

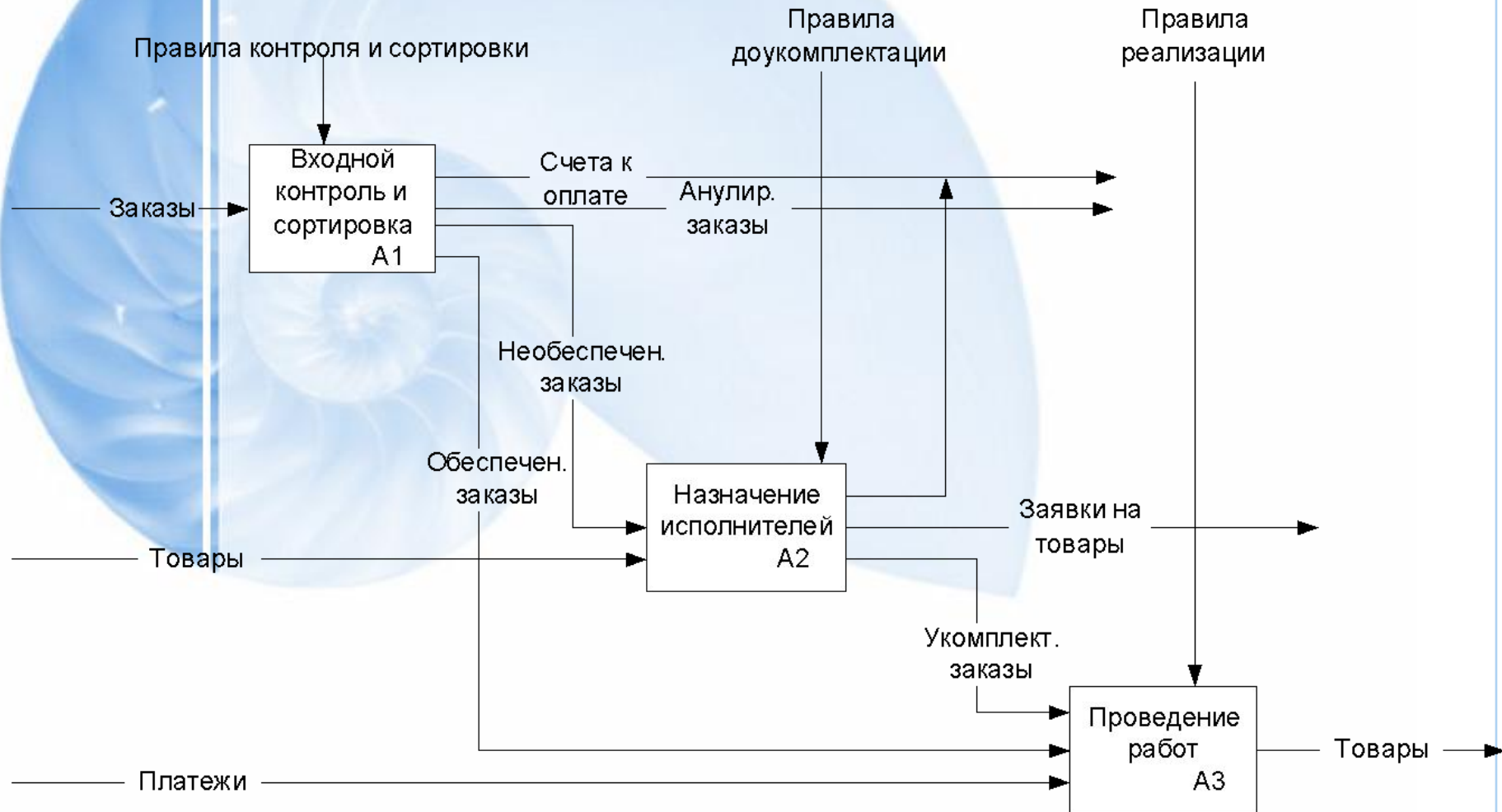
Анализ и структурно- логическое проектирование СИСТЕМ

**Методика Сущность – Связь
Построение структур данных**

**Клевцов С.И. каф. ВС ИРТСУ
ЮФУ**

Функциональное моделирование

4 Пример IDEF0 -диаграммы, моделирующей деятельность компании, занимающейся распределением товаров по заказам



Диаграммы потоков данных

- 4** *Модель DFD имеет два уровня представления:*
- 4** *Первый уровень представления составляют диаграммы верхних уровней иерархии (контекстные диаграммы), которые определяют основные процессы или подсистемы ИС с внешними входами и выходами.*
- 4** *Второй уровень представления составляют собственно диаграммы потоков данных и процессов их обработки. Они детализируют контекстные диаграммы. 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. На этом уровне декомпозиции выполняются мини спецификации.*

Диаграммы потоков данных

- 4 Основные компоненты диаграмм потоков данных:**
- 4 внешние сущности;**
- 4 системы/подсистемы;**
- 4 процессы;**
- 4 накопители данных;**
- 4 потоки данных.**

Диаграммы потоков данных

4 *Внешняя сущность*

4 *Внешняя сущность представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, например, оператор, внешняя система и др.*

Система
сбора
данных
нижнего
уровня

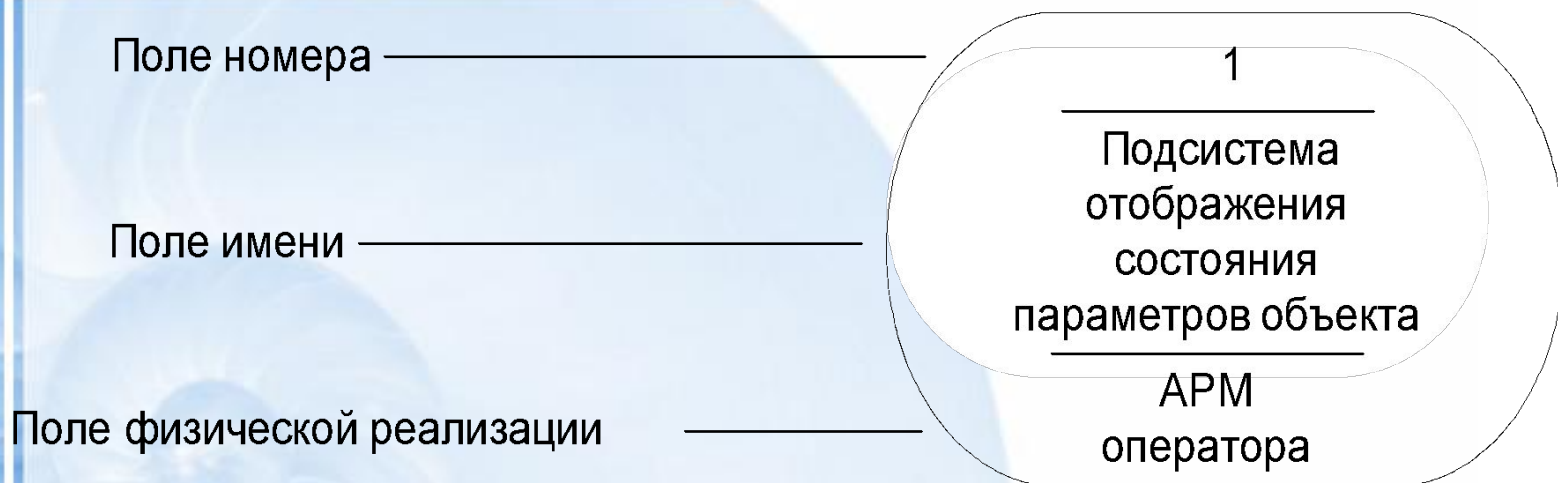
Диаграммы потоков данных

4 *Внешняя сущность*

- 4 *Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС.*
- 4 *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.*

Диаграммы потоков данных

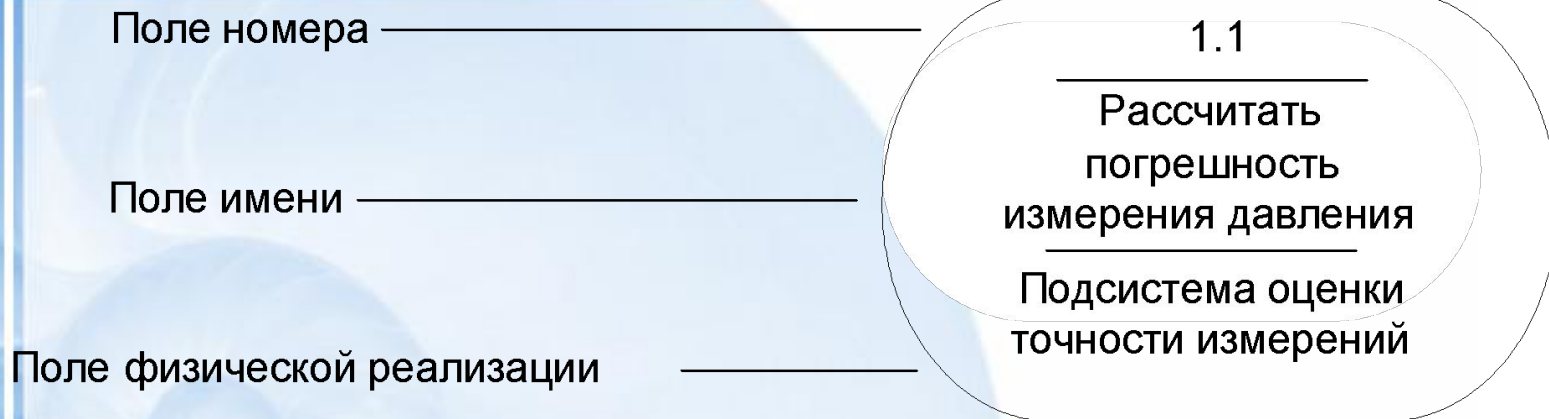
4 Системы и подсистемы



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

Диаграммы потоков данных

4 *Процессы*

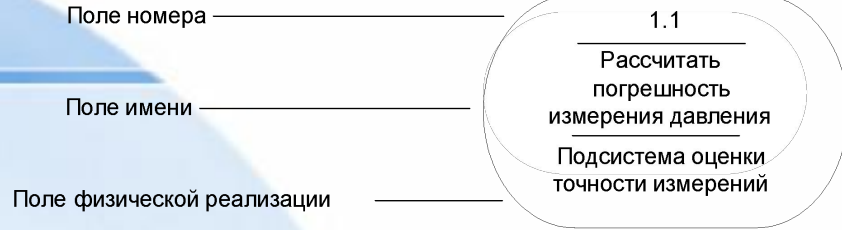


Процесс характеризует конкретное преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

Например, преобразование входного потока данных о значениях сигналов с датчиков в выходной поток вычисленных физических величин (давление, температура и др.).

Диаграммы потоков данных

4 Процессы



- **Номер процесса** служит для его идентификации.
- **В поле имени** вводится наименование процесса в виде предложения с активным глаголом в неопределенной форме (вычислить, рассчитать, ...), за которым следуют существительные в винительном падеже, например:
"Рассчитать давление на объекте";
"Вычислить среднее значение температуры".
- **Фраза**, характеризующая процесс, должна недвусмысленным образом определять конкретную операцию или совокупность операций, которым подвергается входящий поток данных.
- **Информация** в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

Диаграммы потоков данных

4 *Накопители данных*

D1 | Измеренные значения давления

- *Накопитель данных* представляет собой 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

Диаграммы потоков данных

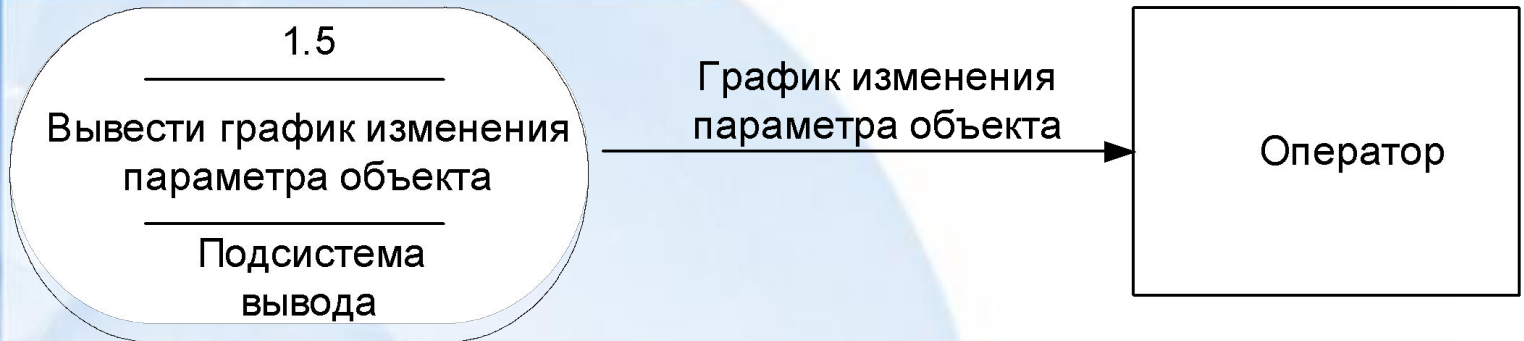
4 *Накопители данных*

D1	Измеренные значения давления
----	------------------------------

- *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.

Диаграммы потоков данных

4 Поток данных



- **Поток данных** определяет информацию, передаваемую через некоторое соединение от источника к приемнику.
- **Реальный поток** данных может быть информацией, передаваемой по линии связи между датчиком и микроконтроллером, по кабелю между двумя вычислительными устройствами и т.д.
- На диаграмме поток данных изображается в виде линии со стрелкой, которая показывает направление потока.
- **Каждый поток** данных должен иметь имя, отражающее его содержание.

Построение иерархии диаграмм потоков данных

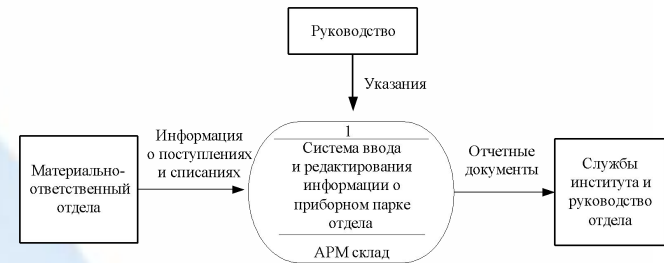
Первый уровень представления модели - уровень контекстных диаграмм.



- **Модель простой ИС** может быть представлена контекстной диаграммой в виде одной системы как единого целого.
- **Модель** в этом случае выглядит как 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 имея, отражающее его содержание.

Построение иерархии диаграмм потоков данных

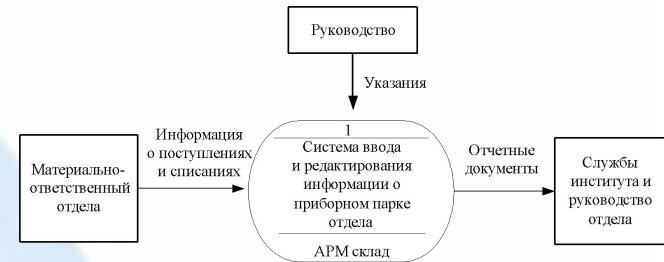
Первый уровень представления модели - уровень контекстных диаграмм.



- **Для сложных ИС** строится иерархия контекстных диаграмм.
- **Для модели сложной ИС** контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а **набор подсистем, соединенных потоками данных.**
- **Признаками сложности ИС** 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.

Построение иерархии диаграмм потоков данных

Первый уровень представления модели - уровень контекстных диаграмм.



- **Начальная диаграмма** последовательно декомпозируется на ряд подсистем нижнего уровня и в итоге может быть представлена в виде иерархии контекстных диаграмм.
- При этом 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 (источниками и приемниками информации), с которыми взаимодействует ИС.

Построение иерархии диаграмм потоков данных

Первый уровень представления модели - уровень контекстных диаграмм.



- ***В заключении полученную иерархию контекстных диаграмм проверяют на полноту исходных данных об объектах системы и изолированность объектов, т.е. отсутствие информационных связей с другими объектами.***

Построение иерархии диаграмм потоков данных

Второй уровень представления модели – уровень диаграмм потоков данных и процессов их обработки

Каждая подсистема, которая присутствует на контекстных диаграммах, детализируется при помощи **DFD – диаграмм** и, как правило, представляется в виде процессов, связанных между собой потоками данных.

Каждый процесс на диаграмме, в свою очередь, может быть детализирован при помощи **DFD – диаграмм** или, если дальнейшая детализация на подпроцессы невозможна или нецелесообразна, представлен в виде спецификации.

Построение иерархии диаграмм потоков данных

Второй уровень представления модели – уровень диаграмм потоков данных и процессов их обработки

- **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.

Построение иерархии диаграмм потоков данных

Второй уровень представления модели – уровень диаграмм потоков данных и процессов их обработки

- **Спецификация** является конечной вершиной иерархии DFD.
- **Решение** о завершении детализации процесса и использовании спецификации принимается 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 спецификации небольшого объема (не более 20-30 строк).**

Построение иерархии диаграмм потоков данных

Второй уровень представления модели – уровень диаграмм потоков данных и процессов их обработки

- **Важно** отметить, что до построения иерархии DFD-диаграмм для детализации процессов необходимо определить содержание всех потоков и накопителей данных, которое описывается при помощи структур данных.
- **Структуры данных** конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации.
 - Условное вхождение** означает, что данный компонент может отсутствовать в структуре.
 - Альтернатива** означает, что в структуру может входить один из перечисленных элементов.
 - Итерация** означает вхождение любого числа элементов в указанном диапазоне.

Построение иерархии диаграмм потоков данных

Второй уровень представления модели – уровень диаграмм потоков данных и процессов их обработки

- **Верификация законченной модели системы (проверка на полноту и согласованность):**
- **В полной модели** все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы.
- **Выявленные недетализированные объекты** следует детализировать, вернувшись на предыдущие шаги разработки.
- **В согласованной модели** для всех потоков данных и накопителей данных должно выполняться правило сохранения информации:
все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

Правила построения диаграмм потоков данных

- **1. Размещать на каждой диаграмме от 3 до 6-7 процессов.**
- **2. Не загромождать диаграммы несущественными на данном уровне деталями.**
- **3. Декомпозицию потоков данных осуществлять параллельно с декомпозицией процессов.**
- **4. Выбирать ясные, отражающие суть дела, имена процессов и потоков для улучшения понимаемости диаграмм, при этом стараться не использовать аббревиатуры.**

Правила построения диаграмм потоков данных

Соблюдать следующие этапы:

- 1) Идентификация внешних объектов, с которыми система должна быть связана.**
- 2) Идентификация основных видов информации, циркулирующей между системой и внешними объектами.**
- 3) Предварительная разработка контекстной диаграммы.**
- 4) Изучение предварительной контекстной диаграммы и внесение в нее изменений по результатам ответов на возникающие при этом изучении вопросы по всем ее частям.**
- 5) Построение контекстной диаграммы путем объединения всех процессов предварительной диаграммы в один процесс, а также группирования потоков.**
- 6) Формирование DFD первого уровня на базе процессов предварительной контекстной диаграммы.**
- 7) Проверка основных требований по DFD первого уровня.**
- 8) Декомпозиция каждого процесса текущей DFD с помощью детализирующей диаграммы или спецификации процесса.**

Правила построения диаграмм потоков данных

Соблюдать следующие этапы:

9) Проверка основных требований по DFD соответствующего уровня.

10) Добавление определений новых потоков в словарь данных при каждом их появлении на диаграммах.

11) Параллельное (с процессом декомпозиции) изучение требований (в том числе и вновь поступающих), разбиение их на элементарные и идентификация процессов или спецификаций процессов, соответствующих этим требованиям.

12) После построения двух-трех уровней проведение ревизии с целью проверки корректности и улучшения понимаемости модели.

13) Построение спецификации процесса (а не простейшей диаграммы) в случае, если некоторую функцию сложно или невозможно выразить комбинацией процессов.

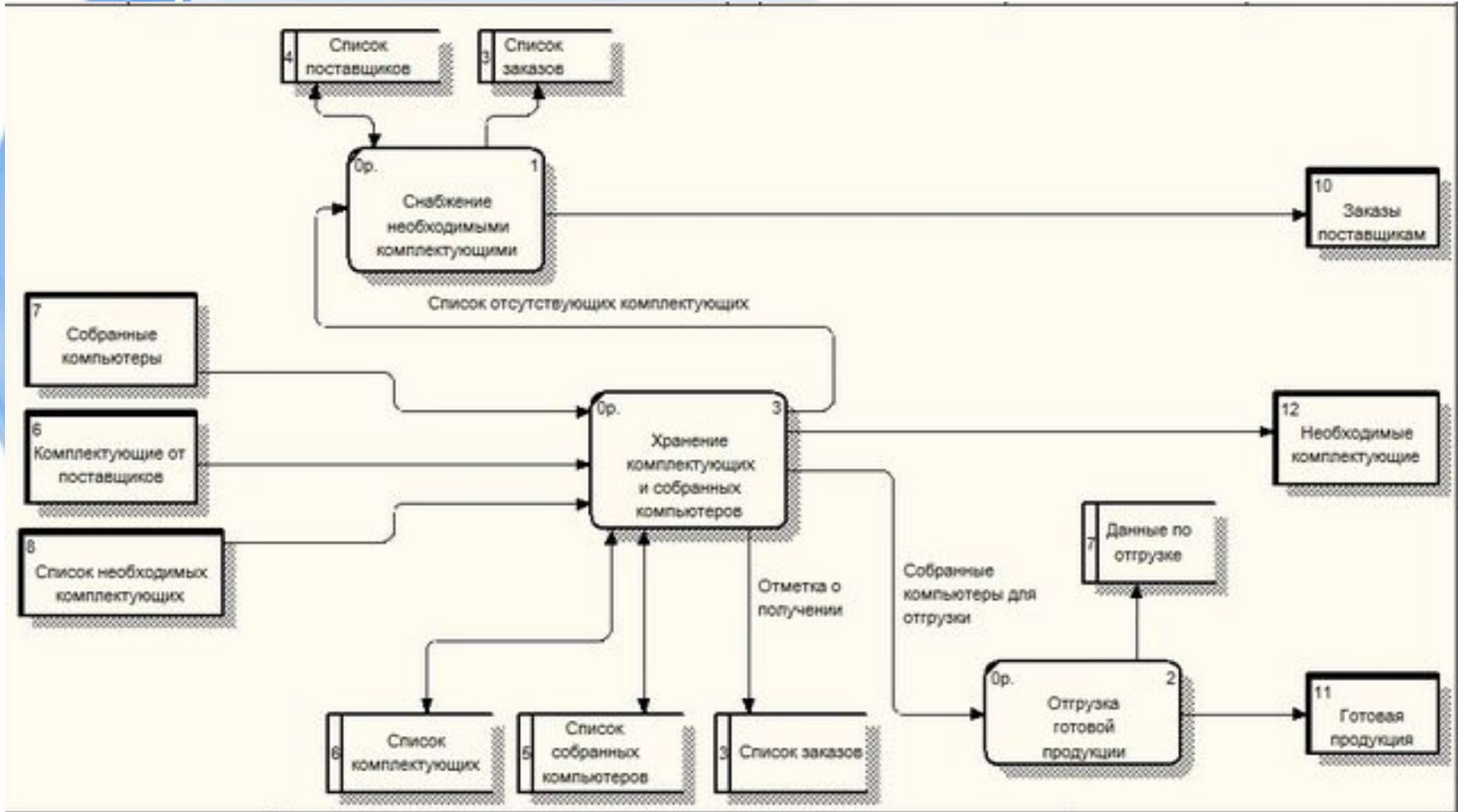
Построение иерархии диаграмм потоков данных

Примеры диаграмм



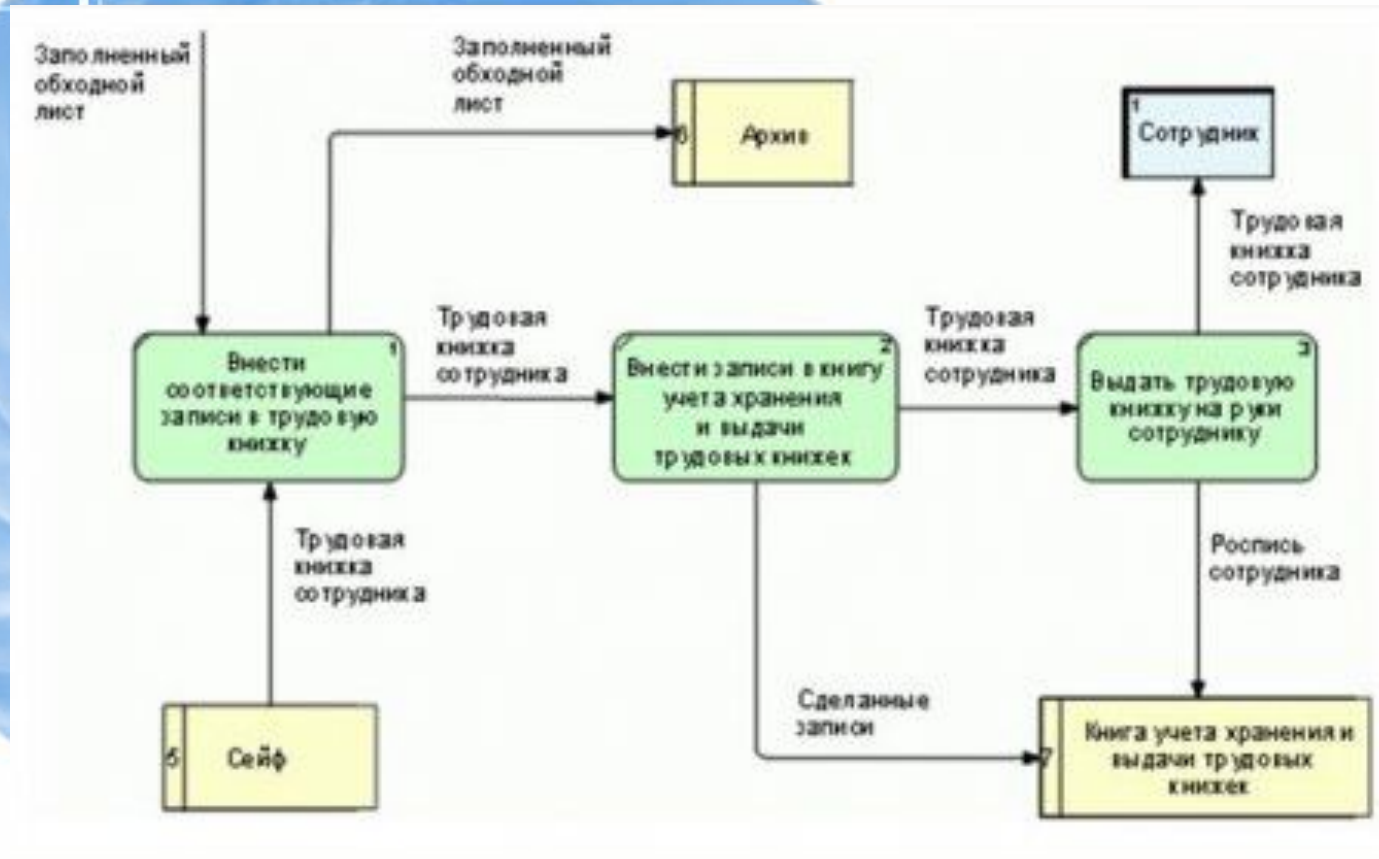
Построение иерархии диаграмм потоков данных

Примеры диаграмм



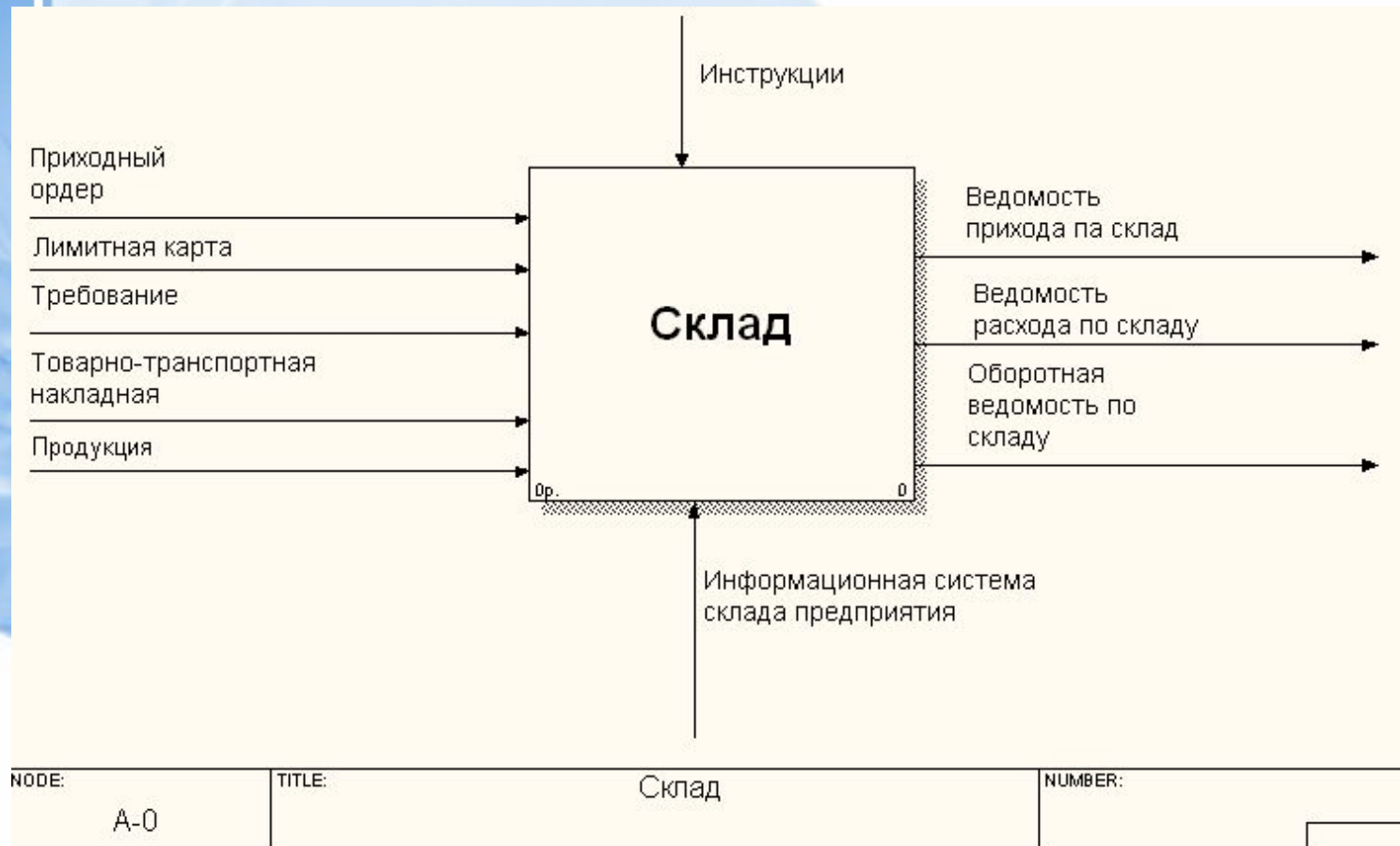
Построение иерархии диаграмм потоков данных

Примеры диаграмм



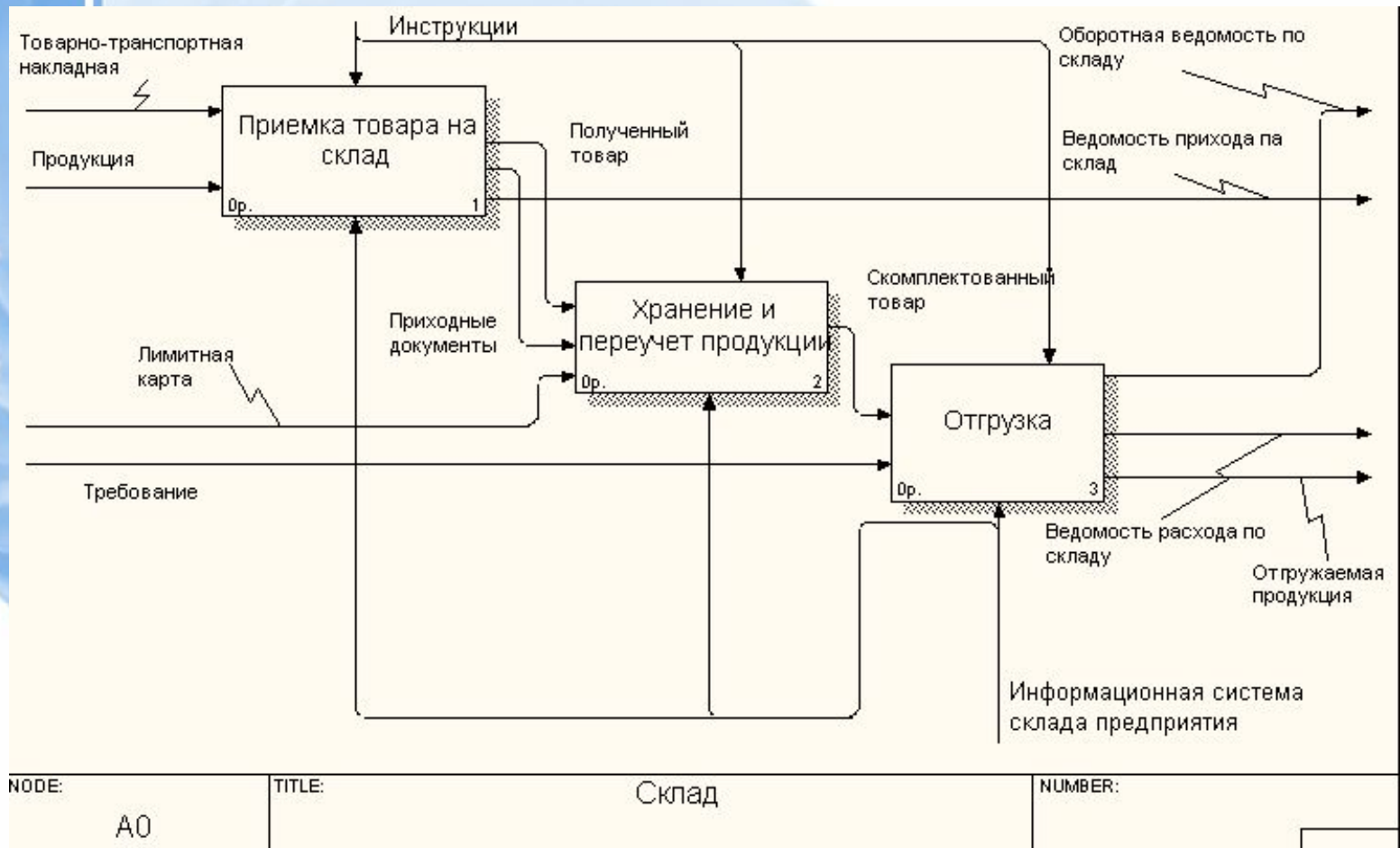
4 Пример IDEF0 -диаграммы, моделирующей деятельность склада

4 Контекстная диаграмма



4 Пример IDEF0 -диаграммы, моделирующей деятельность склада

4 Детализация



- 4 Далее моделировать систему будем, используя диаграммы потоков данных (DFD).
- 4 Декомпозируем функциональный блок «Приемка товара на склад»

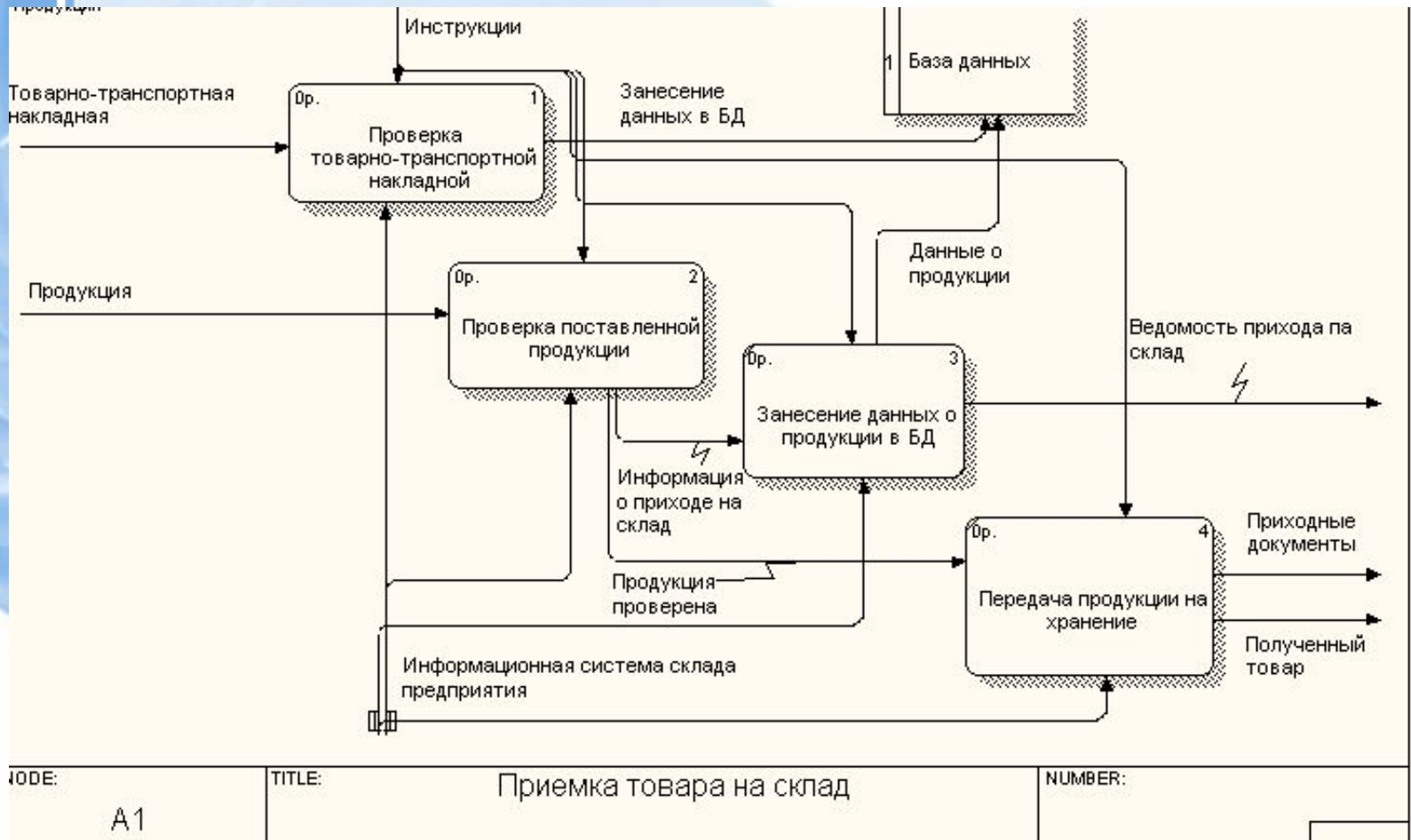


Диаграмма DFD «Приемка товара на склад»

- 4 Далее моделировать систему будем, используя диаграммы потоков данных (DFD).
- 4 Декомпозируем функциональный блок «Хранение и переучет продукции»

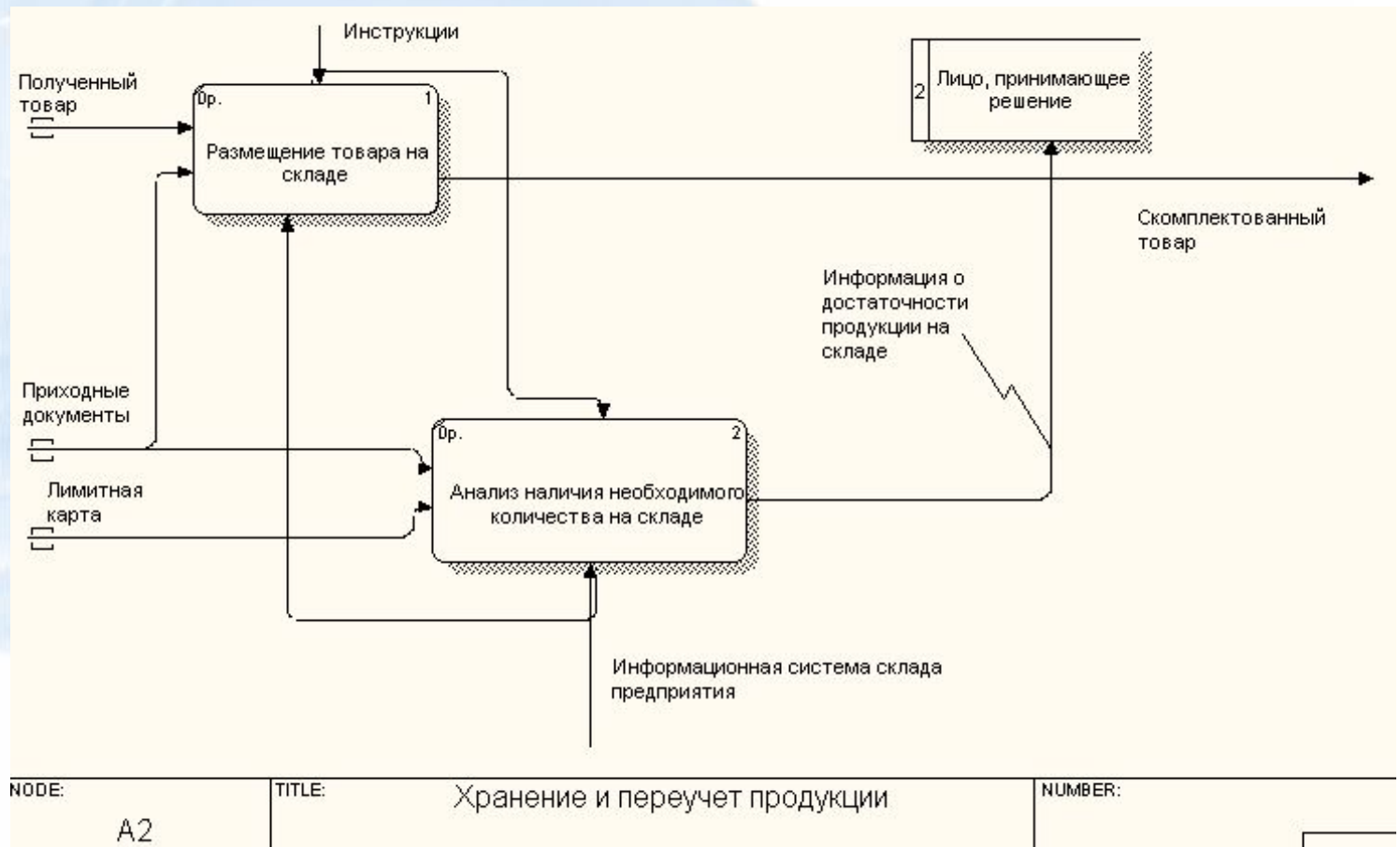


Диаграмма DFD «Хранение и переучет продукции»

- 4 Далее моделировать систему будем, используя диаграммы потоков данных (DFD).
- 4 Декомпозируем функциональный блок «Отгрузка»

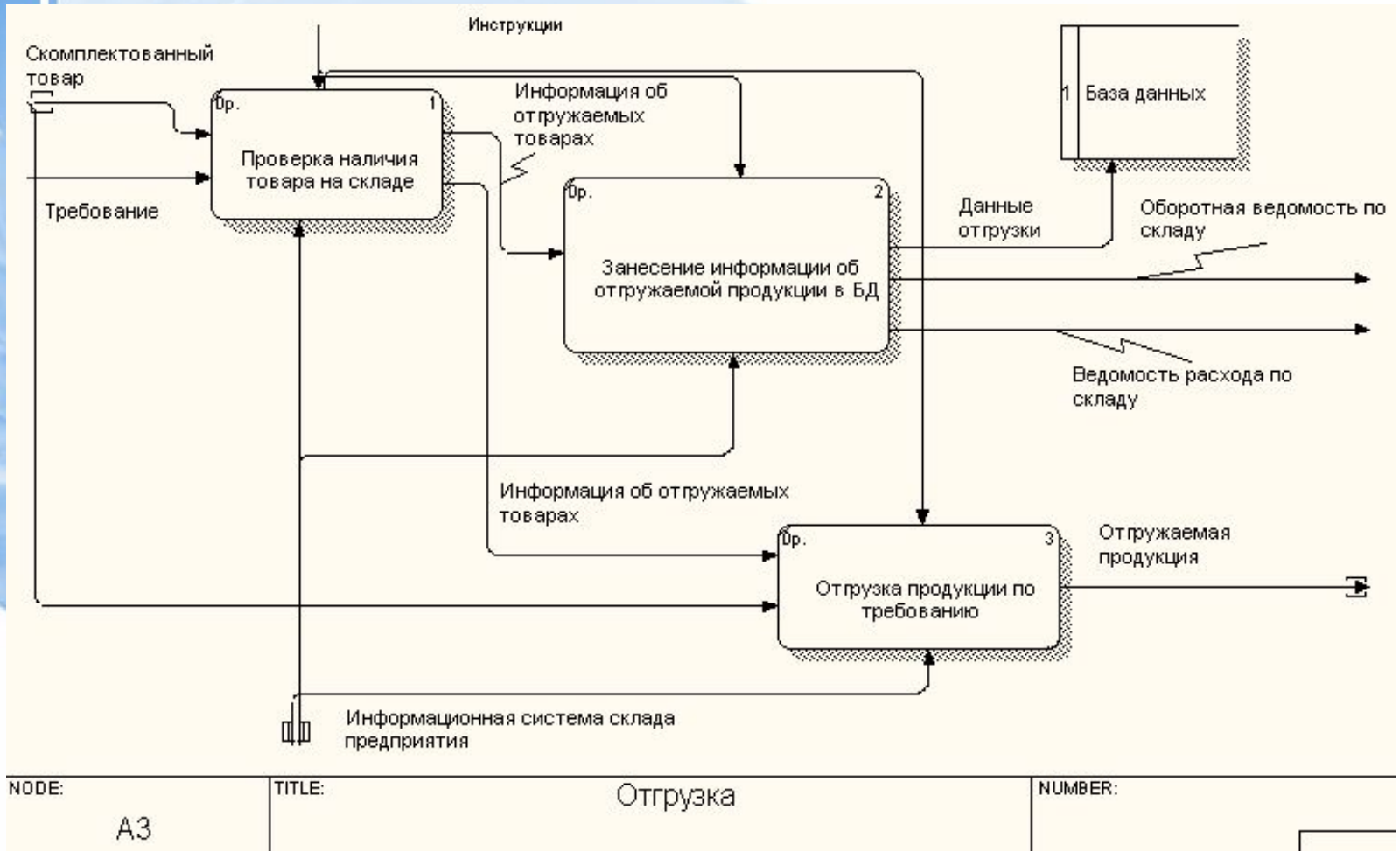


Диаграмма DFD «Отгрузка»

Методика "сущность-связь" построения структур баз данных

Термины, используемые при проектировании БД

База данных — некоторый упорядоченный банк информации, предназначенный для хранения необходимых данных.

Система управления базой данных (СУБД) — программный комплекс, отвечающий за работу с базой данных, как-то: ее сортировка, извлечение информации по запросу пользователя и некоторые другие действия.

Поле — самая маленькая единица хранения информации, из которой строятся другие более крупные единицы данных — записи. В поле обычно хранится один атрибут для описания некоего объекта, информация о котором хранится в базе данных.

Запись — набор полей, объединенных в единую структуру, на значение которой — хранить информацию об экземпляре интересующего нас объекта. Иногда в литературе запись называют **кортежем**.

Методика "сущность-связь" построения структур баз данных

Термины, используемые при проектировании БД

Таблица — набор записей, в который входит информация обо всех экземплярах объекта. Обычно каждая таблица сохраняется в виде отдельного файла.

	Один столбец соответствует одному полю		Поле записи	
	Имя 1	Имя 2	Имя 3	Имя 4
Одна строка соответствует одной записи	<i>ФИО</i>	<i>Начислено</i>	<i>Удержано</i>	<i>Сумма</i>
	Иванов В.А.	8900	1200	7700
<i>Запись БД</i>	Дроздов И.И.	12300	2000	10300
	<u>Жариков А.А.</u>	10870	1340	9530

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Техническое задание:

Создать систему хранения и вычисления заработной платы с учетом различных премий, надбавок и налогов.

В простейшем случае мы можем создать таблицу, состоящую из четырех полей:

"Фамилия",

"Начислено",

"Удержано",

"Сумма",

и такой "плоской" базы данных нам бы вполне могло хватить.

Но есть много недостатков!!!

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Необходимо преобразование плоской базы данных в реляционную.

Реляционная база данных представляет собой набор таблиц, которые содержат всю необходимую информацию.

Чтобы преобразовать старую базу данных в реляционную БД, мы должны произвести **нормализацию, то есть разбиение универсальной таблицы на несколько простых.**

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Для правильной нормализации мы воспользуемся методикой "сущность-связь".

Сущности — это объекты, которыми мы оперируем при помощи СУБД и которые нас интересуют.

Связи - показывают, как эти сущности взаимодействуют между собой.

Для нашей БД одна из связей сущности звучит примерно так: "**Работник получает Деньги**".

Здесь "Работник" и "Деньги" — **сущности**, а "получает" — **связь**, описывающая их взаимодействие.

Связи зачастую выглядят как глаголы с предлогами, если предлог есть.

Определяя сущности, необходимо выделить их **атрибуты**, которые будут использованы в базе данных.

Атрибуты — это свойства, которыми сущности обладают.

Так, для сущности "Работник" нас будут интересовать такие атрибуты, как "Фамилия Имя Отчество" ("ФИО"), "Табельный номер" и "Должность".

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Определим степени связей:

Степень связи — абстрактная характеристика, показывающая, сколько элементов одной сущности связано со сколькими элементами другой сущности в рамках решения конкретной задачи.

Степень связи может принимать значения:

- **Один к одному** – $1 : 1$ – один элемент одной сущности связан только с одним элементом другой сущности;
- **Один к многим** – $1 : M$ – один элемент одной сущности связан со многими элементами другой сущности;
- **Многие к одному** – $M : 1$ – многие элементы одной сущности связаны с одним элементом другой сущности;
- **Многие ко многим** – $M : M$ – многие элементы одной сущности связаны со многими элементами другой сущности.

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Класс принадлежности – характеристика, определяющая, обязательно или нет все элементы сущности должны быть связаны с элементами другой сущности.

Классы принадлежностей:

Обязательный – (O);

Необязательный – (N)

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

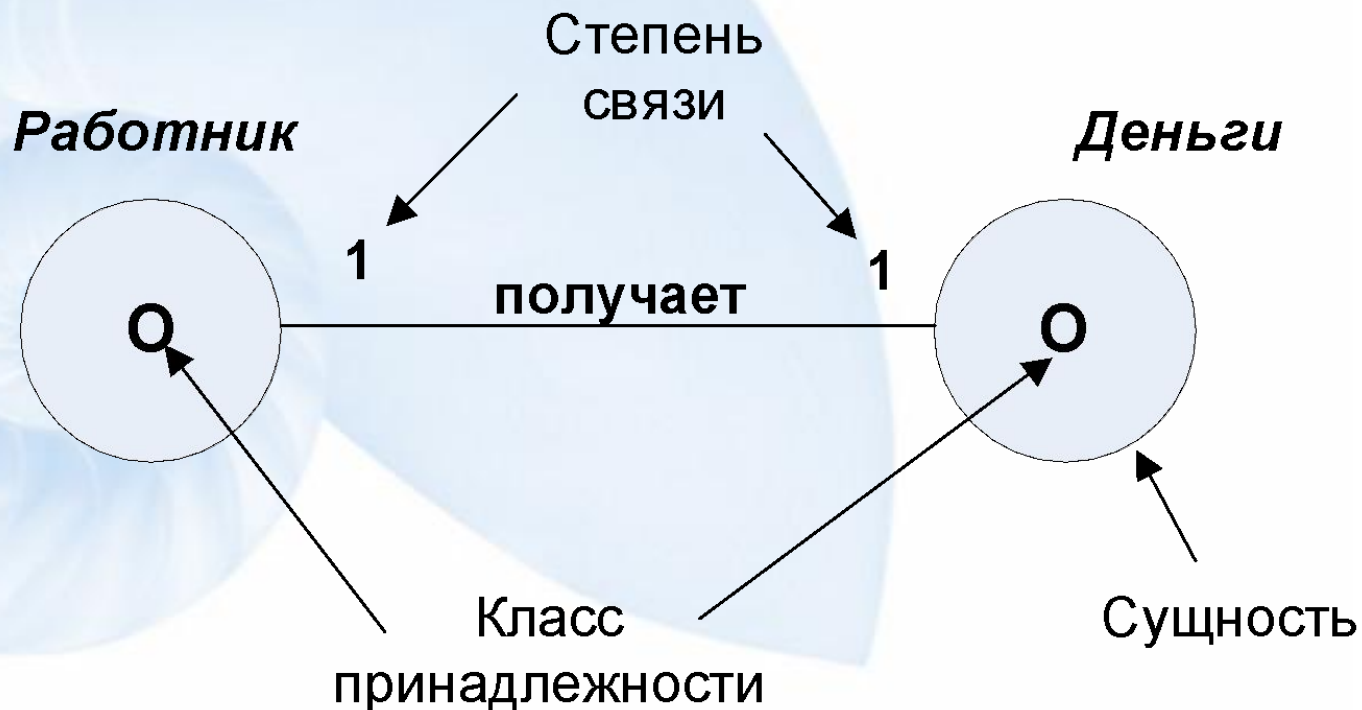


Диаграмма для конструкции "Работник получает Деньги"

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Понятие ключа

Ключ, или **первичный ключ**, как его часто называют, помогает однозначно выделить именно ту запись в БД, которая нам нужна.

Для этого в таблице выбирается одно или несколько полей, по содержимому которого (которых) можно с полной уверенностью установить, ту ли запись мы нашли.

Ключ из нескольких полей называют **композиционным**, то есть **составным**.

Нахождение ключа для таблицы — дело довольно трудное.

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Усложним задачу.

Пусть сумма выплаты работнику будет вычисляться из нескольких сумм, каждая из которых считается за каждую выполненную работу отдельно. Видов работ также может быть несколько.

Таким образом, мы имеем **четыре сущности:**

"Работник" - информация о сотруднике,
"Виды работ" - типовые работы, выполняемые на предприятии, и расценки,
"Выполненные работы" - список закрытых нарядов,
"Баланс" - заработанная сумма, налоги и итоговая сумма для получения.

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Атрибуты сущностей:

"Работник": Фамилия, Имя, Отчество, Табельный номер, Должность.

"Виды работ" : Наименование, Стоимость.

"Выполненные работы": Наименование вида, номер закрытого наряда, объект.

"Баланс": Сумма выплат.

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

Возможные конструкции:

Работник выполняет разные Виды работ.

Выполненные работы состоят из разных Видов работ.

Работник участвовал в Выполнении работ (Выполненные работы).

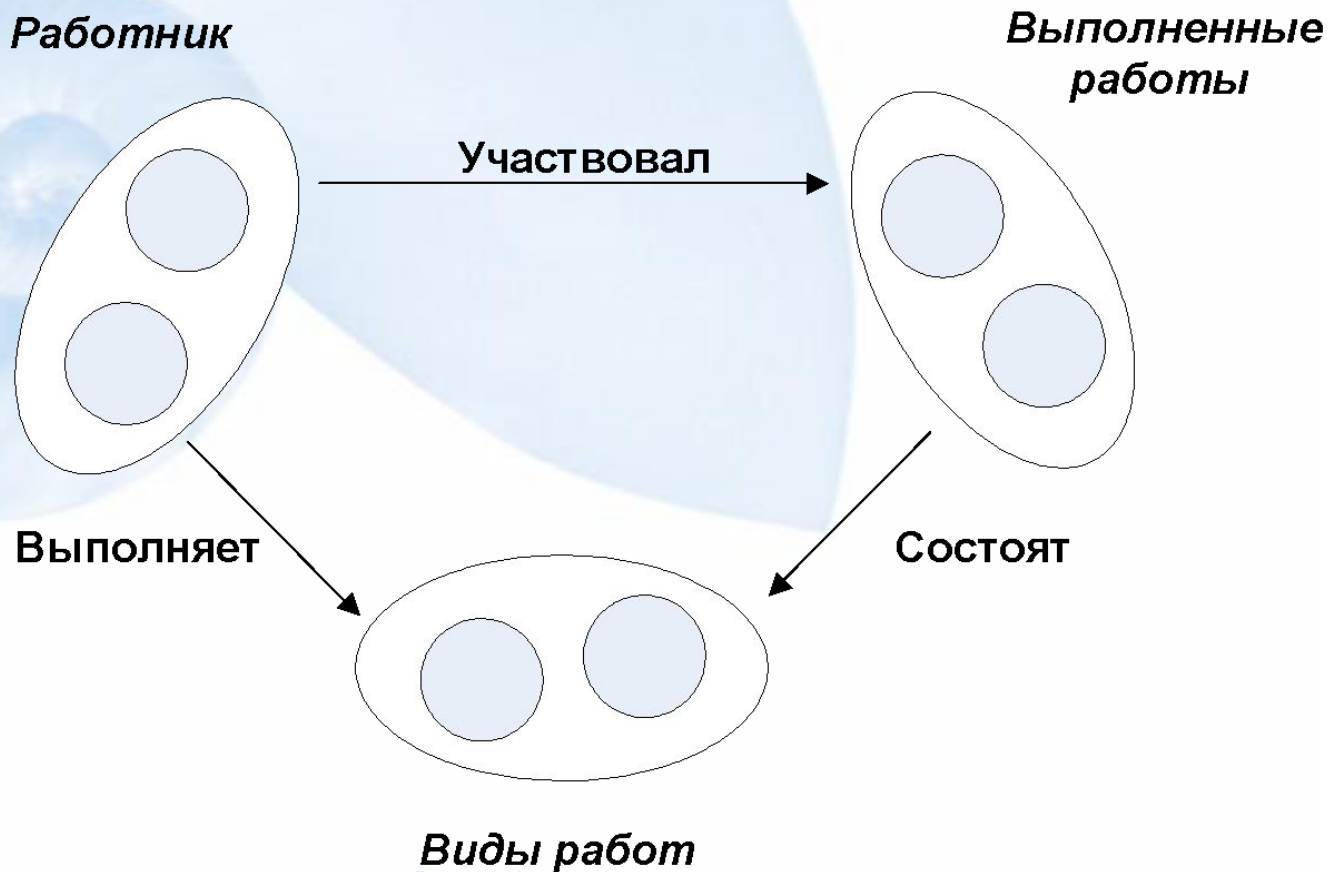
Работник имеет Баланс выплаты.

Выполненные работы составляют Баланс.

Методика "сущность-связь" построения структур баз данных

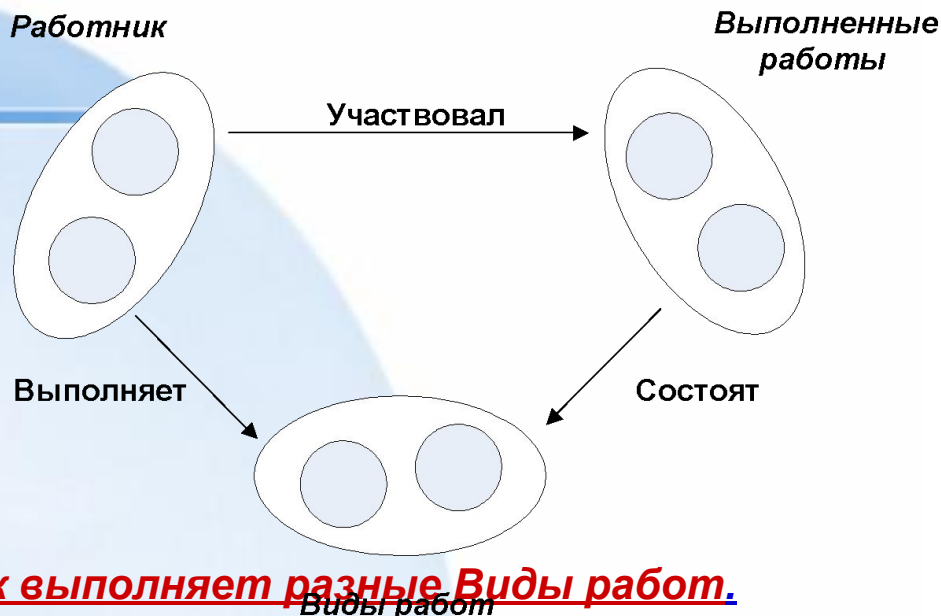
Пример применения методики - Система подсчета зарплаты.

Структура диаграммы будет выглядеть следующим образом



Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.



Анализ конструкции **Работник выполняет разные Виды работ.**

1. Определение класса принадлежности:

каждый работник должен выполнять какой-то вид работы, следовательно, здесь сущность "Работник" имеет обязательный класс принадлежности - О.

работники предприятия должны выполнять весь спектр видов работ (имеется в виду нормальное предприятие). Следовательно, сущность «Виды работ» имеет также обязательный класс принадлежности - О.

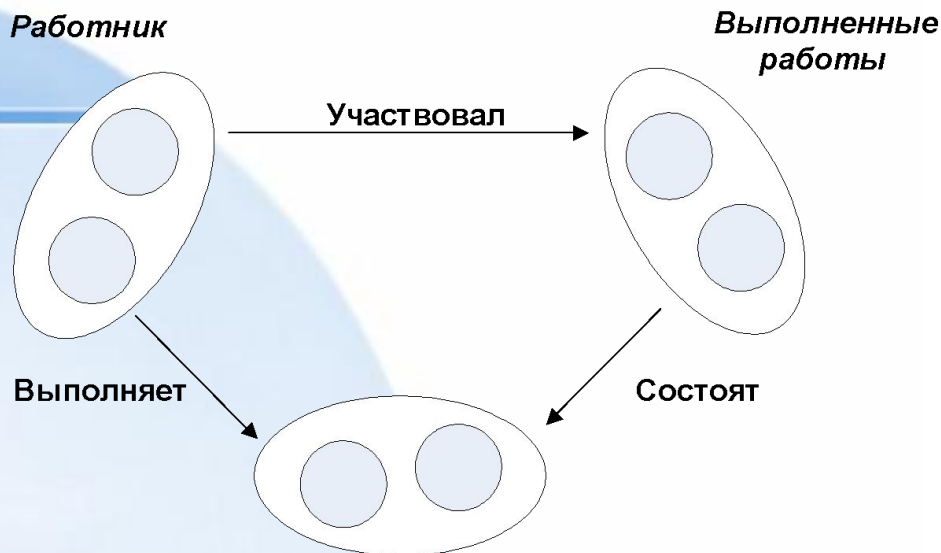
2. Определение степени связи:

каждый работник может выполнять несколько видов работ;
каждый вид работы может быть выполнен несколькими работниками

Следовательно, конструкция имеет степень связи "многие ко многим" (М:М).

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.



Анализ конструкции **Выполненные работы состоят из разных Видов работ.**

1. Определение класса принадлежности:

каждая выполненная работа должна быть определенного вида, следовательно, здесь сущность **Выполненные работы** имеет обязательный класс принадлежности - O.

некоторые виды работ могут в текущем месяце не выполняться совсем, что дает сущности "**Вид работ**" в данной конструкции имеет необязательный класс принадлежности - N.

2. Определение степени связи:

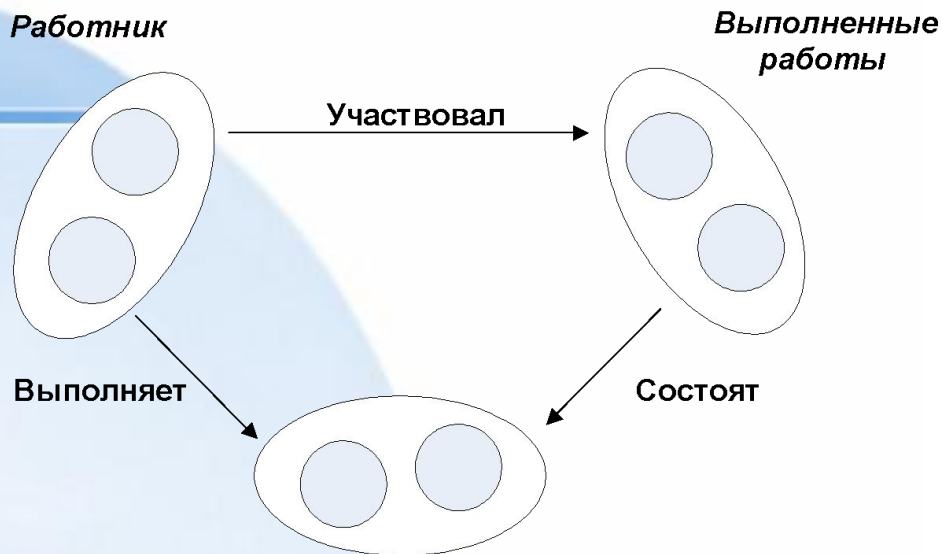
многие выполненные работы могут быть одного вида;

каждая выполненная работа может быть только одного вида.

Следовательно, конструкция имеет степень связи "**многие к одному**" (M:1).

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.



Анализ конструкции **Работник участвовал в выполнении работ**

1. Определение класса принадлежности:

каждый работник должен участвовать в выполнении работ, следовательно, здесь сущность **Работник** имеет **обязательный класс принадлежности - О.**

каждая выполненная работа имеет своего исполнителя, то есть сущность **Выполненные работы** в данной конструкции имеет **обязательный класс принадлежности - О.**

2. Определение степени связи:

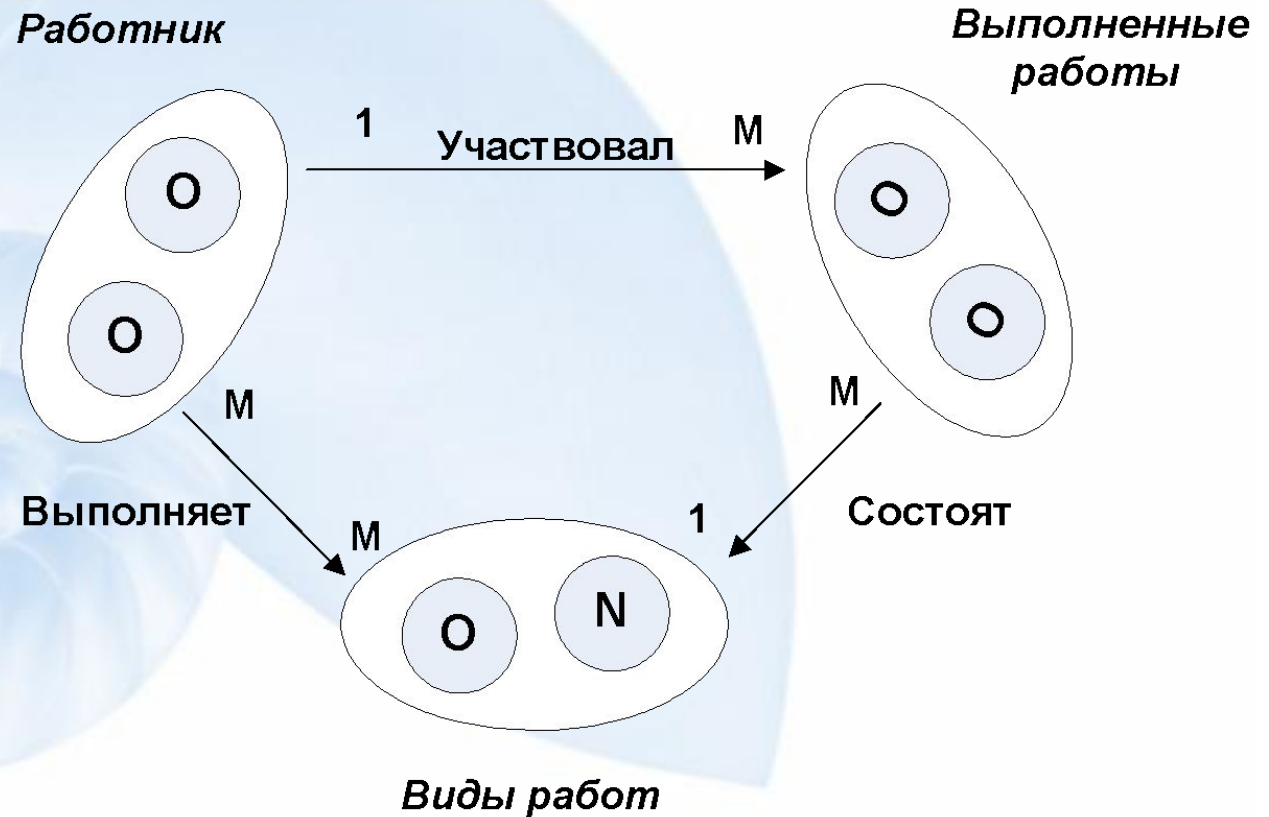
один работник может участвовать в выполнении множества работ;

одна выполненная работа выполняется одним работником.

Следовательно, конструкция имеет степень связи **"один ко многим" (1:М).**

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.



Итоговая диаграмма

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №1

Если имеется связь степени 1:1 и класс принадлежности обеих сущностей обязательный, то требуется всего одна таблица.

Первичным ключом этой таблицы может быть любой из ключей сущностей.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №2

Если имеется связь степени 1:1 и класс принадлежности одной сущности является обязательным, а другой — необязательным, то требуются **две таблицы.**

Для каждой сущности необходимо создать свою таблицу, первичным ключом которой будет ключ этой сущности.

Кроме того, первичный ключ сущности с необязательным классом принадлежности должен быть добавлен в качестве атрибута в таблицу, созданную для сущности с обязательным классом принадлежности.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №3

Если имеется связь степени 1:1 и класс принадлежности обеих сущностей является необязательным, то требуются **три таблицы.**

Для каждой сущности необходимо создать свою таблицу, первичным ключом которой будет ключ этой сущности.

Кроме того, связующая таблица должна содержать в себе ключи других таблиц в качестве атрибутов.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №4

Если имеется связь степени 1 :М и класс принадлежности М-связанной сущности является обязательным, то требуются **две таблицы.**

Для каждой сущности необходимо создать свою таблицу, первичным ключом которой будет ключ этой сущности.

Кроме того, первичный ключ 1-связанной сущности должен быть добавлен в качестве атрибута М-связанной таблицы.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №5

Если имеется связь степени 1:M и класс принадлежности M-связанной сущности является необязательным, то требуются **три таблицы.**

Для каждой сущности необходимо создать свою таблицу, первичным ключом которой будет ключ этой сущности.

Кроме того, связующая таблица должна содержать в себе ключи других таблиц в качестве ее атрибутов.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №6

Если имеется связь степени M:M, то требуются *три* таблицы.

Для каждой сущности необходимо создать свою таблицу, первичным ключом которой будет ключ этой сущности.

Кроме того, связующая таблица должна содержать в себе ключи других таблиц в качестве ее атрибутов.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

Правило №7

При наличии многосторонней связи необходимо создать свою таблицу для каждой сущности, первичным ключом которой будет ключ этой сущности.

Кроме того, необходимо создать еще одну таблицу связи, которая будет содержать первичные ключи остальных таблиц в качестве своих атрибутов.

То есть для *n*-сторонней связи необходимо создать *n+1* таблиц.

Методика "сущность-связь" построения структур баз данных

Правила построения таблиц БД на основе диаграмм «сущность - связь» :

В соответствии с правилом 7 нужно создать четыре таблицы по одной для каждой сущности и дополнительную таблицу связи.

1 таблица: Работник ("Табельный номер", "ФИО", "Отдел").

2 таблица: Виды работ ("Название работы", "Расценка").

3 таблица: Выполненные работы ("Наряд №", "Количество единиц работы", "Сумма").

4 таблица: Таблица связи ("Табельный номер", "Название работы", "Наряд №").