проекта
- ◆ технология должна обеспечивать независимость выполняемых проектных решений от средств реализации ИС (систем управления базами данных (СУБД), операционных систем, языков и систем программирования)
- ◆ технология должна быть поддержана комплексом согласованных CASE-средств, обеспечивающих автоматизацию процессов, выполняемых на всех стадиях ЖЦ

Общие требования к методологии и технологии

Выработка необходимых стандартов:

- ◆ стандарт проектирования
- ◆ стандарт оформления проектной документации
- ◆ стандарт пользовательского интерфейса

Методология RAD

Одним из возможных подходов к разработке ПО в рамках спиральной модели ЖЦ является получившая в последнее время широкое распространение методология быстрой разработки приложений RAD (Rapid Application Development). Под этим термином обычно понимается процесс разработки ПО, содержащий 3 элемента:

- ◆ небольшую команду программистов (от 2 до 10 человек)
- ◆ короткий, но тщательно проработанный производственный график (от 2 до 6 мес.)
- ◆ повторяющийся цикл, при котором разработчики, по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные через взаимодействие с заказчиком

Методология RAD

Жизненный цикл ПО по методологии RAD состоит из четырех фаз:

- ◆ фаза анализа и планирования требований
- ◆ фаза проектирования
- ◆ фаза построения
- ◆ фаза внедрения

Методология RAD

Основные принципы методологии RAD:

- ◆ разработка приложений итерациями
- ◆ необязательность полного завершения работ на каждом из этапов жизненного цикла
- ◆ обязательное вовлечение пользователей в процесс разработки ИС
- ◆ необходимое применение CASE-средств, обеспечивающих целостность проекта
- ◆ применение средств управления конфигурацией, облегчающих внесение изменений в проект и сопровождение готовой системы
- ◆ необходимое использование генераторов кода

Методология RAD

Основные принципы методологии RAD:

- ◆ использование прототипирования, позволяющее полнее выяснить и удовлетворить потребности конечного пользователя
- ◆ тестирование и развитие проекта, осуществляемые одновременно с разработкой
- ◆ ведение разработки немногочисленной хорошо управляемой командой профессионалов
- ◆ грамотное руководство разработкой системы, четкое планирование и контроль выполнения работ

Структурный подход

- ◆ принцип "разделяй и властвуй" - принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения
- ◆ принцип иерархического упорядочивания - принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне
- ◆ принцип абстрагирования - заключается в выделении существенных аспектов системы и отвлечения от несущественных
- ◆ принцип формализации - заключается в необходимости строгого методического подхода к решению проблемы
- ◆ принцип непротиворечивости - заключается в обоснованности и согласованности элементов
- ◆ принцип структурирования данных - заключается в том, что данные должны быть структурированы и иерархически организованы

Структурный подход

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"

Методология функционального моделирования *SADT*

Методология *SADT* разработана *Дугласом Россом*. На ее основе разработана, в частности, известная методология *IDEF0* (*Icam DEFinition*), которая является основной частью программы *ICAM* (Интеграция компьютерных и промышленных технологий), проводимой по инициативе **ВВС США**.

Методология *SADT* представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель *SADT* отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Методология функционального моделирования *SADT*

Основные концепции:

- ◆ **графическое представление блочного моделирования.** Графика блоков и дуг *SADT*-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описываются посредством интерфейсных дуг, выражающих "ограничения", которые в свою очередь определяют, когда и каким образом функции выполняются и управляются
- ◆ **строгость и точность.** Выполнение правил *SADT* требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика.

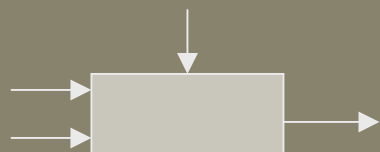
Методология функционального моделирования SADT

Правила SADT включают:

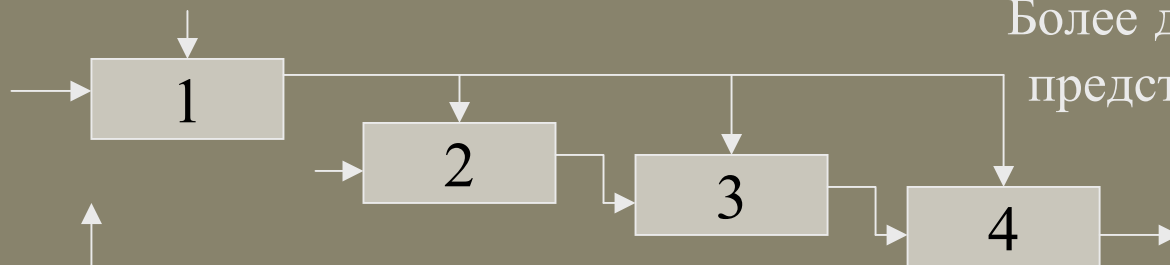
- ◆ **ограничение количества блоков на каждом уровне декомпозиции (правило 3-6 блоков)**
- ◆ **связность диаграмм (номера блоков)**
- ◆ **уникальность меток и наименований (отсутствие повторяющихся имен)**
- ◆ **синтаксические правила для графики (блоков и дуг)**
- ◆ **разделение входов и управлений (правило определения роли данных)**
- ◆ **отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель**

Методология функционального моделирования SADT

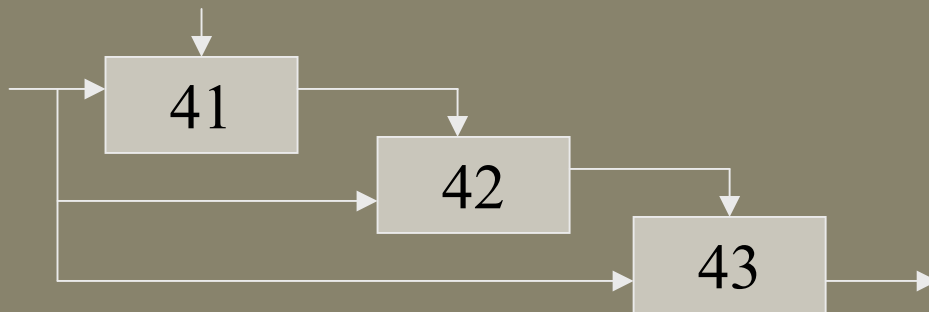
Структура SADT-модели. Декомпозиция диаграмм



Более общее
представление



Более детальное
представление



Эта диаграмма
Является “родителем”
этой диаграммы

Методология функционального моделирования SADT

Значимость	Тип связности	Для функций	Для данных
0	Случайная	Случайная	Случайная
1	Логическая	Функции одного и того же множества или типа	Данные одного и того же множества или типа
2	Временная	Функции одного и того же периода времени	Данные, используемые в каком-либо временном интервале
3	Процедурная	Функции, работающие в одной и той же фазе или итерации	Данные, используемые во время одной и той же фазы или итерации

Методология функционального моделирования SADT

Значимость	Тип связности	Для функций	Для данных
4	Коммуникационная	Функции, использующие одни и те же данные	Данные, на которые воздействует одна и та же деятельность
5	Последовательная	Функции, выполняющие последовательные преобразования одних и тех же данных	Данные, преобразуемые последовательными функциями
6	Функциональная	Функции, объединяемые для выполнения одной функции	Данные, связанные с одной функцией

Моделирование потоков данных (DFD)

В основе данной методологии (методологии *Gane/Sarson*) лежит построение модели анализируемой ИС - проектируемой или реально существующей. В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных (ДПД или **DFD**), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы ИС с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процесс становится элементарными и детализировать их далее невозможно.

Моделирование потоков данных (DFD)

Основные компоненты диаграмм потоков данных:

- ◆ **Внешняя сущность** представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой ИС, если это необходимо, или, наоборот, часть процессов ИС может быть вынесена за пределы диаграммы и представлена как внешняя сущность.



Заказчик

Обозначение на диаграммах

Моделирование потоков данных (DFD)

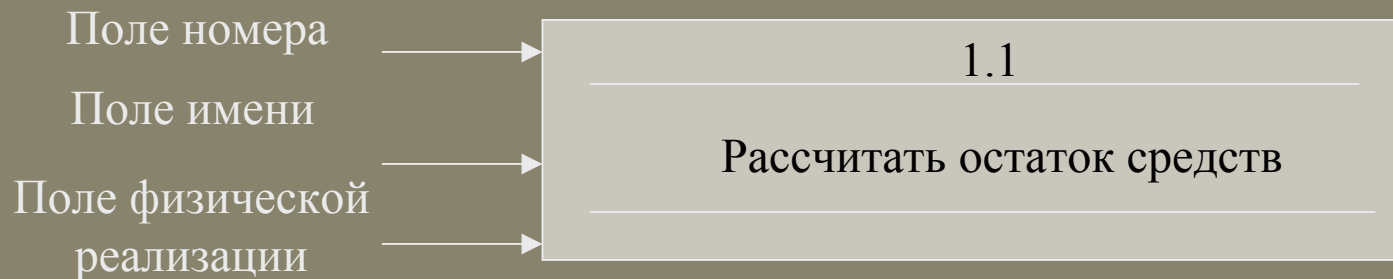
- ◆ **Системы и подсистемы.** При построении модели сложной ИС она может быть представлена в самом общем виде на так называемой контекстной диаграмме в виде одной системы как единого целого, либо может быть декомпозирована на ряд подсистем. Подсистема (или система) на контекстной диаграмме изображается следующим образом



Номер подсистемы служит для ее идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями

Моделирование потоков данных (DFD)

- ◆ **Процесс** представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.



Номер процесса служит для его идентификации. В поле имени вводится наименование процесса, за которым следуют существительные в винительном падеже. Информация в поле физической реализации показывает, какое подразделение организации выполняет данный процесс.

Моделирование потоков данных (DFD)

- ◆ **Накопитель данных** представляет собой абстрактное устройство для хранения абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми. Накопитель данных может быть реализован физически в виде микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д. Накопитель данных на диаграмме потоков данных изображается

D1

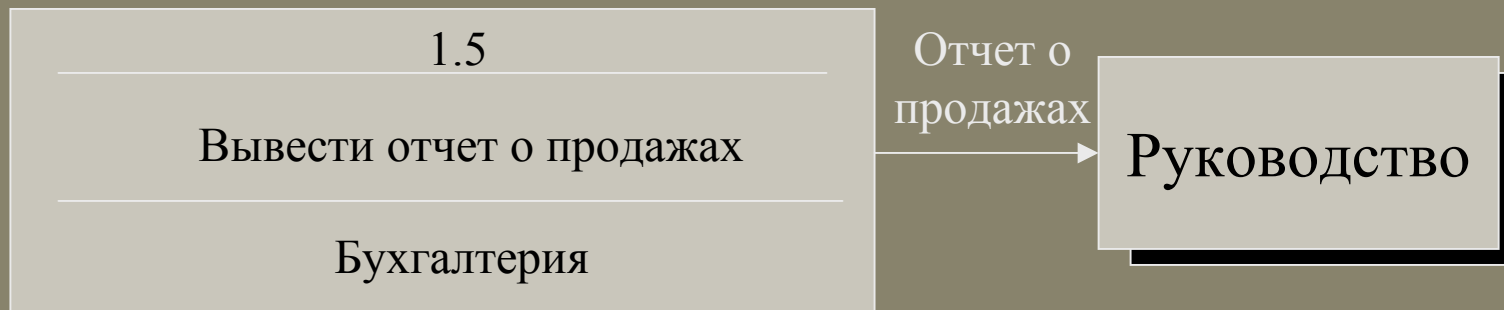
Получаемые счета

Накопитель данных идентифицируется буквой "D" и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей базы данных и описание хранящихся в нем данных должно быть увязано с информационной моделью.

Моделирование потоков данных (DFD)

- ◆ **Поток данных** определяет информацию, передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д. Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока. Каждый поток данных имеет имя, отражающее его содержание



Моделирование потоков данных (DFD)

Первым шагом при построении иерархии *ДПД* является построение контекстных диаграмм. Обычно при проектировании относительно простых ИС строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и кроме того, единственный главный процесс не раскрывает структуры распределенной системы. Признаками сложности (в смысле контекста) могут быть:

- ◆ наличие большого количества внешних сущностей (десять и более);
- ◆ распределенная природа системы;
- ◆ многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.

Моделирование потоков данных (DFD)

Для сложных ИС строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Иерархия контекстных диаграмм определяет взаимодействие основных функциональных подсистем проектируемой ИС как между собой, так и с внешними входными и выходными потоками данных и внешними объектами, с которыми взаимодействует ИС.

Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры ИС на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых участвуют разные организации и коллективы разработчиков.

После построения контекстных диаграмм полученную модель следует проверить на полноту исходных данных об объектах системы и изолированность объектов.

Моделирование потоков данных (DFD)

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи *ДПД*. Каждый процесс на ДПД, в свою очередь, может быть детализован при помощи ДПД или мини-спецификации.

При детализации должны выполняться следующие правила:

- ◆ **правило балансировки** - означает, что при детализации подсистемы или процесса детализирующая диаграмма в качестве внешних источников/приемников данных может иметь только те компоненты (подсистемы, процессы, внешние сущности, накопители данных), с которыми имеет информационную связь детализируемая подсистема или процесс на родительской диаграмме;
- ◆ **правило нумерации** - означает, что при детализации процессов должна поддерживаться их иерархическая нумерация. Например, процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

Моделирование потоков данных (DFD)

Мини-спецификация является конечной вершиной иерархии *ДПД*.

Решение о завершении детализации процесса и использовании мини-спецификации принимается аналитиком исходя из следующих критериев:

- ◆ наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3 потока)
- ◆ возможности описания преобразования данных процессом в виде последовательного алгоритма
- ◆ выполнения процессом единственной логической функции преобразования входной информации в выходную
- ◆ возможности описания логики процесса при помощи мини-спецификации небольшого объема (не более 20-30 строк)

Case-метод Баркера (ERD)

Цель моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть ражены в любую систему баз данных.

нным средством моделирования данных являются *диаграммы сущность-связь* (*ERD*). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). *ERD* нно используются для проектирования реляционных баз данных.

ктирования реляционных баз данных.

введена *П. Ченом (Chen)* и получила дальнейшее развитие в иейшее развитие в работах *Баркера*.

Case-метод Баркера (ERD)

Основные шаги:

- ◆ выделение сущностей
- ◆ идентификация связей
- ◆ идентификация атрибутов

Case-метод Баркера (ERD)

Сущность (Entity) - реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению.

Обозначение на схеме <Имя сущности>

Свойства сущности:

- ◆ каждая сущность должна иметь уникальное имя, и к одному и тому же имени должна всегда применяться одна и та же интерпретация. Одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами
- ◆ сущность обладает одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь
- ◆ сущность обладает одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности
- ◆ каждая сущность может обладать любым количеством связей с другими сущностями модели

Case-метод Баркера (ERD)

Связь (Relationship) - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь - это ассоциация между сущностями. При этом каждый экземпляр одной сущности, называемой родительской сущностью, ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, называемой сущностью-потомком. Каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. Таким образом, экземпляр сущности-потомка может существовать только при существовании сущности родителя.

Связи может даваться имя, выражаемое грамматическим оборотом глагола и помещаемое возле линии связи. Имя каждой связи между двумя данными сущностями должно быть уникальным, но имена связей в модели не обязаны быть уникальными. Имя связи всегда формируется с точки зрения родителя, так что предложение может быть образовано соединением имени сущности-родителя, имени связи, выражения степени и имени сущности-потомка.

Case-метод Баркера (ERD)

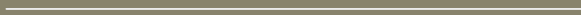
Степень связи и обязательность графически изображаются следующим образом



Один



Много



Обязательная



Необязательная

Case-метод Баркера (ERD)

Атрибут - любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут представляет тип характеристик или свойств, ассоциированных со множеством реальных или абстрактных объектов. Экземпляр атрибута - это определенная характеристика отдельного элемента множества. Экземпляр атрибута определяется типом характеристики и ее значением, называемым значением атрибута. В *ER*-модели атрибуты ассоциируются с конкретными сущностями. Таким образом, экземпляр сущности должен обладать единственным определенным значением для ассоциированного атрибута. Атрибут может быть либо обязательным, либо необязательным

<имя сущности>

*<атрибут - 1>

* обязательный атрибут

(не допускаются null значения)

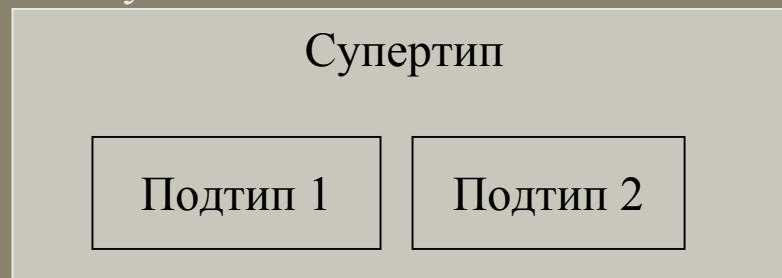
0 необязательный атрибут

(допускаются null значения)

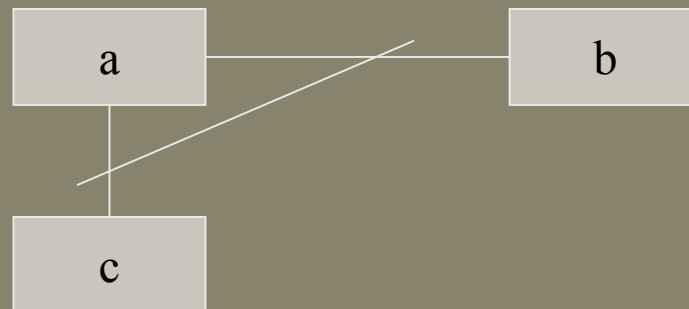
Case-метод Баркера (ERD)

Помимо перечисленных основных конструкций модель данных может содержать ряд дополнительных:

- ◆ **Подтипы и супертипы:** одна сущность является обобщающим понятием для группы подобных сущностей



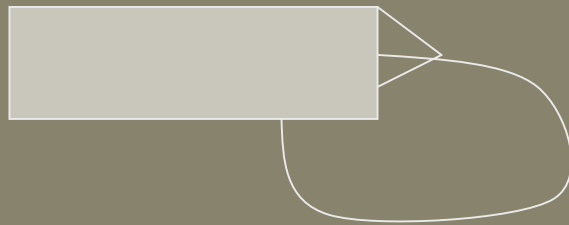
- ◆ **Взаимно исключающие связи:** каждый экземпляр сущности участвует только в одной связи из группы взаимно исключающих связей



Case-метод Баркера (ERD)

Помимо перечисленных основных конструкций модель данных может содержать ряд дополнительных:

- ◆ **Рекурсивная связь:** сущность может быть связана сама с собой



- ◆ **Неперемещаемые (*non-transferrable*) связи:** экземпляр сущности не может быть перенесен из одного экземпляра связи в другой

