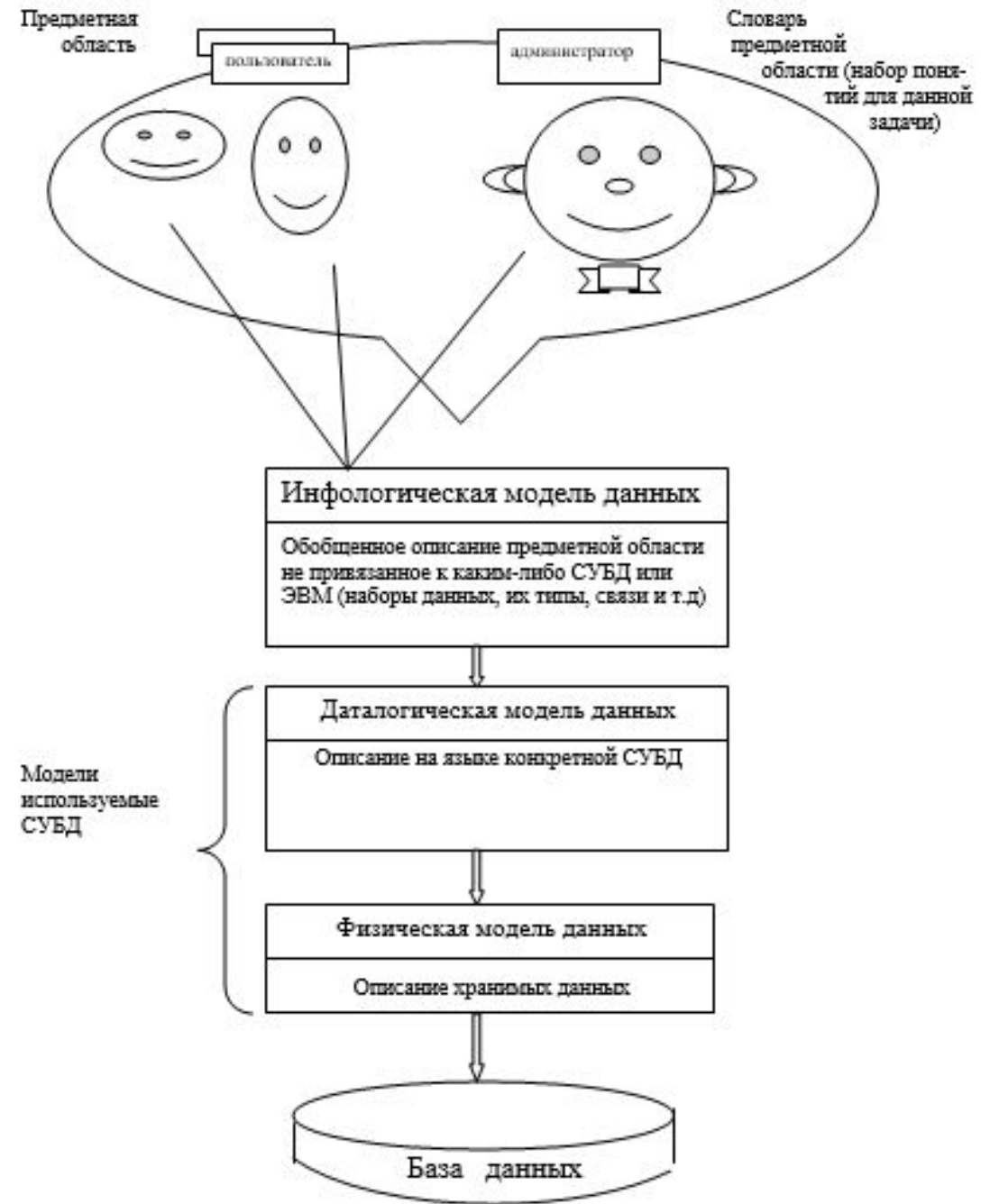


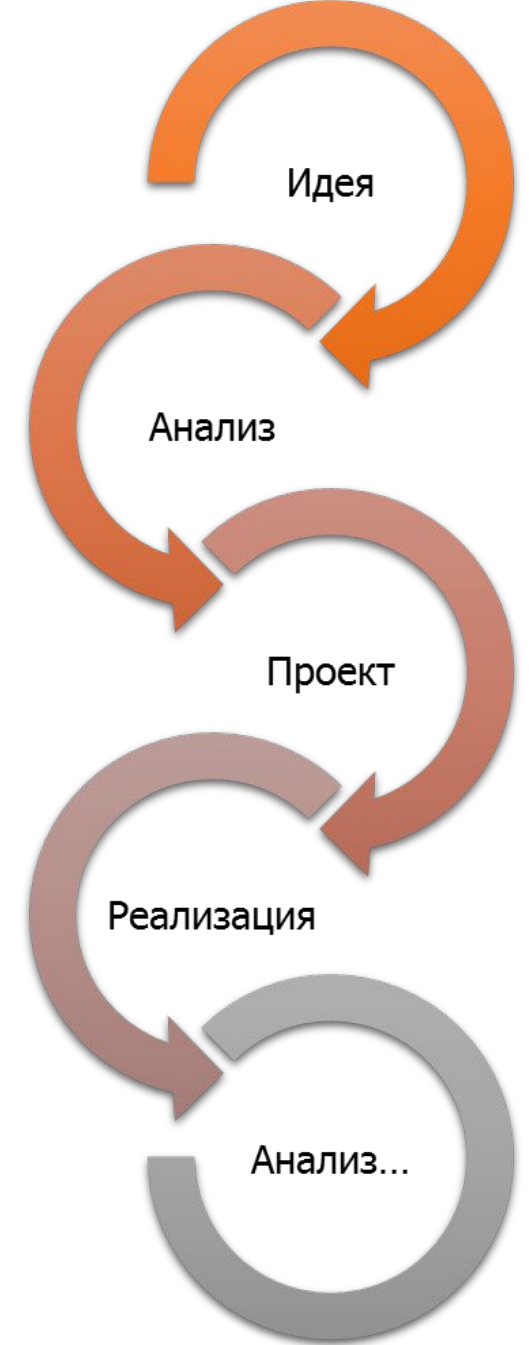
методы проектирования и программирования программного обеспечения

Моделирование данных

Формирование БД



Базовые понятия и определения информационной архитектуры



О важности ИА

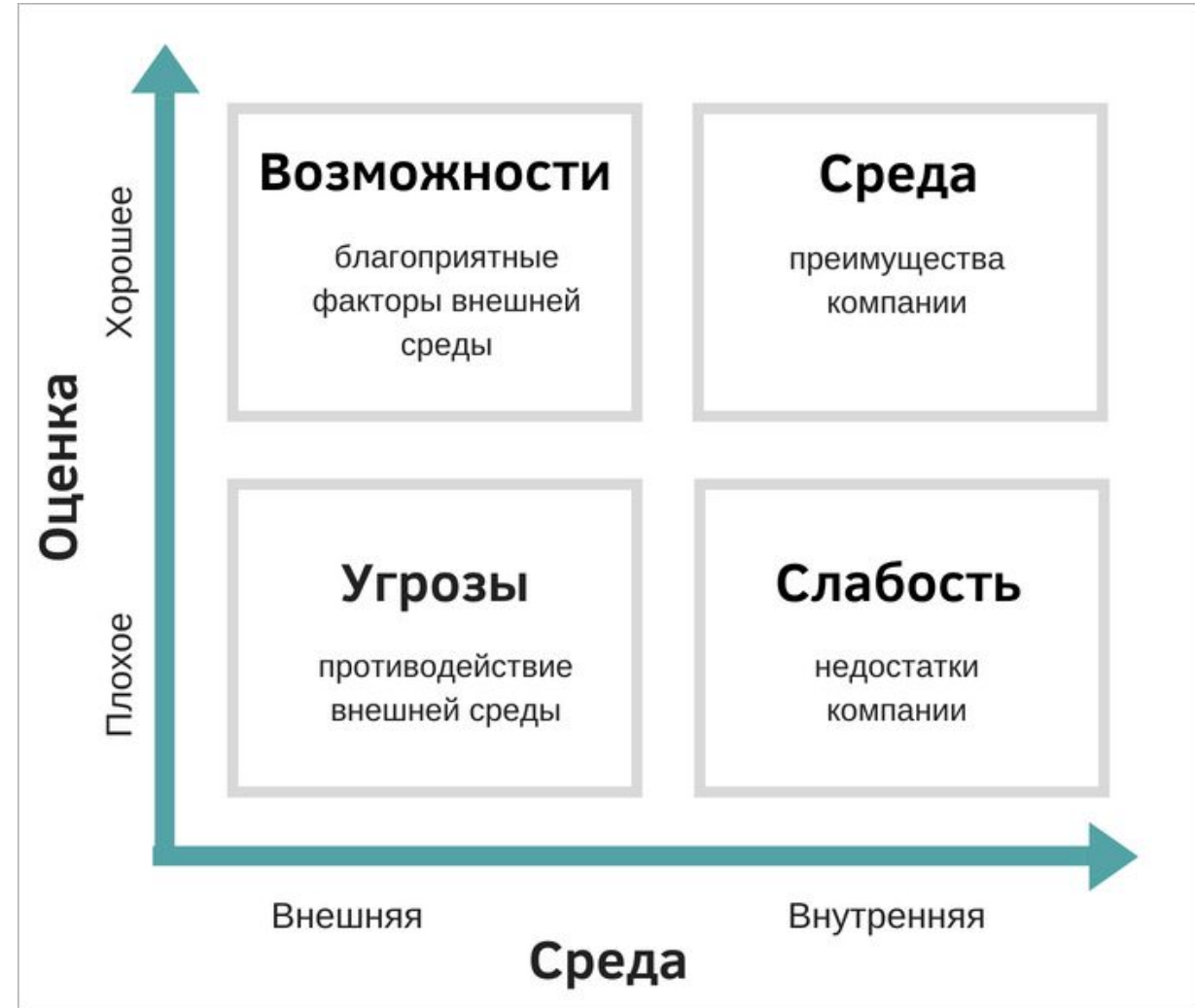
Понимание приложения как единого продукта

ИА начинается с ситуационного анализа.

Первое – для кого?

Ситуационный анализ включает:

- анализ текущей ситуации: понять, на каком этапе вы находитесь;
- анализ целевой аудитории: особенности её восприятия, поведения, потребления;
- обзор рынка конкурентов: правила поведения, привычные пользовательские паттерны;
- точки принятия решений.



Постановка целей и задач

Коммуникационные
цели

Бизнес-цели

Задачи, которые
должны быть для
этого решены

Бизнес-цели касаются денег, окупаемости.

Коммуникационные цели — это то, что не измерить деньгами, но можно измерить какими-то другими показателями. С коммуникационными целями сложнее, потому что могут быть совершенно разные задачи. Например, повысить знание бренда — это вполне измеряемый показатель. Вы хотите привлечь миллион пользователей и удержать около семидесяти процентов или повысить вирусность продукта с определённым показателем?

Из целей формулируете задачи — определённые шаги, которые должны быть сделаны, чтобы достигнуть поставленные цели. Цели и задачи, которые **ОПРЕДЕЛЯЮТСЯ** на этом этапе, **станОВЯТСЯ** основой будущей информационной архитектуры продукта.

Концепция сайта

Концептуальные
инструменты

Навигационная
концепция

Точки принятия
решений

Из этого строится информационная архитектура

Концептуальные инструменты — сквозные решения, которые определяют продукт как единое целое, характерные только для этого продукта. С помощью которых достигаются поставленные цели.

Навигационная концепция — зависит от приоритетов, которые задаются в архитектуре.

Точки принятия решения — необходимо понимать, где возникает главный

Пример:

Концепция сайта

Интернет-
магазин

Концептуальные
инструменты

Навигационная
концепция

Точки принятия
решений

Каталог товаров
Страницы
описаний
Сравнение
выбора

Мобильная-
гамбургер
Горизонтальная
дополнительная

Главная
Страница товара
Корзина

Задача ИА — найти в массиве контента общие признаки и сформировать группы информационных объектов, с которыми можно продолжать работу.

Вопросы, которые надо решить

- Применять маркировку: классифицировать информацию (алфавитном, хронологическом, географическом). Но более важна категоризация контента на основе потребностей пользователя.
- Задача информационного архитектора — опираясь на понимание конечных потребностей пользователей, особенности шаблонов восприятия и конкурентное окружение сформировать группы информационных объектов максимально понятным для пользователя способом.
- Хрестоматийный пример: арбуз и помидор — ягоды только с точки зрения ботаники. Клубника — это орех, но в магазине вы не пойдёте искать её в отдел с орехами. Продукты попадают в ту категорию, которая привычнее покупателям. Наши представления и реальность могут сильно отличаться.

Определение понятия «Информационная архитектура»

- Сочетание схем организации, предметизации и навигации, реализованных в информационной системе.
- Проектирование информационного пространства, способствующее выполнению задач и интуитивному доступу к содержимому.
- Развивающаяся дисциплина и сообщество практиков, ставящее своей задачей распространение принципов проектирования и архитектуры на цифровых пространствах.

Информация в области ИА

- **информационная структура**
 - базы данных (data);
 - базы знаний (knowledge);
- информация в различных формах ее представления – сайты, документы, программы, графика...
- **метаданные** - информация о информации-объекты содержимого - люди, процедуры, организации...

Проблемы, связанные с информацией

1. **Неоднозначность**

Любому языку свойственна неоднозначность. Сколько, например, значений слова «лук» вы знаете? А слова «такса»?

Не говоря уж о профессиональном жаргоне. Как неподготовленному человеку понять, что под услугой «перевозка» скрывается транспортировка груза лишь от склада до склада, а не от двери до двери?

Проблемы, связанные с информацией

2. Гетерогенность

гомогенность - однородность

гетерогенность – это свойство объекта или группы объектов, составленных из несвязных или несхожих частей.

Проблемы, связанные с информацией

3. Различие точек зрения

У владельца ресурса есть четкое представление о том, как функционирует его бизнес. Какие в компании существуют подразделения, какая у нее организационная структура. И зачастую он стремится организовать информацию на сайте в соответствии с этими своими представлениями.

Вот только посетитель ничего не знает о том, как устроена компания.

Приходя на сайт, он имеет в виду свою конкретную цель, например, найти технические характеристики продукта. И как ему узнать в компетенции какого подразделения находится составление этих характеристик?

Структурирование, организация, предметизация

- **Структурирование** – выбор степени детализации для информационных атомов сайта и установление отношений между ними;
- **Организация** – объединение этих компонентов в отдельные смысловые категории и темы;
- **Предметизация** – именованье этих категорий и ведущих к ним навигационных ссылок.

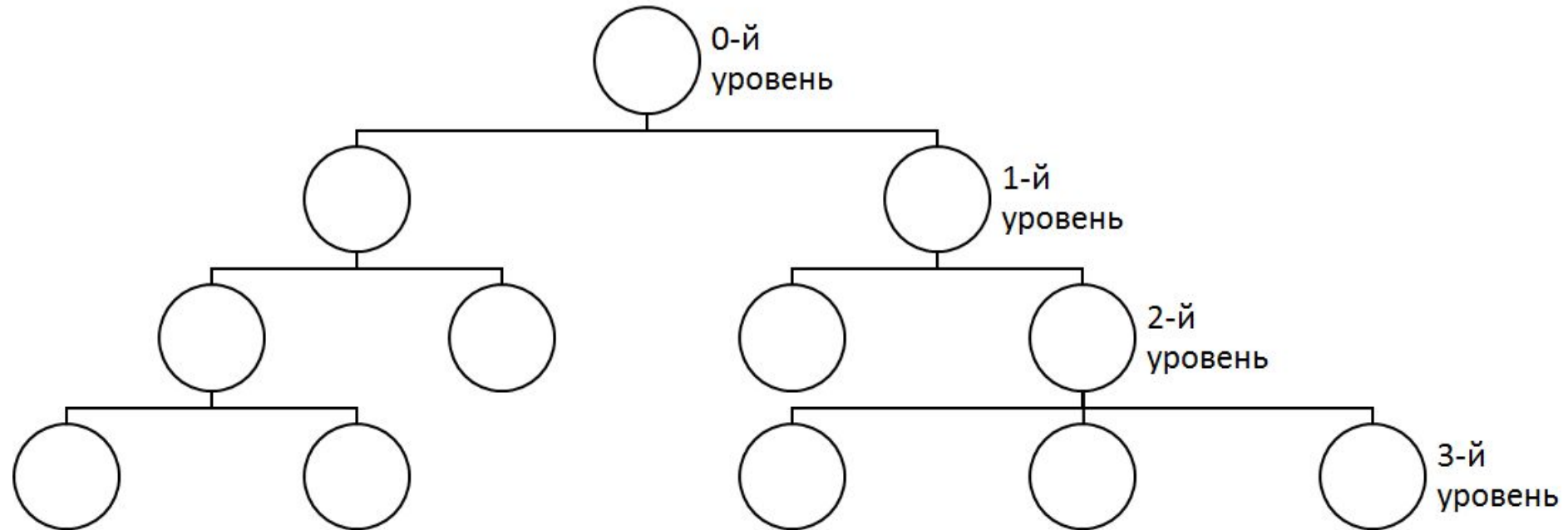
Системы классификации информации

1. иерархическая



2. многоаспектная

Иерархическая



множество объектов последовательно разбивается на подчиненные подмножества или классификационные группировки. На каждом уровне классификации разделение группы объектов на подмножества происходит по одному из признаков.

Особенность - любой объект на каждом уровне классификации может быть отнесен только к одному подмножеству объектов. Создаются жесткие классификационные группировки, обусловленные заранее установленным выбором признаков классификации и порядком их использования по ступеням классификации

Схемы организации, основанные на иерархической системе классификации

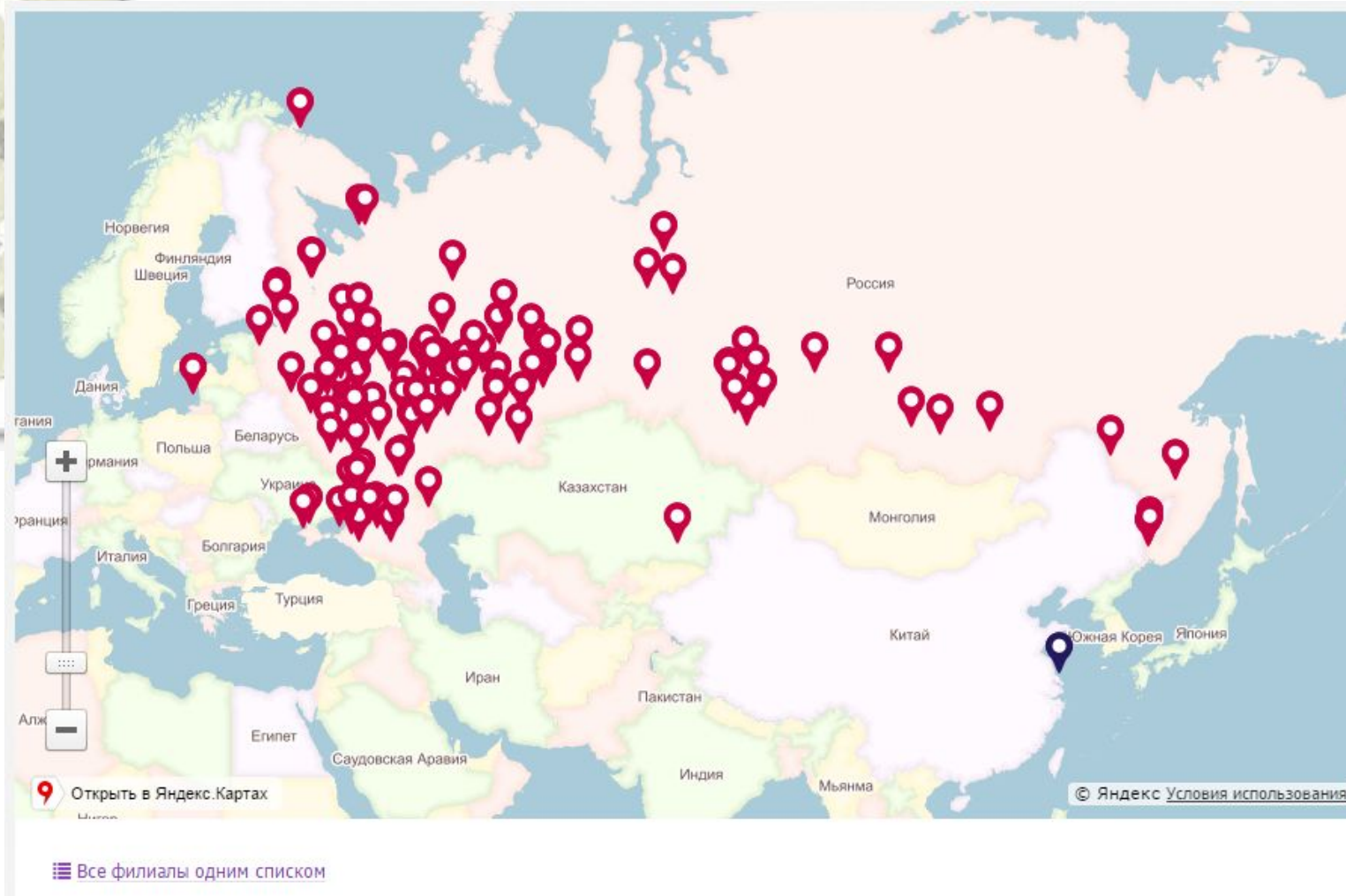
Алфавитная организация

Хронологическая организация

Предполагает организацию информации по дате публикации. Применяется тогда, когда дата публикации составляет важный аспект информации. Пресс-релизы, новости, ленты блогов – все это легко упорядочивается в хронологическом порядке.

Географическая организация

Часто важной характеристикой информации является географическое расположение. Классификация контактной информации по стране, региону, городу, станции метро обеспечит легкий и наглядный доступ. Наглядно представить географию на сайте можно как с помощью карты, так и в виде алфавитного списка, совмещаю алфавитную и географическую организации.



Тематическая организация

Предполагает разделение информации по принципу принадлежности к отдельным темам. Недостатком этого метода является его неоднозначность, поскольку разные люди могут относить один и тот же объект к разным темам.

02.00.00	Философия
03.00.00	История. Исторические науки
04.00.00	Социология
05.00.00	Демография
06.00.00	Экономика. Экономические науки
10.00.00	Государство и право. Юридические науки
11.00.00	Политика. Политические науки
12.00.00	Науковедение
13.00.00	Культура. Культурология
14.00.00	Народное образование. Педагогика
15.00.00	Психология
16.00.00	Языкознание
17.00.00	Литература. Литературоведение. Устное народное творчество
18.00.00	Искусство. Искусствоведение
19.00.00	Массовая коммуникация. Журналистика. Средства массовой информации
20.00.00	Информатика
21.00.00	Религия. Атеизм

Одежда и обувь



Дом и сад



Строительство



Аксессуары и украшения



Материалы для ремонта



Техника и электроника



Товары для детей



Организация по задачам

Если можно выделить некоторые функции или задачи, которые имеют наиболее высокий приоритет для пользователей, то информацию можно организовать по совокупности процессов, функций или задач.

В Интернет организация по задачам чаще всего встречается на сайтах банковских, страховых организаций, в системах онлайн-банкинга.

Организация по аудитории

Когда аудиторию ресурса можно четко разделить хотя бы на две группы, информация, предназначенная для этих групп совершенно разная, и можно быть уверенным в том, что любой посетитель сайта может однозначно отнести себя к той или иной группе, можно применить организацию по аудитории.

**пользователь
пользователь**

администратор

зарегистрированный

[Курсы валют](#) [Курсы валют по карточкам](#) [Драгоценные металлы](#) [Аттестованные бриллианты](#) [Памятные монеты](#) [Интернет-витрина](#)

[О БАНКЕ](#)

[ФИЗИЧЕСКИМ ЛИЦАМ](#)

[КОРПОРАТИВНЫМ КЛИЕНТАМ](#)

[ФИНАНСОВЫМ ИНСТИТУТАМ](#)

[СОТРУДНИКАМ](#)

[СТУДЕНТАМ](#)

[ВЫПУСКНИКАМ](#)

[ПОСТУПАЮЩИМ](#)

[УСЛУГИ](#)



БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНФОРМАТИКИ
И РАДИОЭЛЕКТРОНИКИ

[ENG](#) [RUS](#) [BEL](#) [ARA](#) [VIE](#) [KAZ](#) [UZB](#) [CHI](#) [AZE](#) [TR](#)



Поиск



4 учебная неделя



[УНИВЕРСИТЕТ](#)

[ОБРАЗОВАНИЕ](#)

[НАУКА](#)

[МОЛОДЁЖНАЯ
ПОЛИТИКА](#)

[МЕЖДУНАРОДНОЕ
СОТРУДНИЧЕСТВО](#)

[КОНТАКТЫ](#)

Метафоры

Применяются для того, чтобы объяснить пользователям нечто новое с помощью уже знакомых понятий. Ярким примером использования метафор в Интернет является корзина в

e-commerce.

Отличительной особенностью метафор является их свойство устаревать. Сегодняшние дети уже не воспринимают иконку «Сохранить» как изображение трехдюймовой дискеты.

Material design от Google – современный пример применения метафор.



Гибридные схемы организации информации


Строгие схемы организации информации обладают одним важным преимуществом – они легко воспринимаются пользователями. Легче всего пользователи воспринимают **классификацию по аудитории и по тематике**.

в реальной ситуации проектирования информация из-за своей разнородности редко укладывается в любую строгую схему организации. На практике чаще всего используются **гибридные схемы**.

Гибридная схема может строиться по-разному. Возможен вариант, когда на верхнем уровне используется одна схема, например, тематическая, а на следующем – другая, например, алфавитная, причем в разных разделах могут использоваться разные схемы.

Также могут использоваться **две и более строгих схем** в совокупности на одном уровне классификации.

Частным клиентам Малому бизнесу Корпоративным клиентам Финансовым организациям Акционерам и инвесторам Партнерам

 **СБЕРБАНК**
Организация по аудитории
Всегда рядом

900 8 (800) 555 55 50 +7 (495) 500-55-50

Отделения и банкоматы Пермский край

Поиск

Регистрация Вход

Взять кредит Выбрать карту Накопить и сохранить Оплатить и перевести Вложить и заработать Застраховать себя и имущество

Организация по задачам

Финансы ПРОСТО

Ваше мнение

Многоаспектные системы классификации

Аспект – точка зрения на объект классификации, который характеризуется одним или несколькими признаками.

Многоаспектная система классификации использует для характеристики объекта несколько независимых признаков (аспектов) в качестве основания классификации.

два типа многоаспектных систем

фасетная и дескрипторная.

Фасет — это аспект классификации, который используется для образования независимых классификационных группировок.

Дескриптор — ключевое слово, определяющее некоторое понятие, которое формирует описание объекта и дает принадлежность этого объекта к классу, группе и т.д.

Фасетная классификация

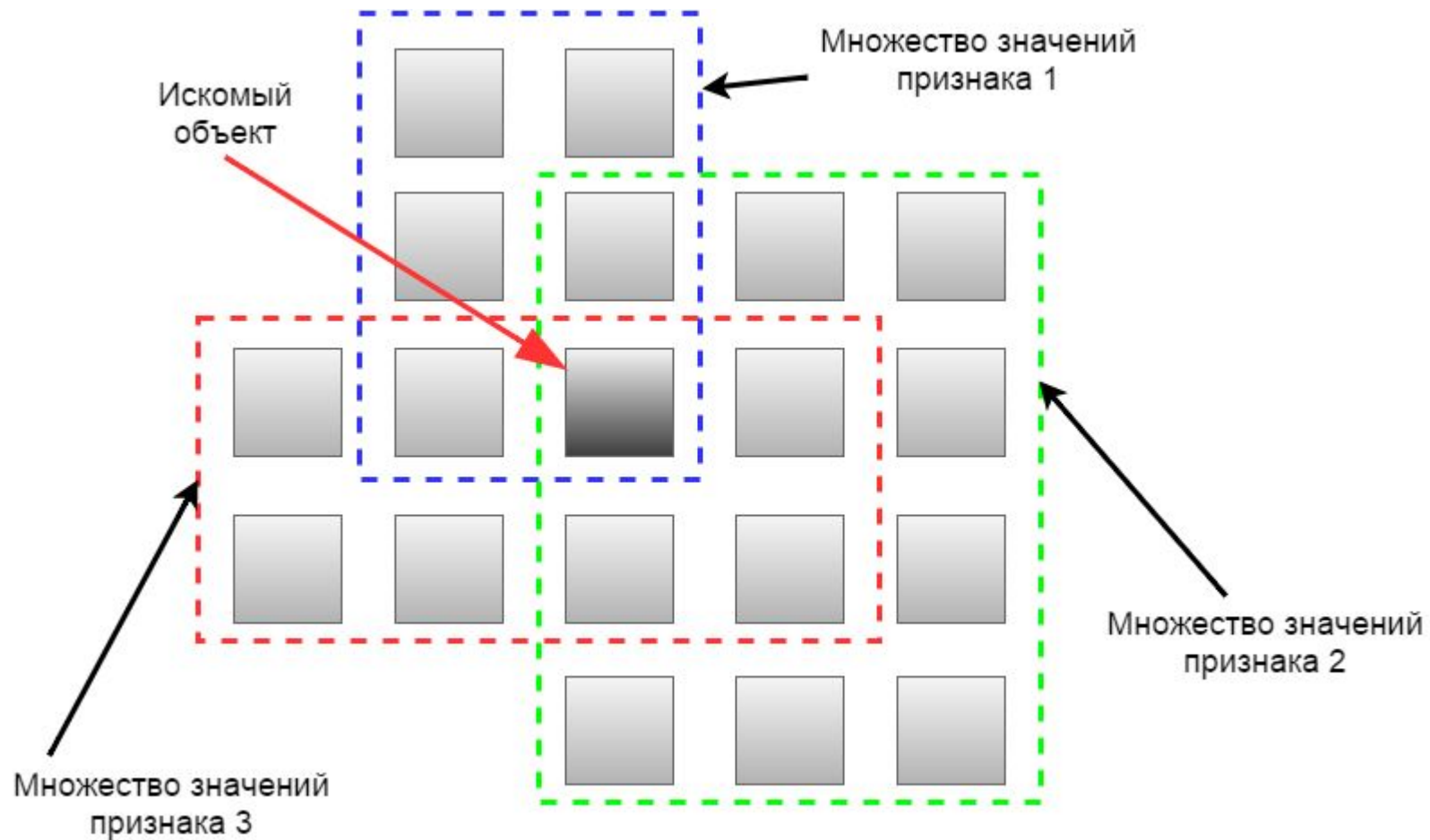
разрабатывается система таблиц признаков классификации, которые называются фасетами. Для идентификации объекта осуществляется выборка признаков из фасетов и их объединение в определенной последовательности.

Наименование признака 1	Наименование признака 2	Наименование признака 3	Наименование признака 4	Наименование признака 5
Значение 1	Значение 1	Значение 1	Значение 1	Значение 1
Значение 2	Значение 2	Значение 2	Значение 2	Значение 2
Значение 3	Значение 3	Значение 3	Значение 3	Значение 3
Значение 4	Значение 4	Значение 4	Значение 4	Значение 4

Фасеты

Значения фасетов

Тип	Жанр	Страна	Премия
Документальный	Боевик	Россия	«Оскар»
Игровой	Комедия	США	«Золотая пальмовая ветвь»
Анимация	Мелодрама	Франция	«Золотой лев»



Разные люди ищут по разным признакам.
Например – обувь – мужская/женская/страна
производитель/сезон/материал/размер/цена

Фото и видеокамеры

ФОТОАППАРАТЫ

- > Фотоаппараты мгновенной печати
- > Фотоаппараты беззеркальные
- > Фотоаппараты зеркальные
- > Фотоаппараты компактные
- > Фотоаппараты ультразумы
- > Все фотоаппараты

Напишите нам, мы онлайн!

ФОТОАКСЕССУАРЫ



Фотоаппараты



Фотоаксессуары



значения
ФАСЕТов

ФАС
ЕТ

фамилия

имя

М/
Ж

семейное

положение
прописка

студент

номер

зачетки

специальность

М/
Ж

спор

т

права на вождение

автомобиль

музыка

б

номер

группы
ср.

баллы

блондинка

факультет

курс

с

знание

английского
проф.

компетенции

Преимущества фасетной навигации :

Не бывает пустых результатов поиска. Поскольку множество значений всех признаков определяется по существующим объектам, указание значения хотя бы одного признака имеет результатом как минимум один объект.

Указывается величина. По каждому значению любого признака можно указать количество объектов, которые обладают данным значением признака.

Множество путей поиска. Пользователи могут выбирать значения разных признаков в любом порядке.

Выбор значений может быть отменен. Поскольку каждый признак имеет множество значений, пользователь в любой момент может построить новую комбинацию значений.

Дескрипторная классификация

используется для поиска информации по не структурируемому данным, например, документам. Для осуществления дескрипторной классификации составляется словарь – **тезаурус** – ключевых слов, включая все их родовидовые отношения, отношения синонимии, омонимии и полисемии, а затем все документы индексируются определенным набором ключевых слов.

Дескрипторная система классификации

- Используется для организации поиска информации, для ведения тезаурусов (словарей).
- Язык дескрипторной системы приближен к естественному.**
 - Отбирается множество ключевых слов и словосочетаний, описывающих определенную предметную область.
- Создается словарь ключевых слов и словосочетаний – словарь дескрипторов.** Между дескрипторами устанавливаются **связи**
 - **синонимические** (синонимы),
 - родовидовые** (включение в более общий класс)
 - **ассоциативные** (связанные с помощью какого-то свойства).
- Используются в поисковых системах (*библиотечной, Интернет-системах – Yandex, Rambler и др.*).

Пример дескрипторной классификации

- **ЦЕЛОЕ - ЧАСТЬ:**
 - *компания – отдел – (рабочая)_ группа – работник;*
 - *баланс – раздел – группа_(статей), – статья;*
- **РОД - ВИД:**
 - *металл - черный_(металл), цветной_(металл);*
 - *черный_(металл) – сталь, чугун;*
 - *сталь – холоднокатаная_(сталь), горячекатаная_(сталь);*
- **ДОПОЛНЕНИЕ:**
 - *отчетность – бухгалтерская, статистическая, налоговая, внутренняя,*
 - *ресурсы – материальные, трудовые, финансовые, информационные;*
- **ПРОТИВОПОСТАВЛЕНИЕ:**
 - *актив – пассив,*
 - *приход – расход,*
 - *прибыль – убытки,*
 - *дебит – кредит;*
- **ОБЪЕКТ - ДЕЙСТВИЕ:**
 - *материалы - расход,*
 - *материалы – приход,*
 - *материалы – остаток,*
- **ДЕЙСТВИЕ - ВРЕМЯ:**
 - *расход – за месяц, расход – за квартал, расход – за год, ...*
- **ДЕЙСТВИЕ – МЕСТО:**
 - *приход – склад, расход – склад, остаток – склад, расход – цех, ...*

Составим примеры

Вид дескрипторной классификации		
Целое-часть	Портал	Сайт/страниц
Род-вид	Блог	Личный/корпоративный
Дополнение	Информационный ресурс	Сайт/газета/журнал

В Интернет дескрипторы реализуются в виде облака тегов. Количество записей по каждому тегу может указываться как числовым значением, так и размером шрифта, используемого для написания того или иного тега.

Рубрикатор

АВТО и Мото **Бизнес** Бизнес Планы Все для офиса Деловые
услуги Реклама Финансы **Дом, семья, дети** Бытовая техника Мебель Сад и
огород Досуг и Юмор Еда Рецепты **Здоровье** Женское
здоровье Препараты Искусство Кино Литература Музыка Фото **Компьютеры и**
Интернет hardware Интернет программное обеспечение Мода,
стиль Наука Образование Общество ЖЗЛ История Милитари Окружающий
Мир Домашние животные, растения, грибы **Промышленное**
оборудование СВЯЗь Оборудование Услуги Секс Спорт Справочная
информация **Страны, путешествия** Россия и ближнее
зарубежье **Строительство** Дизайн Материалы Оборудование Снабжение Хобби

Принципы информационной архитектуры

Принцип выбора:

Надо предлагать пользователям осмысленный выбор.

Нельзя давать слишком много вариантов – это может привести к дезориентации пользователя
выбор должен быть сосредоточен на чем-то конкретном

Информация подается в виде иерархии, категорий и суб-категорий, вместо того, чтобы приводить ее просто длинным списком.

Принципы информационной архитектуры

Принцип объектов

рассматриваем контент как развивающуюся сущность, которая имеет собственный жизненный цикл. Разный контент будет иметь разные атрибуты и поведение, и это нужно учесть при проектировании дизайна.

Текст

Объем
Ширина, высота
Координаты левого верхнего угла
Отступы (осветленное пространство)
поддерживаемость, связность и
цельность, логичность и
лаконичность

Атрибуты и поведение?

Картина маслом

ЯРКОСТЬ
контрастность
НАСЫЩЕННОСТЬ
Б
Тип – растр,
вектор

Принципы информационной архитектуры

Принцип раскрытия. Важно дать пользователю необходимую ему информацию. Однако стоит убедиться, что это действительно то, что ему нужно, а не то, что **ВАМ** захотелось дать. Принцип говорит также о том, что нужно сразу давать пользователю информацию, необходимую для понимания: что он сможет найти на других страницах сайта, а что нет. Информацию нужно подавать постепенно, от страницы к странице, а не пытаться вывалить все и сразу.

Принцип примеров. Использование принципа существенно улучшает пользовательский опыт. Например, зайдя в категорию товара «Электрочайник», выдается 4-5 разных моделей. Это помогает пользователю быстрее сориентироваться, особенно, если он не до конца понимает, что значит название категории.

Принципы информационной архитектуры

- **Принцип парадного входа.** Большинство посетителей попадают на сайт не через главную страницу. Это значит, что любая страница должна содержать необходимый минимум текстовой информации — чтобы пользователи поняли, где они находятся.
- **Принцип множественной классификации.** Этот принцип говорит о том, что разные пользователи используют сайт по-разному, у них могут быть разные методы для нахождения одной и той же информации. Например, одни будут пользоваться поиском, другие предпочтут поблуждать по сайту. Контент нужно адаптировать различным сценариям пользовательского поведения.

Принципы информационной архитектуры

Принцип целенаправленной навигации. Не так важно, где находится меню, важно то, что на нем написано.

Постарайтесь, чтобы ваше меню и панель навигации показывали, где находится пользователь сейчас и куда он может попасть с текущей страницы.

Принцип роста. На подавляющем большинстве сайтов контент — текучая, изменчивая сущность. Количество контента с на сайте сегодня может быть лишь малой долей того, чтобы там может быть завтра. Организуйте контент таким образом, чтобы позволять ему расти в будущем. Причем не только в плане расширения какого-то блока с текстом: контент может добавляться совершенно разных типов.

Основным преимуществом реляционных СУБД является возможность связывания на основе определенных соотношений файлов БД. Со структурной точки зрения реляционные модели являются более простыми и однородными, чем древовидные и сетевые. В реляционной модели каждому объекту предметной области соответствует одно или более соотношений.

Можно выделить несколько типов СУБД, позволяющими управлять большими информационными массивами.

- простейшие СУБД, которые позволяют обрабатывать один массив информации, они обеспечивают ввод, поиск, сортировку, составление отчетов и т.д. , действия в них осуществляются при помощи меню и др. диалоговых средств (PC-File, Reflex.,Q&A)

- более сложные, которые поддерживают и обрабатывают несколько массивов информации, описывающих разнотипные объекты, и связи между ними, они как правило содержат средства программирования (Lotus Approach, Paradox, а для разработки сложных информационных систем Microsoft Access, Fox Pro и др.)

- для создания многопользовательских информационных систем больше всего подходят СУБД типа клиент-сервер, где база располагается на мощном компьютере – сервере, который принимает запросы на получение некоторой информации или ее обработки от программ клиентов, выполняемых на других компьютерах.

Моделирование потоков данных (процессов)

Модель представляет собой иерархию диаграмм потоков данных (ДПД или DFD), описывающих процесс преобразования данных от ввода в систему до выдачи результата пользователю. Контекстные диаграммы (т.е. диаграммы верхних уровней) определяют основные процессы или подсистемы с внешними входами и выходами. Эти диаграммы детализируются при помощи диаграмм нижних уровней. Детализация продолжается до тех пор пока процесс не станет элементарным.

Источники информации или внешние сущности порождают потоки информации, переносящие ее к подсистемам или процессам, которые эту информацию преобразуют и порождают новые потоки, переносящие информацию к другим подсистемам или процессам, накопителям данных или внешним сущностям – потребителям информации.

Моделирование данных « Модель «сущность-связь»

Рассматриваемые вопросы:

- **Элементы модели «сущность-связь»**
- **Диаграммы «сущность-связь»**
- **Слабые сущности**
- **Подтипы сущностей**
- **Пример ER-диаграммы**
- **Диаграммы «сущность-связь» а стиле UML**

Элементы модели «сущность-связь»

Сущность

- Класс сущностей
- Экземпляр сущности

Атрибуты

- Композитные атрибуты
- Многозначные атрибуты

Идентификаторы

- Уникальные/неуникальные
- Композитные

Связи

- Классы связей
- Экземпляры связей
- Рекурсивные связи

Элементы модели «сущность-связь»

Сущность

Сущность (*entity*) – это некоторый объект, идентифицируемый в рабочей среде пользователя, нечто такое, за чем пользователь хотел бы наблюдать.

Обозначение средствами в UML- диаграммах:

Сущность обозначается



Элементы модели «сущность-связь»

Класс сущностей (entity classes) – это совокупность сущностей, описывается структурой или форматом сущностей, составляющих этот класс.

Экземпляр сущности (an instance) представляет конкретную сущность

Обычно класс сущностей держит множество экземпляров сущности.

Элементы модели «сущность-связь»

Пример сущности СТУДЕНТ

СТУДЕНТ

Сущность содержит:

Группа студента

Номер студента

ФИО

Год рождения

Средний бал

Два экземпляра сущности **СТУДЕНТ**

101

03

Бондаренко С.А.

1987

4,5

102

05

Петров О.К.

1986

4,0

Элементы модели «сущность-связь»

Атрибуты

Атрибуты (*свойства*) – описывают характеристики сущности.

Пример композитного атрибута: Адрес, состоящий из группы атрибутов {Улица, Город, Индекс}.

Пример многозначного атрибута: атрибут Имя студента сущности ПРЕПОДАВАТЕЛЬ, который может содержать имена нескольких обучаемых им студентов.

Элементы модели «сущность-связь»

Идентификаторы

Идентификаторы (identifiers) – атрибуты, с помощью которых экземпляры сущностей именуются, или идентифицируются.

Если идентификатор является **уникальным**, его значение будет указывать на один и только один экземпляр сущности.

Если идентификатор является **неуникальным**, его значение будет указывать на некоторое множество экземпляров.

Идентификаторы, состоящие из нескольких атрибутов, называются **композитными идентификаторами** (composite identifiers).

Элементы модели «сущность-связь»

СВЯЗИ

*Взаимоотношения сущностей выражаются **СВЯЗЯМИ**.*

Классы связей (*relationship classes*) — это взаимоотношения между классами сущностей.

Экземпляры связи (*relationship instances*) — взаимоотношения между экземплярами сущностей

Степень связи (*relationship degree*) — число классов сущностей, участвующих в связи.

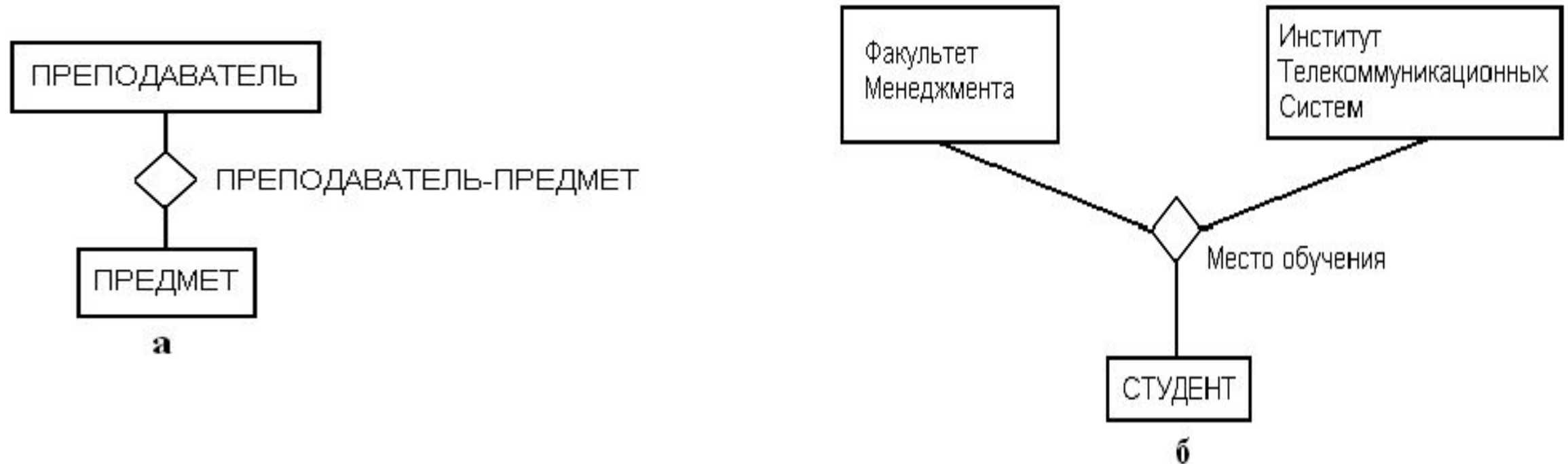
Обозначение средствами в UML-диаграммах:

Связь обозначается



Элементы модели «сущность-связь»

Примеры различных степеней связи: **а – связь степени 2, б – связь степени 3.**




Связи степени 2 весьма распространены, их часто называют еще **бинарными связями** (*binary relationships*).

Элементы модели «СУЩНОСТЬ-СВЯЗЬ»


Три типа бинарных связей

Обозначение средствами в UML- диаграммах:

Связь $1:1$ («один к одному») обозначается 

Связь $1:N$ («один к N» или «один ко многим») – 

Связь $N:M$ (читается «N к M» или «многие ко многим») – 

Связь обладания в обобщенном виде, когда не указан конкретный тип связи · 

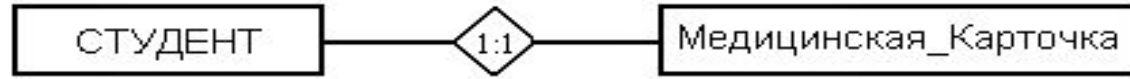
Числа внутри ромба, символизирующего связь, обозначают максимальное количество сущностей на каждой стороне связи. Эти *ограничения* называются **максимальными кардинальными числами**, а совокупность из двух таких ограничений для обеих сторон связи называется **максимальной кардинальностью** (*maximum cardinality*) связи.

ЭЛЕМЕНТЫ МОДЕЛИ «СУЩНОСТЬ-СВЯЗЬ»

Пример бинарных связей: а – бинарная связь 1:1,

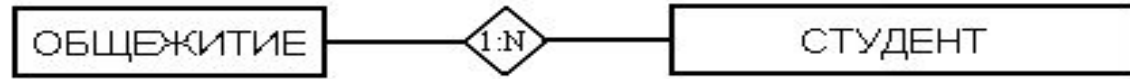
б – бинарная связь 1:N, в – бинарная связь N:M,

г – представление связи с помощью разветвлений.



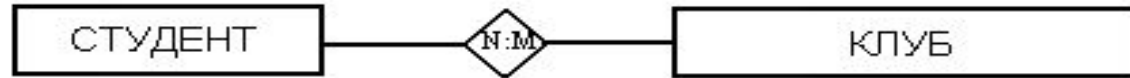
СТУДЕНТ-Медицинская_Карточка

а



ОБЩЕЖИТИЕ-СТУДЕНТ

б



СТУДЕНТ-КЛУБ

в



г

Диаграммы «сущность-связь»

Схемы бинарных связей, изображенных выше, называются диаграммами «сущность-связь», или **ER-диаграммами** (*entity-relationship diagrams, ER-diagrams*).

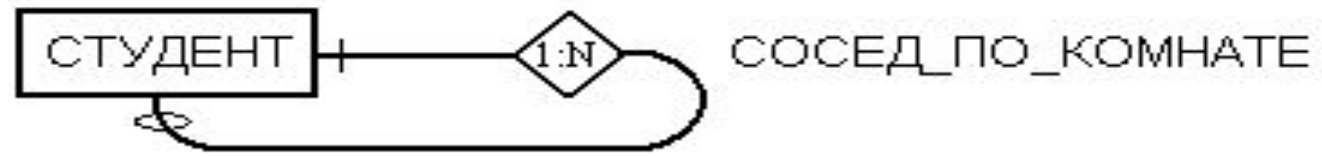
Для указания **минимальной кардинальности** (*minimum cardinality*) существует несколько способов. Один из них, продемонстрирован ниже.

Связь с указанной минимальной кардинальностью



Диаграммы «сущность-связь»

Связи между сущностями одного и того же класса называются иногда **рекурсивными связями** (*recursive relationships*).



а

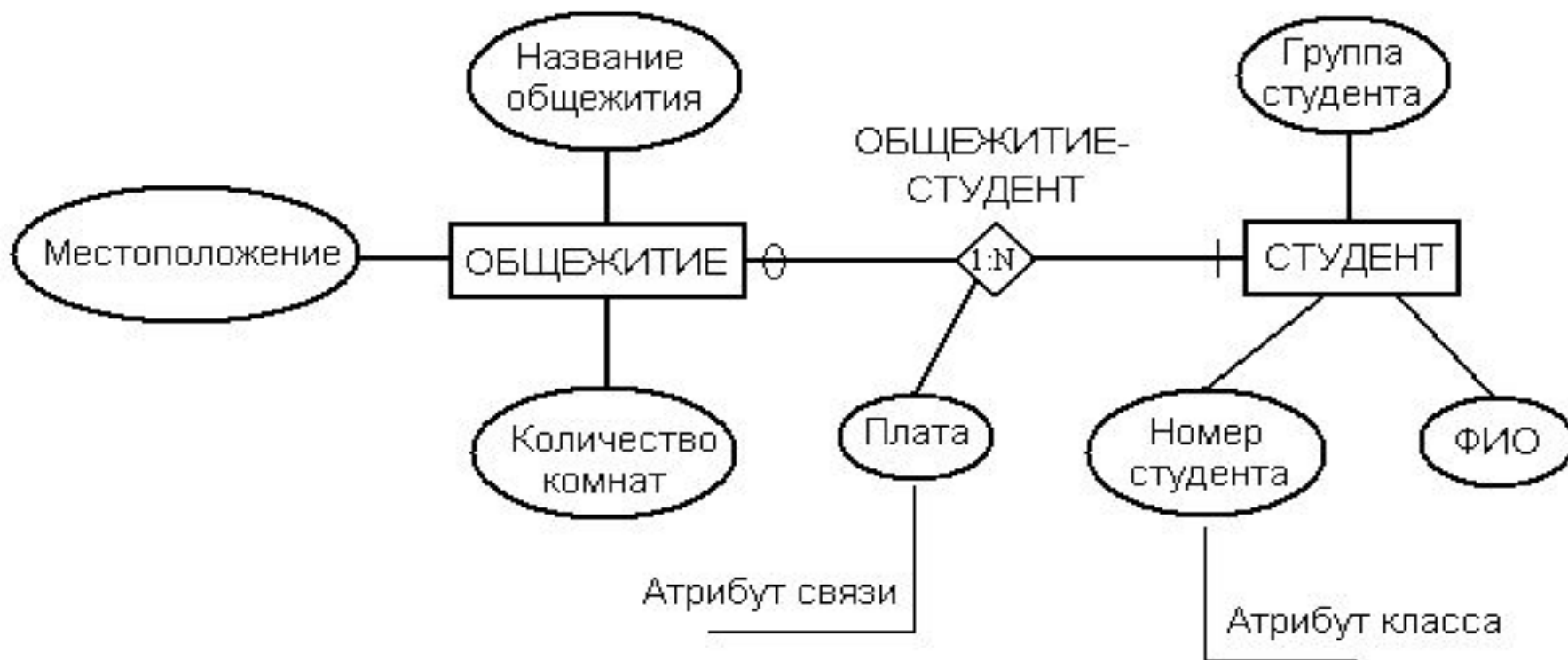


б

Диаграммы «сущность-связь»

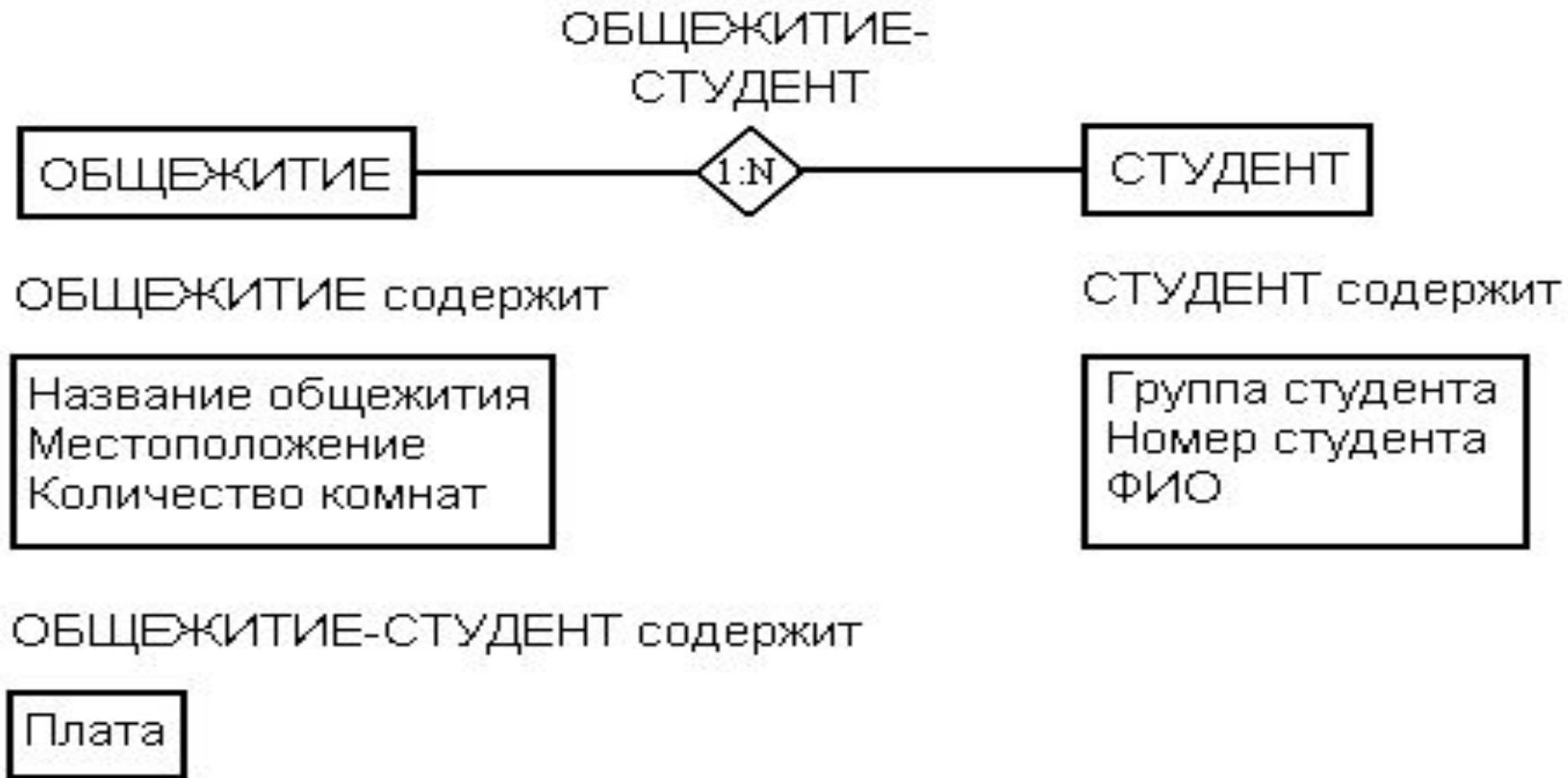
Изображение атрибутов в диаграммах «сущность-связь»

В некоторых версиях ER-диаграмм *атрибуты* обозначаются эллипсами, соединенными с сущностью или связью, которой они принадлежат.



а

диаграммы «сущность-связь»



б

Изображение свойств на диаграммах «сущность-связь»:
а – указание на диаграмме; **б** – отдельное перечисление.

Слабые сущности

Слабые сущности (*weak entity*) - сущности, которые могут существовать в базе данных только в том случае, если в ней присутствует сущность некоторого другого типа.

Сущность, не являющаяся слабой, называется **сильной сущностью** (*strong entity*).

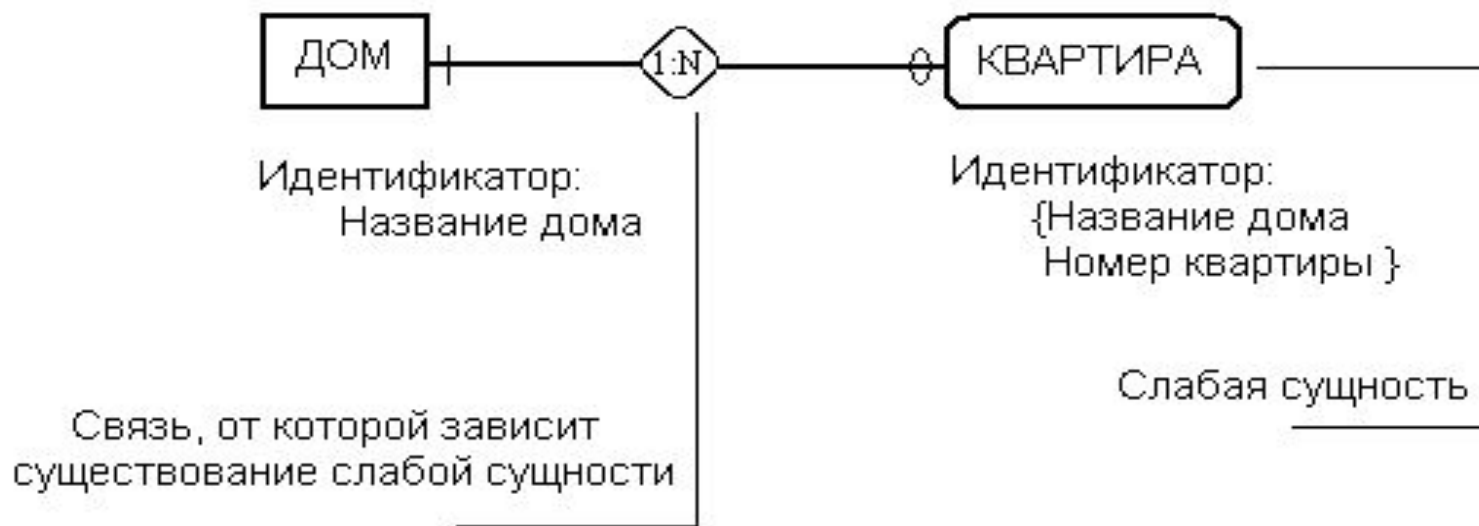
Идентификационно-зависимые сущности (*ID-dependent entities*) - это такие сущности, идентификаторы которых содержат идентификатор другой сущности.

Слабые сущности

Слабые сущности: а – пример слабой сущности,
б – пример идентификационно-зависимой сущности.



а

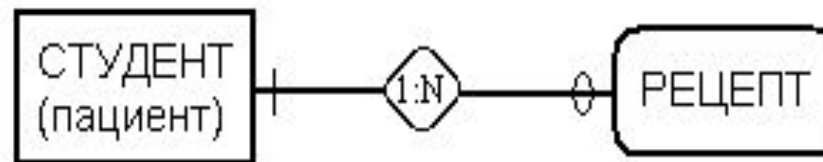


б

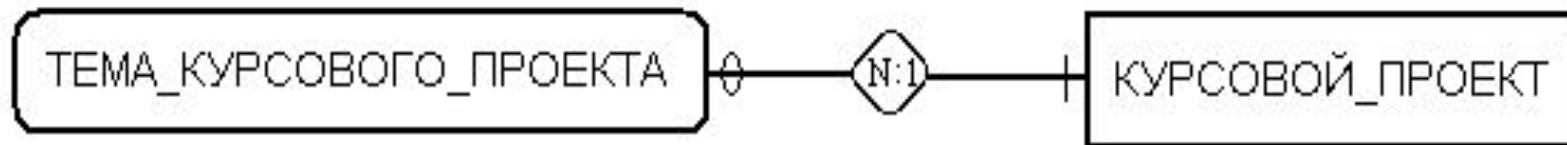
Слабые сущности

Чтобы сущность можно было отнести к разряду **слабых**, она должна логически зависеть от другой сущности.

Пример обязательных сущностей



а



Идентификатор:
{Название курсового проекта,
Название задачи}

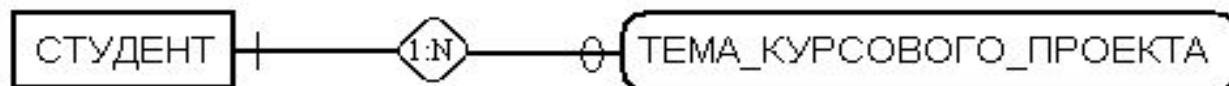
Идентификатор:
Название курсового проекта

б

Слабые сущности

Многозначные атрибуты представляются в модели «сущность-связь» путем создания новой слабой сущности и построения связи вида «один ко многим».

Представление многозначных атрибутов с помощью слабых сущностей



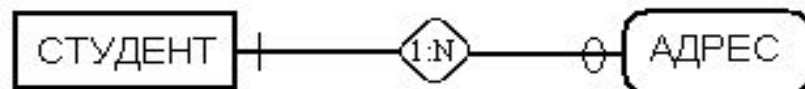
СТУДЕНТ содержит:

Группа студента
Номер студента
и прочие атрибуты

ТЕМА_КУРСОВОГО_ПРОЕКТА содержит:

Тема курсового проекта

а



СТУДЕНТ содержит:

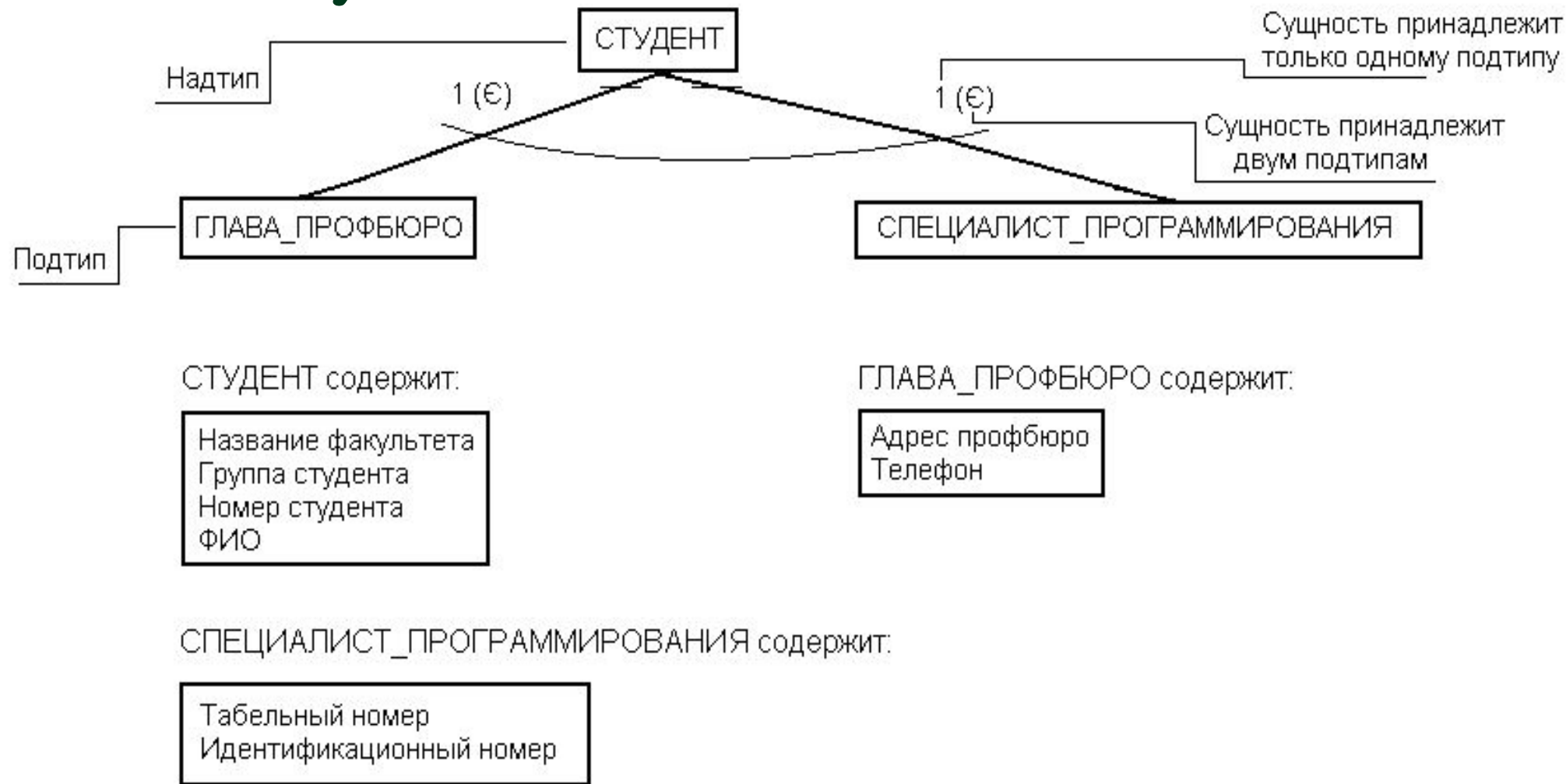
Группа студента
Номер студента
и прочие атрибуты

АДРЕС содержит:

Улица
Город
Индекс

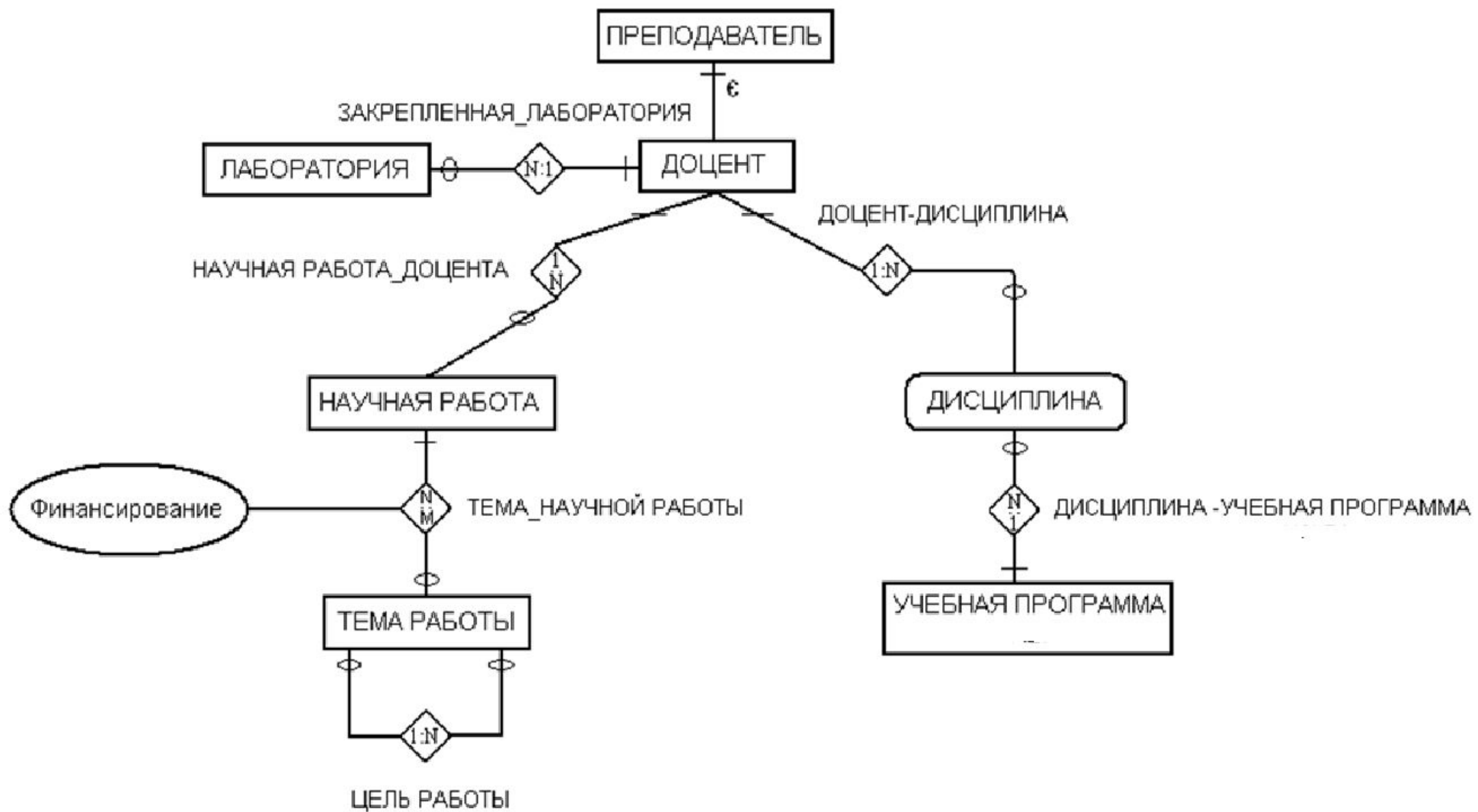
б

Подтипы сущностей



Иерархии генерализации имеют специальную характеристику, называемую **наследованием** (*inheritance*), которая означает, что *подтипы классов сущностей наследуют атрибуты от надтипа*.

Пример ER-диаграммы



Диаграммы «сущность-связь» в стиле UML

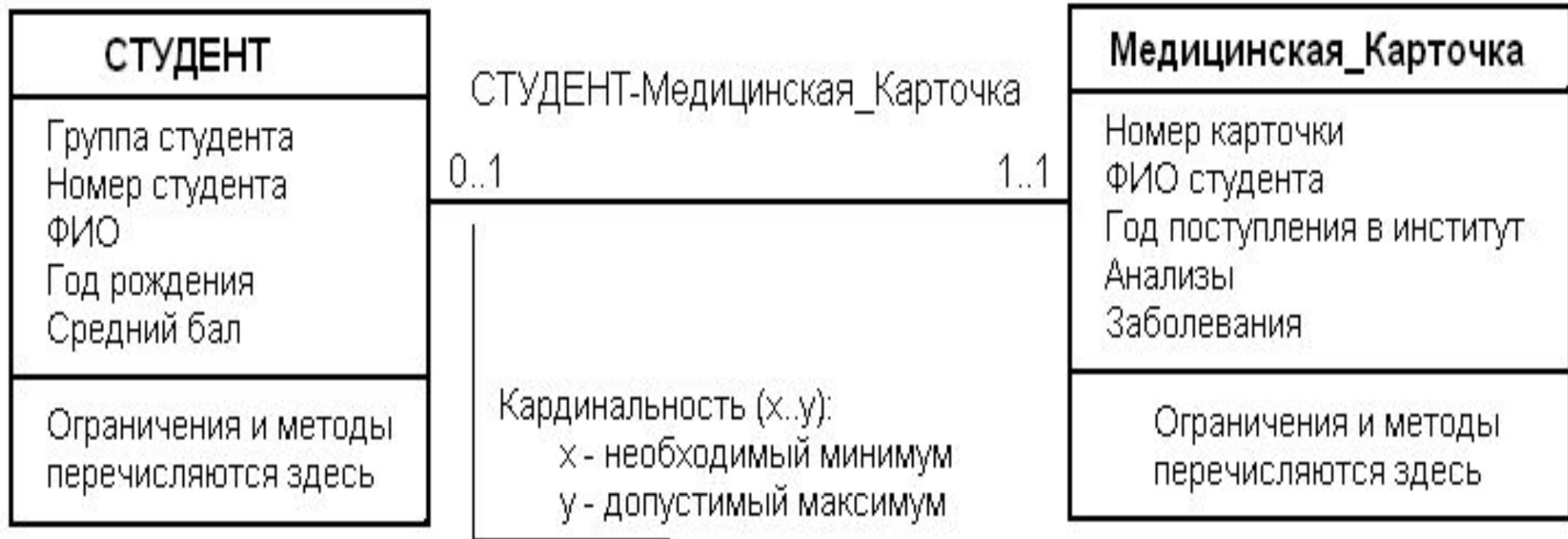
Унифицированный язык моделирования
(UML, Unified Model Language) - это набор структур и методик для моделирования и проектирования объектно-ориентированных программ (ООП) и приложений.

Диаграммы «сущность-связь» в стиле UML

Сущности и связи в UML

Представления различных типов связей в UML:

а – связь 1:1, **б** – связь 1:N, **в** – связь N:M.

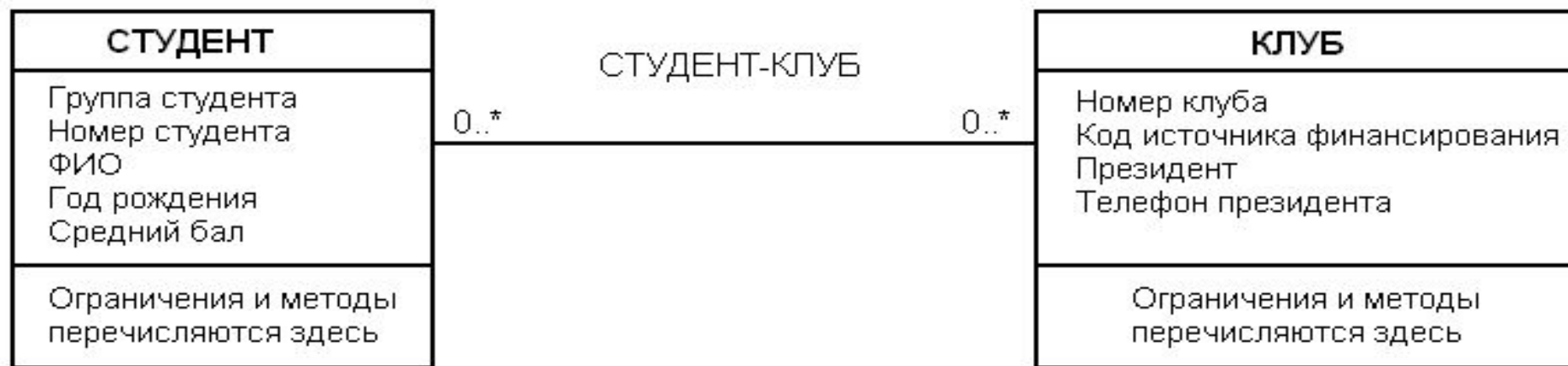


а

Диаграммы «сущность-связь» в стиле UML



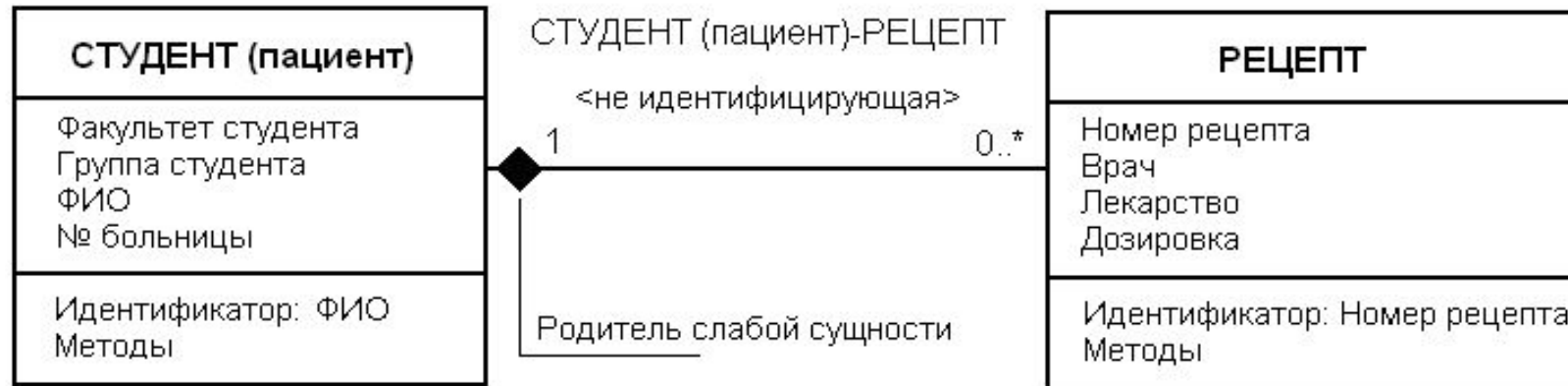
б



в

Диаграммы «сущность-связь» в стиле UML

Представление слабых сущностей



а



б

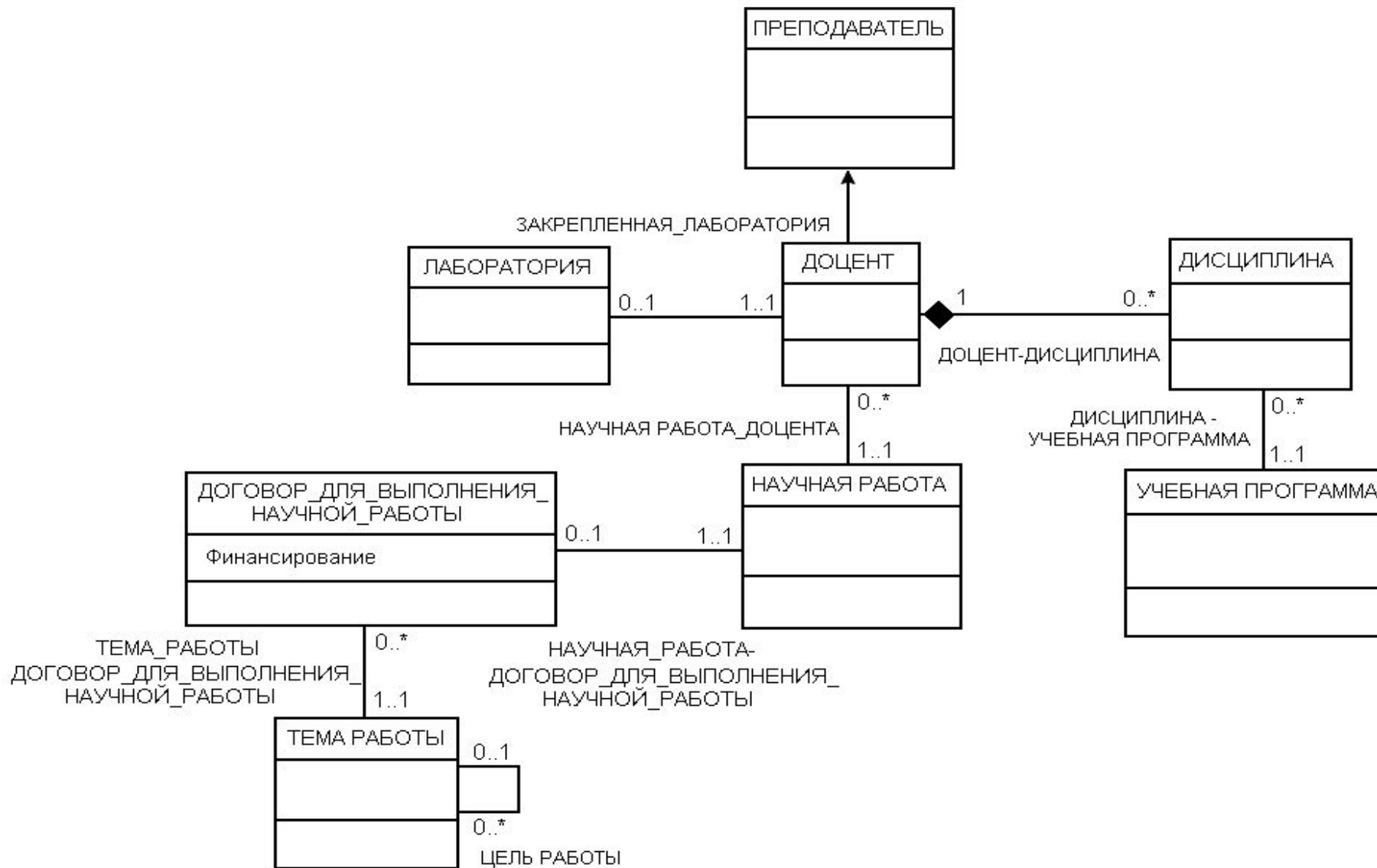
Диаграммы «сущность-связь» в стиле UML

Представление подтипов



Диаграммы «сущность-связь» в стиле UML

UML-версия диаграммы «сущность-связь»



Диаграммы «сущность-связь» в стиле UML

Конструкции ООП, введенные языком UML

- Классы всех сущностей, которые должны храниться в базе данных, помечаются стереотипом «Persistent» (устойчивый)
- UML допускает назначение атрибутов классам сущностей
- UML использует объектно-ориентированную нотацию для обозначения видимости атрибутов и методов
 - «+» - открытые
 - «#» - защищенными
 - «-» - закрытыми

Диаграммы «сущность-связь» в стиле UML

Открытым (public) называется такой атрибут, который может читаться и изменяться любым методом любого объекта.

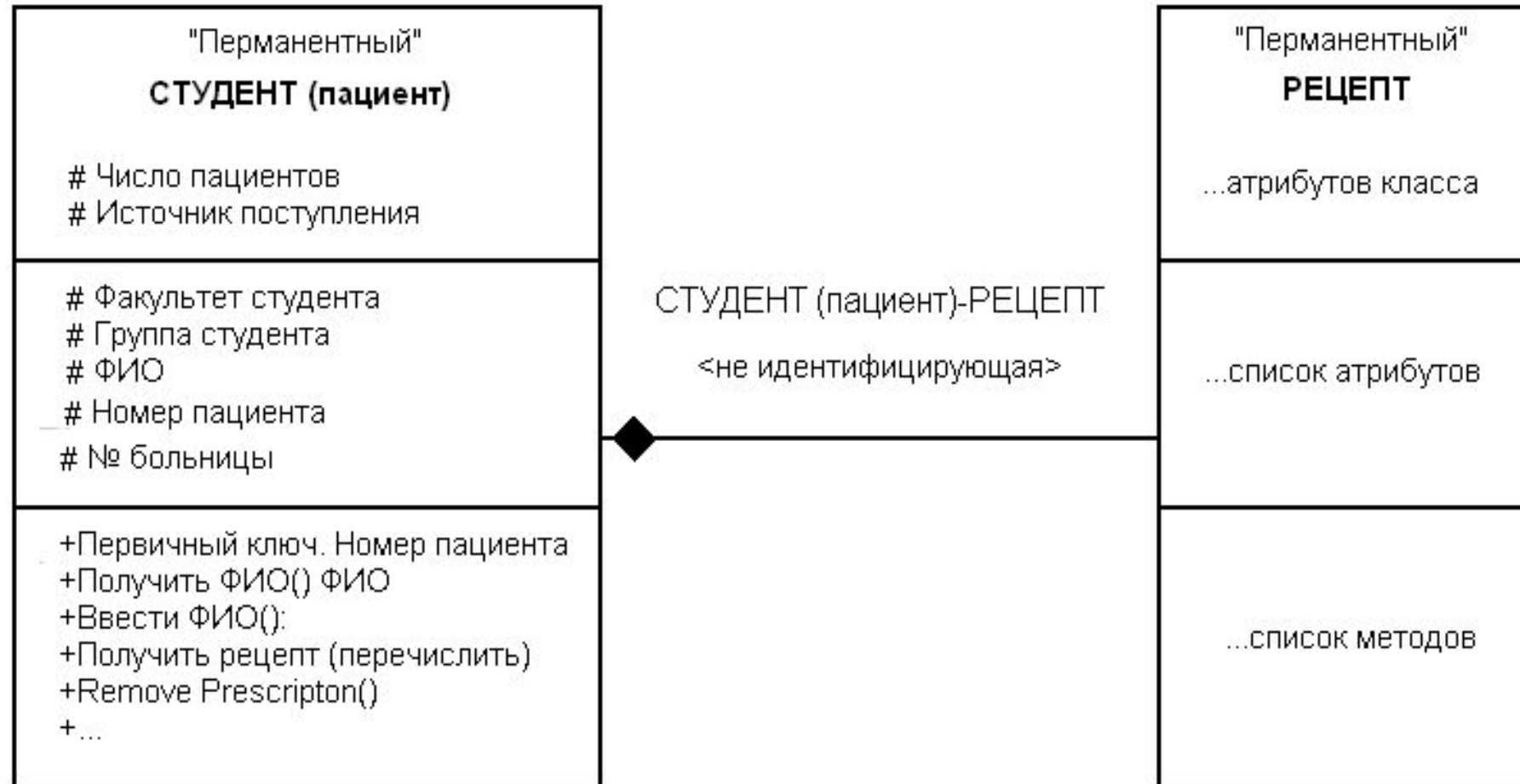
Термин **защищенный** (protected) означает, что атрибут или метод доступен только для методов данного класса и его подклассов.

А термин **закрытый** (private) указывает на то, что соответствующий атрибут или метод доступен только для методов данного класса.

- В UML задаются ограничения и методы.

Диаграммы «сущность-связь» в стиле UML

Представление классов сущностей в UML с помощью конструкций ООП



Моделирование потоков данных (процессов)

Основными компонентами диаграмм потоков данных являются:

- внешние сущности,
- системы и подсистемы,
- процессы,
- накопители данных,
- потоки данных.

Внешние сущности – те сущности, которые являются источниками или потребителями-приемниками информации (поставщик, заказчик, клиент, склад).

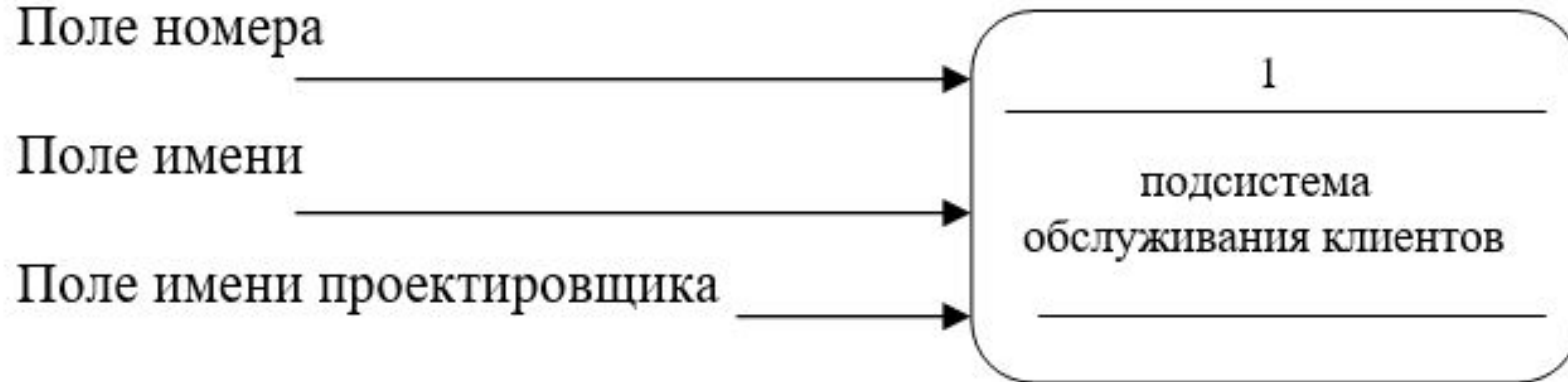
В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой системы, а часть процессов может быть вынесена за ее пределы и быть представлена, как внешняя сущность.

Внешняя сущность изображается в виде прямоугольника, изображенного, как бы над диаграммой и отбрасывающей на него тень.



Моделирование потоков данных (процессов)

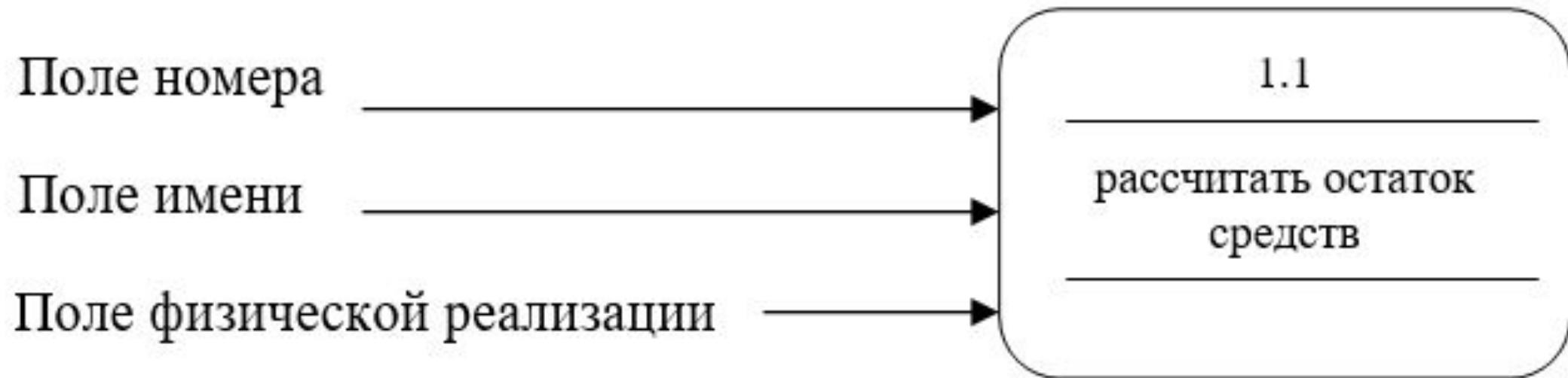
Системы и подсистемы. При построении модели система может быть представлена в виде контекстной диаграммы или может быть декомпозирована на ряд подсистем.



Номер подсистемы служит для ее идентификации, в полях имени вводится наименование подсистемы в форме существительного с определениями и дополнениями.

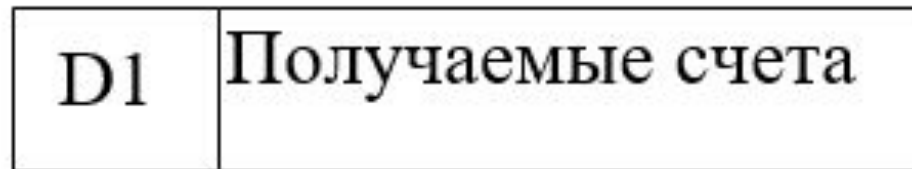
Моделирование потоков данных (процессов)

Процесс– процесс представляет собой преобразование ВХОДНЫХ ПОТОКОВ ДАННЫХ В ВЫХОДНЫЕ В СООТВЕТСТВИИ С определенным алгоритмом.



Моделирование потоков данных (процессов)

Накопитель данных – представляет собой абстрактное устройство для хранения информации, которую можно поместить, а затем при необходимости извлечь оттуда (причем способы помещения и извлечения информации могут быть любыми).носителем может быть - ящик в картотеке, файл на магнитном носителе, таблица в оперативной памяти и т.д. на диаграмме носитель выглядит, как показано ниже

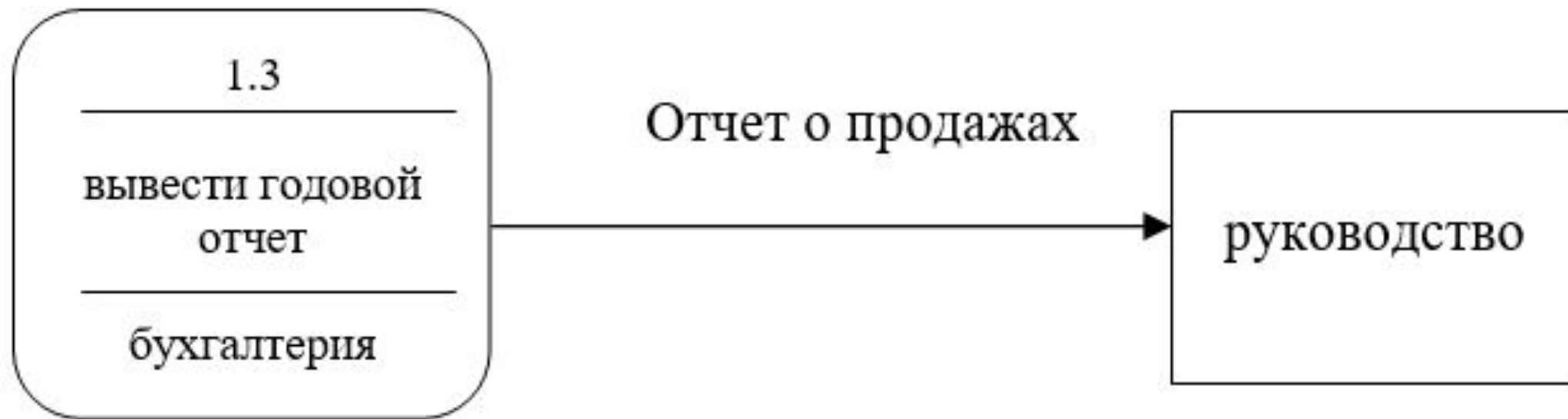


Накопитель данных идентифицируется буквой D и произвольным числом. Имя выбирается из соображения наибольшей информативности.

Накопитель, как правило является прообразом будущей БД, а, следовательно, описание хранящихся в нем данных должно

Моделирование потоков данных (процессов)

Поток данных - определяет информацию, передаваемую через некоторое соединение от источника к приемнику. На диаграмме поток показывается стрелкой, указывающей направление потока.. поток имеет имя отражающее его содержание.

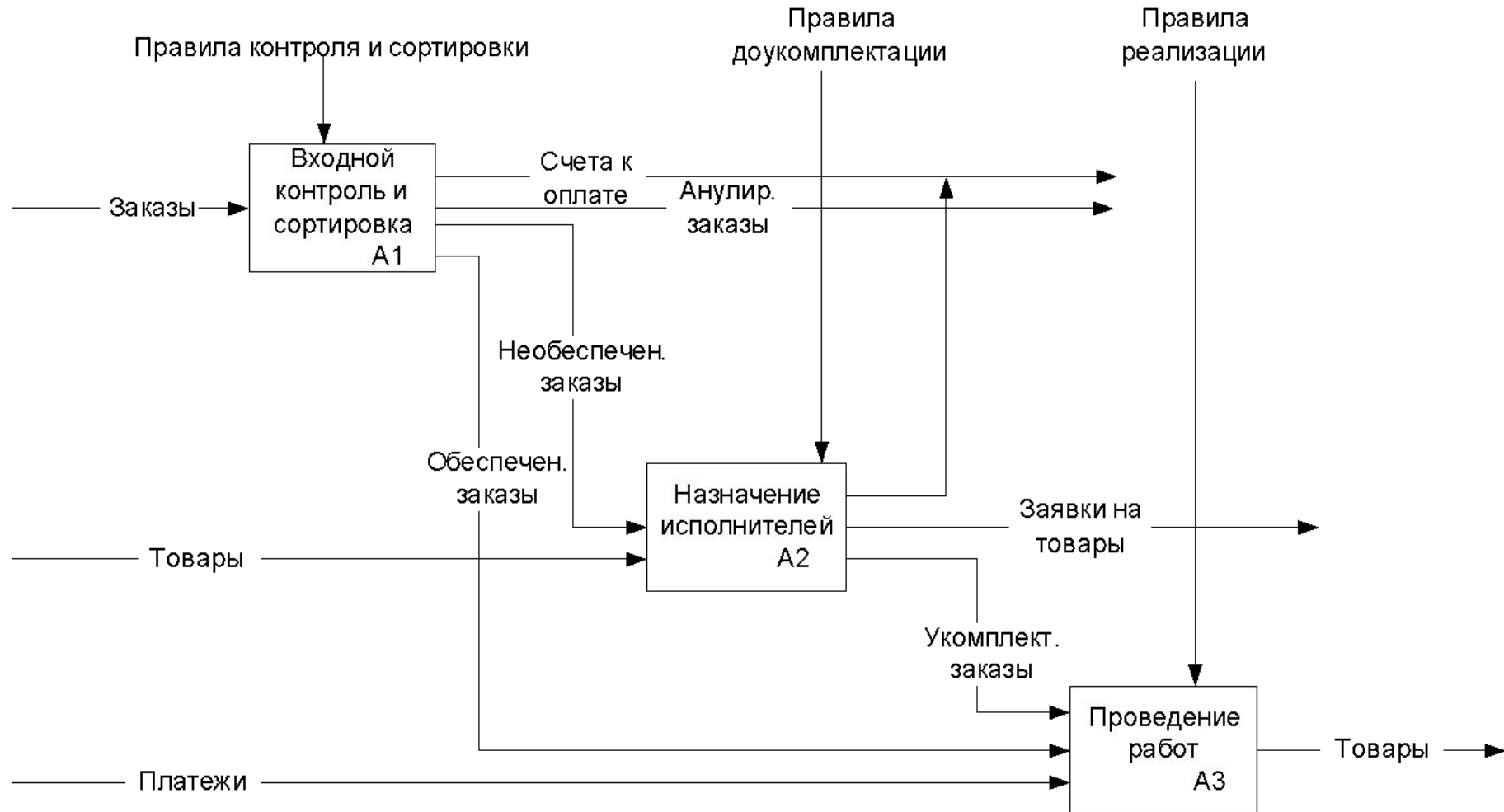


Анализ и структурно-логическое проектирование систем

Методика Сущность – Связь
Построение структур данных

Функциональное моделирование

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



Диаграммы потоков данных

- *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 становятся элементарными и детализировать их далее невозможно. На этом уровне декомпозиции выполняются мини спецификации.

Диаграммы потоков данных

- **Основные компоненты диаграмм потоков данных:**

- 4 **внешние сущности;**
- 4 **системы/подсистемы;**
- 4 **процессы;**
- 4 **накопители данных;**
- 4 **потоки данных.**

Диаграммы потоков данных

- **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

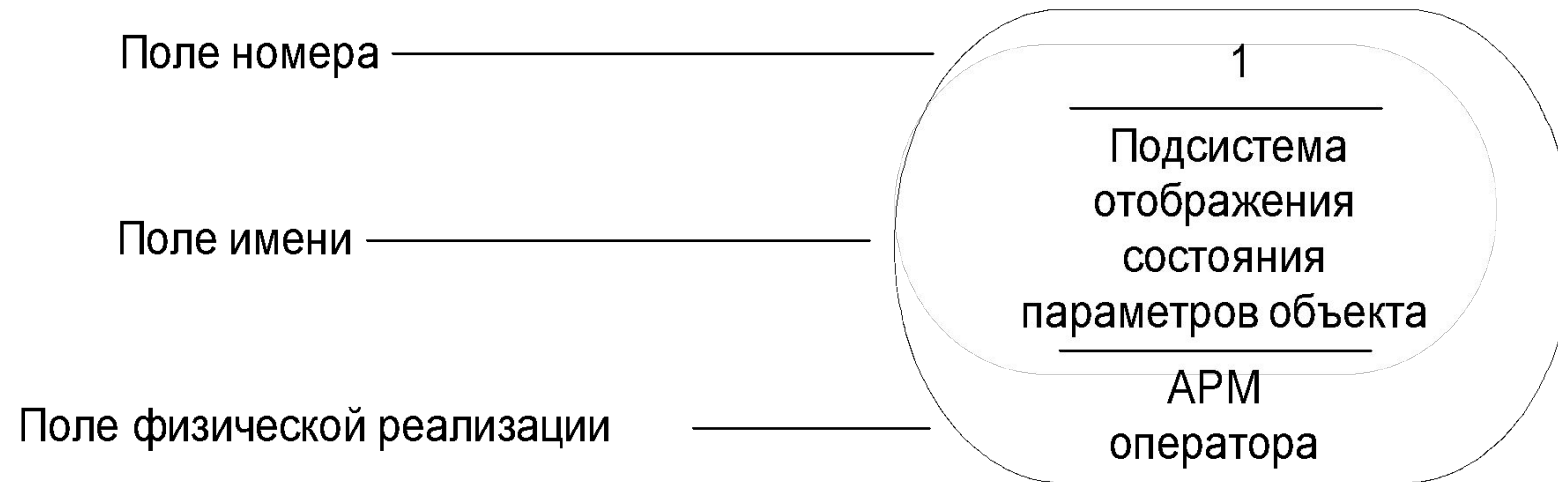
Диаграммы потоков данных

- **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

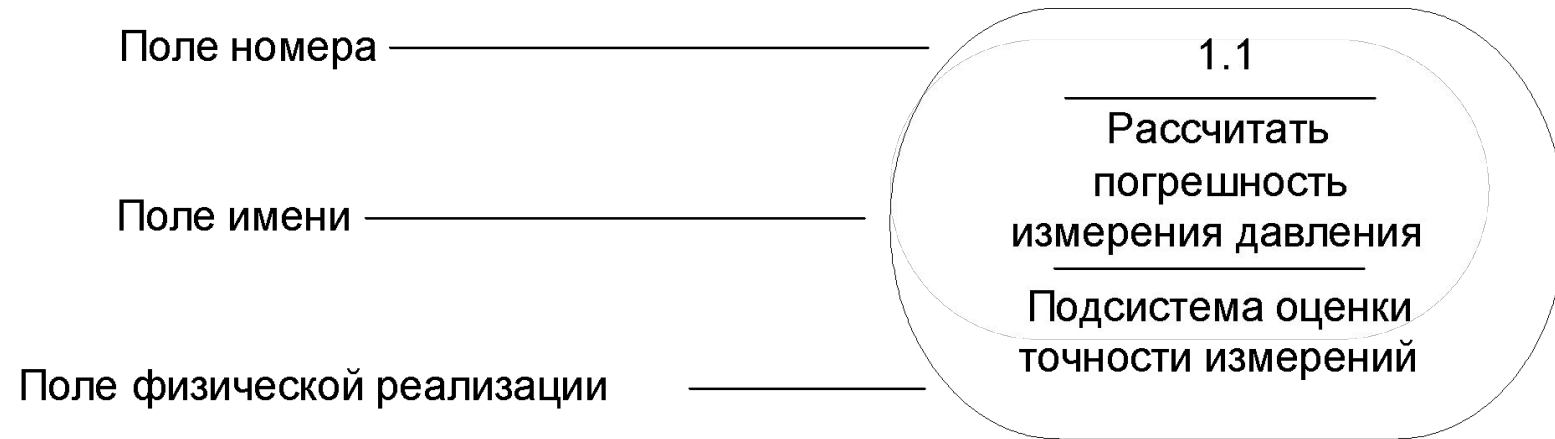
• **Системы и подсистемы**



***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

• **Процессы**

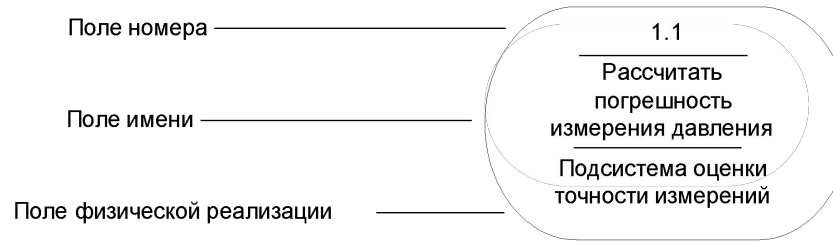


Процесс характеризует конкретное преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

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

Диаграммы потоков данных

• 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.

- **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

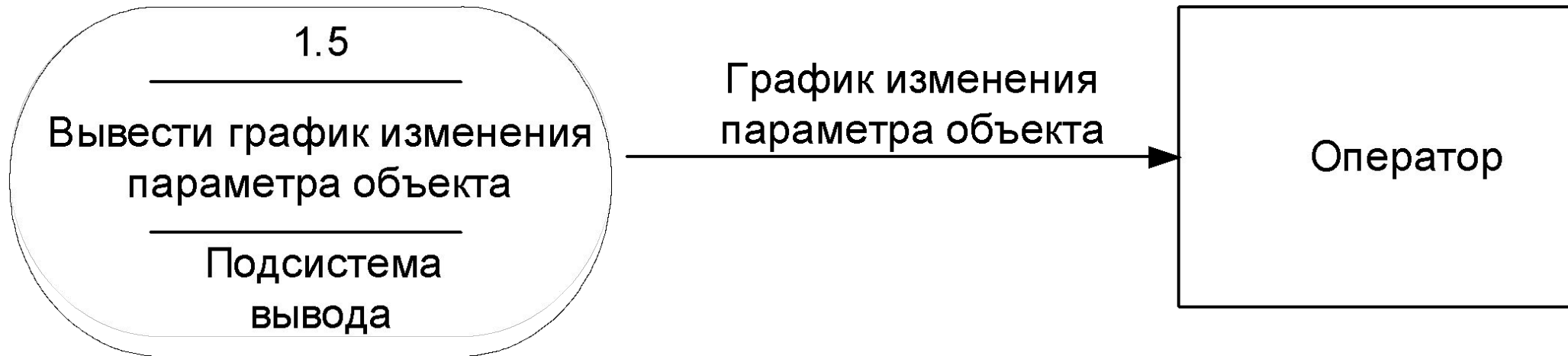
- **Накопители данных**

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

• Поток данных



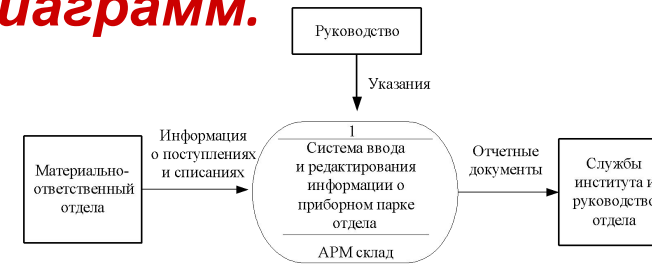
- **Поток данных** определяет информацию, передаваемую через некоторое соединение от источника к приемнику.
- **Реальный поток** данных может быть информацией, передаваемой по линии связи между датчиком и микроконтроллером, по кабелю между двумя вычислительными устройствами и т.д.
- На диаграмме поток 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 имя, отражающее его содержание.

Построение иерархии диаграмм потоков данных
**Первый уровень представления модели -
уровень контекстных диаграмм.**



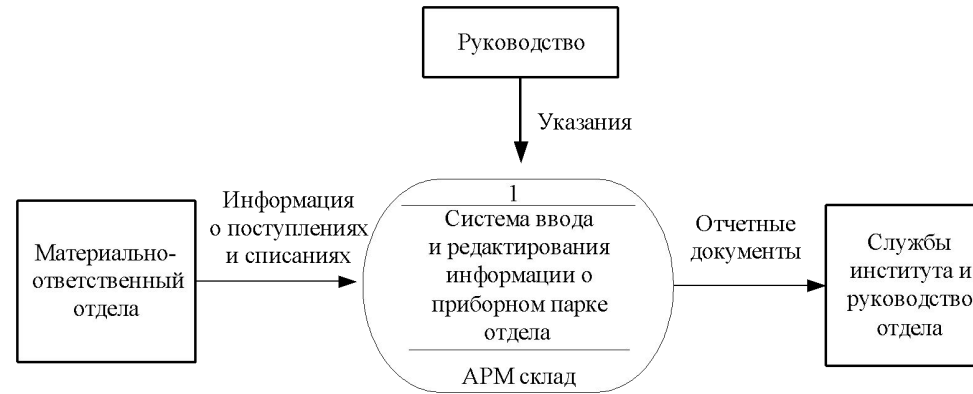
- **Для сложных ИС** строится иерархия контекстных диаграмм.
- **Для модели сложной ИС** контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных.
- **Признаками сложности ИС** могут быть:
 - наличие большого количества внешних сущностей (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.

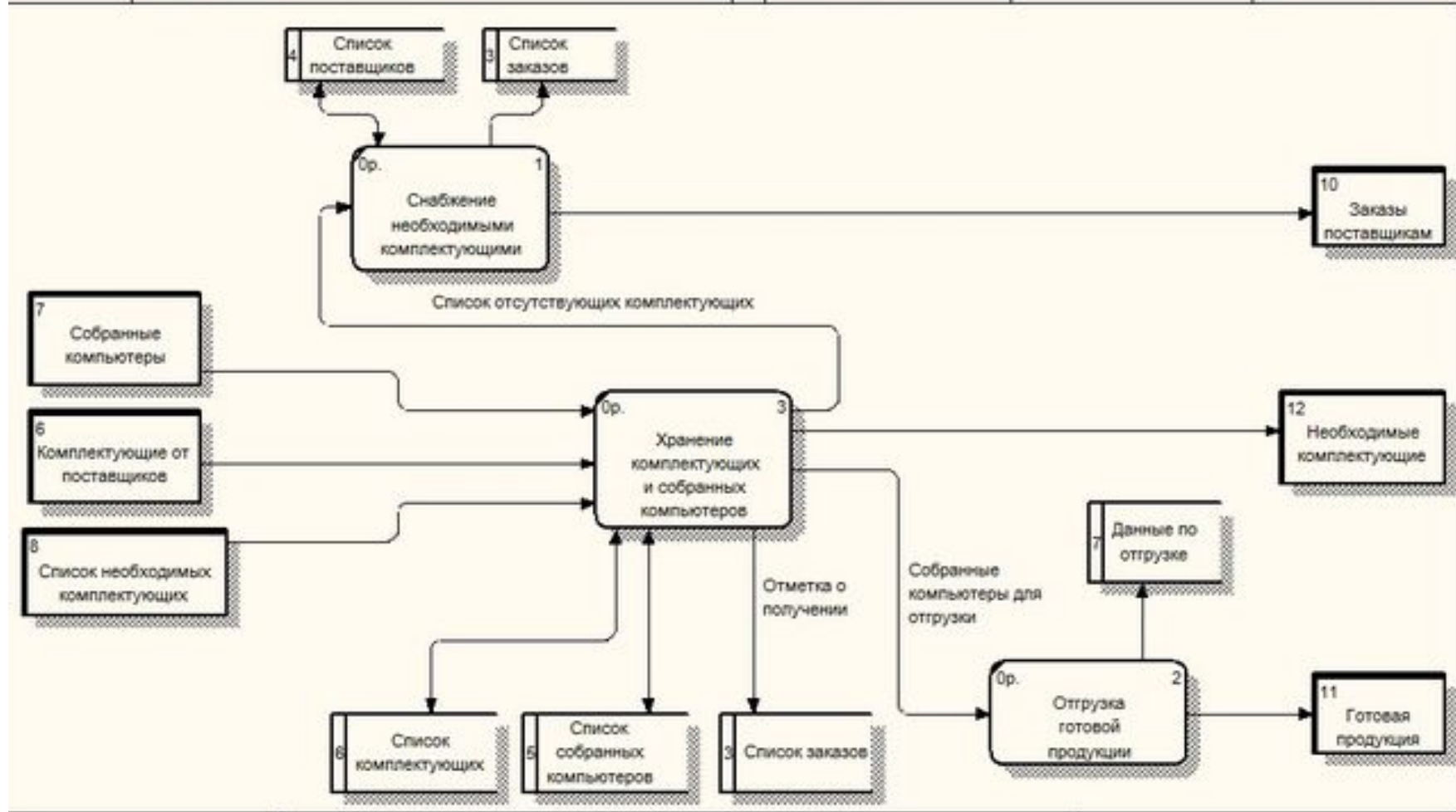
Построение иерархии диаграмм потоков данных

Примеры диаграмм



Построение иерархии диаграмм потоков данных

Примеры диаграмм

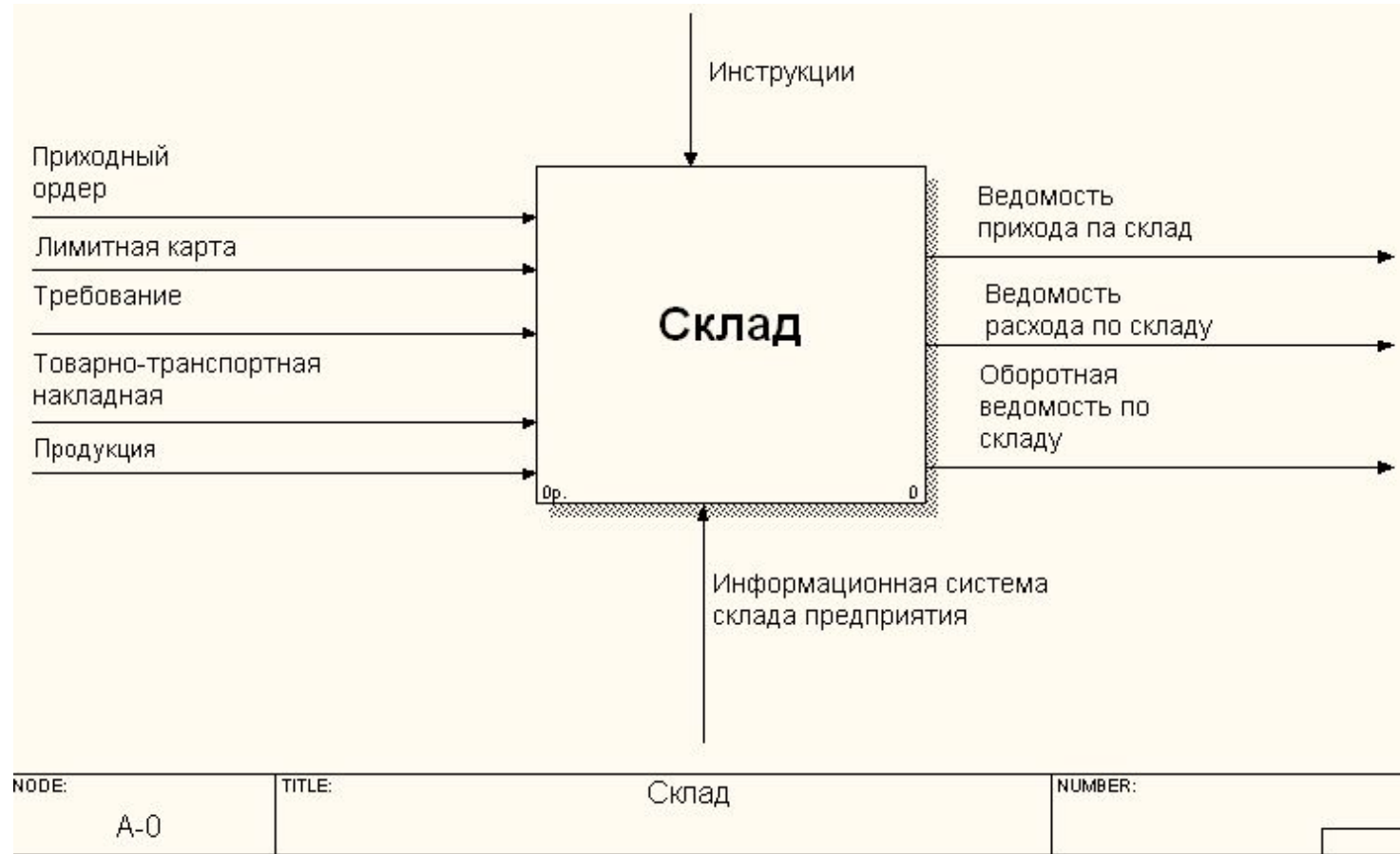


Построение иерархии диаграмм потоков данных

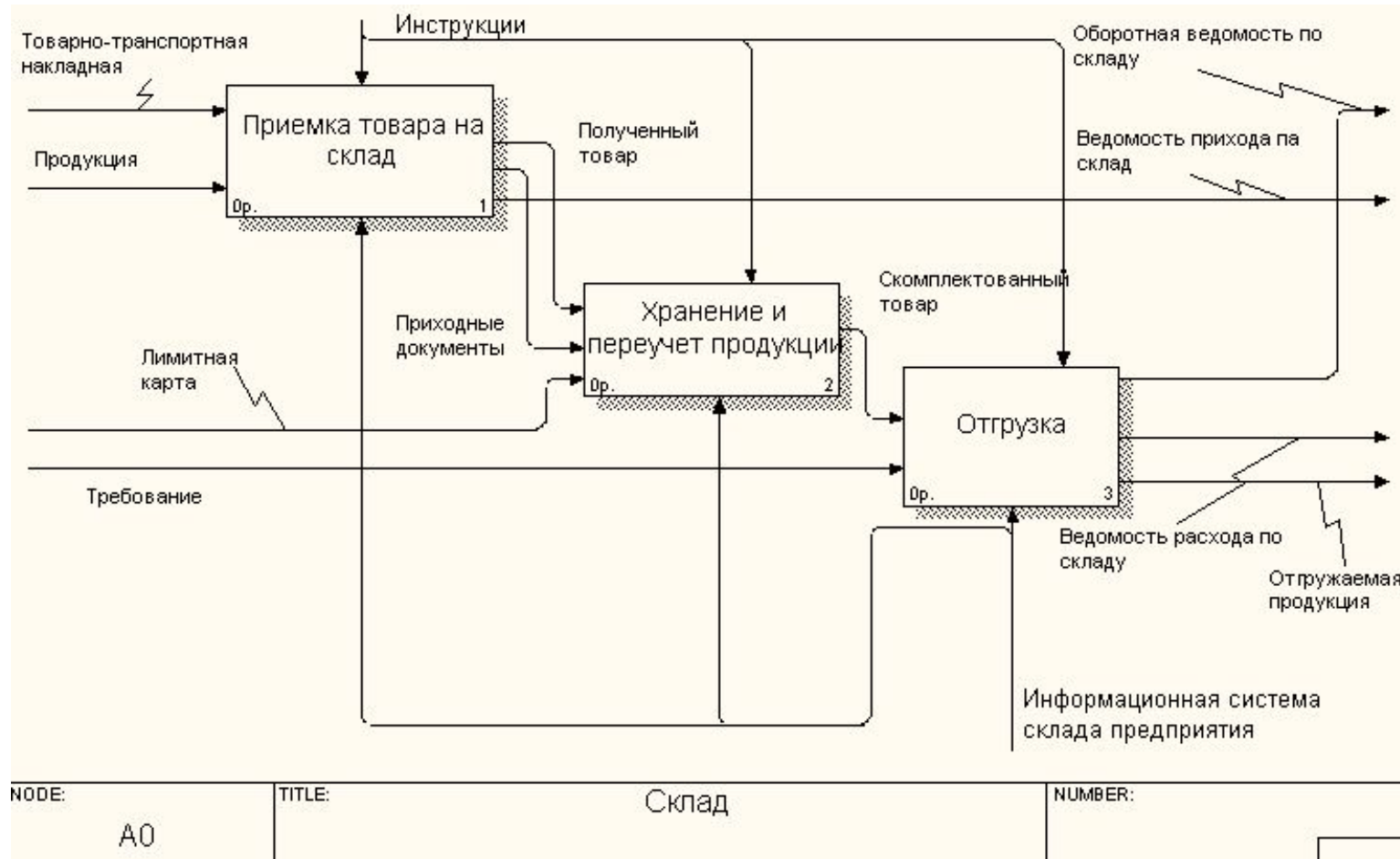
Примеры диаграмм



- Пример IDEF0 -диаграммы, моделирующей деятельность склада
- Контекстная диаграмма



- Пример IDEF0 -диаграммы, моделирующей деятельность склада
- Детализация



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

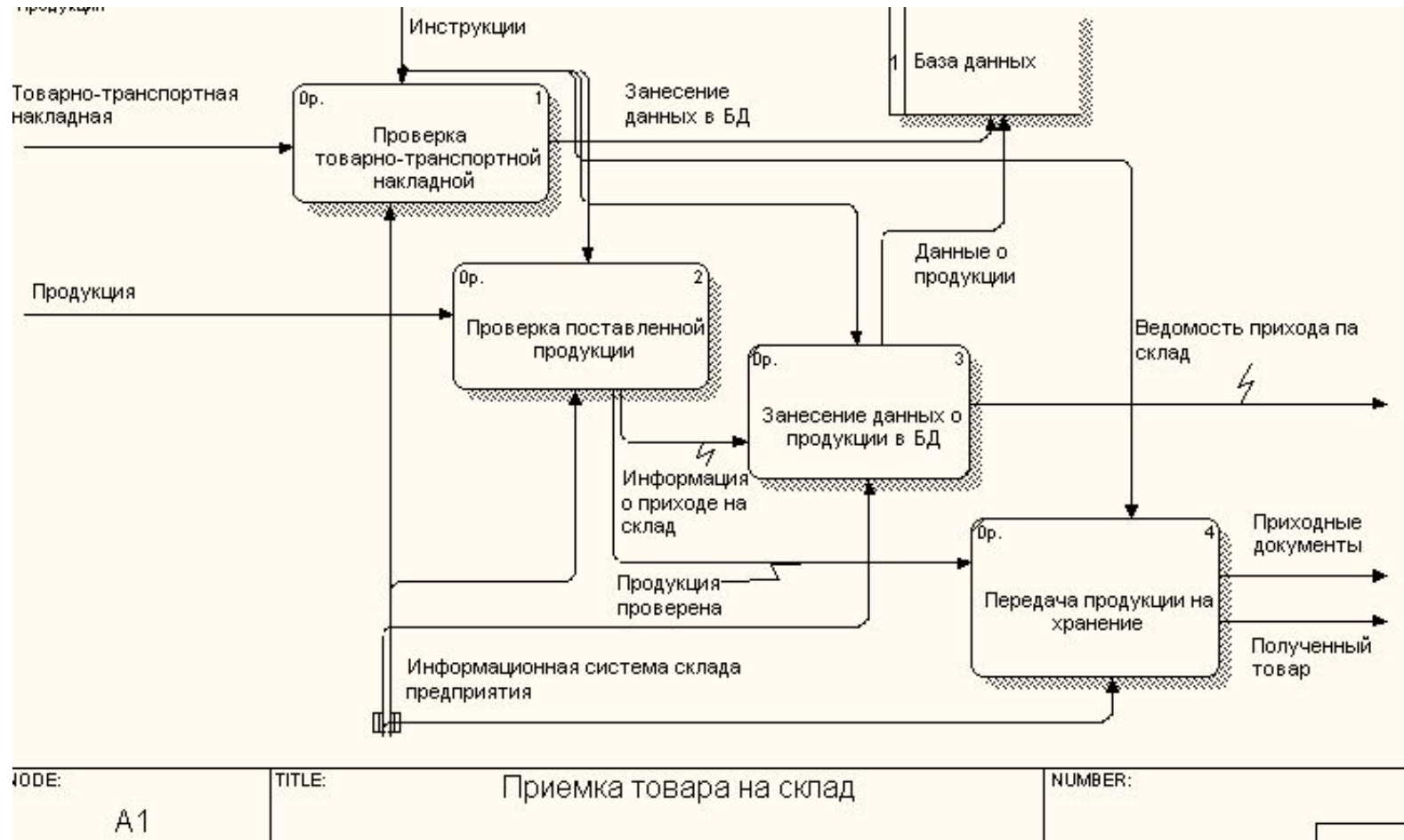


Диаграмма DFD «Приемка товара на склад»

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

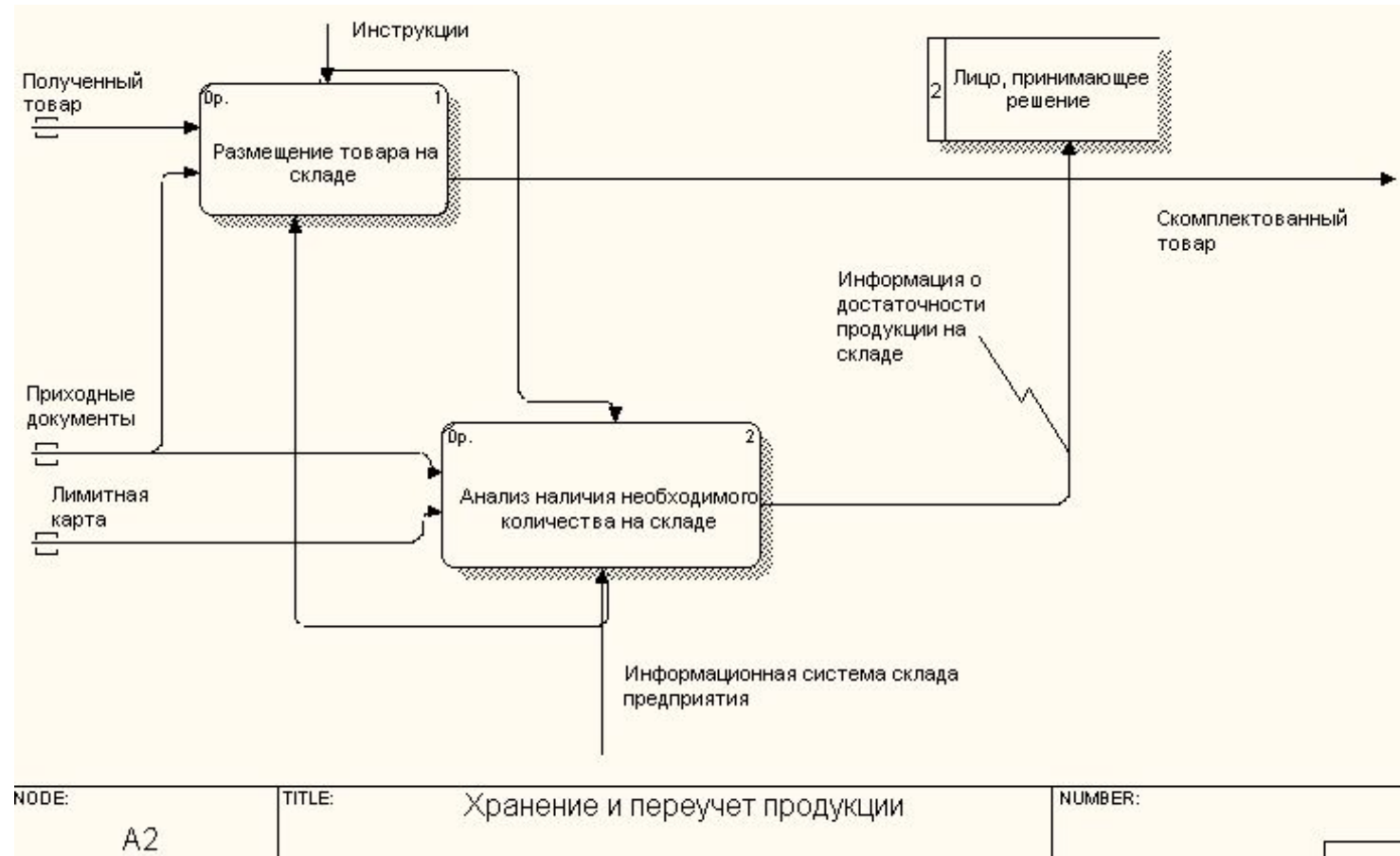


Диаграмма DFD «Хранение и переучет продукции»

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

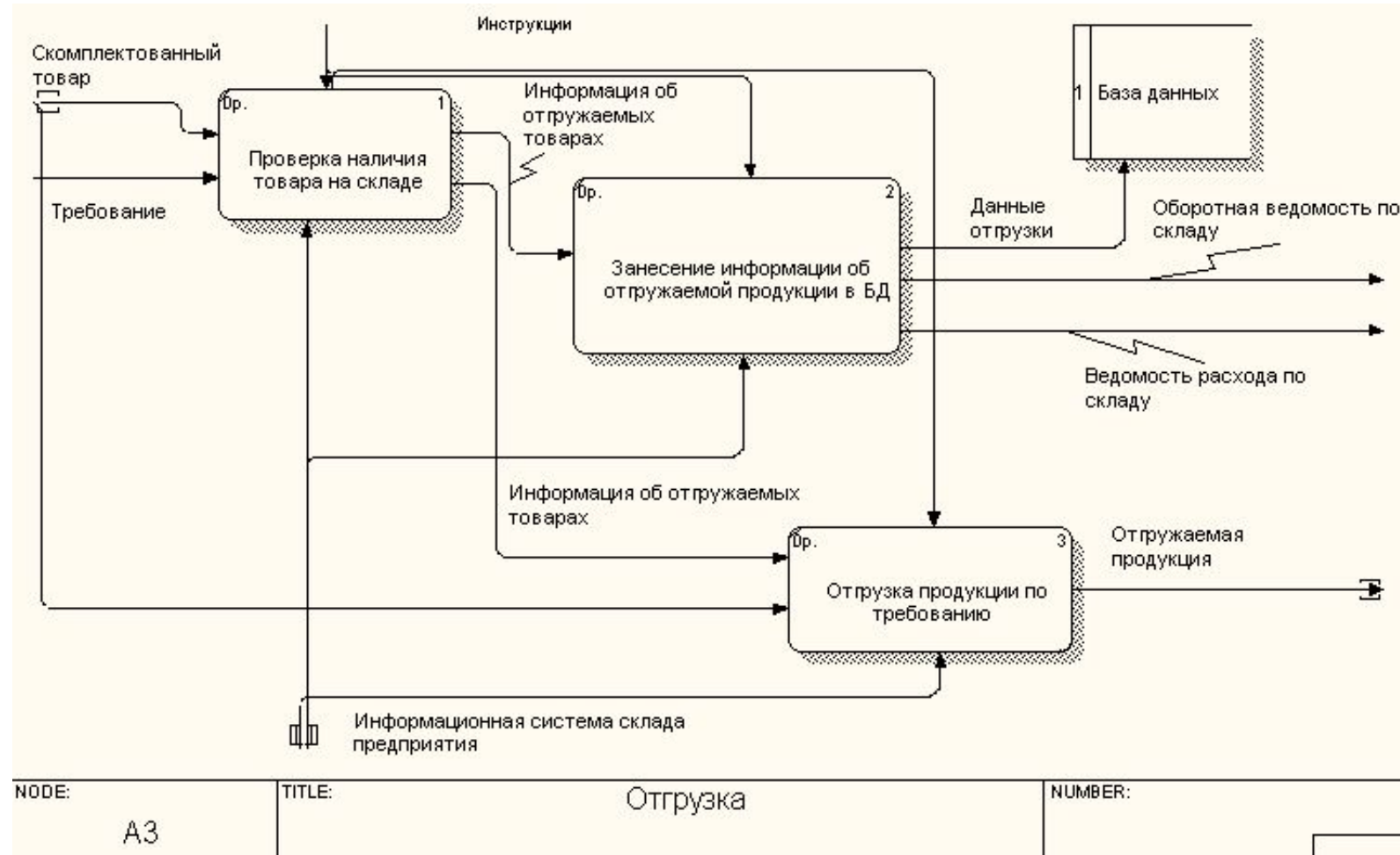


Диаграмма DFD «Отгрузка»

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 принадлежности:

Обязательный – (O);

Необязательный – (N)

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

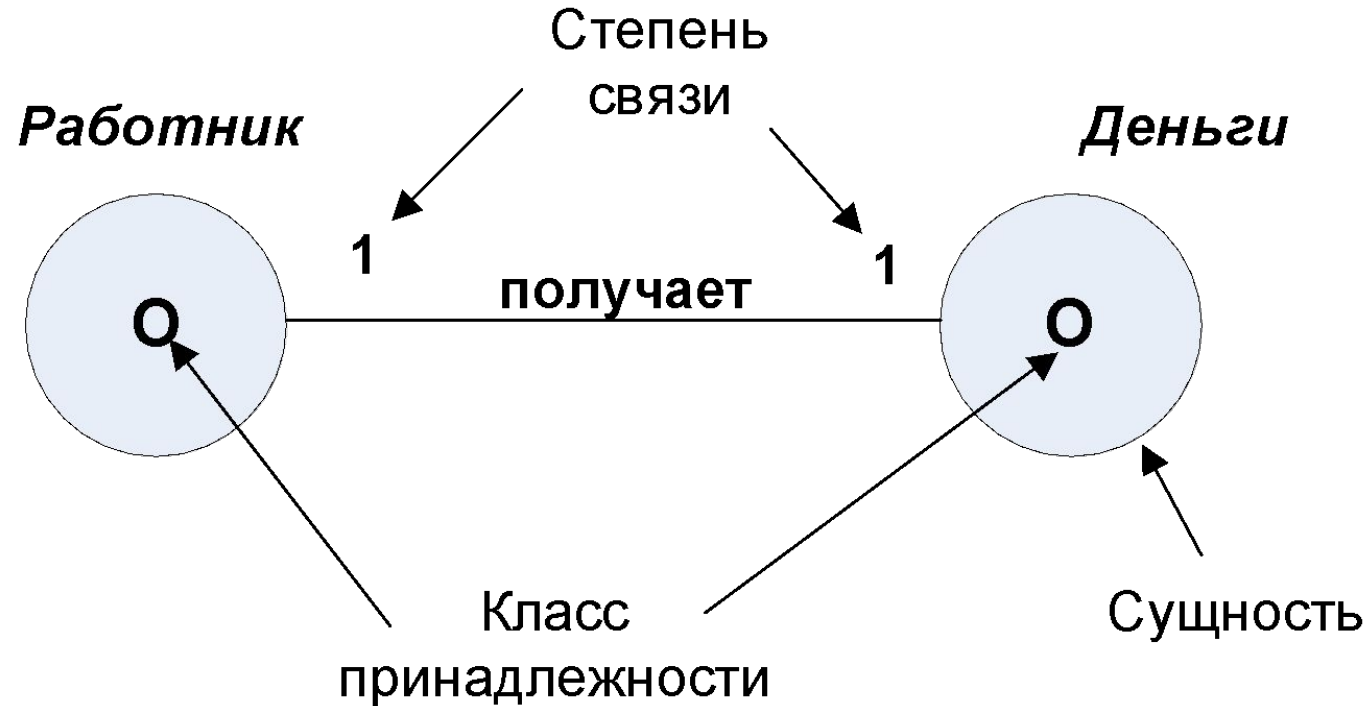


Диаграмма для конструкции "Работник получает Деньги"

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 **компонитным, 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).

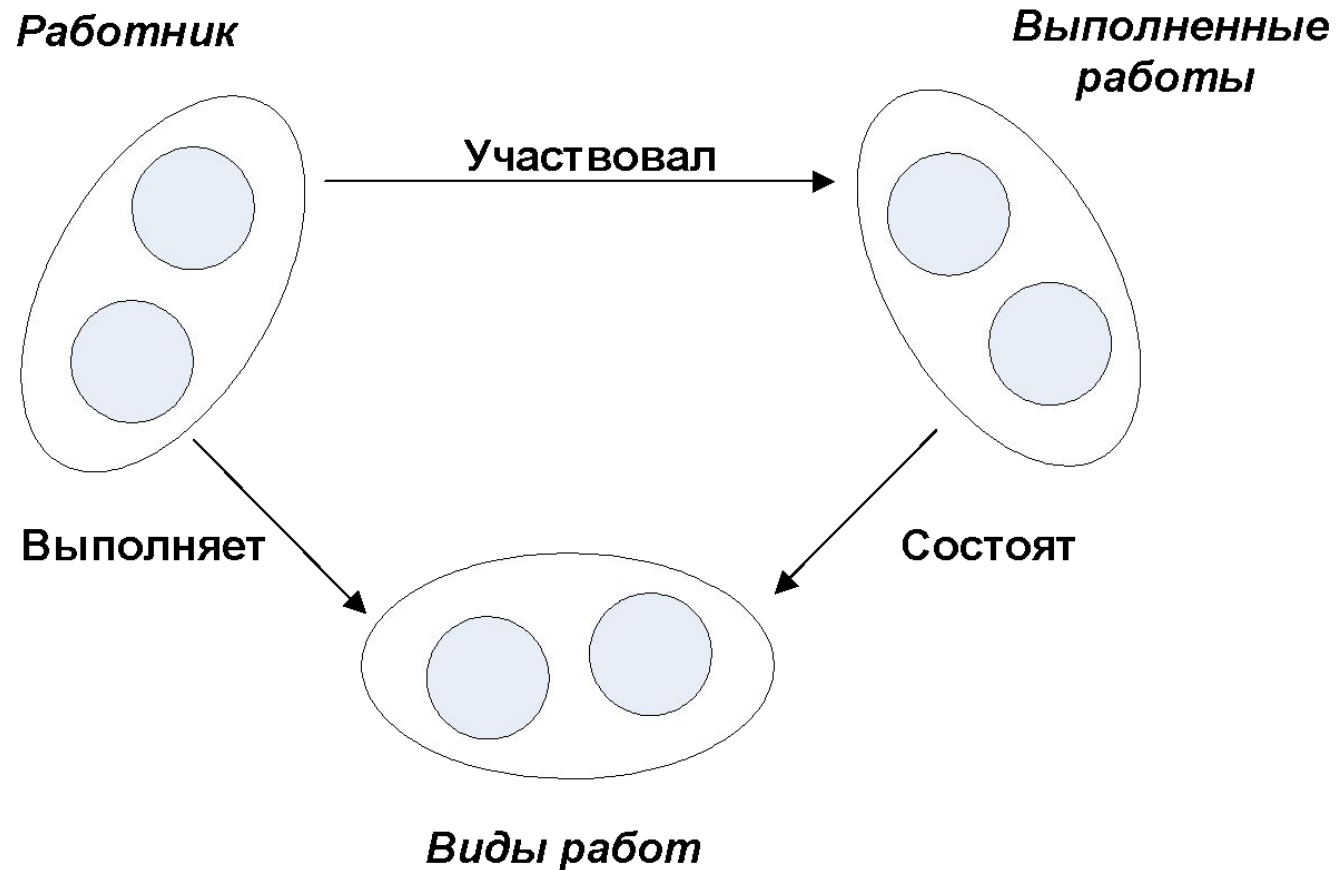
Работник имеет Баланс выплаты.

Выполненные работы составляют Баланс.

Методика "сущность-связь" построения структур баз данных

Пример применения методики - Система подсчета зарплаты.

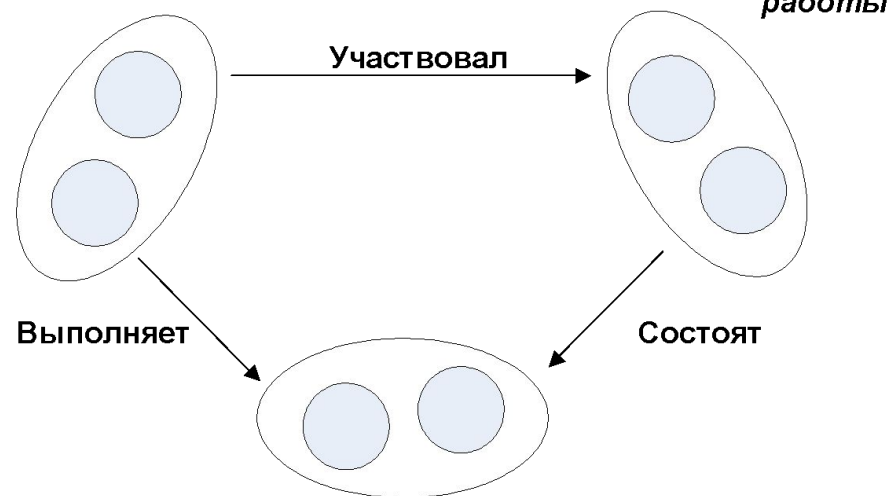
Структура диаграммы будет выглядеть следующим образом



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:

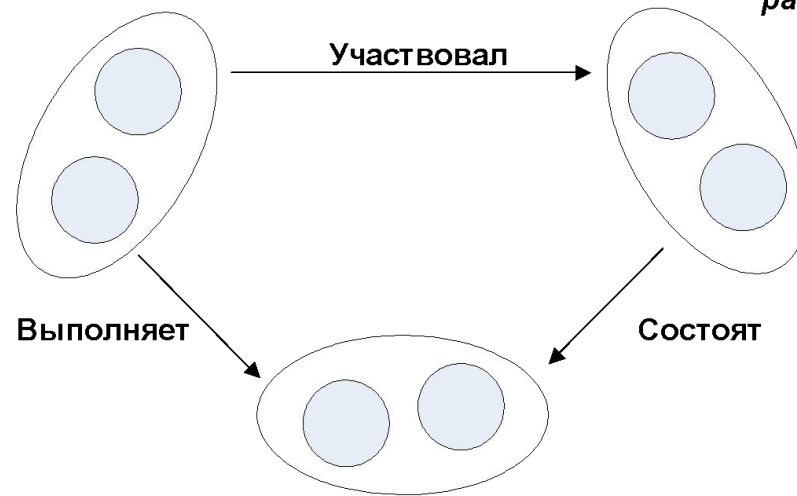
каждый работник может выполнять несколько видов работ;
каждый вид работы может быть выполнен несколькими работниками

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

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;

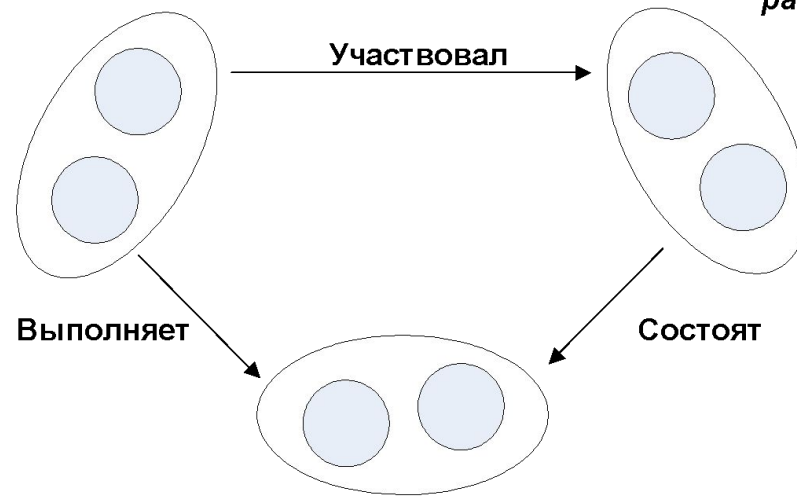
каждая выполненная работа может быть только одного вида.

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 - O.

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

2. Определение степени связи:

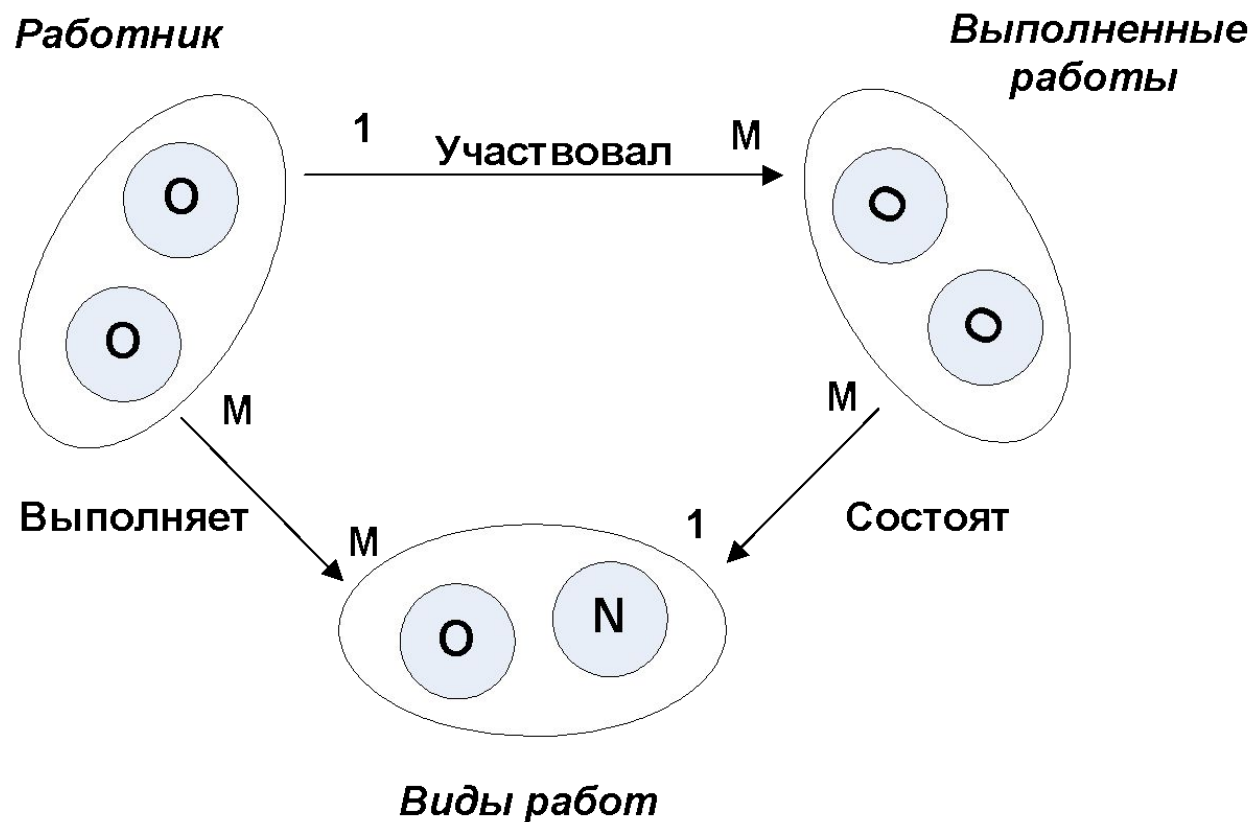
один работник может участвовать в выполнении множества работ;

одна выполненная работа выполняется одним работником.

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

Методика "сущность-связь" построения структур баз данных

**Пример применен
ия 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 №").