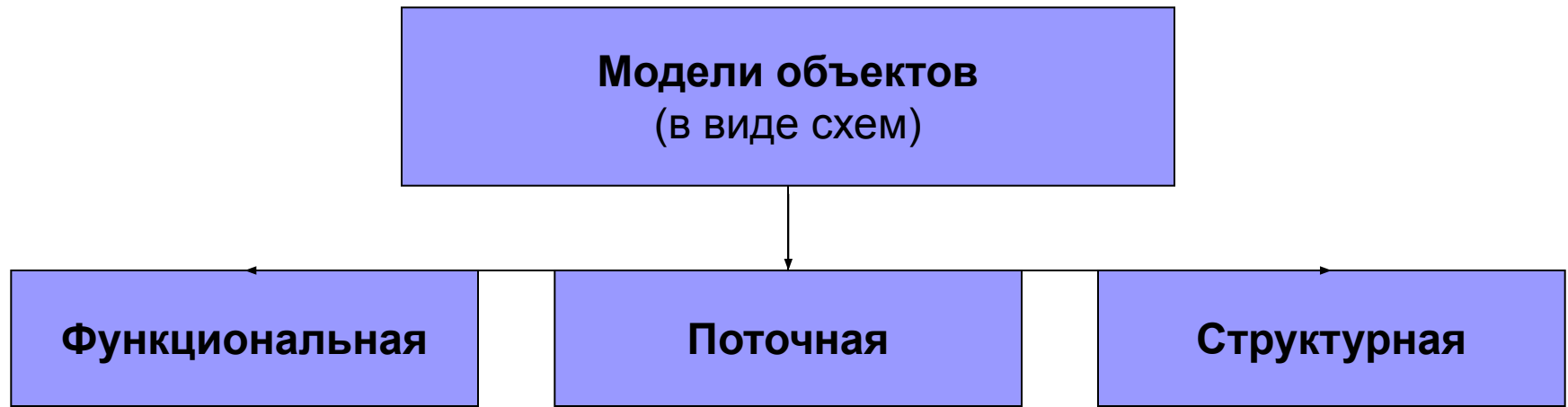


# Моделирование информационных систем

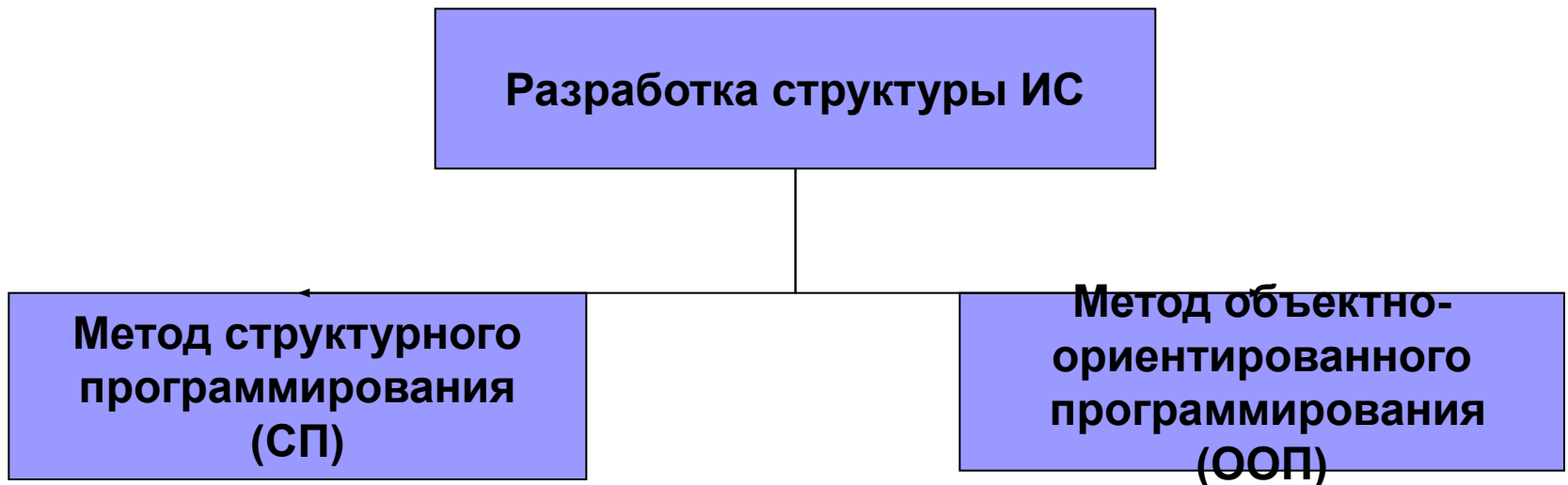
Архитектура информационной системы.  
Нормативное определение архитектуры  
информационной системы по ГОСТ Р 34.320-96.  
Архитектура клиент – сервер.



**Функциональная схема** фиксирует общее представление о системе, независимо от способа ее реализации, и является результатом идеализации системы на основе принципов определенной теории. Каждый элемент схемы выполняет определенную функцию.

**Поточная схема** описывает процессы, протекающие в технической системе и связывающие ее элементы в единое целое. Блоки таких схем отражают различные действия, выполняемые над естественным процессом элементами технической системы в ходе ее функционирования. Поточные схемы отображают не только естественные процессы, но и вообще любые *потоки субстанции (вещества, энергии, информации)*.

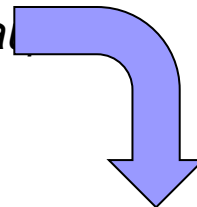
**Структурная схема** фиксирует узловые точки, на которые замыкаются потоки (*процессы функционирования*). Это могут быть единицы оборудования, детали, представляющие собой конструктивные элементы различного уровня, входящие в данную техническую систему, которые могут отличаться по принципу действия, техническому исполнению и ряду других характеристик. **Структурная схема** фиксирует конструктивное расположение элементов и связей (*структуру*) данной технической системы и уже предполагает определенный способ ее реализации.



# Методы структурного моделирования

## Принципы структурного подхода:

- а) Принцип *«разделяй и властвуй»* - решение сложных проблем путем их разбиения на множество меньших, независимых задач;
- б) Принцип *иерархического упорядочивания* – организация составных частей проблемы в иерархические древовидные структуры;
- в) Принцип *абстрагирования* – выделение существенных аспектов системы и отвлечение от несущественных;
- г) Принцип *формализации* – необходимость строго методического подхода к решению проблемы;
- д) Принцип *непротиворечивости* – обоснованность и согласованность элементов;
- е) Принцип *независимости данных* – модели данных должны быть спроектированы и проанализированы независимо от процессов их логической обработки;
- ж) Использование *графических нотаций*



### Виды *графических нотаций*:

- DFD (Data Flow Diagrams) – диаграммы потоков данных (ДПД) совместно со словарями данных и спецификациями процессов;
- ERD (Entity-Relationship Diagrams) – диаграммы «сущность-связь»;
- STD (State Transition Diagrams) – диаграммы переходов состояний

### **Методологии структурного моделирования:**

- SADT - Structured Analysis and Design Technique
- IDEF0

# Методы объектно-ориентированного моделирования

- а) **Принцип онтологизации** системы - декларирует представление системы посредством системы классов, отражающих понятийную структуру предметной области в виде моделей. Классы определяются атрибутами (свойствами) и операциями (методами). Все множество экземпляров класса (объекты) также определяются через атрибуты и операции класса;
- б) **Принцип декомпозиции** - декларирует представление системы (предметной области или ИС) совокупностью взаимодействующих между собой объектов. Объекты являются экземплярами классов и взаимодействуют посредством передачи сообщений;
- в) **Принцип инкапсуляции** – декларирует запрещение любого доступа к атрибутам объекта, кроме как через его операции (методы); в соответствии с этим принципом внутренняя структура объекта скрыта от пользователя, а любое его действие инициируется внешним сообщением, вызывающим выполнение соответствующей операции;
- г). **Принцип наследования** - декларирует создание новых классов от общего к частному; новые классы сохраняют все свойства классов-родителей, а также содержат дополнительные атрибуты и операции, характеризующие их специфику;
- д) **Принцип полиморфизма** - декларирует возможность работы с объектом без информации о конкретном классе, экземпляром которого он является.

## Преимущества ИС, разработанных на основе методов ООП:

- а) возможность *повторного использования* (объектно-ориентированные системы могут быть легко собраны из ранее разработанных программных компонентов);
- б) *расширяемость*, (системы будут легко расширяться без какой либо модернизации повторно используемых компонентов).

**UML** (Unified Modeling Language – Унифицированный Язык Моделирования)

# Архитектура информационной системы

ГОСТ Р 34.320-96

*Концепции и терминология для концептуальной схемы и информационной базы*



Рамочная модель разработки архитектуры по IEEE 1471



# Архитектура ИС

## Системная архитектура

- архитектура ИС предприятия
- технологическая инфраструктура ИС

## Программная архитектура

- архитектура взаимодействия приложений в рамках ИС предприятия (архитектура приложений)
- архитектура программных модулей
- архитектура взаимодействия различных классов в рамках одного приложения

### Уровни описания:

- концептуальный
- логический
- физический

- **Концептуальная архитектура** определяет компоненты системы и их назначения, обычно в неформальном виде. Это представление часто используется для обсуждения с *нетехническими* специалистами (что система должна уметь делать, в основном, с т.з. конечного пользователя).
- **Логическая архитектура** выделяет, прежде всего, вопросы *взаимодействия* компонент системы, интерфейсы и используемые протоколы. Это представление позволяет эффективно организовать параллельную разработку.
- **Физическая реализация** описывает привязку к конкретным *узлам размещения*, типам оборудования, характеристикам окружения, таким как: используемые операционные системы, оборудование и т.п.

## Архитектура ИС по ГОСТ 34.320-96

Внешняя схема	Информационный процессор	Информационная база
Внутренняя схема		
Концептуальная схема		

**Внешняя схема:** Определение форм внешнего представления для возможных совокупностей *предложений* в пределах представления конкретного *пользователя*, а также аспектов манипулирования этими формами.


**предложение:** Лингвистический объект, представляющий определенное высказывание.

Предложения состоят из *термов* и *предикатов*.

**терм:** Лингвистический объект, обозначающий сущность.

**предикат:** Лингвистический объект, аналогичный глаголу, сообщающий что-либо о сущностях, обозначенных термами.

↑  
варианты использования

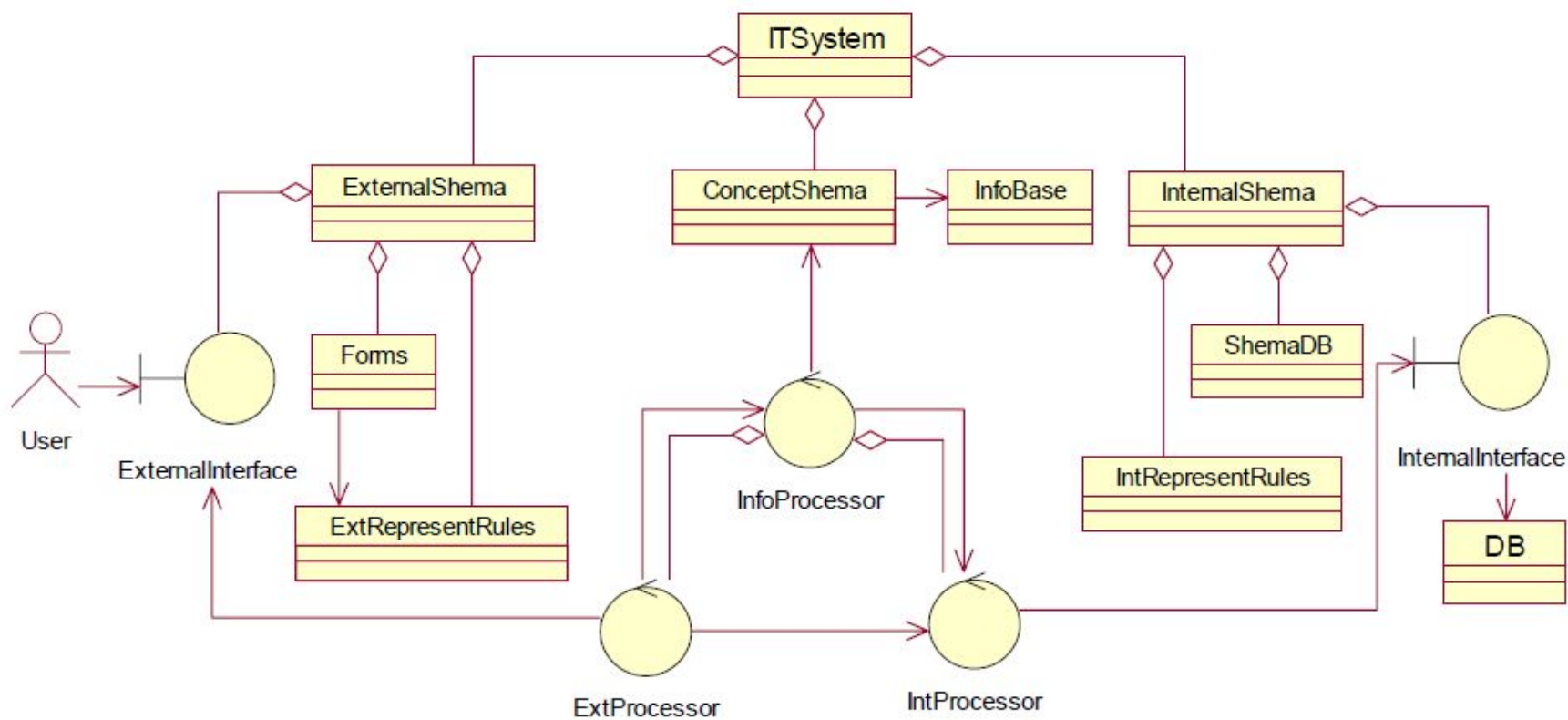


**Внутренняя схема:** Определение форм внутреннего представления в компьютере совокупностей предложений концептуальной схемы и информационной базы, а также аспектов манипулирования этими формами

**Концептуальная схема:** описывает содержимое базы данных, которое включает перечень действий, допустимых над этими данными, относящимися к предметной области.

**Информационная база:** содержит актуальную информацию о сущностях предметной области.

**Информационный процессор:** процессор, который в ответ на команду выполняет действие над концептуальной схемой и/или информационной базой.



Архитектура информационной системы по ГОСТ 34.320-96.

## Интерфейсы ИС:

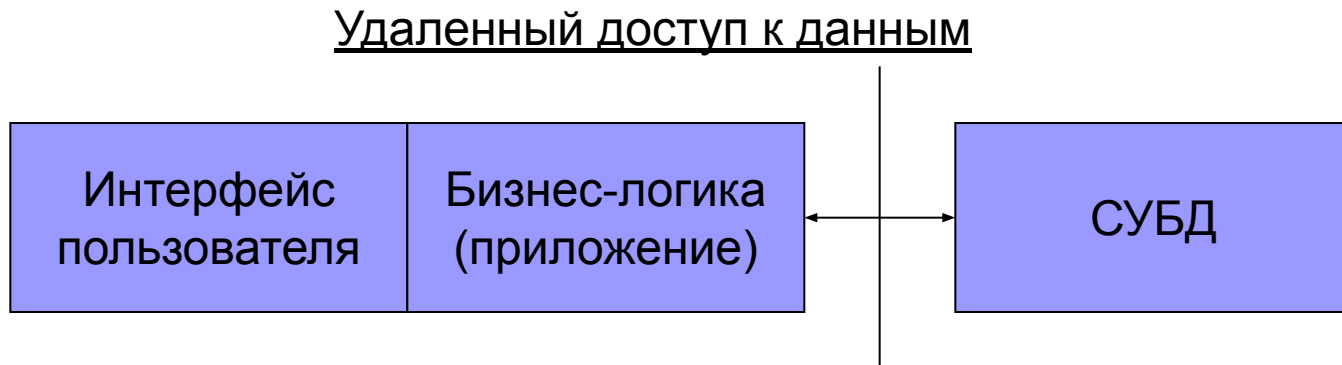
1. **Внешний интерфейс** — интерфейс между *пользователем* в среде и *информационной системой*. Описывается во внешних схемах и по отношению к пользователям информационной системы, обрабатывает внешние формы представления, удобные для конкретного пользователя.

2. **Внутренний интерфейс** — интерфейс между *информационной системой* и средствами *управления данными в компьютере*. Описывается внутренней схемой и связан с аспектами физического представления информации, эффективностью работы программ и механизмами эффективного доступа к хранимым данным, управление параллельным использованием, восстановлением после сбоев и т. д.

# Архитектурные решения при построении клиент – серверных систем

Архитектурные решения:

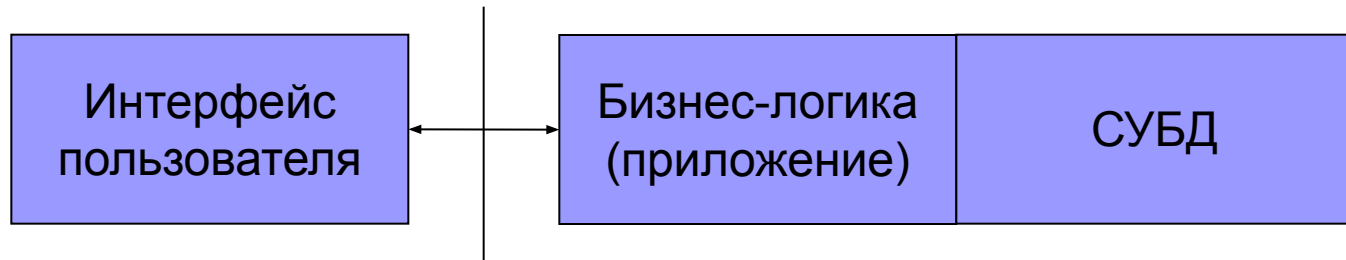
- а) удаленный доступ к данным;
- б) хранимые процедуры;
- в) трехзвенные системы.



Приложение располагается на клиентской машине и осуществляет обмен данными с СУБД.

Создают высокую нагрузку на вычислительную сеть, плохо масштабируются и неудобны в обслуживании.

## Хранимые процедуры



Приложение реализовано средствами самой СУБД.

Отсутствие переносимости, ограничение возможностей и производительности.



## Трёхзвенная архитектура



Приложение располагается в промежутке между СУБД и клиентом (*middleware*) на сервере приложения.

- Переносимость
- Модульность
- Централизация управления
- Надежность
- Простота интеграции
- Производительность

Приложение м.б. разделено на части:

- 1) решение прикладной задачи,
- 2) взаимодействие задачи с остальными компонентами системы.