



КЛАССИФИКАЦИЯ ИС ПО АРХИТЕКТУРЕ

Курс «Информационные
технологии»

Роль структуры управления в информационной системе

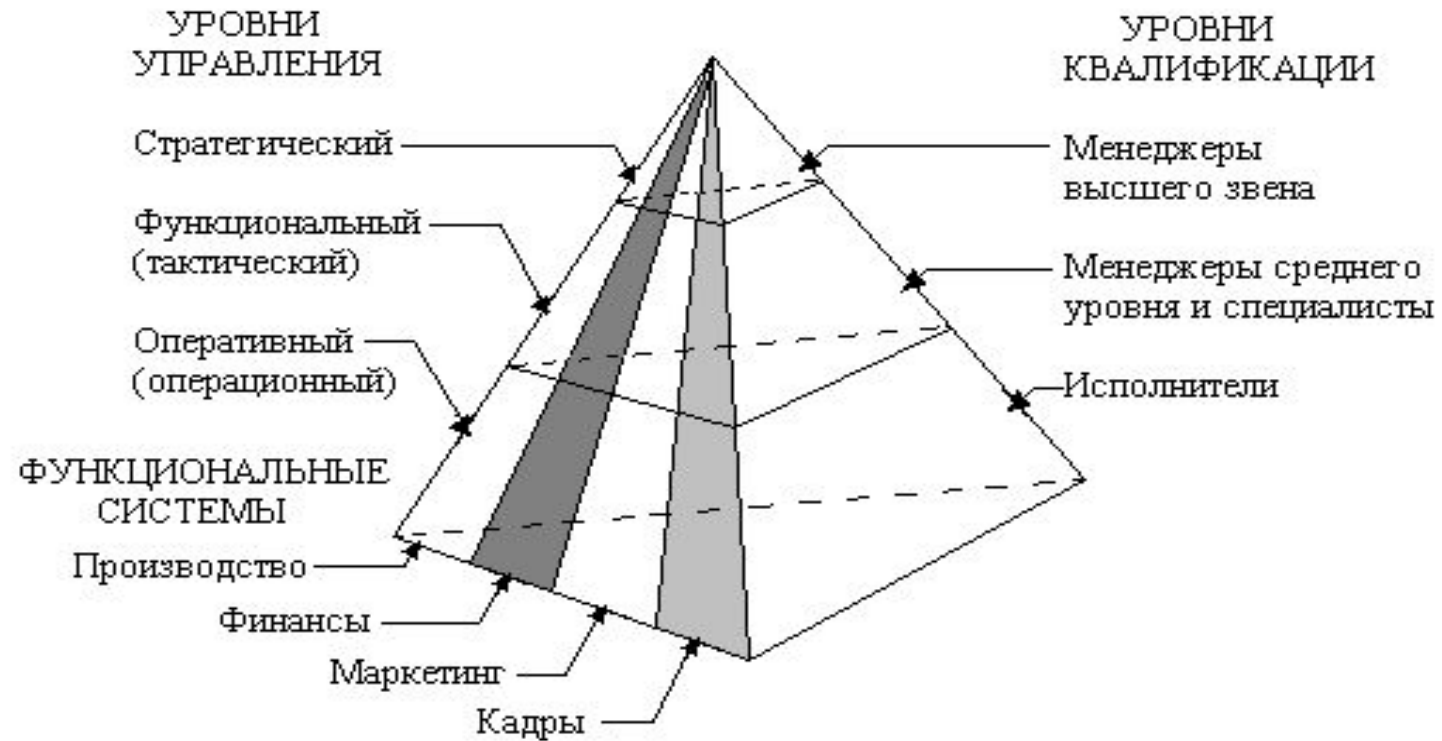


Структура управления предприятием



Квалификация персонала по уровням управления

КЛАССИФИКАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ПО ФУНКЦИОНАЛЬНОМУ ПРИЗНАКУ И УРОВНЯМ УПРАВЛЕНИЯ



Типы ИС в зависимости от функционального признака с учетом уровней управления и квалификацией персонала

Прочие классификации информационных систем



Модель «файл-сервер»



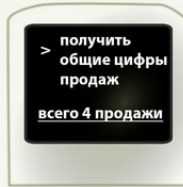
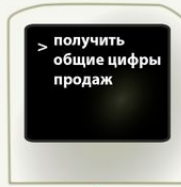
Двухуровневая модель «клиент-сервер»



Модель «файл - сервер» и двухуровневая архитектура «клиент- сервер»

Слой клиента

Самый верхний уровень приложения с интерфейсом пользователя. Главная функция интерфейса представление задач и результатов, понятных пользователю.



Слой логики

Этот слой координирует программу, обрабатывает команды, выполняет логические решения и вычисления, выполняет расчеты. Она также перемещается и обрабатывает данные между двумя окружающими слоями.



получить список продаж за прошлый год



объединить все продажи вместе

ЗАПРОС

1 ПРОДАЖА
2 ПРОДАЖА
3 ПРОДАЖА
4 ПРОДАЖА

Слой данных

Здесь хранится информация и извлекается из базы данных и файловой системы. Информация отправляется в логический слой для обработки и в конечном счете возвращается пользователю.



База Данных



Хранилище



Трехуровневая модель «клиент-сервер»

Функции:
физическое хранение данных,
управление данными

Функции:
логика обработки,
реализация бизнес-правил

Функции:
интерфейс пользователя

ТЕКСТ ХРАНИМОЙ ПРОЦЕДУРЫ

```
CREATE PROCEDURE Имя_Процедуры
  [(Параметр <тип данных> [,
    Параметр <тип данных> ...] )]
  [RETURNS (Параметр <тип данных> [,
    Параметр <тип данных> ...] )]
AS [<список переменных>] <блок> ;
<список переменных> =
  DECLARE [VARIABLE] Переменная <тип данных>;
  [DECLARE [VARIABLE] Переменная <тип данных>; ...]
<блок> =
  BEGIN <составной оператор>
    [<составной оператор>...]
  END
```

<составной оператор> = <блок> | Оператор;

<тип данных> = один из типов данных Firebird

Оператор – любой одиночный оператор языка хранимых процедур и триггеров Firebird.

```
CREATE PROCEDURE MakeGenerator (
  GenName VARCHAR(31),
  GenValue INTEGER
) AS
DECLARE VARIABLE Sql VARCHAR(256);
BEGIN
  Sql = ' CREATE GENERATOR ' || GenName || ' ';
  EXECUTE STATEMENT Sql;
  Sql = ' SET GENERATOR ' || GenName || ' TO ' ||
  CAST(GenValue AS VARCHAR(10)) || ' ';
  EXECUTE STATEMENT Sql;
END;
```

Преимущества в использовании хранимых процедур:

- отпадает необходимость синтаксической проверки каждого запроса и его компиляции перед выполнением, что повышает скорость выполнения запросов;
- отпадает необходимость реализации в клиентской программе запросов, определенных в теле хранимых процедур;
- увеличивается скорость обработки транзакций, так как вместо подчас длинного SQL-запроса по сети передается относительно короткое обращение к хранимой процедуре.

Oracle, Firebird, Interbase, IBM DB2, Informix, MS SQL Server, Sybase Adaptive Server Enterprise, PostgreSQL, MySQL, Caché, ЛИНТЕР.

Достоинства трехуровневой архитектуры:

- разгрузка сервера баз данных от выполнения части операций, перенесенных на сервер приложений;
- уменьшение размера клиентских приложений за счет разгрузки их от лишнего кода;
- единое поведение всех клиентов;
- упрощение настройки клиентов
- при изменении общего кода сервера приложений автоматически изменяется поведение клиентских приложений.



Многозвенные архитектуры:

а — классическая трехзвенная,

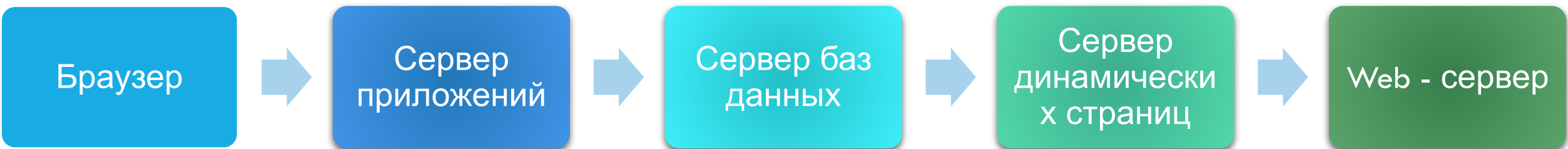
б — пятизвенная,

в — четырехзвенная,

г — трехзвенная архитектура с хранилищем «ключ-значение».

Сплошными линиями обозначены отдельные компоненты, пунктирные показывают, что компоненты объединены друг с другом

Соединение интернет (интранет) технологий с архитектурой клиент - сервер



Спасибо за внимание!