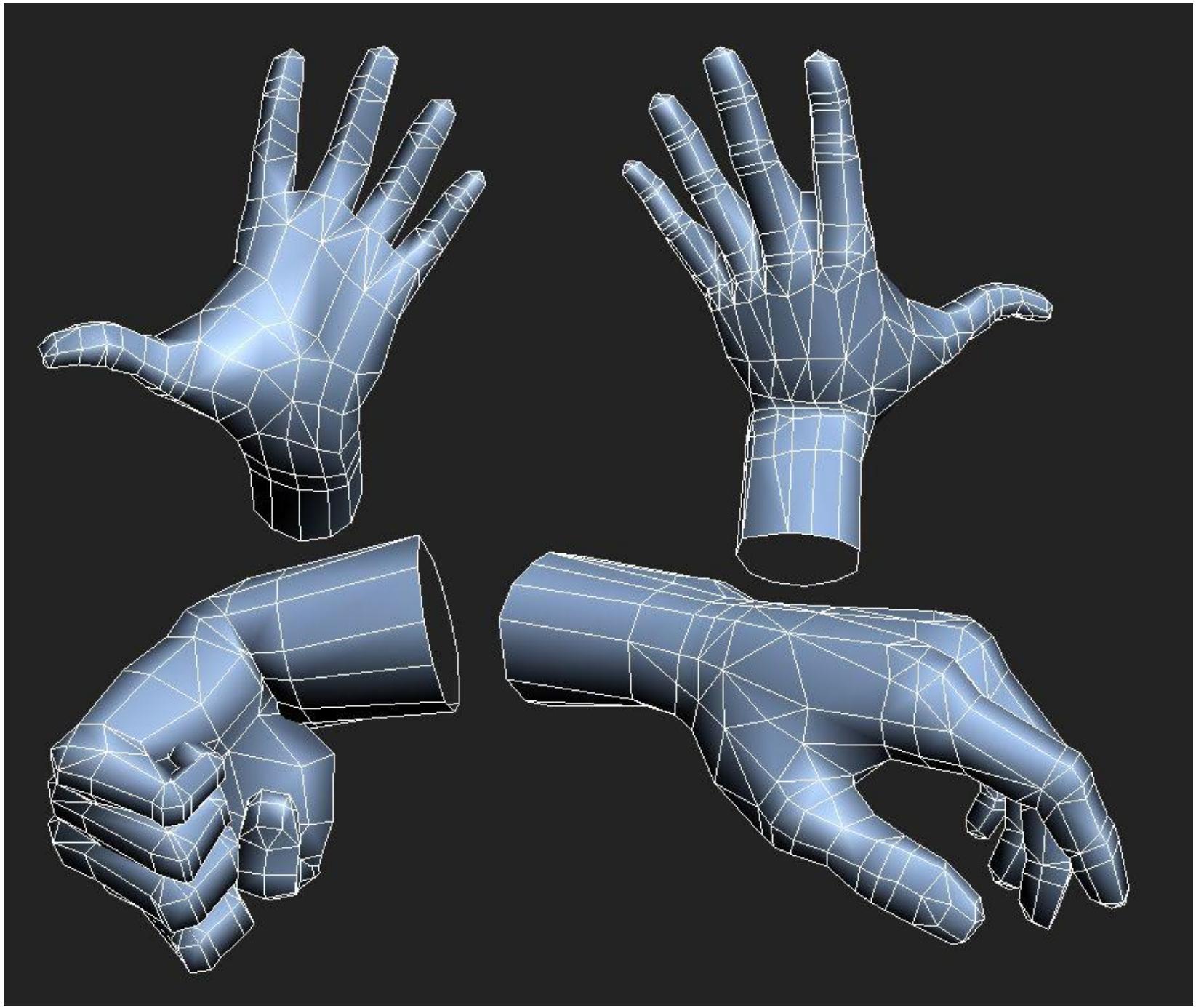
Устройство и функционирование информационных систем

Раздел 3. Основные принципы моделирования ИС

Тема 3.1 Информационное моделирование и формализация

Понятие моделей



- Человечество в своей деятельности (научной, образовательной, технологической, художественной) постоянно создает и использует многообразные модели окружающего мира.

Социумом накоплен богатый опыт моделирования различных объектов и процессов.

Понятие моделей

- Термин «модель» :
  - некоторое упрощенное подобие реального объекта;
  - воспроизведение предмета в уменьшенном или увеличенном виде (макет);
  - схема, изображение или описание какого-либо явления или процесса в природе и обществе;

- Модель
  - физический или информационный аналог объекта, функционирование которого по определенным параметрам подобно функционированию реального объекта;

- Модель
  - некий объект-заместитель, который может заменять объект-оригинал, воспроизводя интересующие нас свойства и характеристики и имеет существенные преимущества или удобства;

- Модель
  - новый объект (реальный, информационный или воображаемый), отличный от исходного, который обладает существенными для целей моделирования свойствами;
  - в рамках этих целей заменяет исходный объект.

Классификация моделей

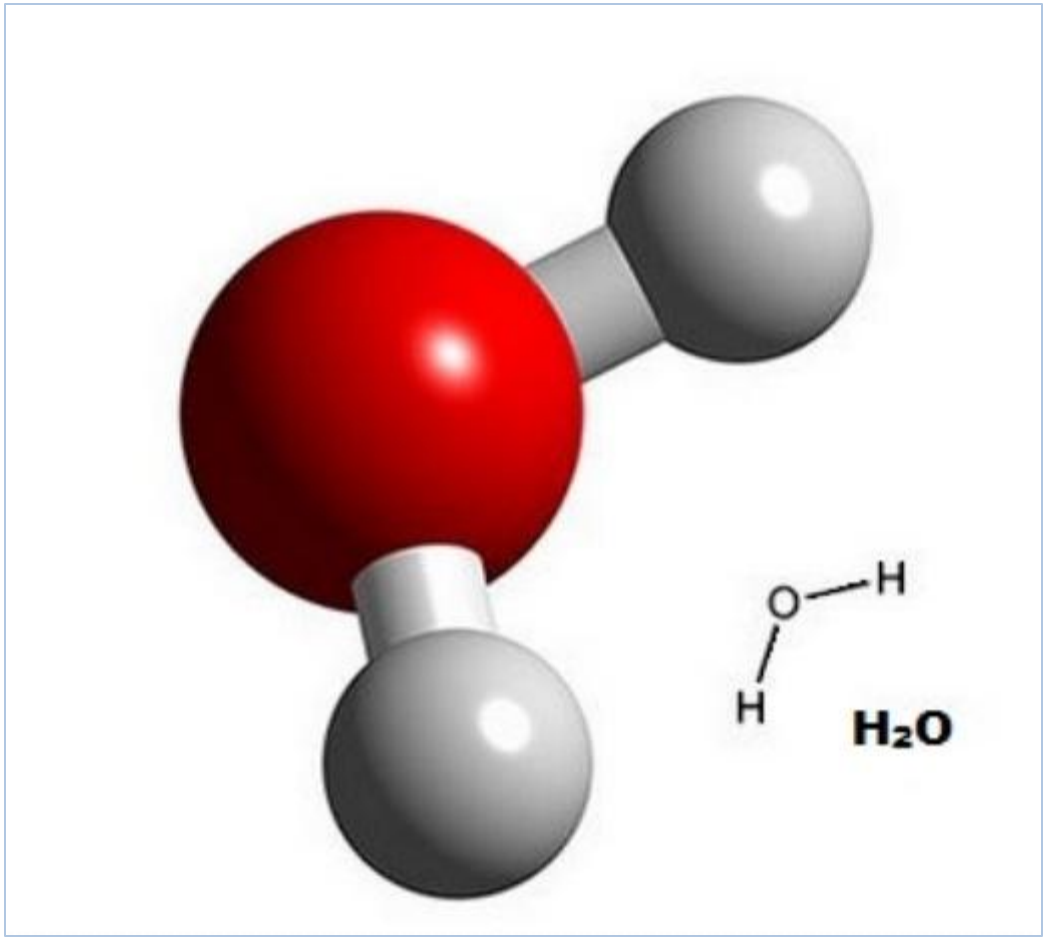
- Модели делятся на:
  - *материальные (натурные) модели* (некие реальные предметы – макеты, муляжи, эталоны) – уменьшенные или увеличенные копии, воспроизводящие внешний вид моделируемого объекта, его структуру или поведение;

Классификация моделей

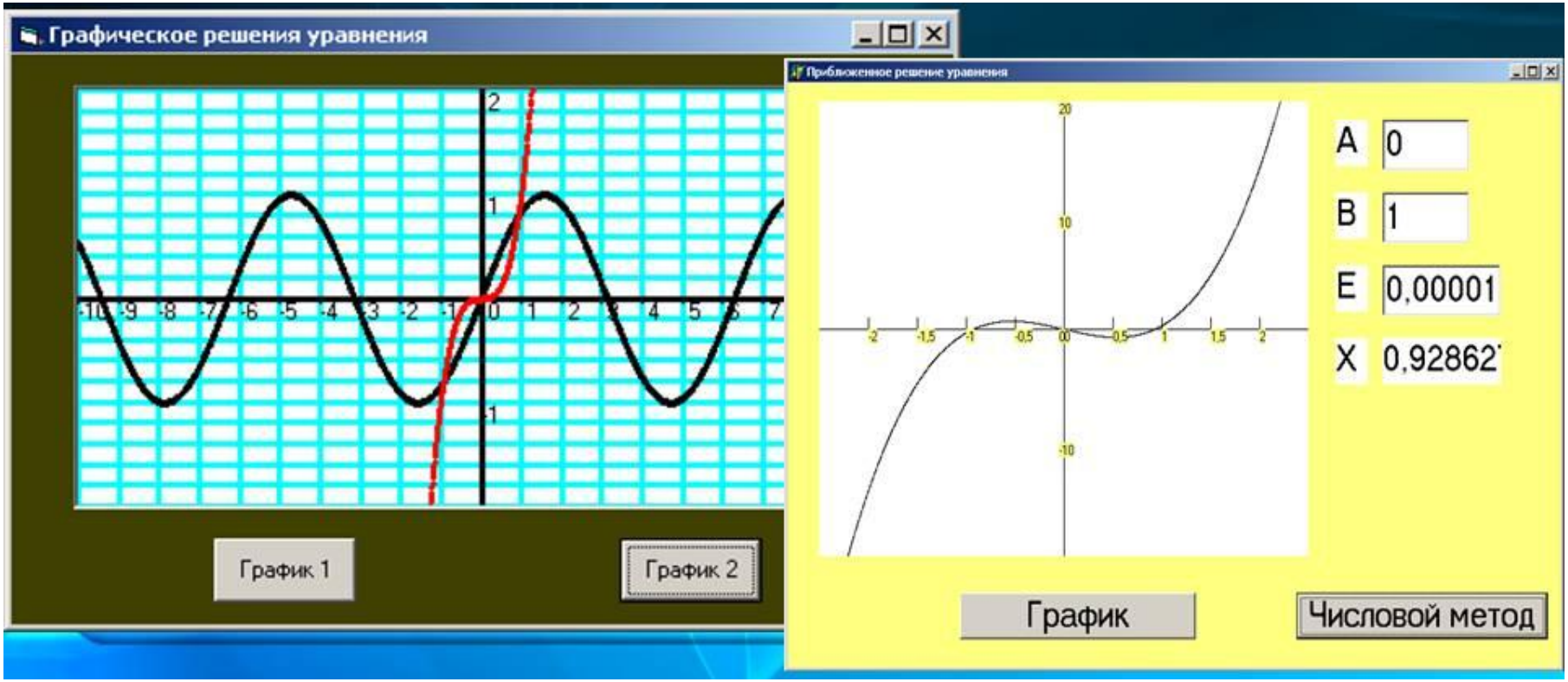
- *воображаемые модели* (геометрическая точка, математический маятник, идеальный газ, бесконечность);
- *информационные модели* – описания моделируемого объекта на одном из языков кодирования информации (словесное описание, схемы, чертежи, карты, рисунки, научные формулы, программы и пр.)

Классификация информационных моделей

- *Информационная (абстрактная)* модель – описание объекта на каком-либо языке
  - *Абстрактность* модели проявляется в том, что ее компонентами являются сигналы и знаки (заложенный в них смысл), а не физические тела.

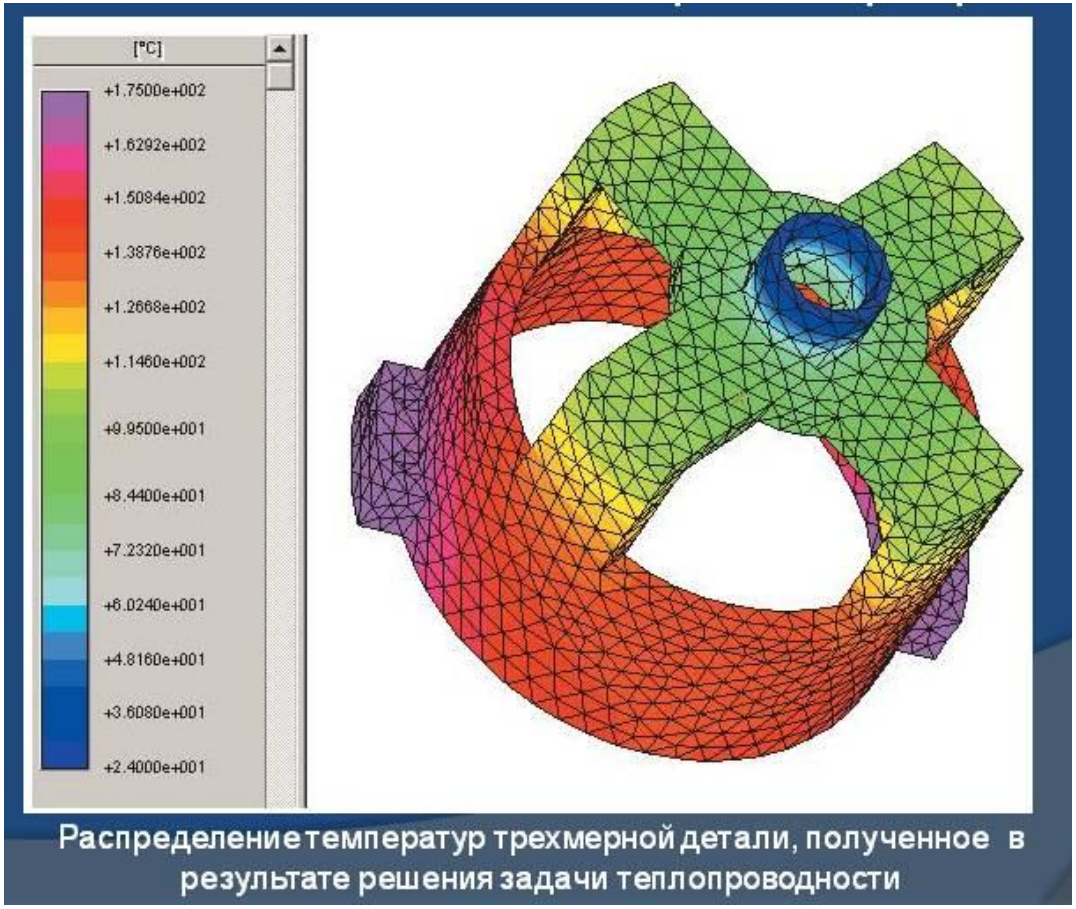


Дескриптивная модель – словесное описание объекта, выраженное средствами того или иного языка.



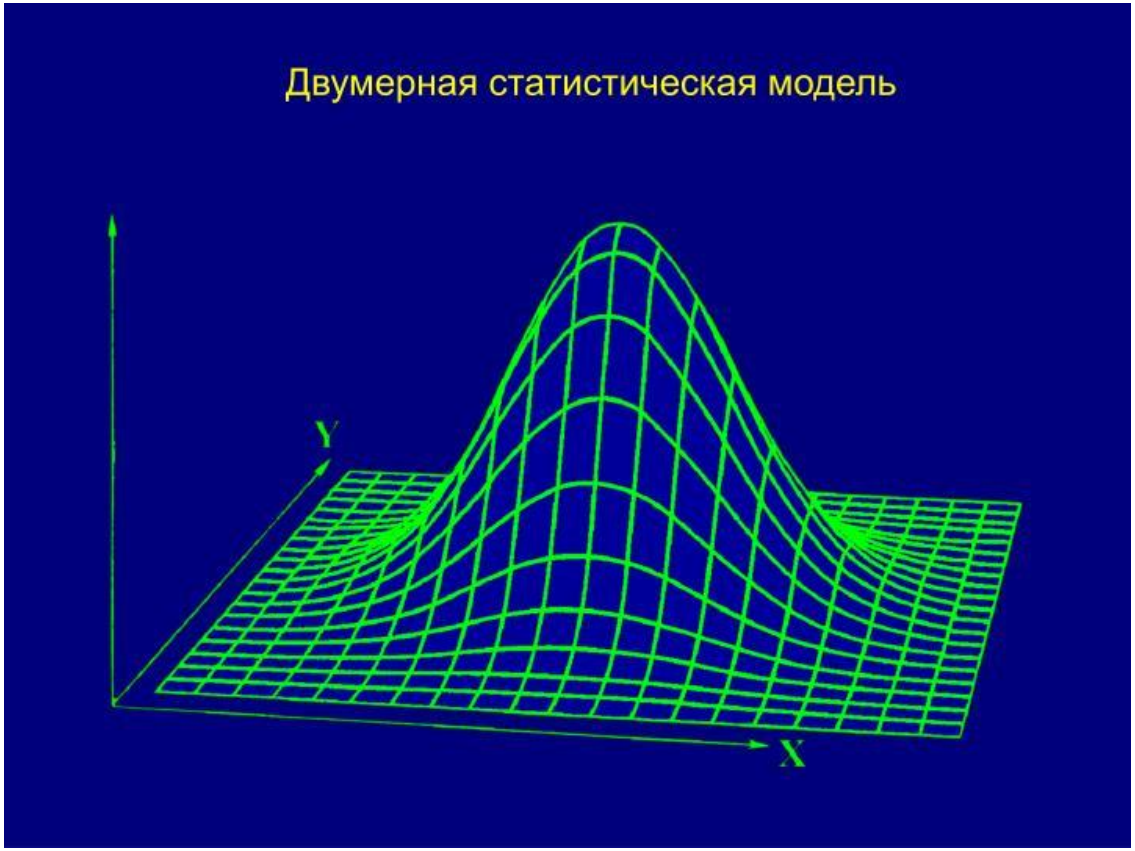
Математическая модель –

- совокупность записанных на языке математики соотношений (формул, неравенств, уравнений, логических соотношений), определяющих характеристики состояния объекта в зависимости от его элементов, свойств, параметров, внешних воздействий;
- приближенное описание объекта, выраженное с помощью математической символики.



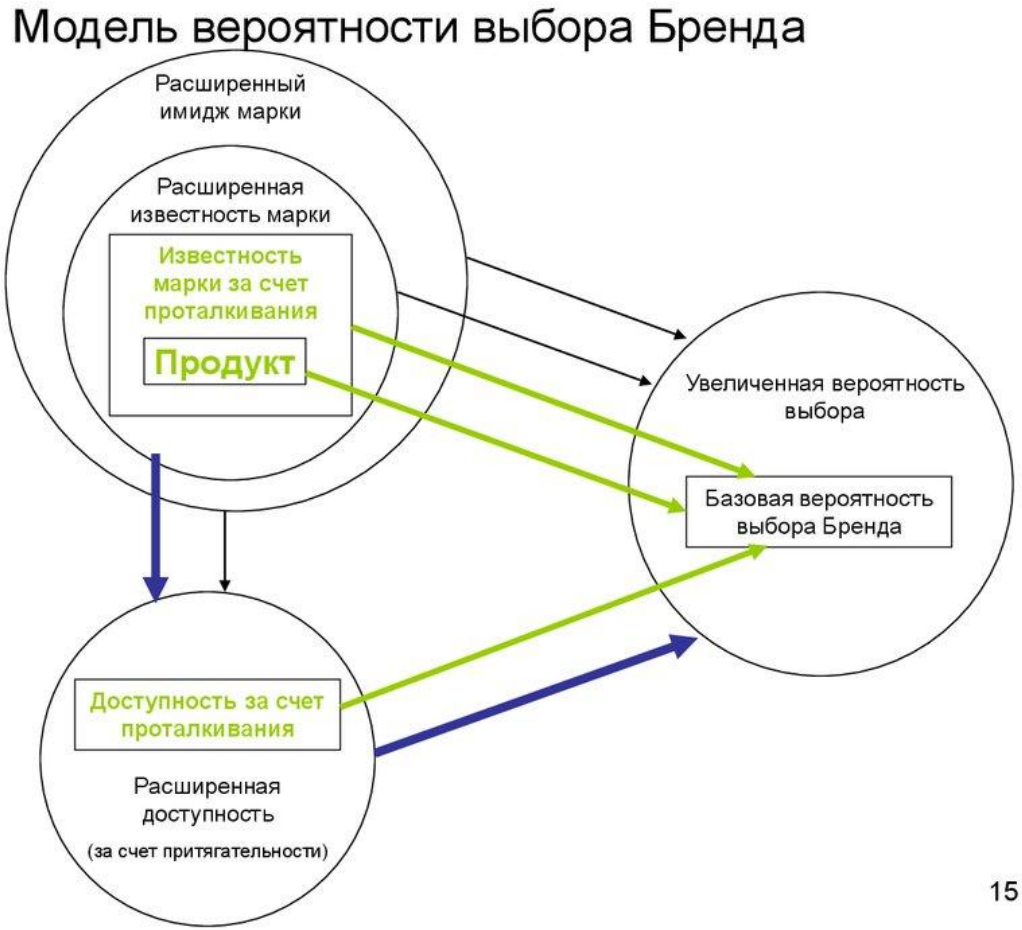
Распределение температур трехмерной детали, полученное в результате решения задачи теплопроводности

Свойство компьютерных математических моделей – <u>визуализация</u> расчётов.



Статические модели отображают объект в какой-то момент времени без учета происходящих с ним изменений - находящийся в состоянии покоя или равновесия (отсутствует параметр времени).

- *Динамические модели* описывают поведение объекта во времени
- *Детерминированные модели* отображают процессы, в которых отсутствуют случайные воздействия.



Вероятностные (стохастические) модели – описание объектов, поведение которых определяется случайными воздействиями;

описания вероятностных процессов и событий, характер изменения которых во времени точно предсказать невозможно.

Примеры событий

1. ПОСЛЕ ЗИМЫ НАСТУПАЕТ ВЕСНА.
2. ПОСЛЕ НОЧИ ПРИХОДИТ УТРО.
3. КАМЕНЬ ПАДАЕТ ВНИЗ.
4. ВОДА СТАНОВИТСЯ ТЕПЛЕЕ ПРИ НАГРЕВАНИИ.

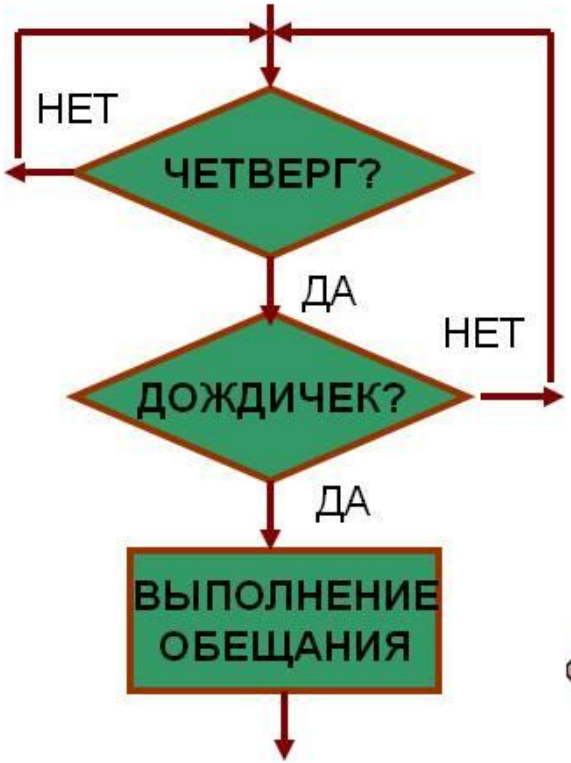
1. НАЙТИ КЛАД.
2. БУТЕРБРОД ПАДАЕТ МАСЛОМ ВНИЗ.
3. В ШКОЛЕ ОТМЕНИЛИ ЗАНЯТИЯ.
4. ПОЭТ ПОЛЬЗУЕТСЯ ВЕЛОСИПЕДОМ.
5. В ДОМЕ ЖИВЕТ КОШКА.

1. 30 ФЕВРАЛЯ ДЕНЬ РОЖДЕНИЯ.
2. ПРИ ПОДБРАСЫВАНИИ КУБИКА ВЫПАДАЕТ 7 ОЧКОВ.
3. ЧЕЛОВЕК РОЖДАЕТСЯ СТАРЫМ И СТАНОВИТСЯ С КАЖДЫМ ДНЕМ МОЛОЖЕ.

- *Имитационная компьютерная модель* – отдельная программа, совокупность программ, программный комплекс, позволяющий с помощью последовательности вычислений и графического отображения их результатов воспроизводить (имитировать) процессы функционирования объекта.

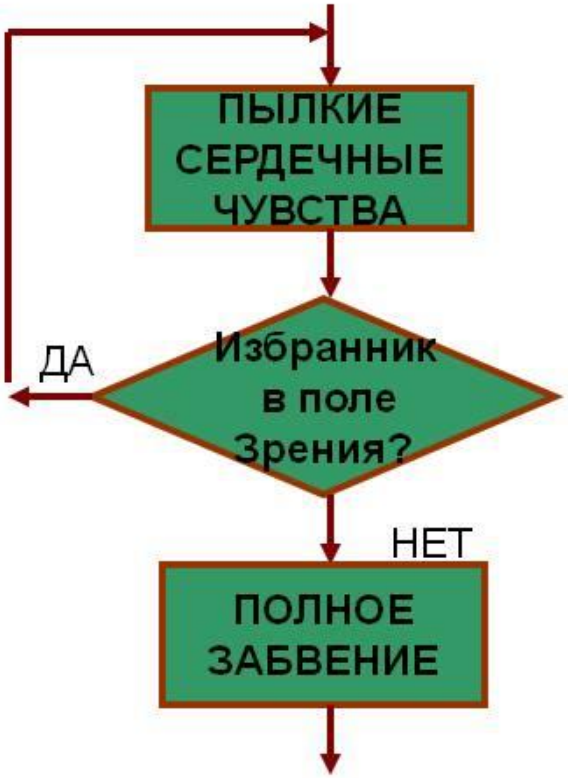
- *Имитационная алгоритмическая модель* - содержательное описание объекта в форме алгоритма, отражающее структуру и процессы функционирования объекта во времени, учитывающее воздействие случайных факторов.

БЛОК-СХЕМЫ ПОСЛОВИЦ



После дождичка

в четверг



С глаз долой –

Из сердца вон.

Гносеологическая модель – описание объективных законов природы.

Концептуальная модель описывает выявленные причинно-следственные связи и закономерности, присущие исследуемому объекту и существенные в рамках определенного исследования.

- *Сенсуальные модели (от англ. Sensor - сенсор, датчик)* – модели чувств, эмоций;
  - модели, оказывающие воздействие на чувства человека (музыка, поэзия, живопись, танец).

- *Аналоговая модель* – аналог объекта, который ведет себя как реальный объект, но не выглядит как таковой.

Моделирование

Моделирование

- *Моделирование* – это:
  - построение моделей реально существующих объектов (предметов, явлений, процессов);
  - замена реального объекта его подходящей копией;
  - исследование объектов познания на их моделях.

Объекты моделирования характеризуются свойствами:

- внешний вид - набор признаков
- структура – перечень элементов и указание отношений между ними;
- поведение – изменение внешнего вида и структуры с течением времени.

Этапы моделирования

Этапы моделирования

- 1. Постановка целей моделирования.
- 2. Анализ объекта моделирования и выделение всех его известных свойств.
- 3. Анализ выделенных свойств и определение существенных из них.

Этапы моделирования

- 4. Выбор формы представления модели.
- 5. Формализация
- 6. Анализ полученной модели на противоречивость.
- 7. Анализ адекватности полученной модели объекту и цели моделирования.

Общая схема моделирования:

Формализация

Формализация – это приведение свойств и признаков объекта моделирования к выбранной форме.

Формализация

Суть формализации состоит в принципиальной возможности изучения объекта и его обозначения.

Для того, чтобы обозначить объект, вводится некоторый набор знаков - язык.

Знак

Значение слова «Знак» по Ожегову:

Знак - Внешнее обнаружение, признак чего-л.

Знак Жест, движение которым сигнализируют, сообщают что-то.

Знак Пометка, изображение, предмет, которыми отмечается, обозначается что-л.

Знак

Основные характеристики знаков:

- Способность знака выступать в качестве заместителя денотата (объекта).
- Нетождественность знака и денотата – знак никогда не может полностью заменить обозначаемое.
- Многозначность соответствия «знак – денотат».

Знак

Значение знака (экстенсионал) — предмет, представляемый (репрезентируемый) данным знаком.

Знак

Смысл знака (интенсионал) — информация о репрезентируемом предмете, которую содержит сам знак или которая связывается с этим знаком в процессе общения или познания.

Взаимосвязь этих характеристик можно графически представить в виде семантического треугольника.

Язык

Язык – знаковая система, используемая в целях познания и коммуникации.

Языки могут быть естественными и искусственными.

Язык

Правила искусственного языка являются строго и однозначно определенными, поэтому такой язык называется формализованным.

Моделирование

Процесс формализации текстовой информации (представление информации в форме графа, чертежа, схемы и т.д.) осуществляется с целью ее однозначного понимания, облегчения и ускорения ее обработки.

Моделирование

Например, процесс формализации оформления текста заключается в использовании бланков, формуляров, шаблонов определенной и часто законодательно утвержденной формы.

Моделирование

*Например, **таблицы** – форма представления информации в удобном для анализа и обработки виде.*

Таблица характеризуется названием, количеством столбцов и их названиями, количеством строк и их названиями, содержимым ячеек.

Моделирование

Граф *– форма представления модели в виде совокупности точек, соединенных между собой линиями - вершинами графа.*

Моделирование

Линии, соединяющие вершины, называются дугами, если задано направление от одной вершины к другой, или ребрами, если направленность двусторонняя.

- Компьютерное моделирование

Компьютерное моделирование

Компьютер не «мыслит» - он способен реализовывать программы, составленные человеком.

Поэтому, чтобы использовать ВТ необходимо:

- четко поставить проблему;

Компьютерное моделирование

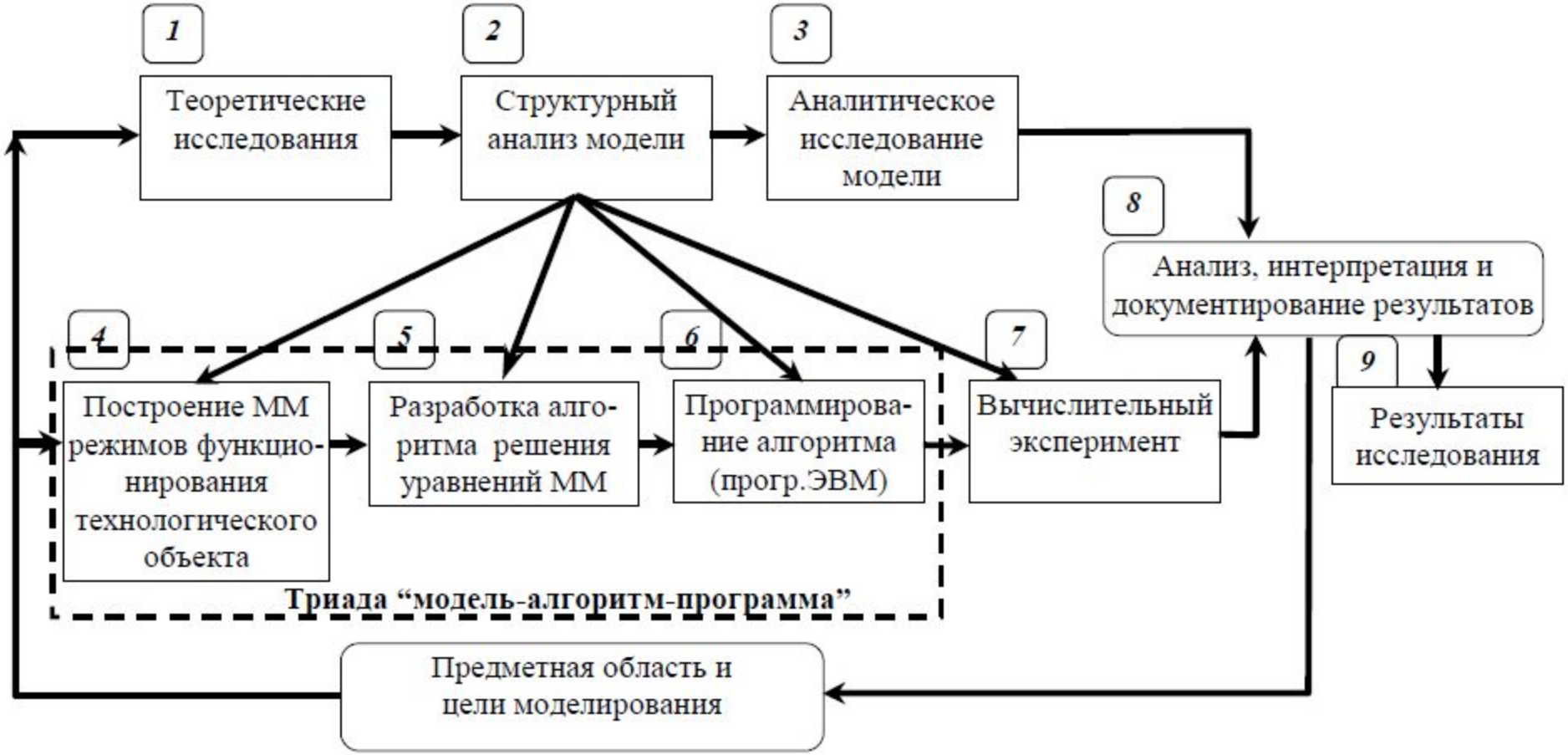
- разработать модель исходных данных;
- определить модель представления результатов;

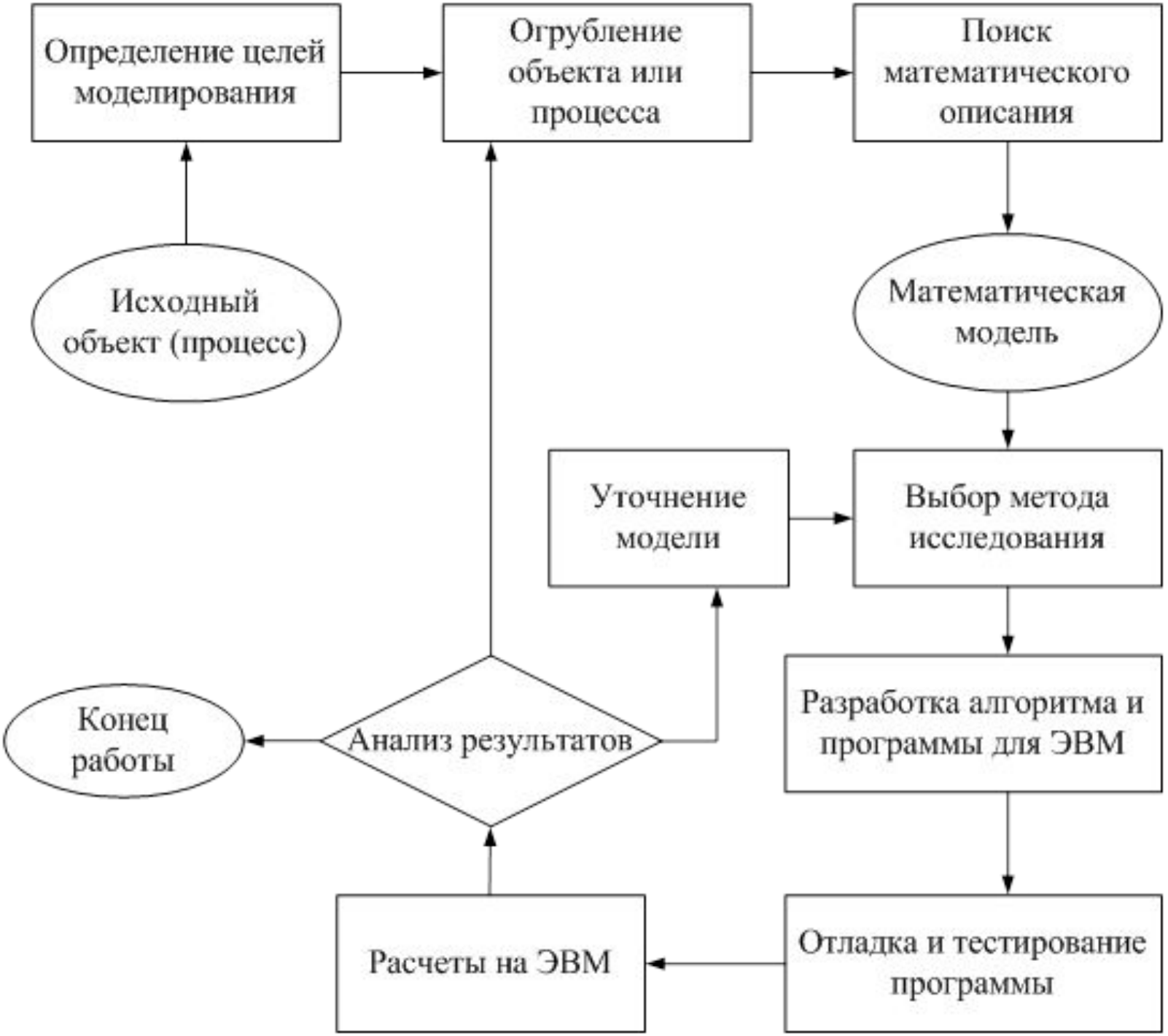
Компьютерное моделирование

- разработать алгоритм решения задачи;
- написать программу;
- ввести программу и исходные данные в память;

Компьютерное моделирование

- отладить программу, запустить на выполнение и получить результаты.
- Т.О. возникает классическая для информатики триада – **модель – алгоритм – программа.**


1
Теоретические исследования
2
Структурный анализ модели
3
Аналитическое исследование модели
8
Анализ, интерпретация и документирование результатов
4
Построение ММ режимов функционирования технологического объекта
5
Разработка алгоритма решения уравнений ММ
6
Программирование алгоритма (прогр.ЭВМ)
7
Вычислительный эксперимент
9
Результаты исследования
Триада “модель-алгоритм-программа”
Предметная область и цели моделирования


Определение целей моделирования
Огрубление объекта или процесса
Поиск математического описания
Исходный объект (процесс)
Математическая модель
Уточнение модели
Выбор метода исследования
Конец работы
Анализ результатов
Разработка алгоритма и программы для ЭВМ
Расчеты на ЭВМ
Отладка и тестирование программы

Моделирование ИС

Вспомним о CASE

Основной целью CASE-технологии является разграничение процесса проектирования ПО (ИС) от процесса кодирования и последующих этапов разработки, максимально автоматизировать процесс разработки.

Для выполнения поставленной цели CASE-технологии используют два принципиально разных подхода к проектированию: структурный и объектно-ориентированный.

В CASE-технологиях используют два принципиально разных подхода к проектированию:

1. Структурный

2. Объектно-ориентированный.

Структурный подход предполагает декомпозицию (разделение) поставленной задачи на функции, которые необходимо автоматизировать.

В свою очередь, функции также разбиваются на подфункции, задачи, процедуры.

В результате получается упорядоченная иерархия функций и передаваемой информацией между функциями.

Структурный подход подразумевает использование определённых общепринятых методологий при моделировании различных информационных систем:

- SADT (structured analysis and design technique);
- DFD (data flow diagrams);
- ERD (entity-relationship diagrams).

Основным инструментом объектно-ориентированного подхода является язык UML — унифицированный язык моделирования, который предназначен для визуализации и документирования объектно-ориентированных систем с ориентацией их на разработку программного обеспечения.

UML - язык включает в себя систему различных диаграмм, на основании которых может быть построено представление о проектируемой системе.

Классификация Современных Case средств :

1. **Классификация Case средств по поддерживаемым методологиям:**

   - **функциональные или структурно-ориентированные;**
   - **объектно-ориентированные;**
   - **комплексно-ориентированные.**

Структурный метод моделирования ИС

Структурный метод

Суть структурного метода моделирования и разработки ИС заключается в ее разбиении на отдельные функции (функциональные подсистемы).

Структурный метод

Подсистемы делятся на подфункции, подфункции делятся на задачи и т.д. до конкретных процедур.

Структурный метод

При этом система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны.

Структурный метод

Базовые принципы структурного метода:

- Принцип «разделяй и властвуй» - представление сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;

Структурный метод

- Принцип иерархического построения – организация составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне;

Структурный метод

- Принцип абстрагирования – выделение существенных проблем системы и отвлечение от несущественных;

Структурный метод

- Принцип формализации – необходимость строгого методического подхода к решению проблемы, разработка алгоритма;

Принцип формализации –

1. *разработка неформального описания модели (словесное описание существенных для рассматриваемой задачи характеристик изучаемого объекта и связей между ними);*

Принцип формализации –

2. *составление формализованного описания на некотором языке кодирования (с использованием математических соотношений и текстов);*
3. *реализация формализованного описания в виде программы на некотором языке программирования.*

*Например, формула F=m*a является формализованным описанием второго закона Ньютона.*

Структурный метод

- Принцип непротиворечивости – обоснованность и согласованность всех элементов системы.

Средства структурного метода моделирования

Структурный метод

Для структурного моделирования ИС используются различные средства (методики):

- **SADT** - диаграммы функционального моделирования;
- **DFD** – диаграммы потоков данных;
- **ERD** – диаграммы «Сущность-связь».

SADT-диаграммы функционального моделирования

SADT – методология

Методология SADT (Structured Analysis and Design Technique - методология структурного анализа и проектирования) предложена более 40 лет назад американцем Дугласом Россом, и опробована на практике в период с 1969 по 1973 г.

SADT – методология

SADT - это методология, разработанная для облегчения описания и понимания (информационной) системы средней сложности и ее среды до определения требований к различным видам обеспечения (ИС).

SADT – методология

В основе этого метода моделирования систем лежит описание системы, создаваемого с помощью естественного языка, позволяющего представить функционирование моделируемой системы.

SADT – методология

SADT-модель - это текстовое и графическое описание системы, выполненное в виде иерархически организованной совокупности диаграмм.

SADT – методология

На основе графических средств **SADT** описание системы снабжается изображением ее модели.

SADT – методология

Модель представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм, организованных в виде древовидной структуры, где верхняя диаграмма является наиболее общей, а самые нижние наиболее детализированы.

SADT – методология

В основе SADT лежат два основных принципа:

1. Создание SA-блоков, на основе которых создается иерархическая многоуровневая модульная система, каждый уровень которой представляет собой законченную систему (блок), поддерживаемую и контролируемую системой (блоком), находящейся над ней.

SADT – методология

2. Декомпозиция. Каждый блок, понимаемый как единое целое, разделяется на свои составляющие, описываемые на более детальной диаграмме.

Процесс декомпозиции проводится до достижения нужного уровня подробности описания.

SADT – методология

Диаграмма ограничивается 3-6 блоками для постепенной детализации.

SADT – методология

Вместо одной громоздкой модели используется несколько небольших взаимосвязанных моделей, значения которых взаимно дополняют друг друга, делая понятной структуризацию сложного объекта.

SADT – методология

Т.о. одна из наиболее важных особенностей методологии – постепенное введение всё больших уровней детализации по мере создания диаграмм, отображающих модель.

SADT – методология

Построение модели начинается с изображения всей системы в виде простейшей компоненты.

SADT – методология

Диаграммы первого уровня представляют важнейшие подсистемы с их взаимосвязями, а диаграммы самого нижнего уровня представляют детализированные функции, с помощью которых работает система.

Диаграммы законченной SADT-модели описывают все важные компоненты и детали системы.

SADT – методология

Применение SADT методологии предполагает разбиение на фазы:

- анализ - определение того, что система будет делать;
- проектирование - определение подсистем и их взаимодействие;
- реализация - разработка подсистем по отдельности;

SADT – методология

- объединение - соединение подсистем в единое целое;
- тестирование - проверка работы системы;
- установка - введение системы в действие;
- функционирование - использование системы.

SADT – методология

Обычно SADT - методология применяется на ранних этапах жизненного цикла информационной системы.

SADT – методология

SADT - модель может быть сосредоточена либо на функциях системы, либо на ее объектах.

SADT – методология

SADT-модели, ориентированные на функции, принято называть функциональными моделями, а ориентированные на объекты системы – моделями данных.

SADT – методология

В результате строится модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.

SADT – методология

Диаграммы - главные компоненты модели, все функции ИС и интерфейсы на них представлены как блоки и дуги.

SADT – методология

Место соединения дуги с блоком определяет тип интерфейса.

SADT – методология

Управляющая информация входит в блок сверху; информация, которая подвергается обработке, размещается с левой стороны блока, результаты выхода - с правой стороны.

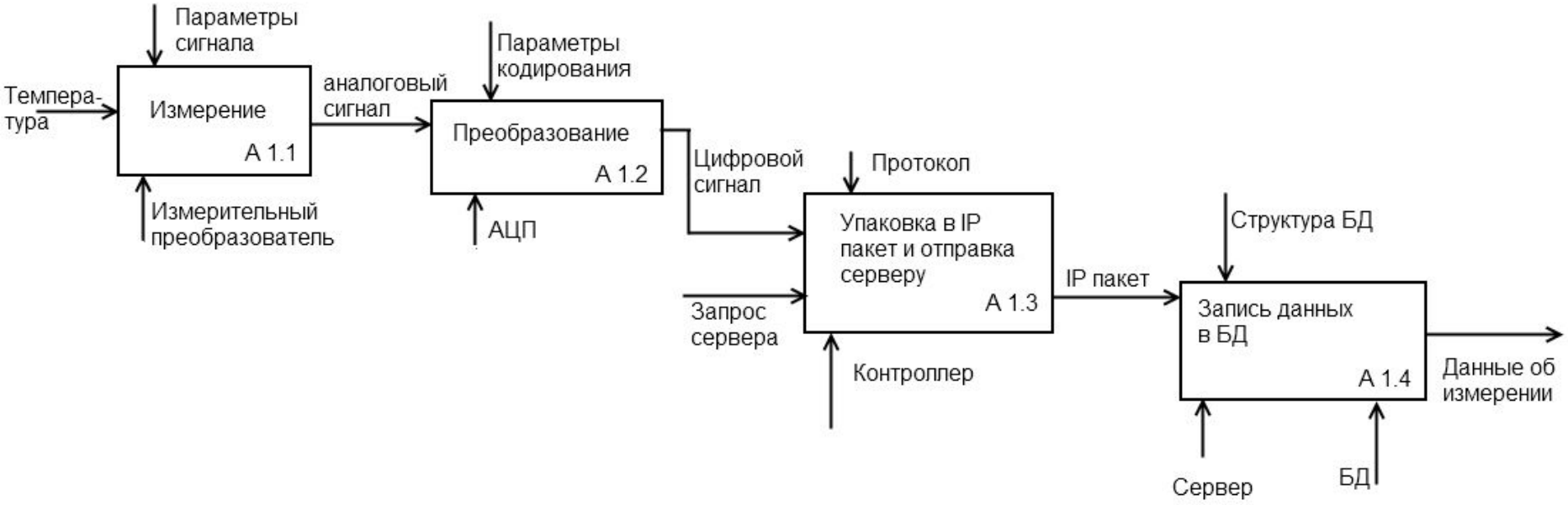
SADT – методология

Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу.

Рис. 3.8. Функциональный блок SADT-диаграммы

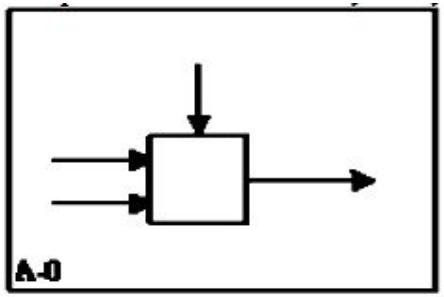
Функциональный блок и интерфейсные дуги

Процесс оформления кредита
Заявка от клиента на выдачу кредита
Процесс оформления кредита
Кредит
Служащий банка



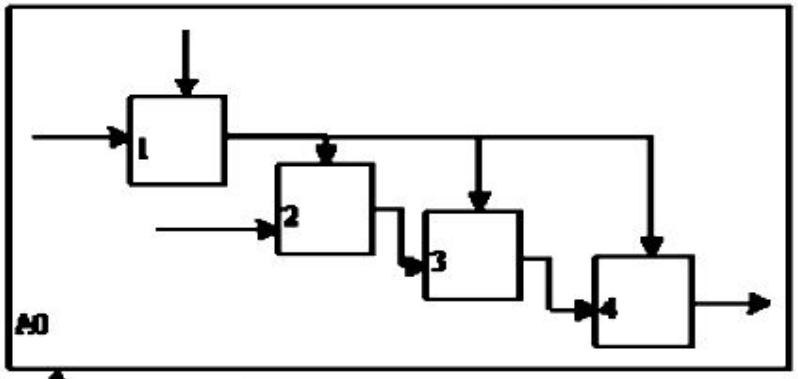
SADT-диаграмма «измерение температуры»

SADT – методология

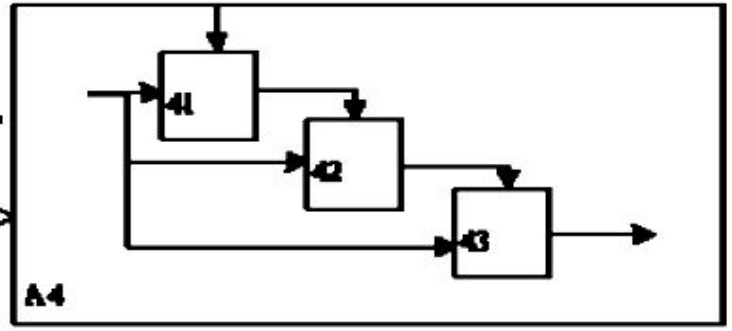


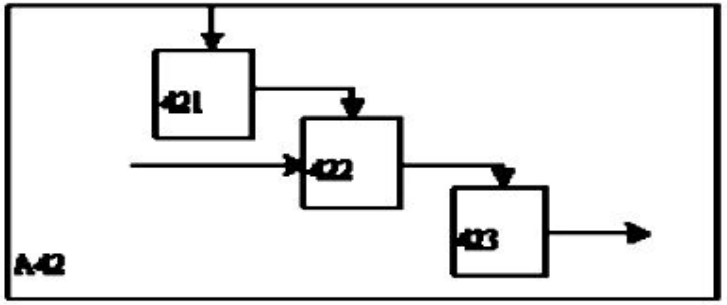
Более общее представление

Более детальное представление

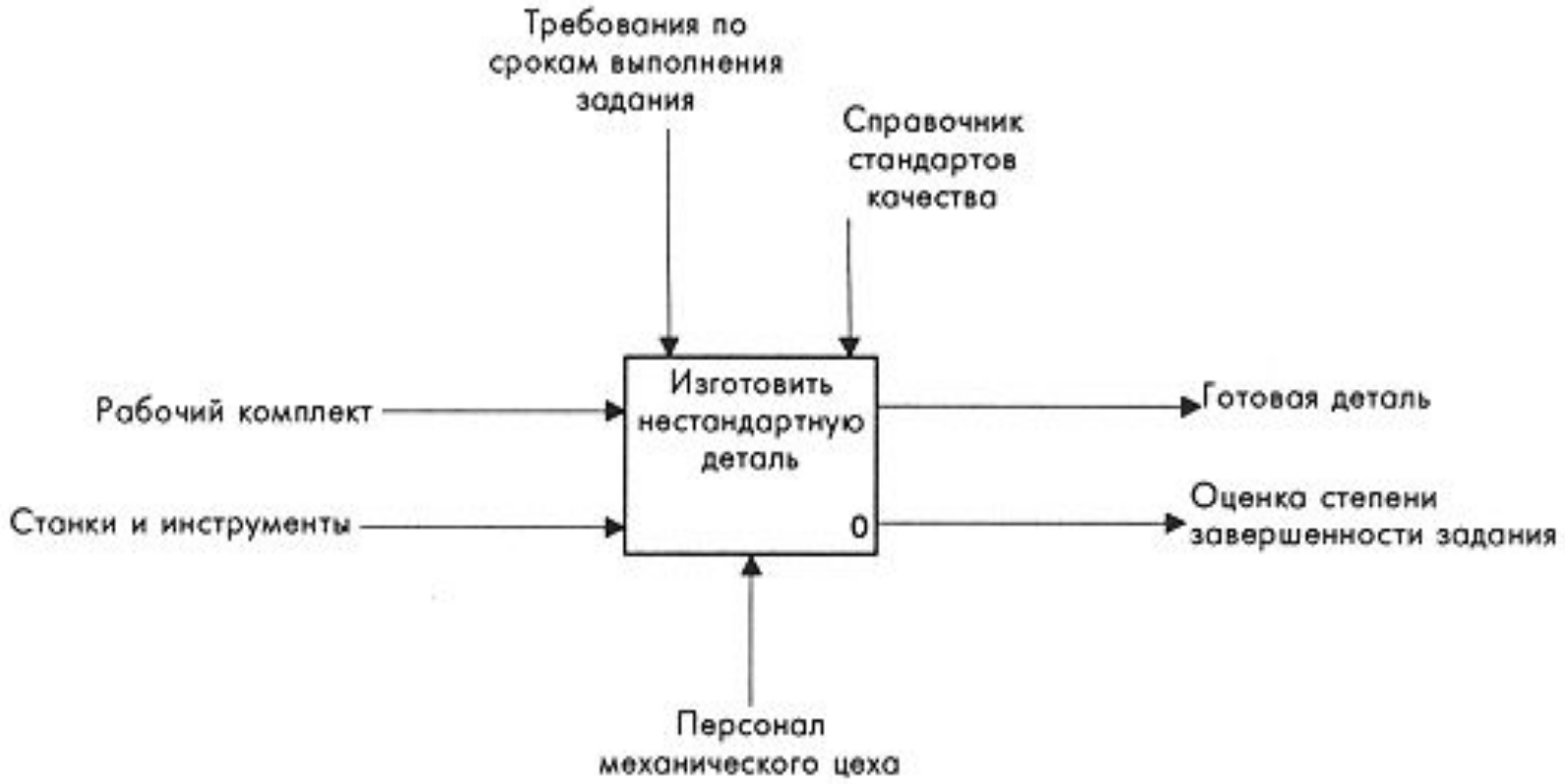


Эта диаграмма является "родителем" этой диаграммы





ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 03/16/92 ПЕРЕСМОТР:	✔ РАБОЧАЯ ВЕРСИЯ ЭСКИЗ РЕКОМЕНДОВАНО ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ ДАТА	КОНТЕКСТ: Top

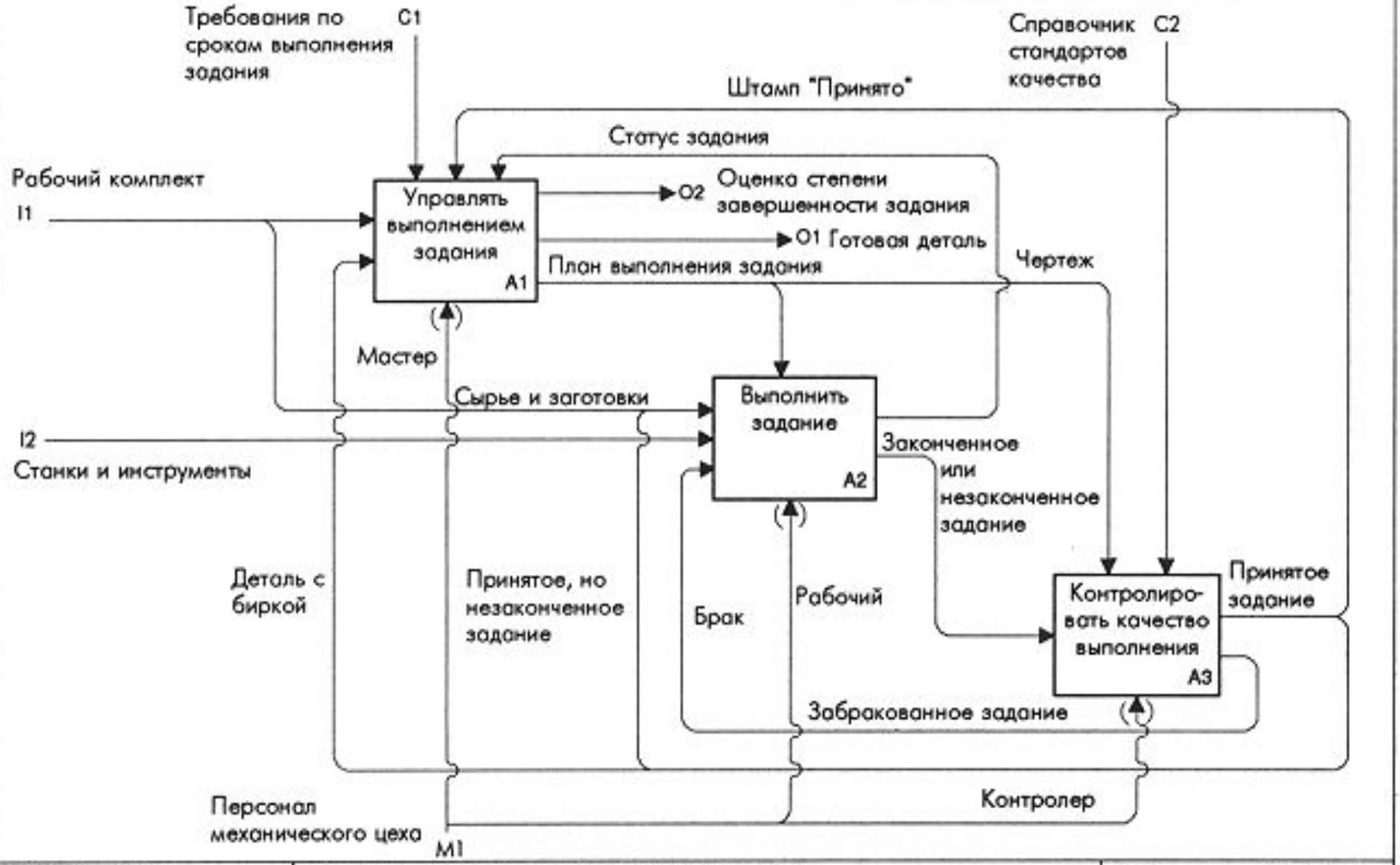


Цель: Понять, какие функции должны быть включены в процесс изготовления нестандартной детали и как эти функции взаимосвязаны между собой с тем, чтобы написать учебное пособие для персонала механического цеха.

Точка зрения: Начальника цеха

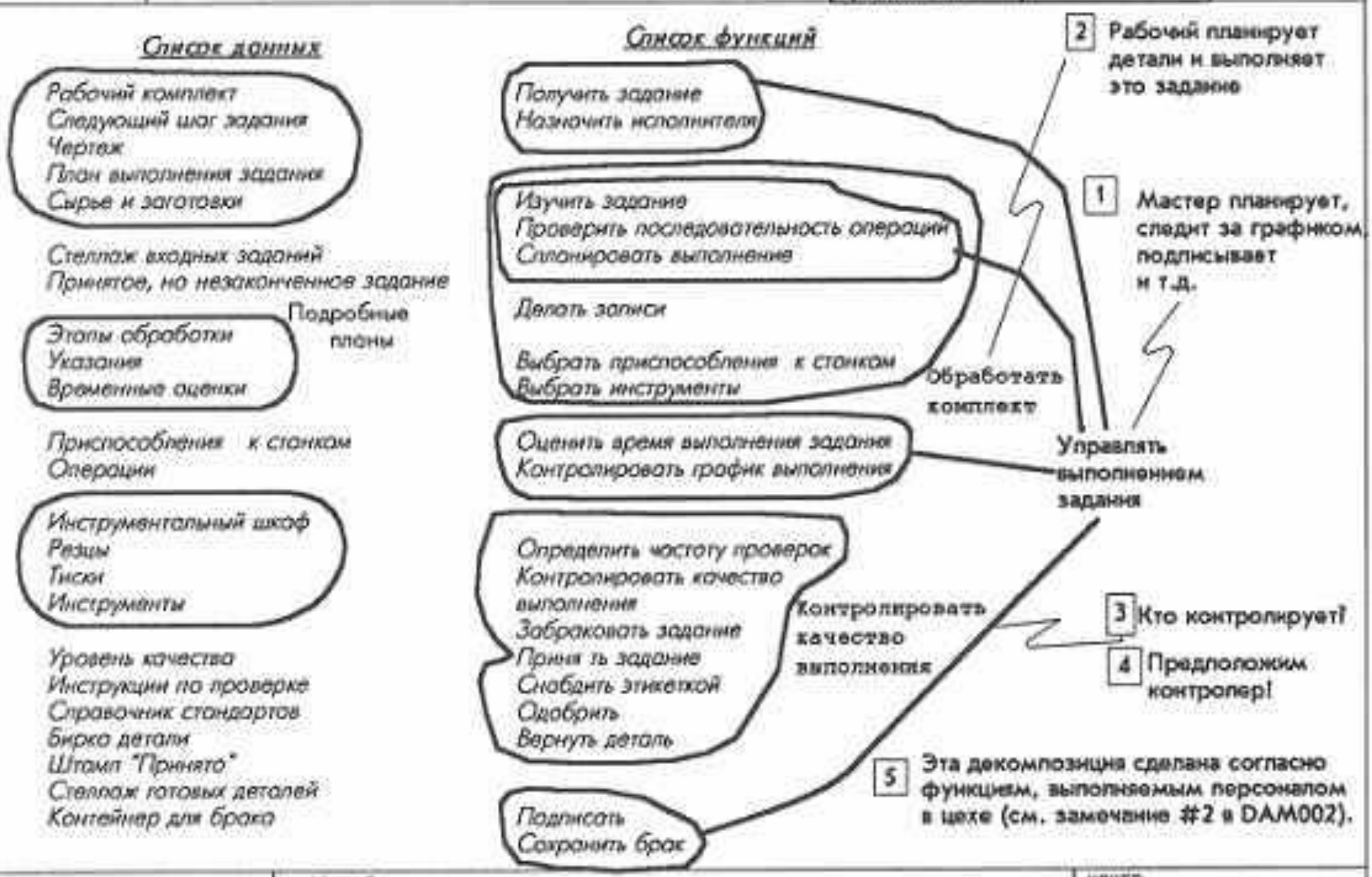
УЗЕЛ: ЭМЦ/ A-0	НАЗВАНИЕ: Изготовить нестандартную деталь	НОМЕР: (DAM004) DAM005

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 03/16/92 ПЕРЕСМОТР:	✔ РАБОЧАЯ ВЕРСИЯ / ЭСКИЗ / РЕКОМЕНДОВАНО / ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ ДАТА	КОНТЕКСТ: ■



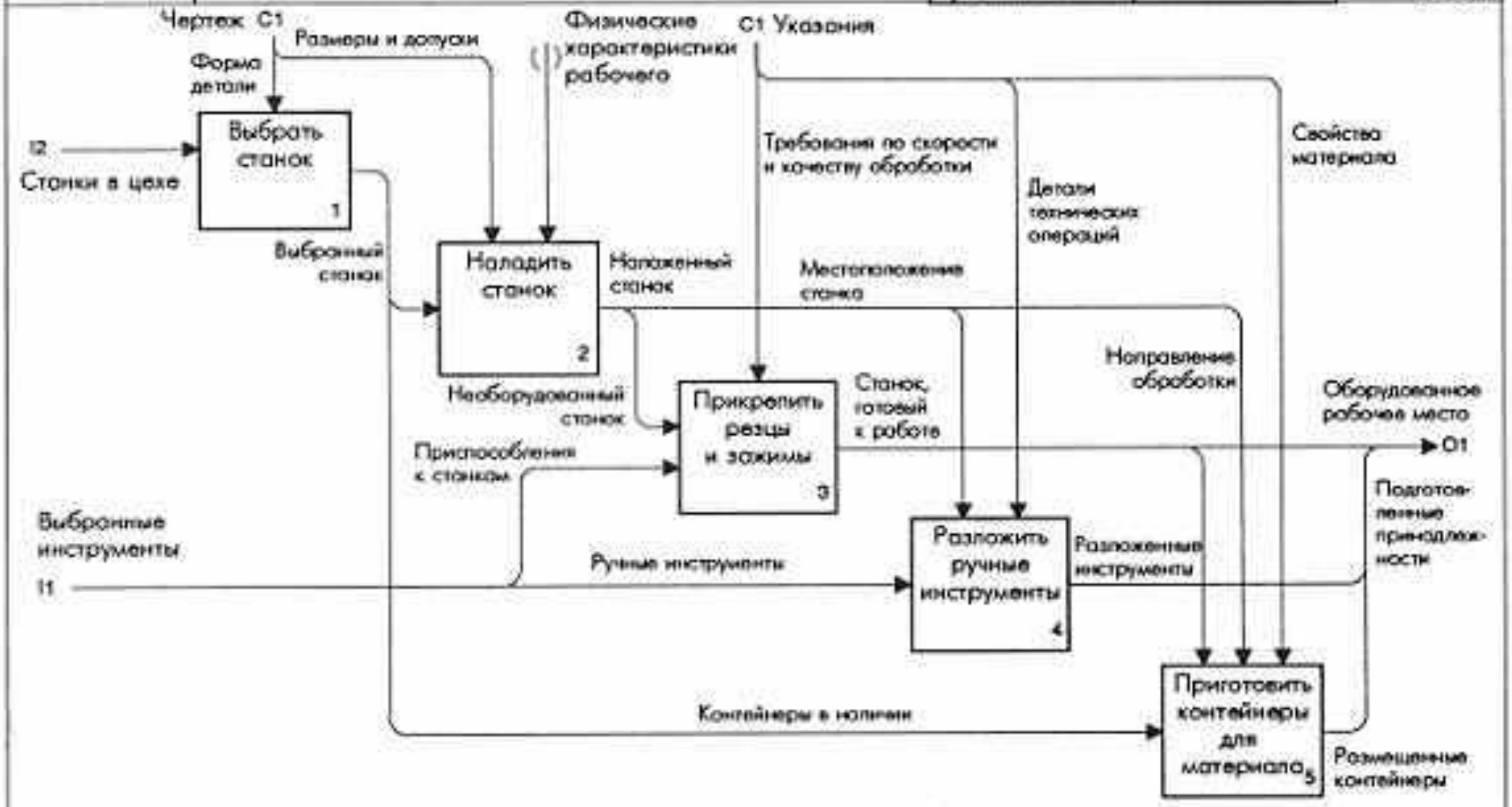
УЗЕЛ: ЭМЦ/ A0	НАЗВАНИЕ: Изготовить нестандартную деталь	НОМЕР: (DAM003) DAM006

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 02/26/93 ПЕРЕСМОТР:	✓ РАБОЧАЯ ВЕРСИЯ / ЧИТАТЕЛЬ / ДАТА ЭСКИЗ РЕКОМЕНДОВАНО ПУБЛИКАЦИЯ	КОНТЕКСТ: Top



УЗЕЛ: ЭМЦ/А-0	НАЗВАНИЕ: Изготовить нестандартную деталь	НОМЕР: DAM001

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 02/18/93 ПЕРЕСМОТР:	РАБОЧАЯ ВЕРСИЯ / ЧИТАТЕЛЬ / ДАТА ЭСКИЗ РЕКОМЕНДОВАНО ПУБЛИКАЦИЯ	КОНТЕКСТ: DAM015



УЗЕЛ: ЭМЦ/A23	НАЗВАНИЕ: Подготовить рабочее место	НОМЕР: DAM016

Форма детали – общий вид детали.
Размеры и допуски – величина детали и значения допустимых отклонений от размеров.
Физические характеристики рабочего – все характеристики рабочего, которые влияют на его способность правильно расположить и наладить станок
Требования по скорости и качеству обработки – критические требования к обработке на станке, влияющие на выбор резцов и зажимов.
Детали технических операций – точная последовательность операций при обработке.
Свойства материалов – свойства обрабатываемого материала, влияющие на способы обработки детали.
Выбранный станок – станок, подходящий для выполнения следующего шага задания.
Налаженный станок – станок, который может использовать оператор.
Необорудованный станок – станок, годный к употреблению, но без приспособлений.
Приспособления к станку – приспособления, необходимые для обработки детали.
Ручные инструменты – то, что оператор держит в руках во время обработки детали.
Контейнеры в наличии – контейнеры для материала и для брака.
Местоположение станка – место в цехе, где стоит станок.
Станок, готовый к работе – станок, готовый к обрабтке детали.

A231 Выбрать станок

Осмотреть форму детали и определить тип станка, наиболее подходящего для обработки деталей такой формы.

A232 Наладить станок

Осмотреть расположение станка и его высоту, опустить или поднять станок так, чтобы было удобно работать. Затем развернуть его так, чтобы загрузка, обработка и выгрузка детали была удобна и безопасна.

A233 Прикрепить резцы и зажимы

В зависимости от типа необходимой обработки прикрепить к станку необходимые зажимы и резцы. Будьте особенно внимательны к требуемым скоростям и качеству обработки.

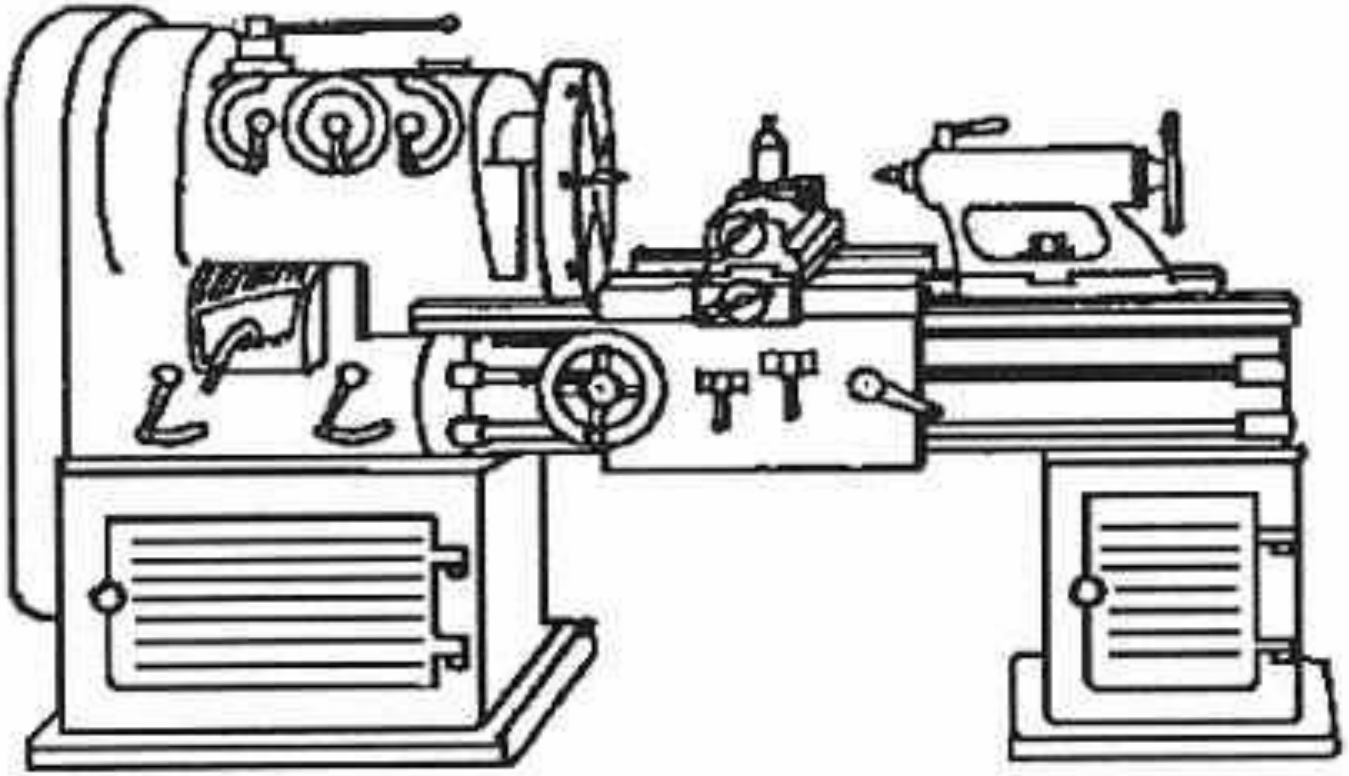
A234 Разложить инструменты

Просмотрите свои записи по обработке на станке, чтобы определить, какие конкретные операции нужно сделать. Разложите инструменты вокруг станка в том же порядке, в каком идут эти операции. Учтите расположение станка, чтобы инструменты были всегда под рукой.

A235 Подготовить контейнеры для материала и брака

Расположите контейнеры так, чтобы в них было легко сгружать материал и собирать отходы. Учтите свойства материала и направления резки, чтобы определить куда будут направляться отходы.

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 06/05/93 ПЕРЕСМОТР:	✔ РАБОЧАЯ ВЕРСИЯ	ЧИТАТЕЛЬ ДАТА	КОНТЕКСТ: Top
			ЭСКИЗ		
			РЕКОМЕНДОВАНО		
			ПУБЛИКАЦИЯ		



УЗЕЛ: ЭМЦ/FEO	НАЗВАНИЕ: Станок, готовый к работе	НОМЕР:

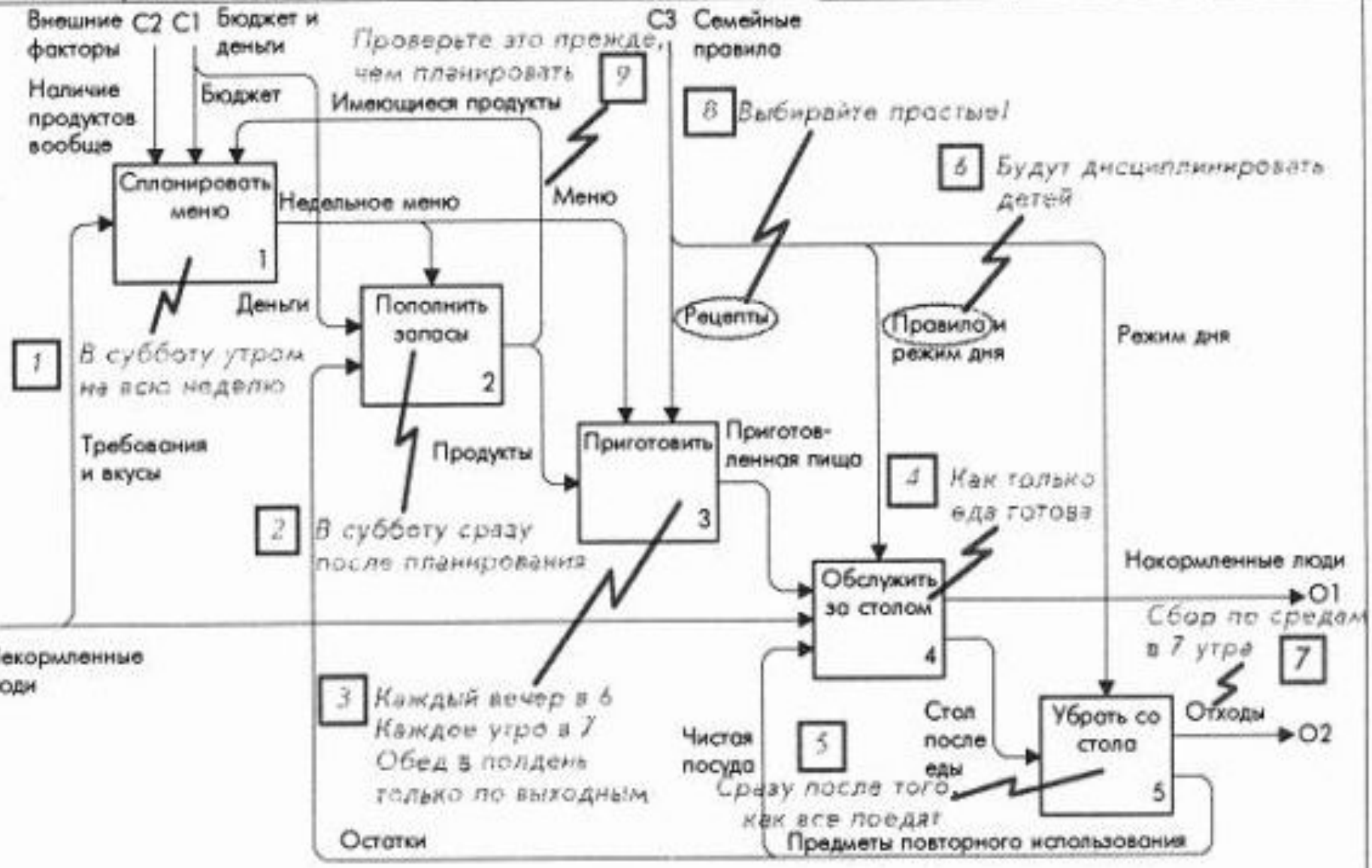
ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: Питание семьи ЗАМЕЧАНИЯ: 6 7 8 9 10	ДАТА: 1/18/93 ПЕРЕСМОТР:	✔ РАБОЧАЯ ВЕРСИЯ / ЭСКИЗ / РЕКОМЕНДОВАНО / ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ: Top

Цель: Определить действия, необходимые для организации питания семьи, чтобы написать инструкции, которым подросток сможет следовать самостоятельно

Точка зрения: Родителей

УЗЕЛ: ПС/А–0	НАЗВАНИЕ: Организовать питание семьи	НОМЕР: DAM005 (DAM004)

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Marca ПРОЕКТ: Питание семьи ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 01/29/93 ПЕРЕСМОТР:	✔ РАБОЧАЯ ВЕРСИЯ ✔ ЭСКИЗ РЕКОМЕНДОВАНО ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ ДАТА	КОНТЕКСТ:



УЗЕЛ: ПС/А0	НАЗВАНИЕ: Организовать питание семьи	НОМЕР: DAM011 (DAM006)

SADT

Преимущества SADT:

- Отражаются системные характеристики: управление, обратная связь и исполнители.

SADT

- Развитые процедуры поддержки коллективной работы;

SADT

- Применение на ранних стадиях создания системы;
- SADT можно сочетать с другими структурными методами.

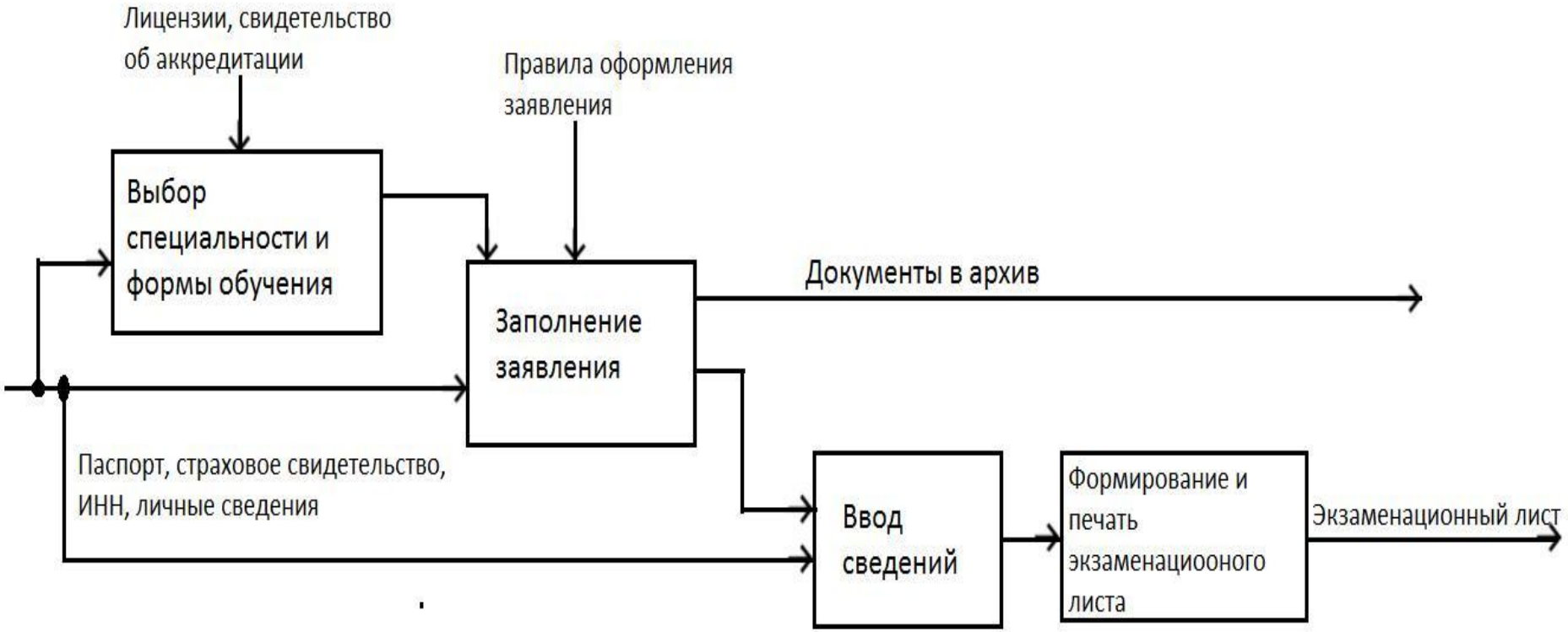
Схемы SADT связывают воедино различные методы, примененные для описания определенных частей системы с различным уровнем детализации.

SADT

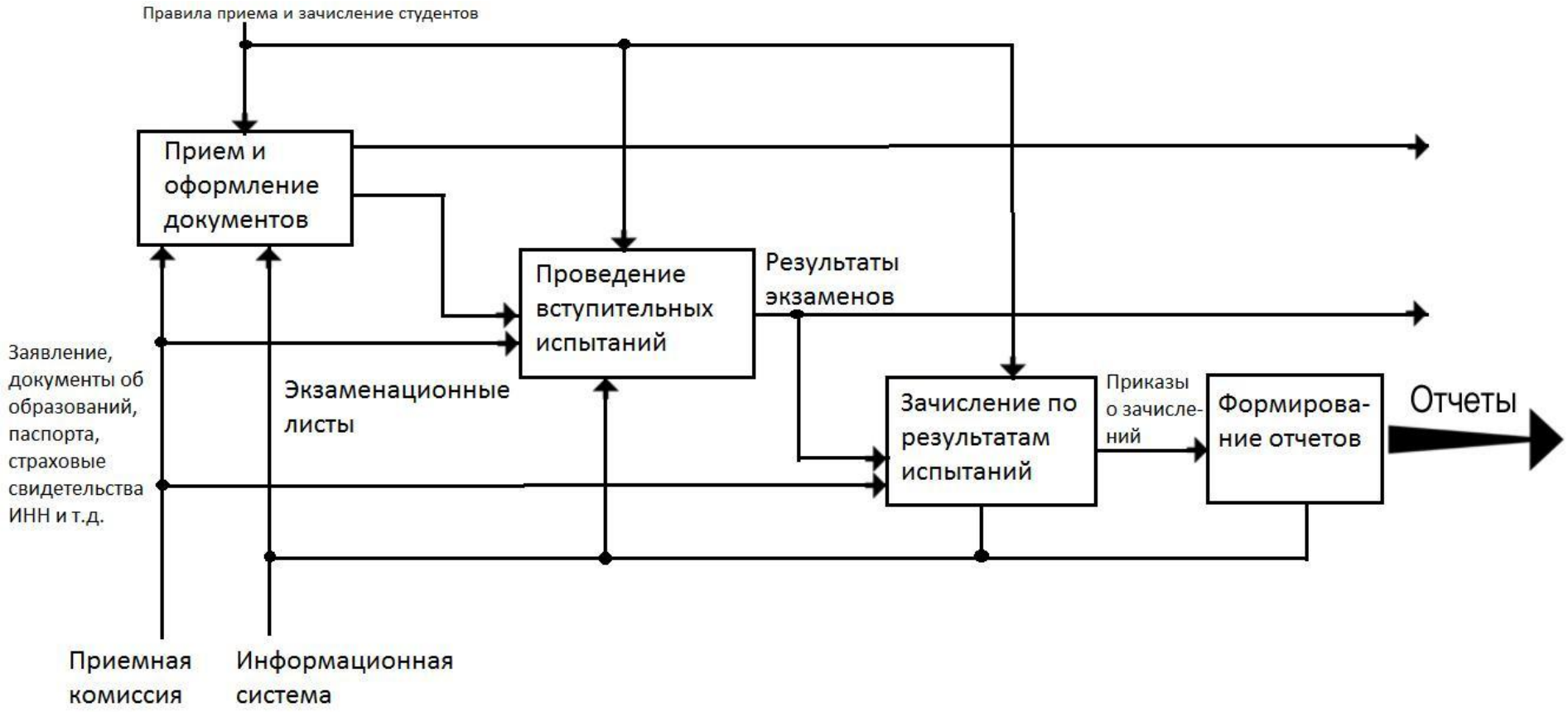
В программе интегрированной компьютеризации производства (ICAM) Мин. обороны США была признана полезность SADT, что привело в 1993 году к стандартизации ее в качестве федерального стандарта в США, а в 2000 году - в качестве руководящего документа по стандартизации в Российской Федерации.

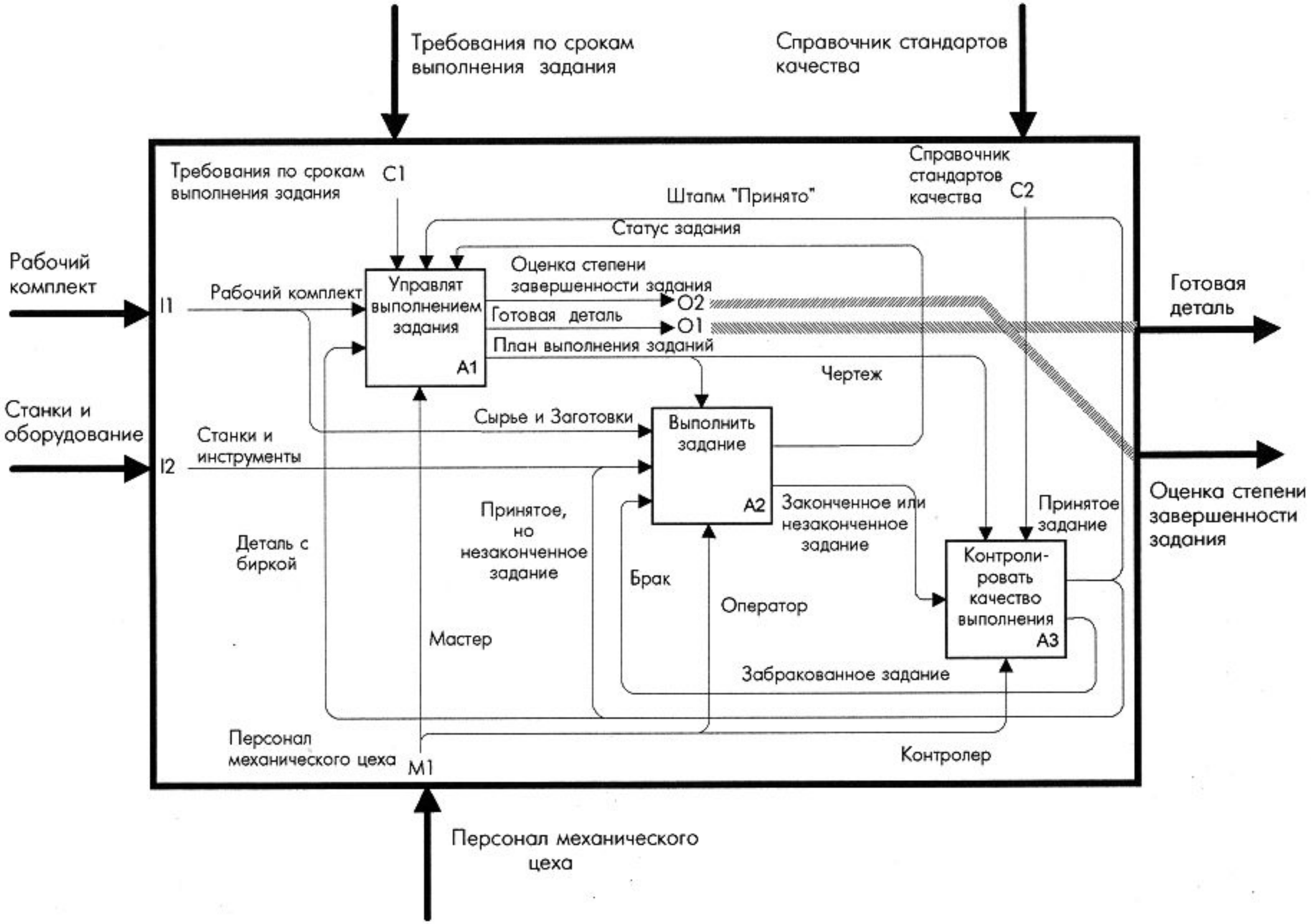
SADT

SADT стандартизована под названием **IDEF0**.


Лицензии, свидетельство об аккредитации
Правила оформления заявления
Выбор специальности и формы обучения
Заполнение заявления
Документы в архив
Паспорт, страховое свидетельство, ИНН, личные сведения
Ввод сведений
Формирование и печать экзаменациооного листа
Экзаменационный лист

Правила приема и зачисление студентов
Заявление, документы об образовании, паспорта, страховые свидетельства, ИНН и т.д.
Правила и зачисление абитуриентов
Отчеты по приему и зачислению студентов
Информационная система
Приемная комиссия


Правила приема и зачисление студентов
Прием и оформление документов
Проведение вступительных испытаний
Результаты экзаменов
Заявление, документы об образований, паспорта, страховые свидетельства ИНН и т.д.
Экзаменационные листы
Зачисление по результатам испытаний
Приказы о зачисле-ний
Формирова-ние отчетов
Отчеты
Приемная комиссия
Информационная система


Требования по срокам выполнения задания
Справочник стандартов качества
Требования по срокам выполнения задания
C1
Справочник стандартов качества
C2
Штапм "Принято"
Статус задания
Рабочий комплект
I1
Рабочий комплект
Управлят выполнением задания
A1
Оценка степени завершенности задания
O2
Готовая деталь
O1
План выполнения заданий
Готовая деталь
Чертеж
Станки и оборудование
I2
Станки и инструменты
Сырье и Заготовки
Выполнить задание
A2
Принятое, но незаконченное задание
Законченное или незаконченное задание
Принятое задание
Оценка степени завершенности задания
Деталь с биркой
Брак
Оператор
Контроли-ровать качество выполнения
A3
Мастер
Забракованное задание
Персонал механического цеха
M1
Контролер
Персонал механического цеха

DFD – диаграммы потоков данных

DFD

DFD — сокращение *Data Flow Diagrams* — диаграммы потоков данных.

DFD

DFD — один из основных инструментов *структурного* анализа и проектирования ИС.

DFD

В современных условиях произошёл переход от структурного к объектно-ориентированному подходу анализа и проектирования систем, но «старинные» структурные нотации по-прежнему используются в бизнес-анализе, и в моделировани ИС.

DFD

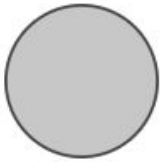
DFD – это нотация, предназначенная для моделирования информационный систем с точки зрения хранения, обработки и передачи данных.

DFD

Для DFD используются две нотации — Йодана (Yourdon) и Гейна-Сэрсона (Gane-Sarson).

Далее в примерах используется нотация Гейна-Сэрсона.

Эти нотации незначительно отличаются друг от друга графическим изображением символов .

Нотация	Юрдан и Коад	Гейн и Сарсон
Внешняя сущность		
Процесс		
Хранилище данных		
Поток данных		

DFD

Т.е. нотация должна наглядно ответить на вопросы:

- Из чего состоит информационная система?
- Что нужно, чтобы обработать информацию?

Синтаксис DFD

Основными компонентами DFD являются:

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

Синтаксис DFD

Внешние сущности (англ. External Entity) - любые объекты, которые не входят в саму систему, но являются для нее источником или приемником информации из системы после обработки данных.

Это может быть человек, внешняя система, какие-либо носители информации и хранилища данных.

Синтаксис DFD

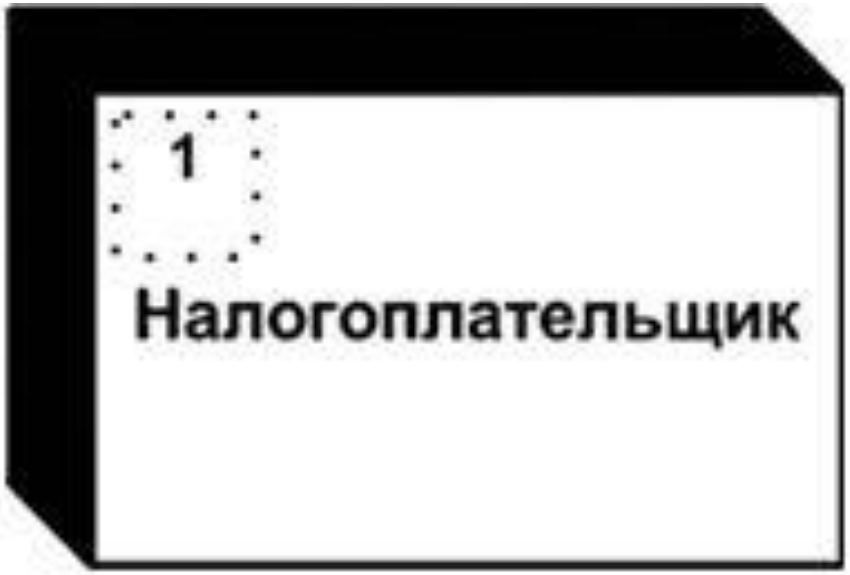
Определение объекта *или системы* в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы.

Синтаксис DFD

При моделировании некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой системы или, наоборот, часть процессов может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

Синтаксис DFD

Внешняя сущность обозначается квадратом «с тенью».



Синтаксис DFD

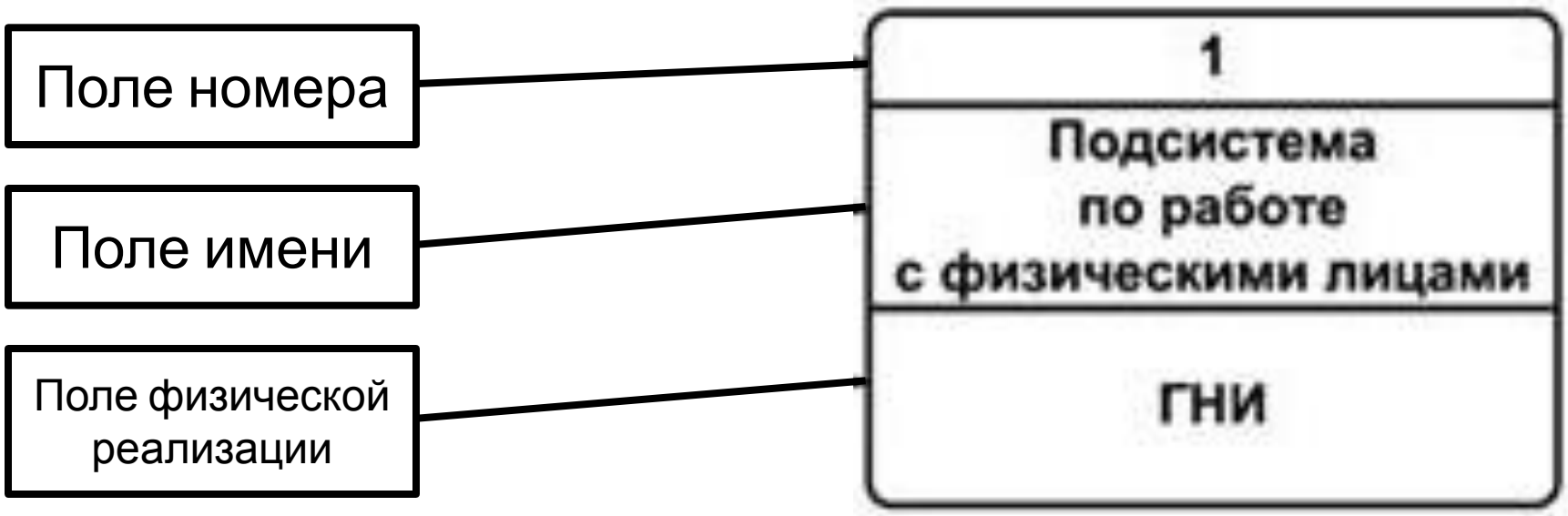
Процессы представляют собой функции системы (подсистемы), преобразующие входы в выходы.

Номер подсистемы служит для ее идентификации.

Синтаксис DFD

В поле имени процесса вводится наименование подсистемы в виде предложения с подлежащим, определениями и дополнениями, т.е. принято использовать глаголы, т.е. «Создать клиента» (а не «создание клиента») или «обработать заказ» (а не «проведение заказа»).

Синтаксис DFD



Пример: подсистема по работе с физическими лицами (*ГНИ — Государственная налоговая инспекция*)

Синтаксис DFD

Хранилище (накопитель) данных (англ. Data store) — некое абстрактное устройство для хранения информации, которую можно в любой поместить в накопитель или извлечь, *причем способы помещения и извлечения могут быть любыми*.

Синтаксис DFD

Это базы данных, таблицы или любой другой вариант организации и хранения данных.

Здесь будут храниться данные о клиентах, заявки клиентов, расходные накладные и любые другие данные, которые поступили в систему или являются результатом обработки процессов.

Синтаксис DFD

Накопитель данных может быть реализован физически в виде ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д.

Синтаксис DFD

Поток данных определяет информацию, передаваемую *через некоторое соединение* от источника к приемнику.

Синтаксис DFD

Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д.

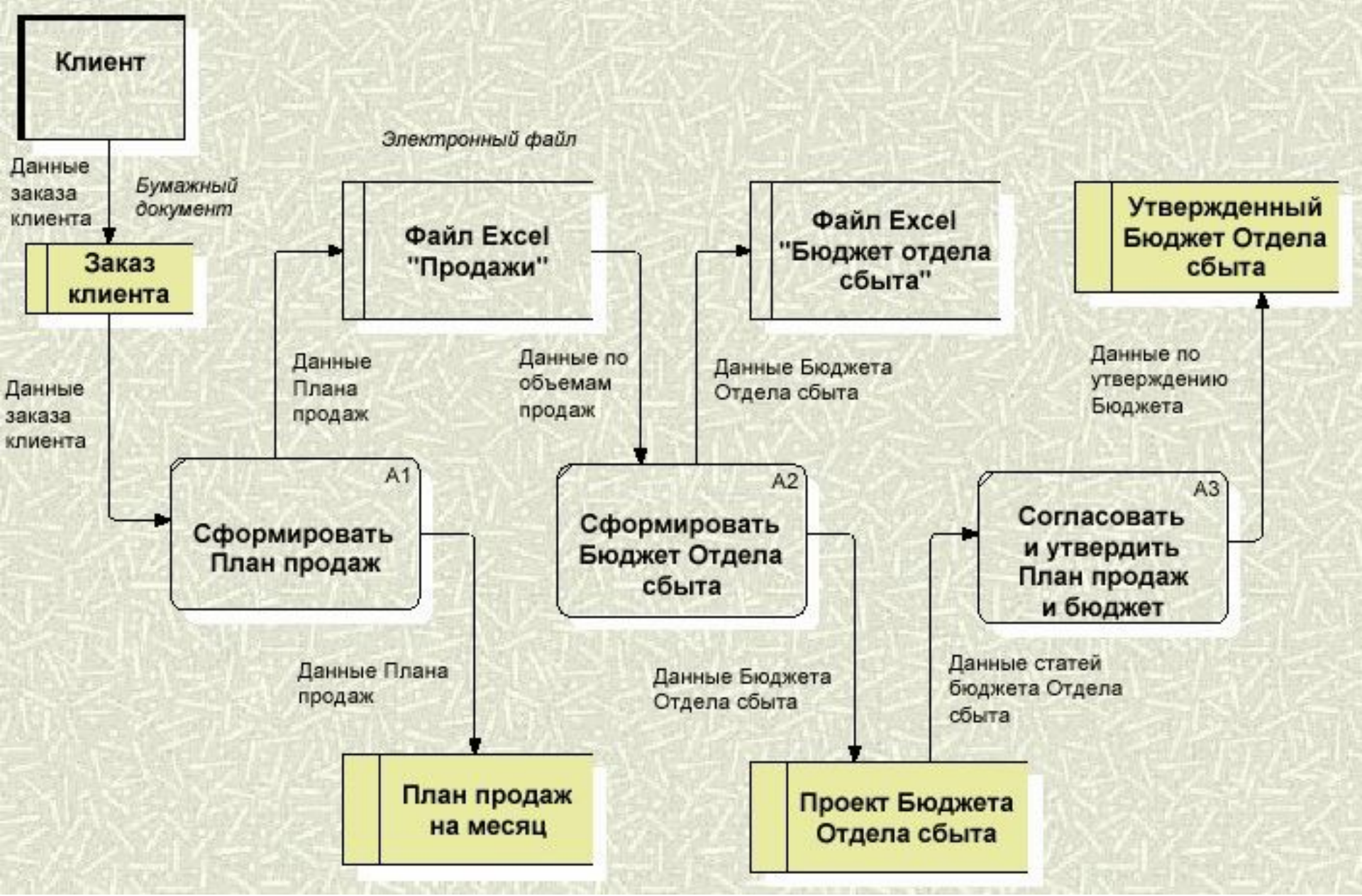
Синтаксис DFD

Поток данных изображается линией, оканчивающейся стрелкой, которая показывает направление потока и имеет имя, отражающее его содержание.

Синтаксис DFD

Правила определяют как следует применять компоненты.

Диаграммы обеспечивают формат графического и словесного описания модели.



DFD-диаграмма ИС «продажи»

Построение модели

Главная цель построения модели DFD - сделать описание системы ясным и понятным на каждом уровне детализации, а также разбить его на части с точно определенными отношениями между ними.

Построение модели

Для построения модели рекомендуется:

- Размещать на диаграмме 3 - 7 процессов (аналогично SADT).

Верхняя граница (6-7) соответствует человеческим возможностям одновременного восприятия и понимания структуры сложной системы с множеством внутренних связей, нижняя граница (3) выбрана по соображениям здравого смысла: нет необходимости детализировать процесс диаграммой, содержащей всего один или два процесса.

Построение модели

- Не загромождать диаграммы несущественными на данном уровне деталями.

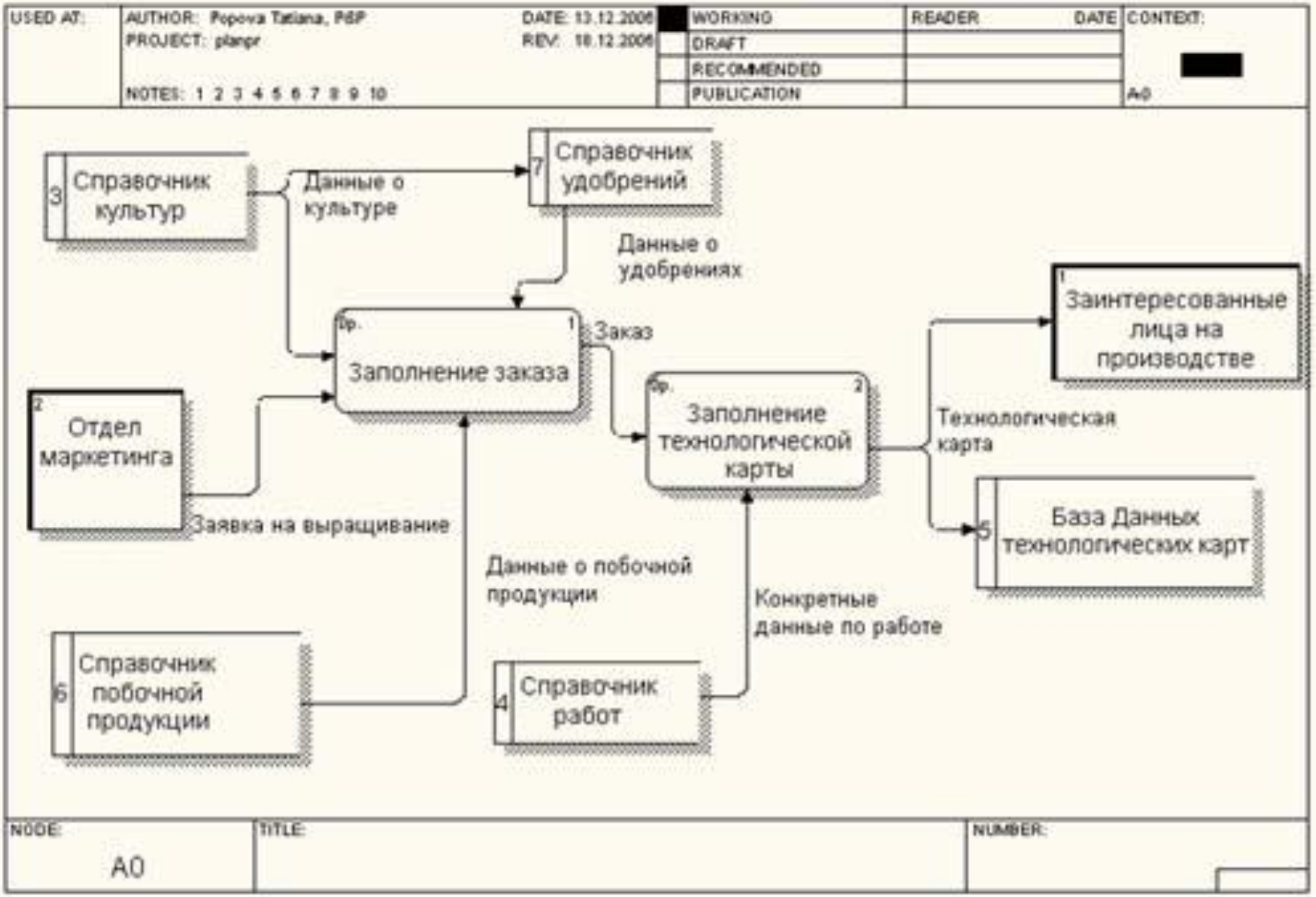
Построение модели

- Декомпозицию потоков данных осуществлять параллельно с декомпозицией процессов, декомпозиции должны выполняться одновременно, а не одна после завершения другой.

Построение модели

- Выбирать ясные, отражающие суть дела имена процессов и потоков, при этом стараться не использовать аббревиатуры.

DFD-диаграмма «Увольнение» ИС «КАДРЫ»



Производство удобрений

Как создавать нотации DFD

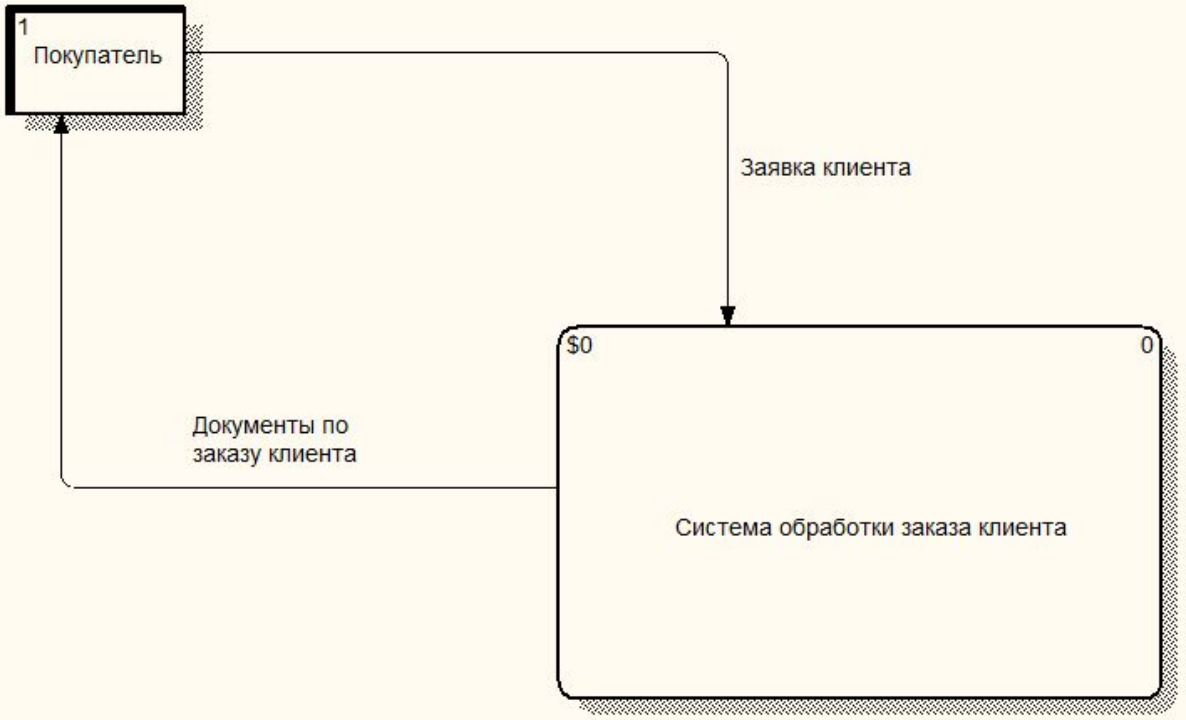
Допустим, у нас есть клиент, который делает заявку.

Менеджер регистрирует эту заявку.

Т.о., в системе появляются данные – клиент и его заказ.

Работник склада должен это увидеть и произвести отгрузку товара с оформлением всех необходимых документов и передать документы клиенту.

USED AT:	AUTHOR: Кинзябулатов Рамиль	DATE: 10/8/2017		WORKING	READER DATE	CONTEXT:
	PROJECT: Продажа товара	REV: 10/9/2017		DRAFT		TOP
				RECOMMENDED		
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION		



NODE:	TITLE:	NUMBER:
A-0	Система обработки заказа клиента	4

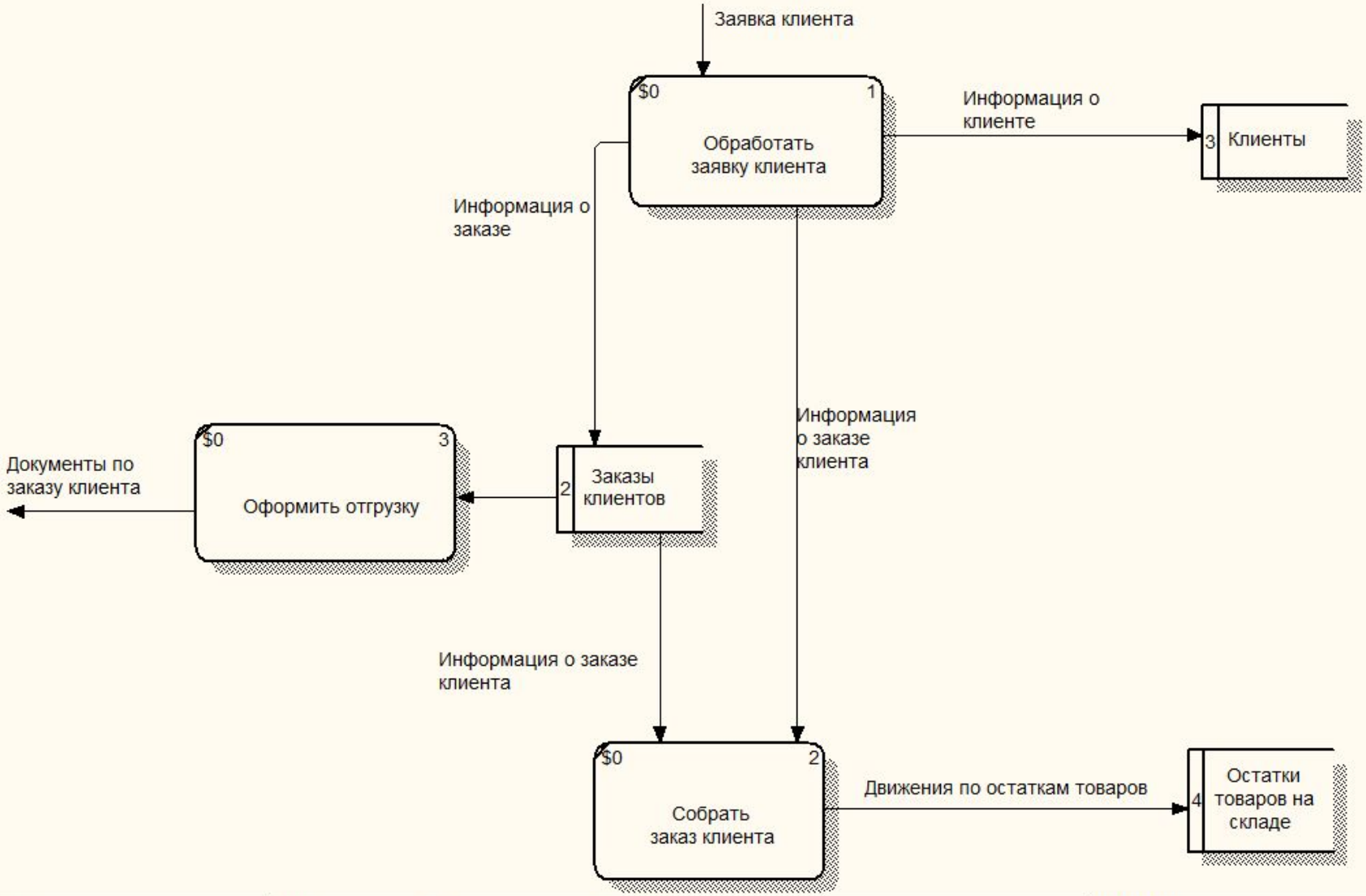
1. Клиент предоставляет свои данные и заявку.
2. Менеджер проверяет и вносит полученные данные в систему.
3. Работник склада формирует документы, например, расходную накладную, и отгружает товар.
4. Клиент получает товар и пакет документов к нему.

С точки зрения DFD у нас имеются:

- Покупатель – это внешняя сущность, которая является источником данных и получением результата.
- Процесс обработки заказа (подтверждение и проводка данных в системе менеджером).
- Сбор заказа на складе (после получения заявки).
- Оформление отгрузки (создание необходимых документов).

Декомпозиция основного элемента нашей диаграммы:

USED AT: Для статьи	AUTHOR: Кинзябулатов Рамиль	DATE: 10/8/2017		WORKING	READER DATE	CONTEXT:
	PROJECT: Продажа товара	REV: 10/9/2017		DRAFT		
			■	RECOMMENDED		■
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION		A-0



NODE:	TITLE:	NUMBER:
A0	Система обработки заказа клиента	4

Критерии оценки модели DFD

Критерии оценки DFD

Завершением моделирования считается выполнение следующих критериев:

- наличия у процесса небольшого количества входных и выходных потоков данных (2-3 потока);

Критерии оценки DFD

- возможности описания процессов в виде последовательного алгоритма;

Критерии оценки DFD

- выполнения процессом единственной логической функции преобразования входной информации в выходную;

Критерии оценки DFD

- возможности описания логики процесса при помощи спецификации небольшого объема (не более 20-30 строк).

Преимущества DFD

Диаграммы DFD обеспечивают удобный способ описания передаваемой информации как между частями моделируемой системы, так и между системой и внешним миром.

Преимущества DFD

Поэтому область применения DFD - создание моделей информационного обмена организации, например, модели документооборота.

На диаграмме видно, где хранятся данные, каким образом производится обмен документацией, где в этом процессе допущены ошибки организации бизнес-процессов и пр.

Т.е. как вспомогательный вариант, в том числе, для наглядной демонстрации клиенту существующих проблем и методов оптимизации работы.

ERD – диаграммы «Сущность-связь»

ERD

В реальном проектировании ИС применяется метод, называемый ***семантическое моделирование***.

ERD

Семантическое моделирование представляет собой моделирование структуры данных, опираясь на смысл этих данных.

ERD

Инструмент семантического моделирования – ***диаграммы сущность-связь, ER*** – *(Entity-Relationship*).

Эти диаграммы часто используются для проектирования реляционных баз данных.

ERD

Нотация была введена Ченом (Chen) в 1976 г. и получила дальнейшее развитие в работах Баркера (Barker). Нотация Чена предоставляет богатый набор средств моделирования данных, включая собственно ERD, а также диаграммы атрибутов и диаграммы декомпозиции.

Элементы ERD - диаграмм

Основные понятия ER-диаграмм (в нотации Баркера)

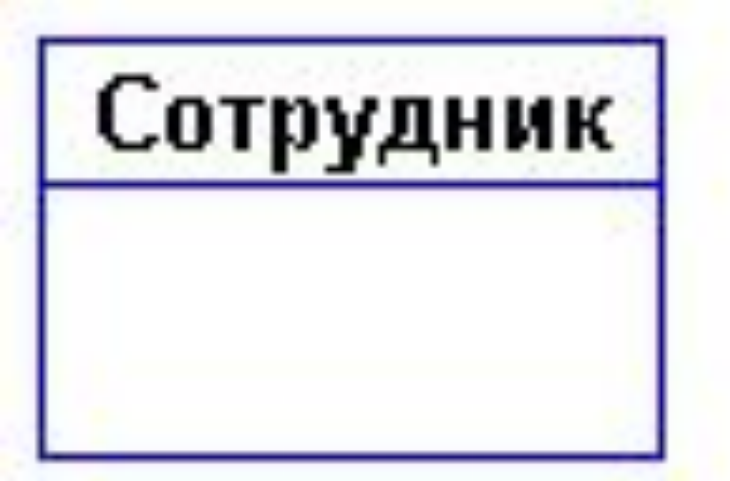
Сущность - это класс однотипных объектов, которая имеет наименование, выраженное существительным в единственном числе.

Примерами сущностей могут быть такие классы объектов как "Поставщик", "Сотрудник", "Накладная".

Сущность — это реальный, либо воображаемый объект, информацию о котором необходимо хранить в базе данных.

ER-диаграммы

Сущность в модели изображается в виде прямоугольника с наименованием.



ER-диаграммы

Экземпляр сущности - это конкретный представитель данной сущности.

Например, представителем сущности "Сотрудник" может быть "Сотрудник Иванов".

ER-диаграммы

Экземпляры сущностей должны быть *различимы,* т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

ER-диаграммы

Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности.

ER-диаграммы

Наименование атрибута должно быть выражено существительным в единственном числе (возможны дополнительные прилагательные).

ER-диаграммы

Например, у сущности "Сотрудник" могут быть такие атрибуты как "Табельный номер", "Фамилия", "Имя", "Отчество", "Должность", "Зарплата" и т. п.

Атрибуты изображаются в пределах прямоугольника, определяющего сущность:

ER-диаграммы

Ключ сущности - это *неизбыточный* набор атрибутов, значения которых в совокупности являются *уникальными* для каждого экземпляра сущности.

Неизбыточность ключа заключается в том, что удаление любого атрибута нарушает его уникальность.

ER-диаграммы

Ключевые атрибуты обозначаются на диаграмме подчеркиванием:

ER-диаграммы

Связь - это некоторая ассоциация между *двумя* сущностями.

Одна сущность может быть связана с другой сущностью или сама с собою.

ER-диаграммы

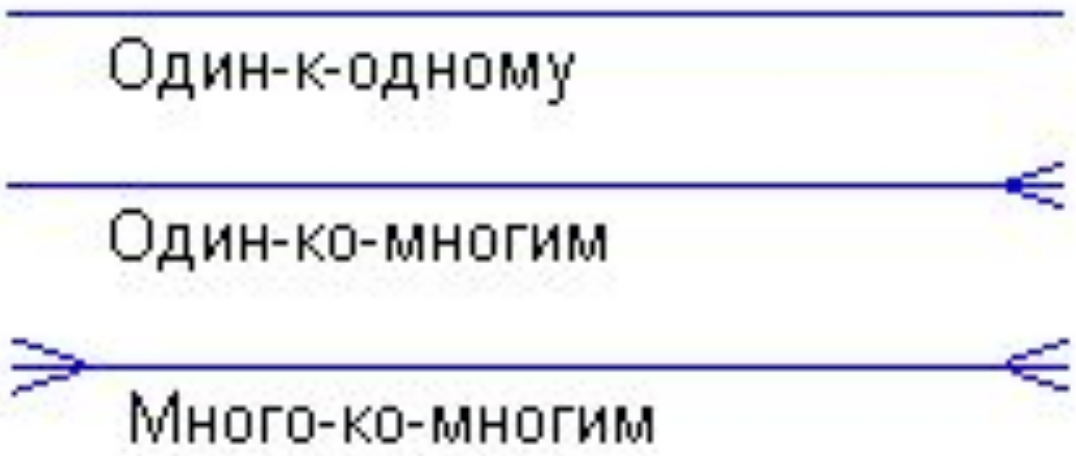
Например, связи между сущностями могут выражаться следующими фразами - "СОТРУДНИК может иметь несколько ДЕТЕЙ", "каждый СОТРУДНИК обязан числиться ровно в одном ОТДЕЛЕ".

ER-диаграммы

Графически связь изображается линией, соединяющей две сущности:

ER-диаграммы

Каждая связь может иметь один из следующих ***типов связи***:



ER-диаграммы

Связь типа ***один-к-одному*** означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой).

ER-диаграммы

Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

ER-диаграммы

Связь типа ***один-ко-многим*** означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой).

ER-диаграммы

Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется ***родительской***, правая (со стороны "много") - ***дочерней***.

ER-диаграммы

Связь типа ***много-ко-многим*** означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности.

ER-диаграммы

Тип связи много-ко-многим является *временным* типом связи, допустимым на ранних этапах разработки модели.

В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

ER-диаграммы

Каждая связь может иметь одну из двух *<u>модальностей связи</u>*:

ER-диаграммы

Модальность "***может***" означает, что экземпляр одной сущности *может быть связан* с одним или несколькими экземплярами другой сущности, *а может быть и не связан* ни с одним экземпляром.

ER-диаграммы

Модальность "***должен***" означает, что экземпляр одной сущности *обязан быть связан не менее чем с одним* экземпляром другой сущности.

Связь может иметь *разную модальность* с разных концов.

ER-диаграммы

Описанный графический синтаксис позволяет *однозначно* читать диаграммы, пользуясь следующей схемой построения фраз:

<Каждый экземпляр СУЩНОСТИ 1> <МОДАЛЬНОСТЬ СВЯЗИ> <НАИМЕНОВАНИЕ СВЯЗИ> <ТИП СВЯЗИ> <экземпляр СУЩНОСТИ 2>

Следующая диаграмма читается слева направо: "каждый сотрудник может иметь несколько детей".

Справа налево: "Каждый ребенок обязан (должен) принадлежать ровно одному сотруднику".

- ER-модель реляционной базы данных с приведением связей к виду «один-ко-многим» называется второй нормальной формой.

Пример разработки простой ER-модели

Простая ER-модель

При разработке ER-моделей требуется получить следующую информацию о предметной области:

- Список сущностей предметной области.
- Список атрибутов сущностей.
- Описание взаимосвязей между сущностями.

Другими словами

выявить:

1) Сущности, о которых хранятся данные в организации (предприятии), например, люди, места, идеи, события и т.д., (будут представлены в виде блоков);

2) Свойства этих сущностей (будут представлены в виде имен атрибутов в этих блоках.

3) Связи между этими сущностями (будут представлены в виде линий, соединяющих эти блоки).

Простая ER-модель

Например, существует задача разработать ИС заказов для оптовой торговой фирмы.

Простая ER-модель

Сначало необходимо изучить предметную область и процессы, происходящие в ней.

Для этого опрашиваются сотрудники фирмы, читается документация, изучаются формы заказов, накладных и т.п.

Простая ER-модель

Например, выяснилось, что проектируемая ИС должна выполнять следующие действия:

- Хранить информацию о покупателях.
- Печатать накладные на отпущенные товары.
- Следить за наличием товаров на складе.

Простая ER-модель

Все существительные в этих предложениях будут потенциальными кандидаты на сущности и атрибуты:

Простая ER-модель

- *Покупатель* - явный кандидат на сущность.
- *Накладная* - явный кандидат на сущность.
- *Товар* - явный кандидат на сущность

Простая ER-модель

- *(?)Склад* - а вообще, сколько складов имеет фирма? Если несколько, то это будет кандидатом на новую сущность.
- *(?)Наличие товара* – это, скорее всего, атрибут, но атрибут какой сущности?

Возникает связь между сущностями - "покупатели могут покупать много товаров" и «товары могут продаваться многим покупателям»:

Простая ER-модель

Добавляем фирме несколько складов, *каждый товар может храниться на нескольких складах и быть проданным с любого склада*.

Добавляем в диаграмму сущность «Накладная».

Связываем сущности "Накладная" и "Склад" "Покупатель" и "Товар»:

Простая ER-модель

Покупатели покупают товары, получая при этом накладные, в которые внесены данные о количестве и цене купленного товара.

Каждый покупатель может получить несколько накладных.

Каждая накладная обязана выписываться на одного покупателя.

Каждая накладная обязана содержать несколько товаров (не бывает пустых накладных).

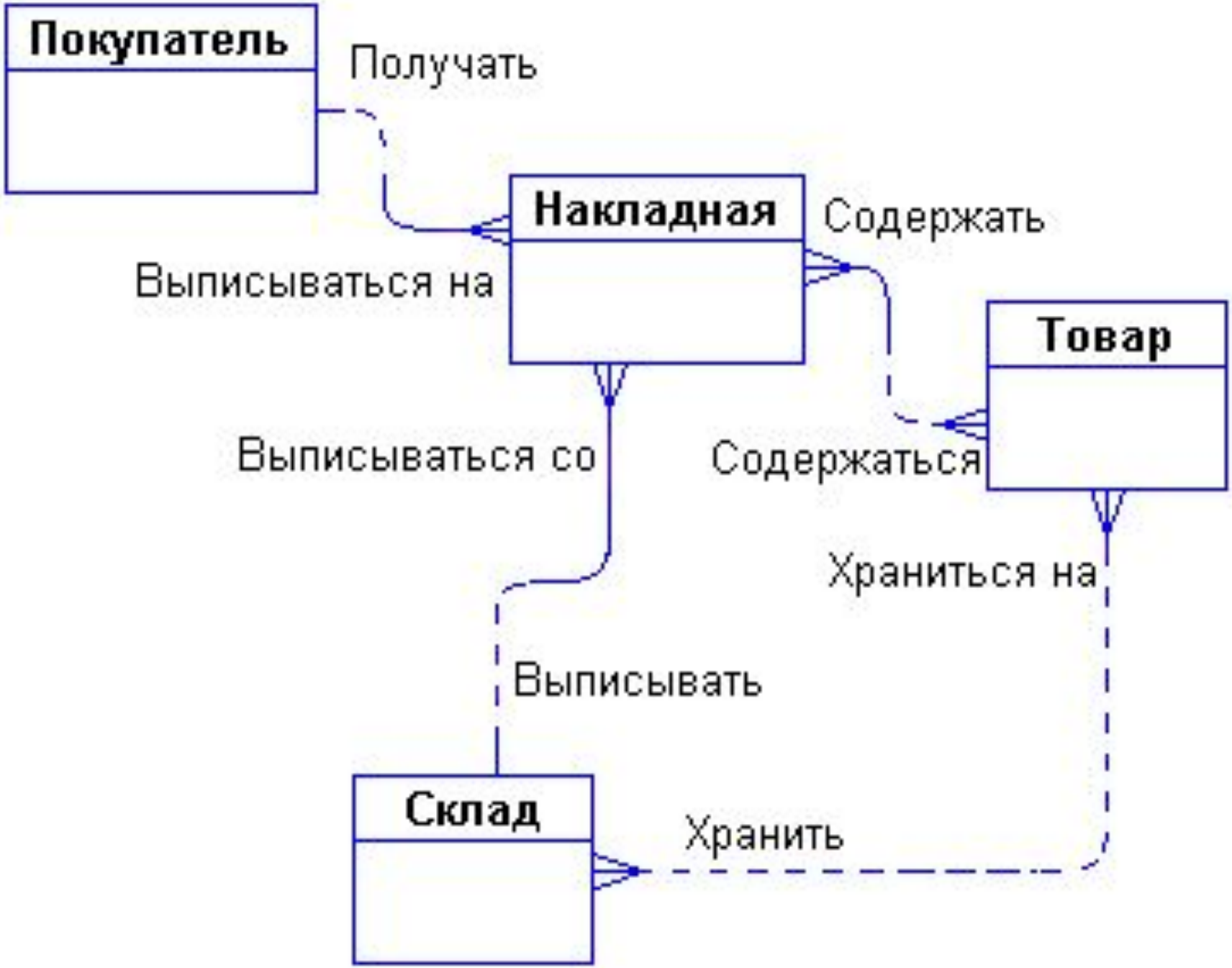
Простая ER-модель

Каждый товар, в свою очередь, может быть продан нескольким покупателям через несколько накладных.

Кроме того, каждая накладная должна быть выписана с определенного склада, и с любого склада может быть выписано много накладных.

Простая ER-модель

Таким образом, после уточнения, диаграмма будет выглядеть следующим образом:


Покупатель
Получать
Накладная
Содержать
Выписываться на
Товар
Выписываться со
Содержаться
Храниться на
Выписывать
Склад
Хранить

Простая ER-модель

После дополнительного анализа добавляем атрибуты сущностей:

- Каждый покупатель является юридическим лицом и имеет наименование, адрес, банковские реквизиты.

Простая ER-модель

- Каждый товар имеет наименование, цену, а также характеризуется единицами измерения.

Простая ER-модель

- Каждая накладная имеет уникальный номер, дату выписки, список товаров с количествами и ценами, а также общую сумму накладной.

Простая ER-модель

- Накладная выписывается с определенного склада и на определенного покупателя.
- Каждый склад имеет свое наименование.

  И т.д.

Простая ER-модель

Сущности "Накладная" и "Товар" связаны друг с другом отношением типа много-ко-многим. Такая связь должна быть расщеплена на две связи типа один-ко-многим.

Для этого требуется дополнительная сущность - "Список товаров в накладной".

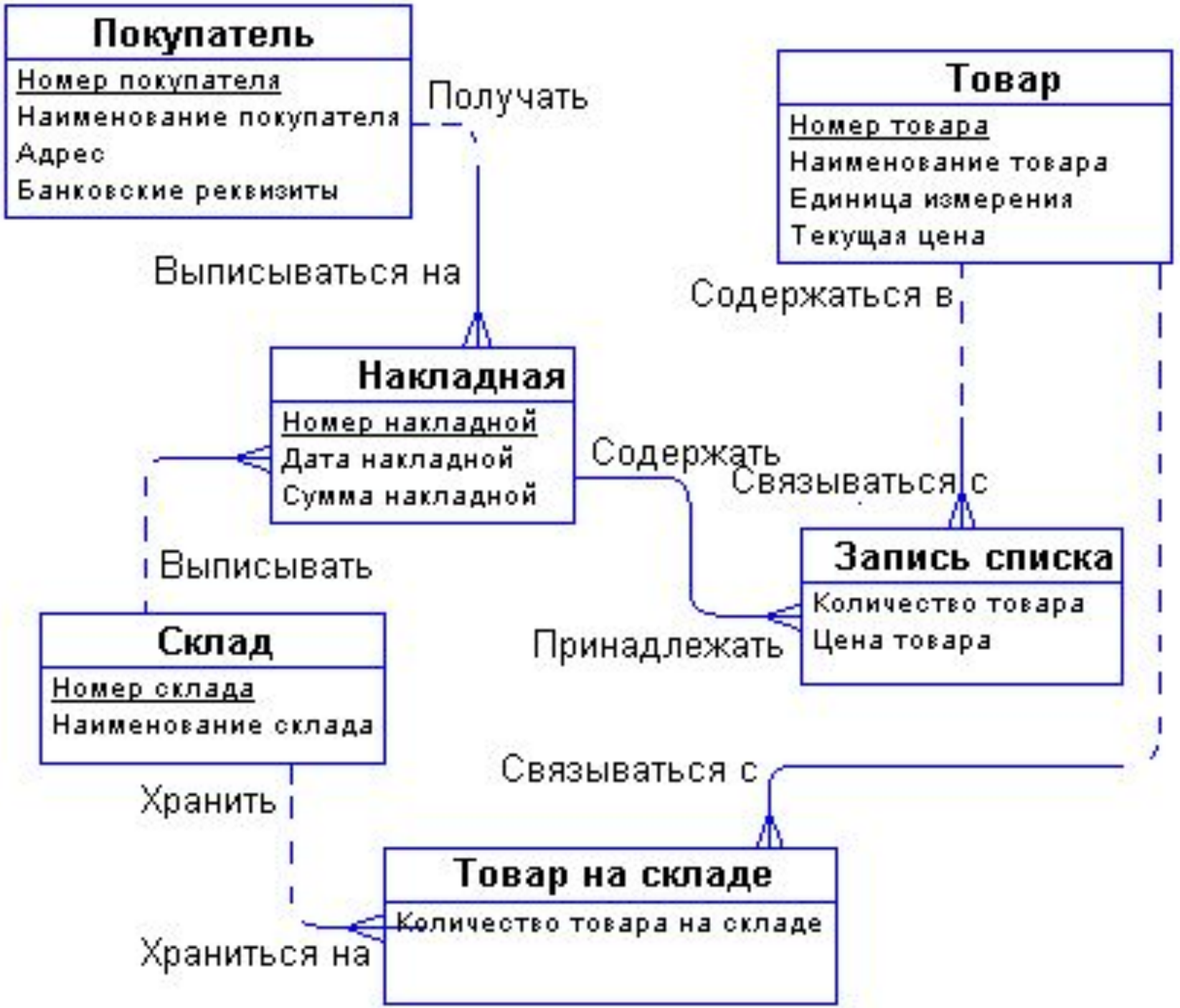
Простая ER-модель

- *Связь ее с сущностями "Накладная" и "Товар" характеризуется следующими фразами - "каждая накладная обязана иметь несколько записей из списка товаров в накладной", "каждая запись из списка товаров в накладной обязана включаться ровно в одну накладную", "каждый товар может включаться в несколько записей из списка товаров в накладной", " каждая запись из списка товаров в накладной обязана быть связана ровно с одним товаром". Атрибуты "Количество товара в накладной" и "Цена товара в накладной" являются атрибутами сущности " Список товаров в накладной".*

Простая ER-модель

- Точно также вводится дополнительная сущность "Товар на складе".

  Атрибутом этой сущности будет "Количество товара на складе".


Покупатель
Номер покупателя
Наименование покупателя
Адрес
Банковские реквизиты
Получать
Товар
Номер товара
Наименование товара
Единица измерения
Текущая цена
Выписываться на
Содержаться в
Накладная
Номер накладной
Дата накладной
Сумма накладной
Содержать
Связываться с
Выписывать
Запись списка
Количество товара
Цена товара
Принадлежать
Склад
Номер склада
Наименование склада
Связываться с
Хранить
Товар на складе
Количество товара на складе
Храниться на

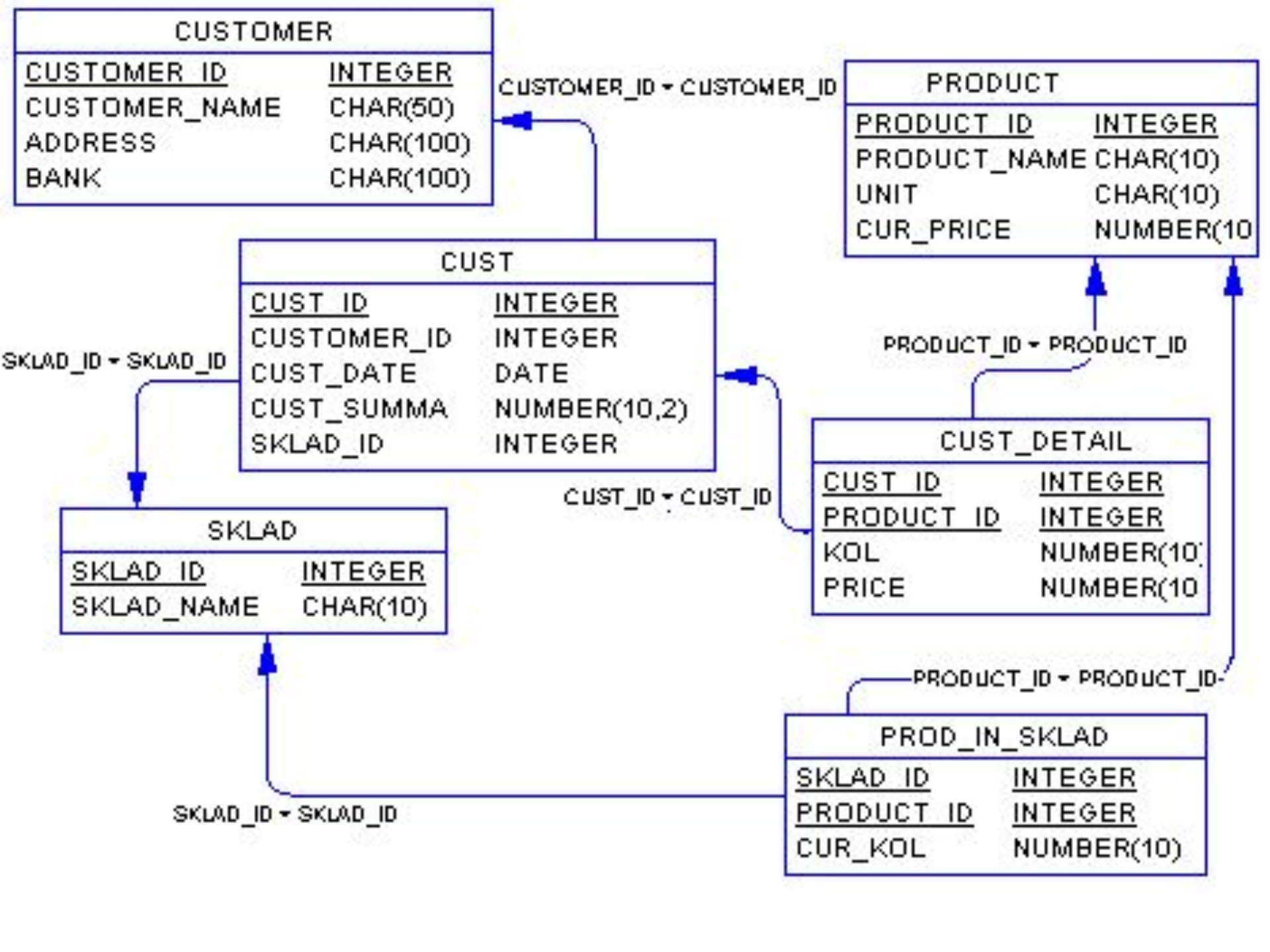
Простая ER-модель

- Разработанный выше пример ER-диаграммы является примером ***концептуальной диаграммы***.

Это означает, что диаграмма *не учитывает* особенности конкретной программной среды (СУБД).

Простая ER-модель

По данной концептуальной диаграмме можно построить ***физическую диаграмму***, которая уже будут учитывать конкретные особенности СУБД - допустимые типы и наименования полей и таблиц, ограничения длины т.п.:


CUSTOMER
CUSTOMER_ID INTEGER
CUSTOMER_NAME CHAR(50)
ADDRESS CHAR(100)
BANK CHAR(100)
CUSTOMER_ID - CUSTOMER_ID
PRODUCT
PRODUCT_ID INTEGER
PRODUCT_NAME CHAR(10)
UNIT CHAR(10)
CUR_PRICE NUMBER(10
CUST
CUST_ID INTEGER
CUSTOMER_ID INTEGER
CUST_DATE DATE
CUST_SUMMA NUMBER(10,2)
SKLAD_ID INTEGER
SKLAD_ID - SKLAD_ID
PRODUCT_ID - PRODUCT_ID
CUST_DETAIL
CUST_ID INTEGER
PRODUCT_ID INTEGER
KOL NUMBER(10
PRICE NUMBER(10
CUST_ID - CUST_ID
SKLAD
SKLAD_ID INTEGER
SKLAD_NAME CHAR(10)
PRODUCT_ID - PRODUCT_ID
PROD_IN_SKLAD
SKLAD_ID INTEGER
PRODUCT_ID INTEGER
CUR_KOL NUMBER(10)
SKLAD_ID - SKLAD_ID

Простая ER-модель

ER-диаграммы удобны тем, что процесс выделения сущностей, атрибутов и связей является итерационным.

Простая ER-модель

Разработав первый приближенный вариант диаграмм, мы уточняем их, опрашивая экспертов предметной области.

При этом документацией, в которой фиксируются результаты бесед, являются сами ER-диаграммы.

Пример ER-модели