

# CASE технологии

## Лекция 5

# Методология структурного анализа Gane/Sarson

- Особенность методологии  
рассмотрение Информационной Системы (существующей или создаваемой) с точки зрения моделирования информационных потоков
- Источники сведений об информационных потоках:  
документооборот  
информационные процессы на предприятии

# Методология структурного анализа Gane/Sarson

- Компоненты модели ИС по Гейну/Сарсону
  - Информационный процесс
  - Событие

# Компоненты модели ИС по Gane/Sarson

- Информационный процесс
  - процесс преобразования входных данных системы в выходные в соответствии с определенными правилами
- Событие
  - Изменение состояния предметной области, возникающее в информационном процессе

# Диаграммы потоков данных

- Один из способов визуализации модели информационного процесса – построение диаграммы потоков данных

Data Flow Diagram (DFD)

# Диаграммы потоков данных

- Модель информационной системы определяется как **иерархия диаграмм потоков данных** (иерархия DFD), которые описывают асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю
- Таким образом, модель ИС – модель сложного информационного процесса

# Диаграммы потоков данных и принципы структурного анализа

- 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 **процессы становятся элементарными** и детализировать их далее невозможно

# Вспомним: принципы структурного анализа

- «Разделяй и властвуй»
- Иерархическое упорядочивание диаграмм
- Структурирование данных
- Абстрагирование
- Формализация
- Непротиворечивость



# Содержание диаграммы потоков данных

- Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или *процессам*.
- Те в свою очередь преобразуют информацию и порождают новые потоки,
- которые переносят информацию к другим *процессам* или подсистемам, накопителям данных или внешним сущностям - потребителям информации

# Компоненты диаграммы потоков данных

- внешние сущности
- системы/подсистемы
- процессы
- накопители данных
- потоки данных

# Внешние сущности

- **Внешняя сущность** - 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.**

# Внешние сущности

- 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

# Внешние сущности: Нотация

- **Внешняя сущность** обозначается квадратом, расположенным как бы "над" диаграммой и бросающим на нее тень, для того, чтобы можно было выделить этот символ среди других обозначений

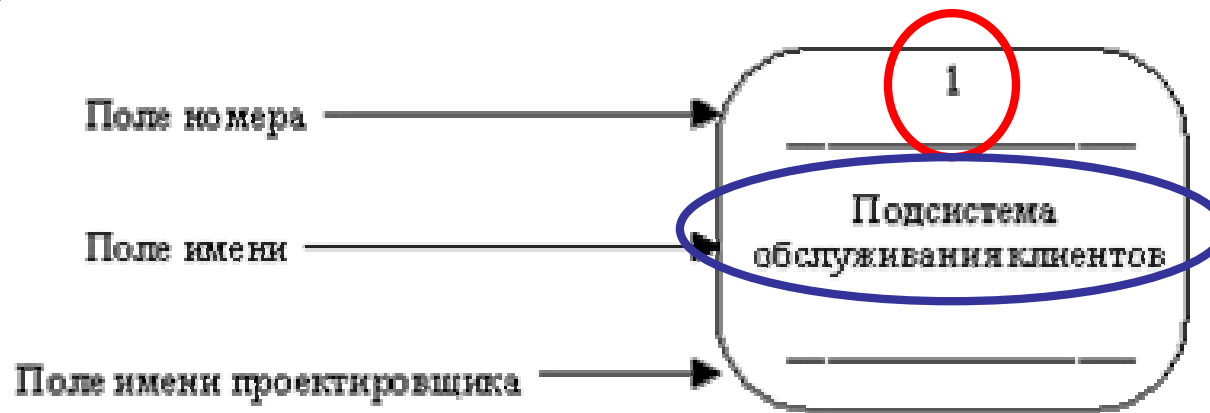


# Системы и подсистемы

- 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: нотация

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



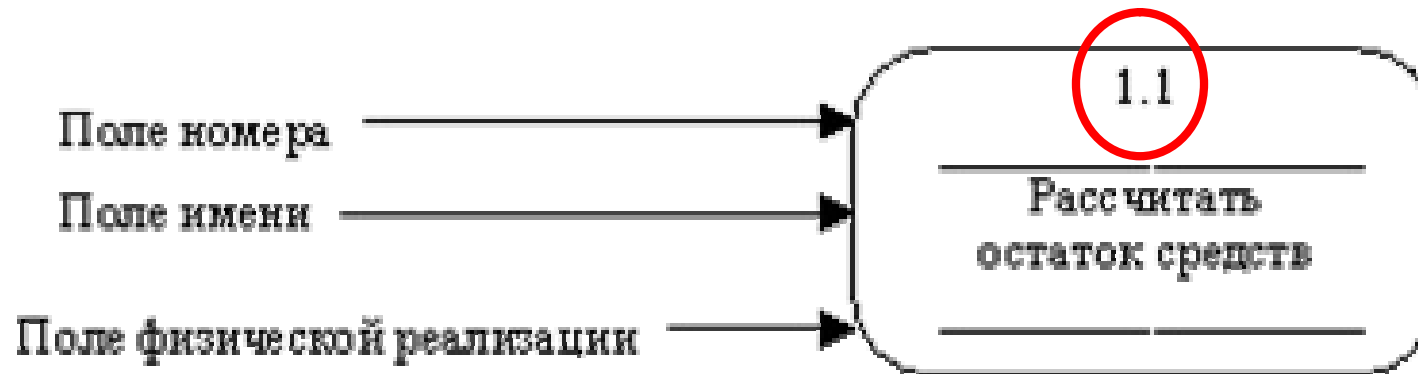
# Процессы

- **Процесс** - 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



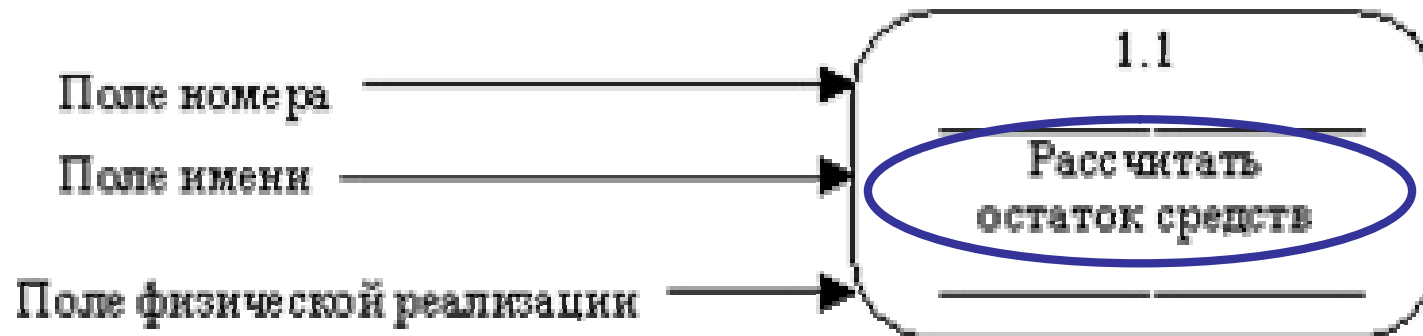
# Процессы: нотация

- **Номер** процесса служит для его идентификации



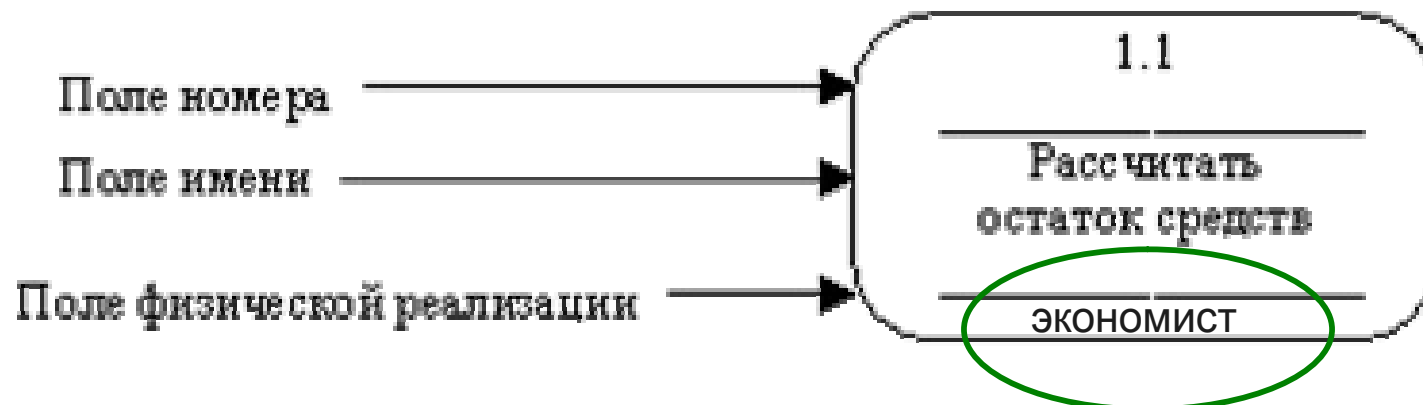
# Процессы: нотация

- **Поле имени** - наименование **процесса** в виде предложения с активным **недвусмысленным глаголом** в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют **существительные в винительном падеже**

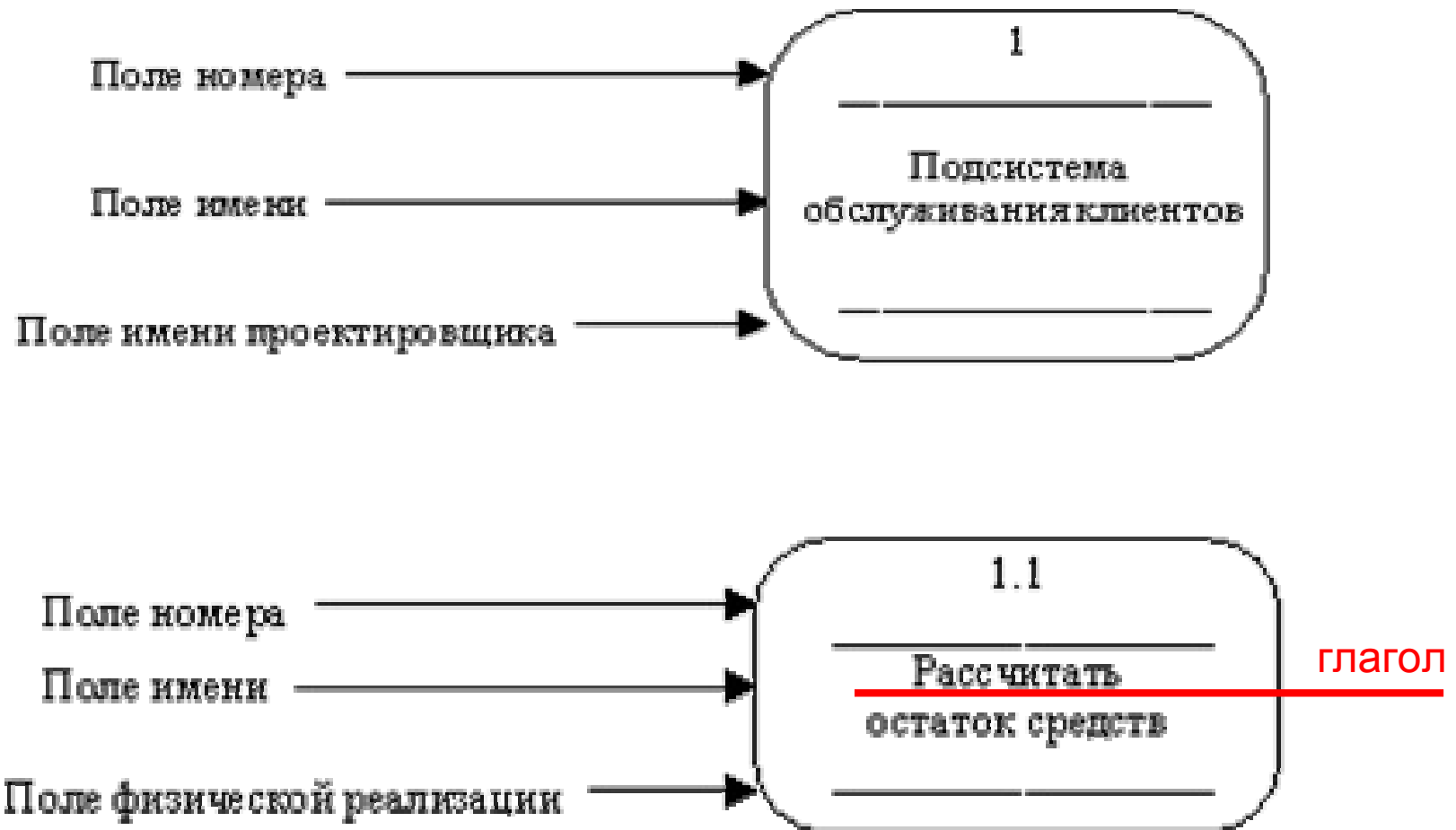


# Процессы: нотация

- поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство **выполняет** данный процесс



# Процессы и подсистемы



# Процессы

- Примеры хороших названий процессов:
  - Ввести сведения о клиентах
  - Выдать информацию о текущих расходах
  - Проверить кредитоспособность клиента
- Примеры плохих названий (нечеткая формулировка действия):
  - обработать ...
  - модернизировать ...
  - отредактировать ...

# Накопители данных

- **Накопитель данных** - 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 ящика в картотеке, таблицы в оперативной памяти, файла на сменном носителе и т.д.

# Накопители данных: нотация

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

D 1	Получаемые счета
-----	------------------

# Накопители данных

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

D 1	Получаемые счета
-----	------------------

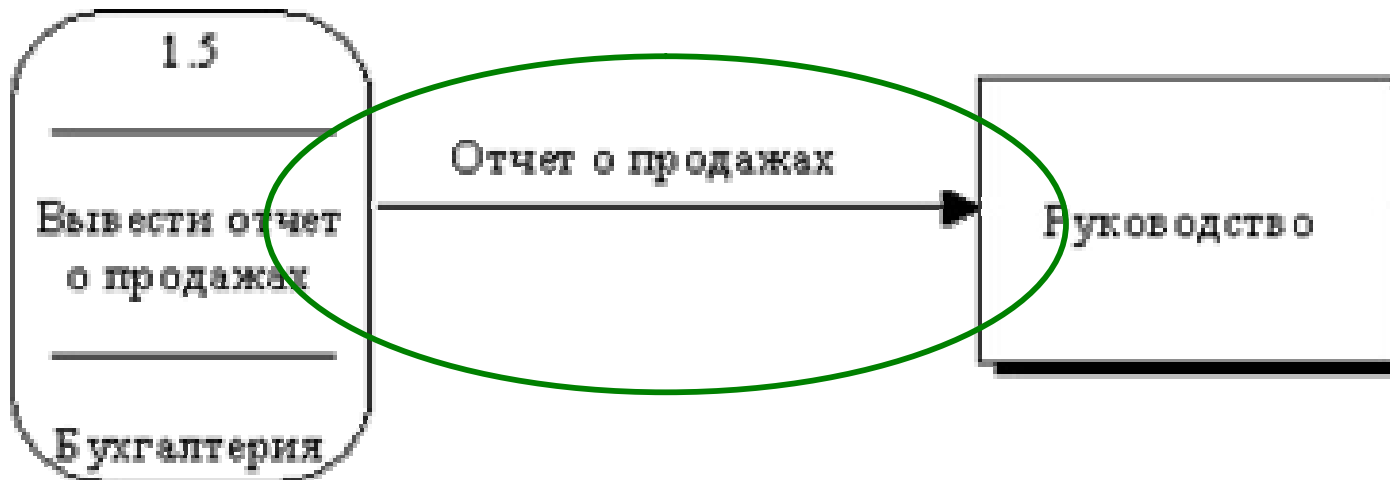


# Потоки данных

- **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,
  - дисками, переносимыми с одного компьютера на другой и т.д.

# Потоки данных: нотация

- **Поток данных** на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока
- Каждый поток данных имеет имя, отражающее его содержание



# Построение иерархии диаграмм потоков данных

- Шаг 1: 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

# Построение иерархии диаграмм ПОТОКОВ ДАННЫХ

- Признаки сложности (в смысле контекста):
  - наличие большого количества внешних сущностей (>10)
  - распределённая природа системы
  - многофункциональность системы с уже сложившейся группировкой функций в отдельные подсистемы

# Построение иерархии диаграмм ПОТОКОВ ДАННЫХ

- Шаг 2: 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

# Построение иерархии диаграмм ПОТОКОВ ДАННЫХ

- Шаг 3: После построения контекстных диаграмм полученную **модель следует проверить на полноту исходных данных** об объектах системы и изолированность объектов (отсутствие информационных связей с другими объектами)

# Построение иерархии диаграмм потоков данных

- Шаг 4: Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи DFD
- Каждый процесс на DFD, в свою очередь, может быть детализирован при помощи:
  - другой DFD
  - либо
  - миниспецификации

# Правила детализации

- **правило балансировки:**  
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:**  
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.



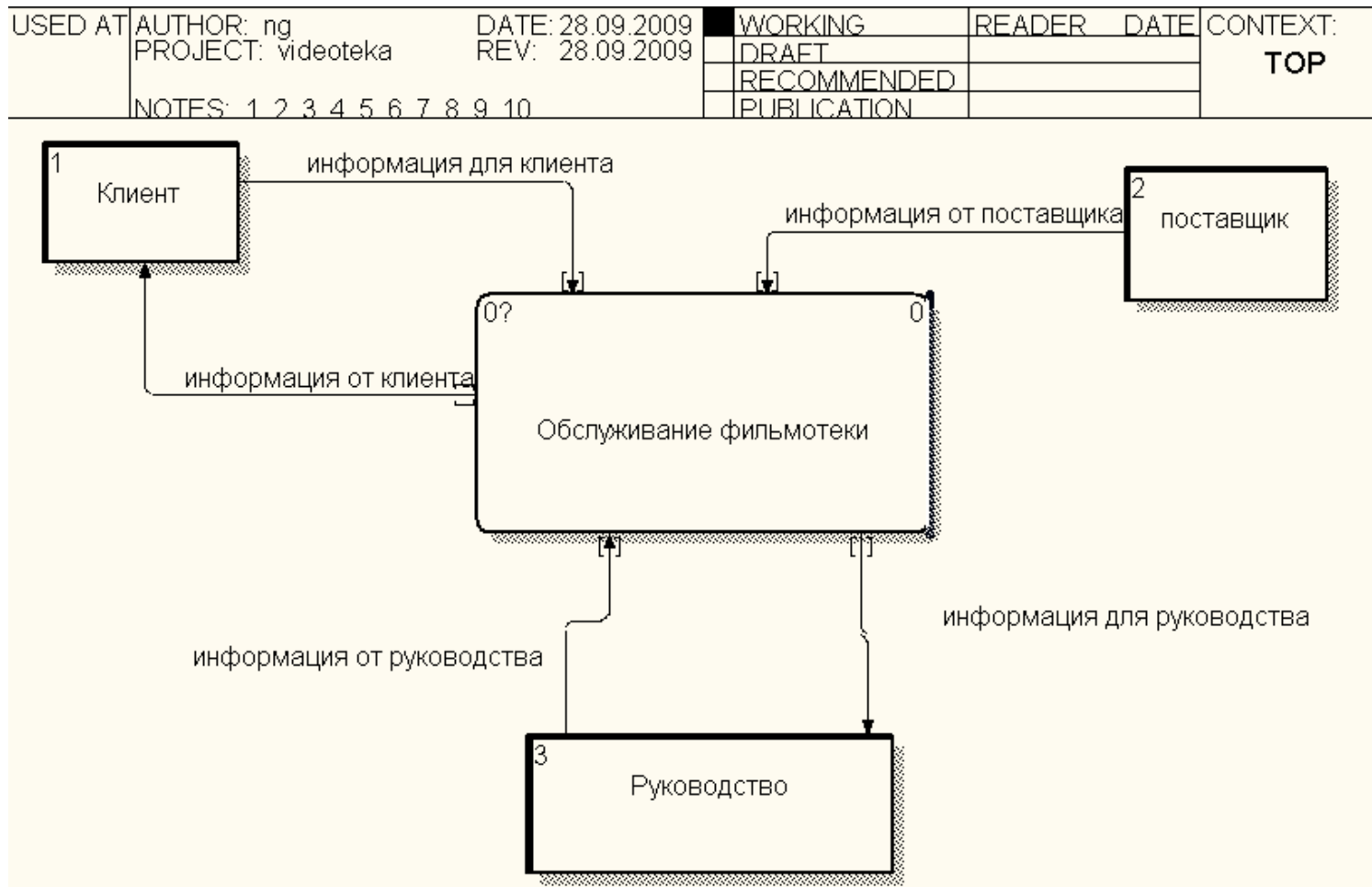
# Миниспецификация

- **Миниспецификация** – «листовая» вершиной иерархии DFD.
- Решение о завершении детализации процесса и использовании миниспецификации принимается в случае:
  - наличия у процесса 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 миниспецификации небольшого объема (не более 20-30 строк)

# Пример

- Рассмотрим работу **видеотеки**, которая получает запросы от клиентов на фильмы, проверяет членство клиентов, контролирует возврат лент, не допуская выдачу фильмов тем, кто просрочил аренду фильма.
- За аренду начисляется плата, за просрочку возврата - пени. Информация об аренде лент хранится отдельно от записей о членстве клиентов.
- Новые фильмы видеотека получает от поставщиков, фиксируя информацию о них.
- Служащие регулярно готовят отчеты для руководства за определенный период времени о членах видеотеки, поставщиках лент, выдаче фильмов и приобретенных лентах.

# Контекстная диаграмма



# Содержание потоков данных

- Информация **от клиента** включает данные о клиенте и запрос на фильм;
- Информация **для клиента** включает ответ на запрос об аренде фильма и членскую карточку.
- Информация **от поставщика** включает данные о поставщике и о новых фильмах;
- Информация **от руководства** включает: запросы отчетов о новых членах, о новых поставщиках, о новых фильмах; об аренде фильмов, о составе видеотеки, о поставщиках вообще.
- Информация **для руководства** включает все эти виды отчетов.

# Таблица «событие-реакция»

- Каждый входной поток данных порожден каким-либо событием, а выходной поток данных является ответом (реакцией) системы

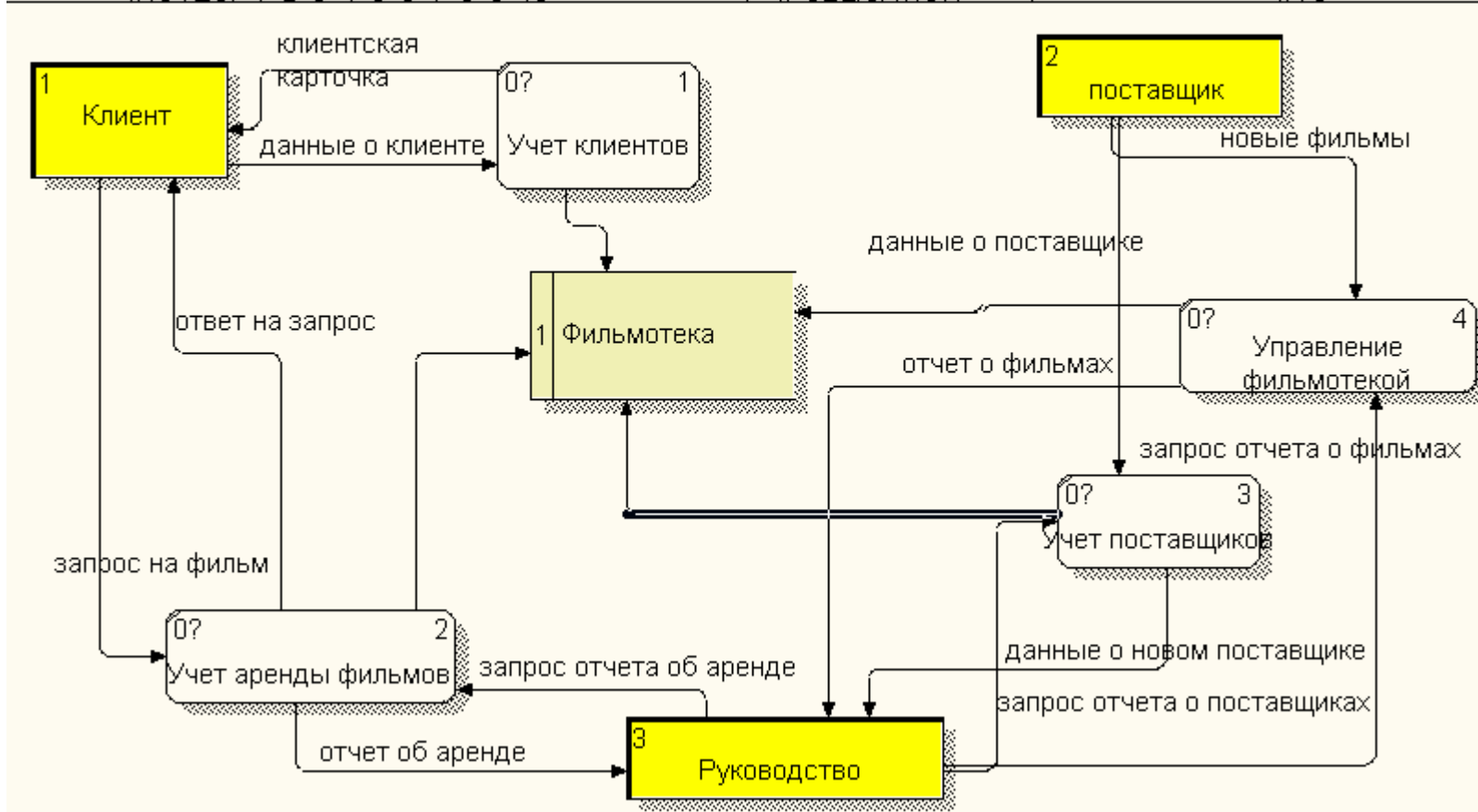
Событие	Реакция
Новый клиент хочет стать членом видеотеки	Регистрация клиента
Клиент сообщает об изменении адреса	Регистрация нового адреса
Клиент просит фильм в аренду	Рассмотрение запроса
Клиент возвращает фильм	Регистрация возврата
Руководство находит нового поставщика	Регистрация нового поставщика
Изменение данных о поставщике	Регистрация изменений
Поставщик передает новый фильм	Получение нового фильма
Руководство запрашивает отчет о работе	Формирование требуемого отчета

# Разбиение на диаграммы 1-го уровня

- Можно разбить процесс 0-го уровня "Обслуживание видеотеки" на 4 процесса, отражающие основные виды деятельности фильмотеки:
  - учет клиентов,
  - учет поставщиков,
  - учет аренды фильмов,
  - управление фондом фильмов.

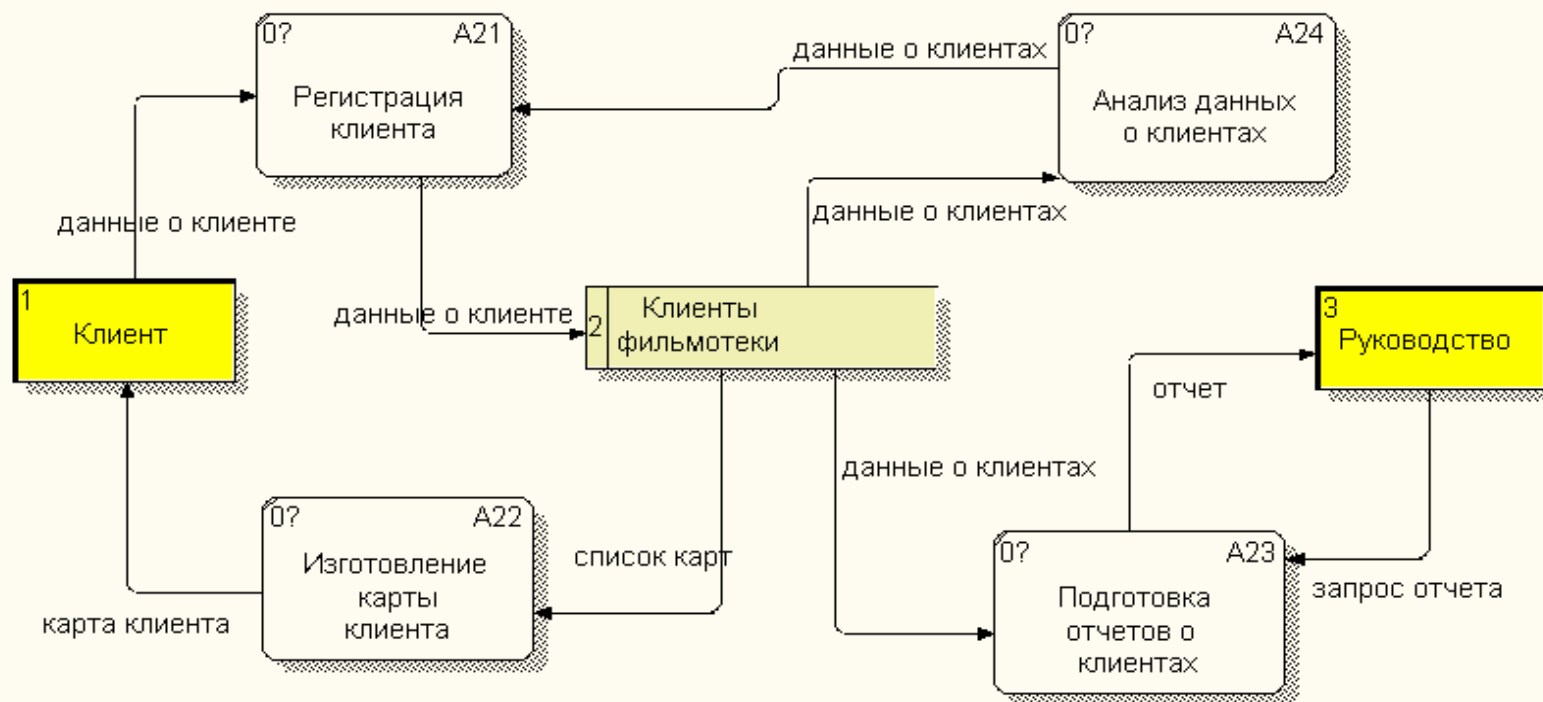
# Результат разбиения

USED AT:	AUTHOR: ng	DATE: 28.09.2009	WORKING	READER	DATE	CONTEXT:
	PROJECT: videoteka	REV: 28.09.2009	DRAFT			
			RECOMMENDED			
NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION			A-0



# Диаграммы 1-го уровня: учет клиентов фильмотеки

USED AT:	AUTHOR: ng	DATE: 28.09.2009	<input checked="" type="checkbox"/> WORKING	READER	DATE	CONTEXT:
	PROJECT: videoteka	REV: 28.09.2009	<input type="checkbox"/> DRAFT			<input type="checkbox"/>
			<input type="checkbox"/> RECOMMENDED			<input type="checkbox"/>
	NOTES: 1 2 3 4 5 6 7 8 9 10		<input type="checkbox"/> PUBLICATION			<input type="checkbox"/>



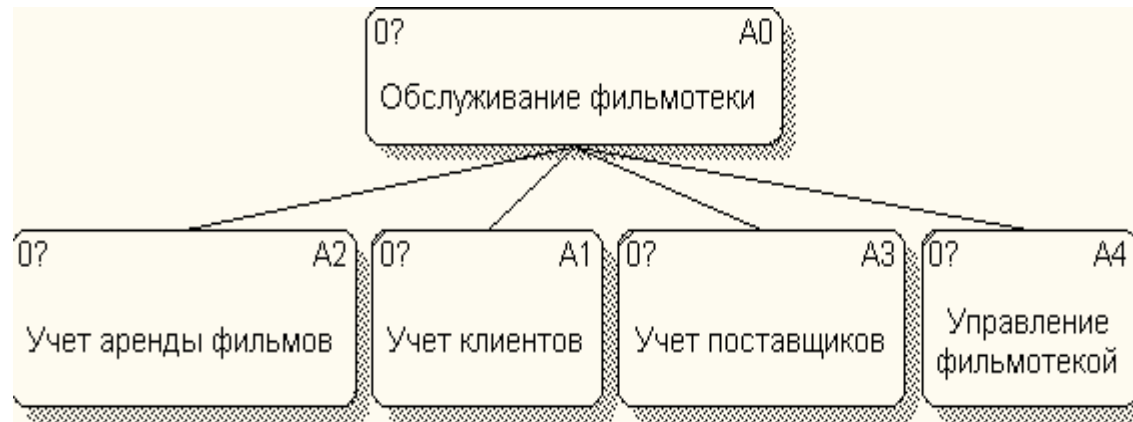
NODE:	TITLE:	NUMBER:
A2	Видеотека	



# Что дальше?

- Каждый процесс на диаграмме «Учет клиентов фильмотеки» имеет 2-3 входных и выходных потока данных, **поэтому их дальнейшая детализация нецелесообразна.**
- Создаем миниспецификацию:
- а) Начало
  - б) Получить от клиента данные для регистрации
  - в) сравнить данные клиента со списком членов видеотеки (процесс А14)
  - г) проверить: был ли клиент зарегистрирован ранее,
    - д) если нет - то занести данные клиента в накопитель "Члены видеотеки"
    - е) в противном случае проверить: является ли клиент должником;
    - ж) если да - отказать клиенту в повторной регистрации и потребовать возврата долга; затем перейти к шагу "и";
  - з) в противном случае - обновить информацию о клиенте в накопителе "Члены видеотеки" и перейти к процессу А12.
  - и) конец

# Дополнительные диаграммы



- Регистрация клиента
- Изготовление карты клиента
- Подготовка отчетов о клиентах
- Анализ данных о клиентах

- По данной диаграмме потоков данных можно построить так называемую «Структуру разбивки работ» (work breakdown structure, WBS), в виде Node-Tree Diagram

# Стандарт IDEF3

- Назначение IDEF3

стандарт документирования технологических процессов, происходящих на предприятии, и инструмент наглядного исследования и моделирования их сценариев

- Источники сведений о процессах:

- документооборот
- информационные процессы на предприятии
- технологические процессы на предприятии

# Сценарии

- Сценарий (Scenario) – описание последовательности изменений свойств объекта, в рамках рассматриваемого процесса.
- Например, описание последовательности этапов обработки детали в цеху и изменение её свойств после прохождения каждого этапа

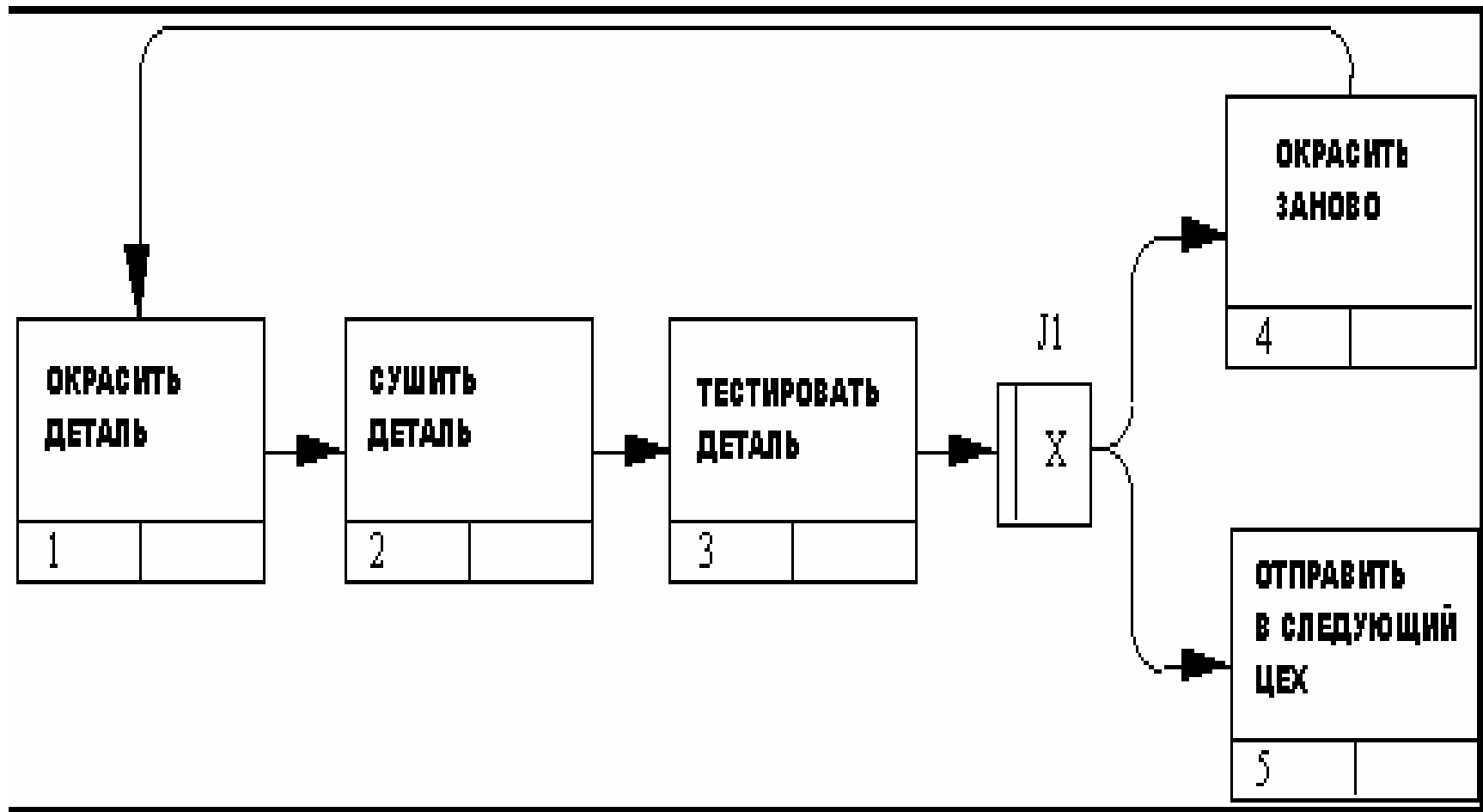
# Сценарии

- Исполнение каждого сценария сопровождается соответствующим документооборотом, который состоит из двух основных потоков:
  - документов, определяющих структуру и последовательность процесса (технологических указаний, описаний стандартов и т.д.),
  - документов, отображающих ход его выполнения (результатов тестов и экспертиз, отчетов о браке, и т.д.).

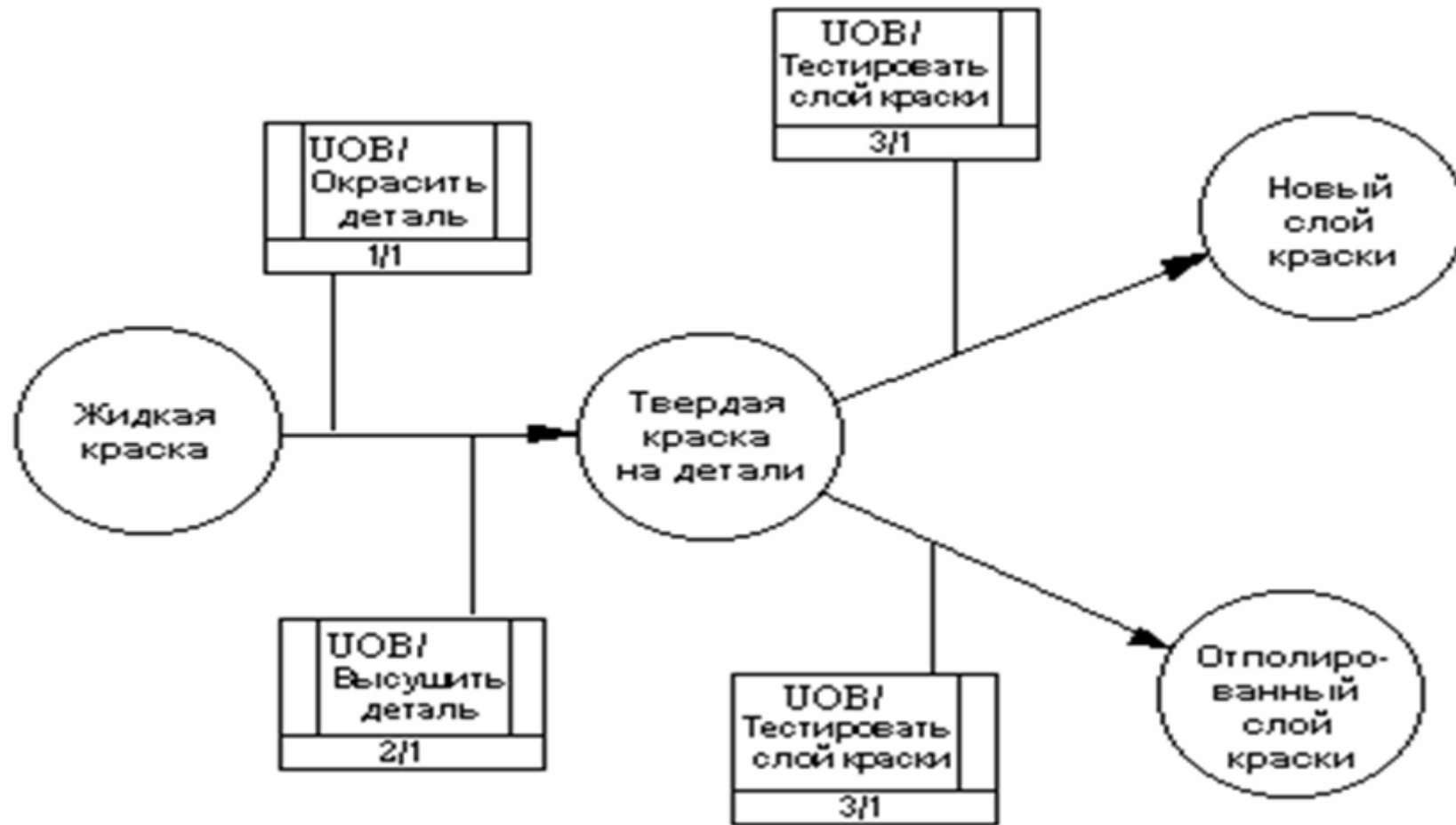
# Типы диаграмм в IDEF3

- Существует два типа диаграмм в IDEF3:
  - диаграммы Описания Последовательности Этапов Процесса  
(Process Flow Description Diagrams, **PFDD**)
  - Диаграммы Состояния Объекта и его Трансформаций в Процессе  
(Object State Transition Network, **OSTN**)

# Пример PFDD диаграммы



# Пример OSTN диаграммы





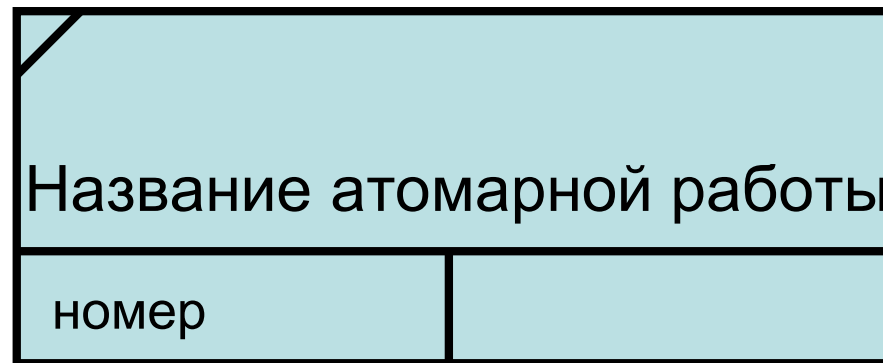
# Компоненты IDEF3 PFDD диаграммы

- Атомарные (неделимые) работы
- Объекты ссылки
- Связи предшествования
- Связи отношения
- Поток объектов
- Перекрестки

# Атомарные работы

- Атомарная работа (Unit of Behavior, UOB) - служит для описания функций (процедур, работ), выполняемых подразделениями/сотрудниками предприятия
- Предполагается, что работа эта неделима, понятна исполнителю и не допускает разночтений (т.е. является шагом некоторого алгоритма)

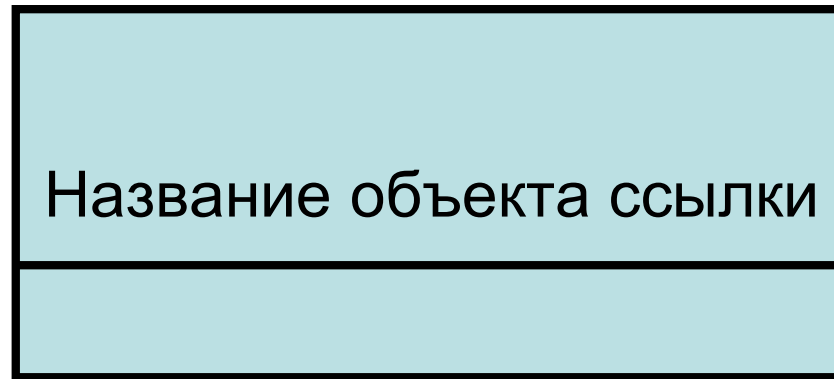
# Атомарные работы: нотация



# Объекты ссылки

- Объект ссылки (referent) – объект, используемый для описания ссылок на другие диаграммы модели, циклические переходы в рамках одной модели, различные комментарии к функциям

# Объекты ссылки: нотация



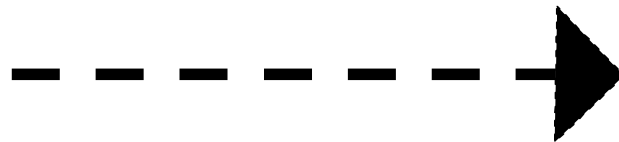
# Связи предшествования

- Связь предшествования (Precedence) – показывает, что прежде чем начнется работа-приемник, должна завершиться работа-источник
- Нотация: сплошная линия со стрелкой, ведущей, от источника к приёмнику



# СВЯЗИ ОТНОШЕНИЯ

- Связь предшествования (Precedence) – показывает связь между двумя работами или между работой и объектом ссылки.
- Нотация: пунктирная линия со стрелкой, ведущей, от источника к приёмнику



# Потоки объектов

- Поток объектов (Object Flow) – показывает участие некоторого объекта в двух или более работах

Например, если объект производится в ходе выполнения одной работы и потребляется другой работой.

- Нотация: сплошная линия с двойной стрелкой, ведущей, от источника к приёмнику



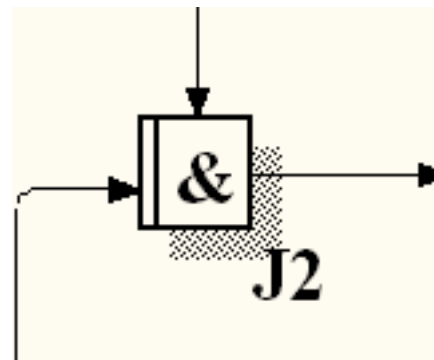


# Перекрестки (Junctions)

- Перекрестки используются в диаграммах IDEF3, чтобы показать ветвления логической схемы процесса и альтернативные пути развития этого процесса

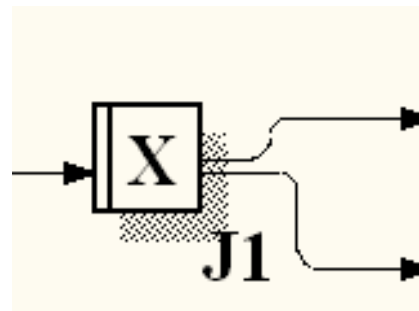
# Виды перекрёстков: слияние

- Перекресток слияния (Fan-in Junction) – узел, собирающий множество стрелок в одну, указывая на необходимость условия завершенности работ-источников стрелок для продолжения процесса



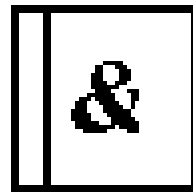
# Виды перекрёстков: ветвление

- Перекресток ветвления (Fan-out Junction) – узел, в котором единственная входящая в него стрелка ветвится, показывая, что работы, следующие за перекрестком, выполняются параллельно или альтернативно



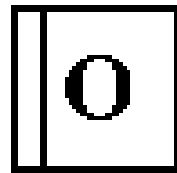
# Виды перекрёстков: AND

- Логическое «И» – логический оператор, определяющий связи между функциями в рамках процесса. Позволяет описать ветвление процесса



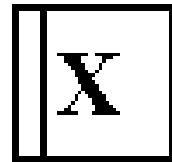
# Виды перекрёстков: OR

- Логическое «ИЛИ» – логический оператор, определяющий связи между функциями в рамках процесса. Позволяет описать ветвление процесса

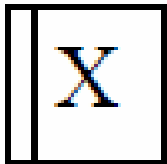
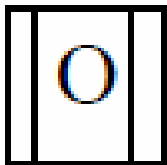
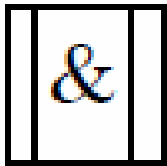


# Виды перекрёстков: XOR

- Логическое «исключающее ИЛИ» – логический оператор, определяющий связи между функциями в рамках процесса. Позволяет описать ветвление процесса

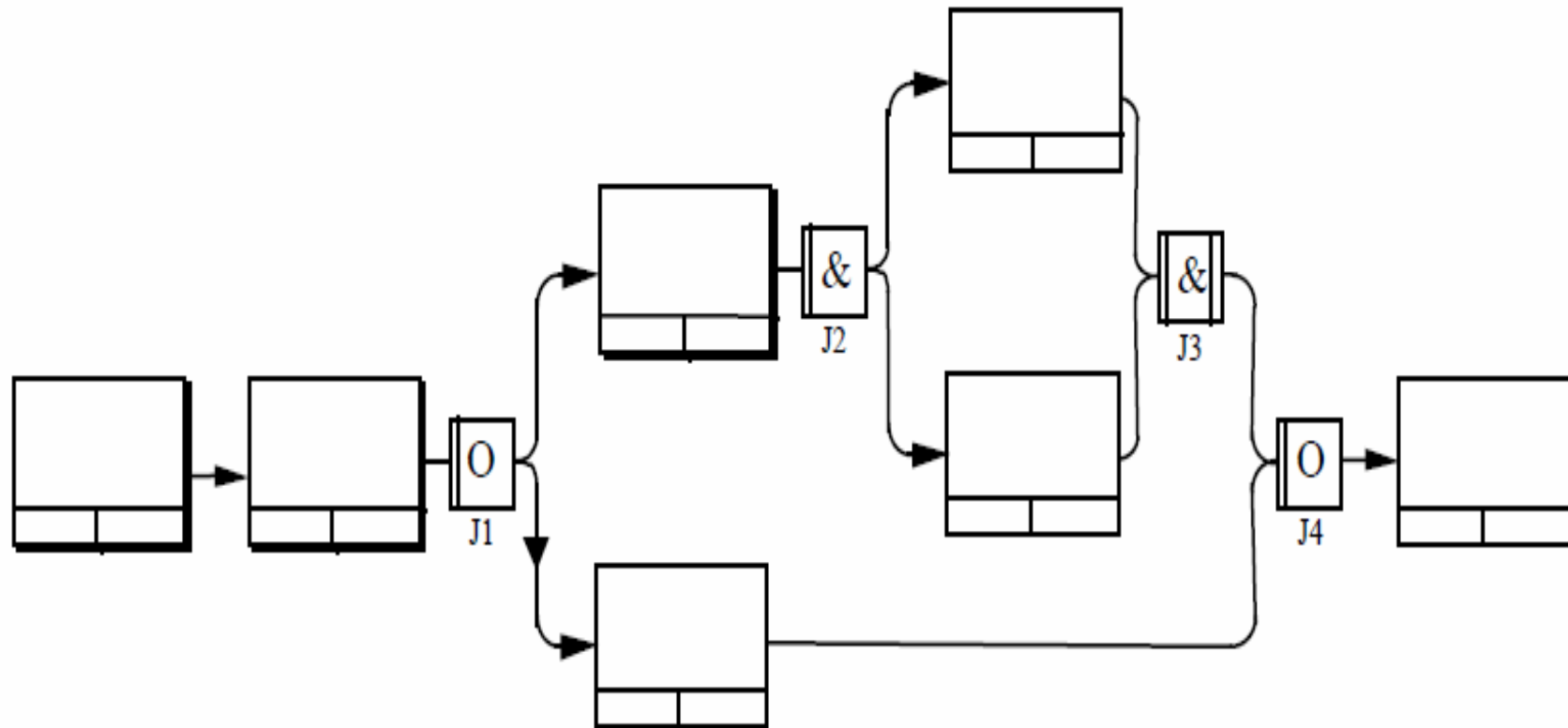


# Виды перекрёстков: асинхронные и синхронные



- Синхронные перекрёстки требуют, чтобы все предшествующие работы (если это перекрёсток слияния) закончились одновременно

# Пример IDEF3 PFDD диаграммы





# Резюме: что узнали на лекции?

- Познакомились с методологией структурного анализа и проектирования:
  - Моделирование потоков данных (Гейна/Сарсона, DFD)
- Познакомились со стандартом документирования процессов
  - IDEF3