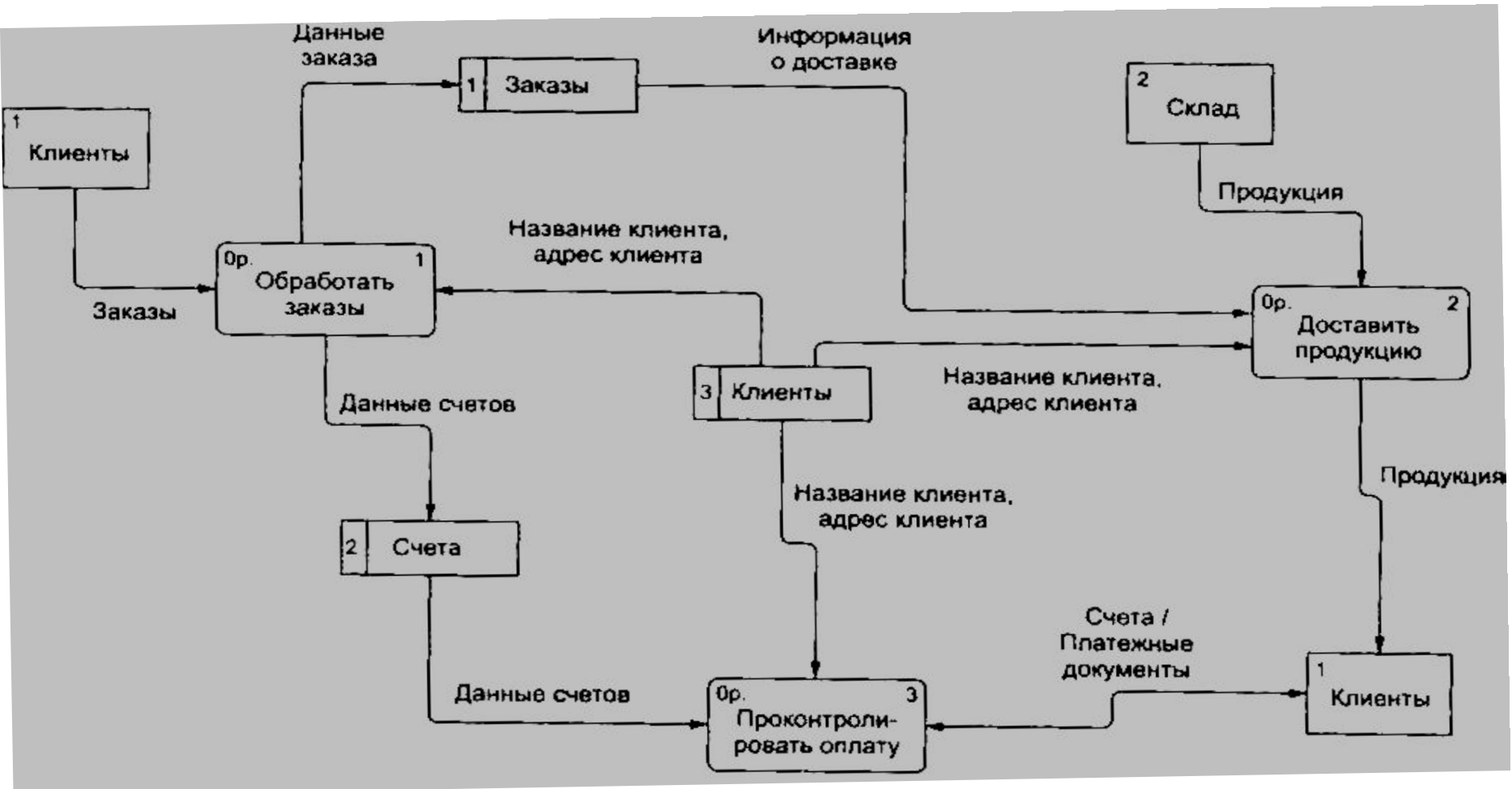


**Методика структурного анализа  
ПОТОКОВ ДАННЫХ  
(DFD — Data Flow Diagrams)**

# Назначение диаграмм потоков данных

- Так же, как и диаграммы IDEF0, диаграммы потоков данных моделируют систему как набор **действий**, соединенных друг с другом **стрелками**.
- Диаграммы потоков данных также могут содержать два новых типа объектов:
  - ✓ **объекты**, собирающие и хранящие информацию — *хранилища данных*;
  - ✓ **внешние сущности** — объекты, которые моделируют взаимодействие с теми частями системы (или другими системами), которые выходят за границы моделирования.
- В отличие от стрелок в IDEF0, которые иллюстрируют отношения, **стрелки** в DFD показывают, как объекты (включая и данные) реально перемещаются от одного действия к другому. Это представление потока вкупе с *хранилищами данных* и *внешними сущностями* обеспечивает отражение в DFD-моделях таких физических характеристик системы, как движение объектов (*потоки данных*), хранение объектов (*хранилища данных*), источники и потребители объектов (*внешние сущности*).

# Пример диаграммы потоков данных



# Синтаксис и семантика диаграмм потоков данных

- В отличие от IDEF0, рассматривающего систему как множество взаимопересекающихся действий, в названиях объектов DFD-диаграмм преобладают имена существительные.
- **Контекстная** DFD-диаграмма часто состоит из одного функционального блока и нескольких внешних сущностей.
- **Функциональный блок** на этой диаграмме обычно имеет имя, совпадающее с именем всей системы

Контекстная диаграмма DFD



# Функциональные блоки

- **Функциональный блок DFD** моделирует некоторую функцию, которая преобразует какое-либо сырье в какую-либо продукцию (или, в терминах IDEF, ВХОД в ВЫХОД).
- Хотя функциональные блоки DFD и изображаются в виде прямоугольников с закругленными углами, они почти идентичны функциональным блокам IDEF0 и действиям IDEF3.
- Как и действия IDEF3, функциональные блоки DFD **имеют входы** и **выходы**, но **не имеют** управления и механизма исполнения как IDEF0.
- В некоторых интерпретациях нотации DFD Гейна — Сарсона механизмы исполнения IDEF0 моделируются как ресурсы и изображаются в нижней части прямоугольника

Элемент DFD-диаграммы, построенной в нотации Гейна — Сарсона



# Внешние сущности

- **Внешние сущности** обеспечивают необходимые входы для системы и/или являются приемниками для ее выходов.
- Одна внешняя сущность может одновременно предоставлять входы (функционируя как поставщик) и принимать выходы (функционируя как получатель).
- Внешние сущности изображаются как прямоугольники и обычно размещаются у краев диаграммы. Одна внешняя сущность может быть размещена на одной и той же диаграмме в нескольких экземплярах. Этот прием полезно применять для сокращения количества линий, соединяющих объекты на диаграмме.

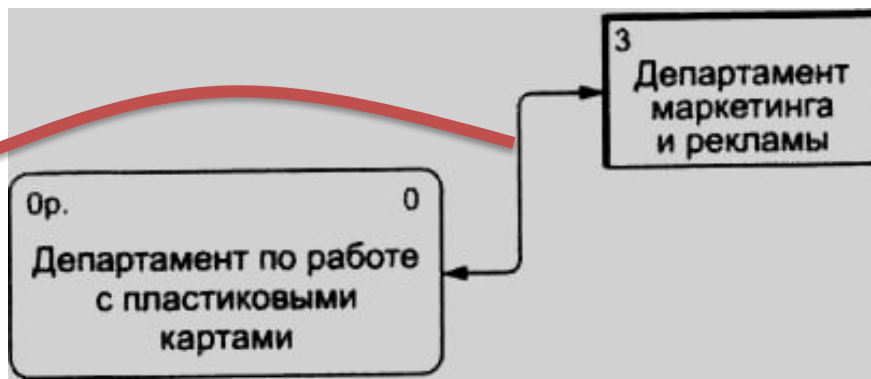
**Обозначение внешней сущности**



# Стрелки (потоки данных)

- **Стрелки** описывают передвижение (поток) объектов от одной части системы к другой.
- Поскольку все стороны обозначающего функциональный блок DFD прямоугольника равнозначны (в отличие от IDEF0), стрелки могут начинаться и заканчиваться в любой части блока.
- В DFD также используются двунаправленные стрелки, которые нужны для отображения взаимодействия между блоками (например, диалога типа приказ — результат выполнения).

Двунаправленный поток  
между блоком и внешней  
сущностью



# Хранилища данных

- В то время как **потоки данных** представляют объекты в процессе их *передвижения*, **хранилища данных** моделируют их во всех *остальных состояниях*.
- При моделировании *производственных систем* **хранилищами данных** служат *места временного складирования*, где хранится продукция на промежуточных стадиях обработки.
- В *информационных системах* **хранилища данных** представляют любой механизм, который поддерживает *хранение данных* для их промежуточной обработки.

Обозначение хранилища данных на DFD-диаграмме

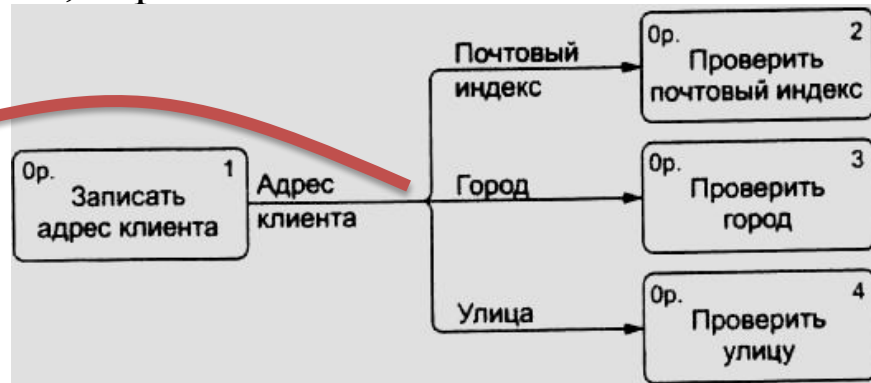




# Ветвление и объединение

- Стрелки на DFD-диаграммах могут быть разбиты (**разветвлены**) на части, и при этом каждый получившийся сегмент может быть переименован таким образом, чтобы показать декомпозицию данных, переносимых данным потоком

Разветвление стрелки, иллюстрирующее декомпозицию данных



- Стрелки могут и соединяться между собой (**объединяться**) для формирования так называемых **комплексных объектов**.



Объединение потока в один

# Построение диаграмм потоков данных

- Диаграммы DFD можно строить с использованием подхода, аналогичного структурному методу анализа и проектирования, применяемому в IDEF0:

Вначале строится модель *физической реализации реальной* системы, которая используется пользователями в настоящее время.

Затем создается *логическая модель текущего состояния* системы для моделирования основных требований существующей системы.

После этого создается *новая логическая модель* для отражения основных параметров *предлагаемой* разрабатываемой системы.

Наконец, создается *новая физическая модель*, реализующая логическую модель *новой* системы.

# Построение диаграмм потоков данных

- В настоящее время при разработке информационных систем завоевывает все большую популярность альтернативный подход, известный как **разделение событий**, в котором для моделирования системы строится несколько моделей DFD.

Вначале строится логическая модель, отображающая систему как набор *действий* и описывающая, *что* должна делать система.



Затем строится *модель окружения*, *включающая* систему как объект, отвечающий на события, порождаемые внешними элементами.

Такая модель обычно состоит из описания назначения системы, одной диаграммы контекстного уровня и списка событий.

Контекстная диаграмма содержит один функциональный блок, представляющий систему в целом, и внешних сущностей (окружения), с которыми система взаимодействует.



На заключительном этапе создается *модель поведения*, показывающая, как система обрабатывает те или иные события.

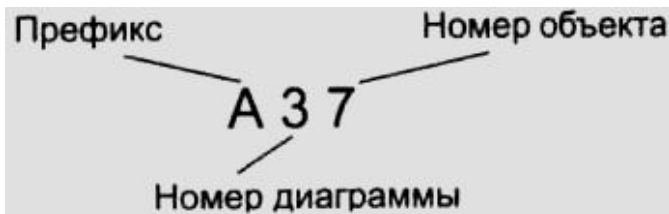
Эта модель начинается с единственной диаграммы с одним функциональным блоком на каждый ответ системы на событие, описанное в модели окружения.

Хранилища данных в модели поведения используются для моделирования данных, которые должны сохраняться в промежутках между обработкой событий.

Потоки применяются для соединения элементов диаграмм между собой и для проверки согласованности моделей поведения и окружения.

# Нумерация объектов

- В DFD каждый **номер функционального блока** может включать в себя *префикс*, *номер родительской диаграммы* и *собственно номер объекта* (рисунок).
- *Номер объекта* уникальным образом идентифицирует функциональный блок на диаграмме. *Номер родительской диаграммы* и *номер объекта* в совокупности обеспечивают уникальную идентификацию каждого блока модели.
- Уникальные номера присваиваются также каждому **хранилищу данных** и каждой **внешней сущности** вне зависимости от расположения объекта на диаграмме.
- Каждый номер хранилища данных содержит префикс D (от английского Data Store) и уникальный номер хранилища в модели (например, D3).
- Аналогично каждый номер каждой внешней сущности содержит префикс E (от английского External entity) и уникальный номер сущности в модели (например, E5).



**Компоненты номера функционального блока DFD**

# Заключение

- Итак, **диаграммы потоков данных (DFD)** обеспечивают удобный способ описания передаваемой информации как между частями моделируемой системы, так и между системой и внешним миром.
- Это качество определяет область применения DFD — они используются для создания *моделей информационного обмена организации*, например модели документооборота.
- Кроме того, различные вариации DFD широко применяются при построении *корпоративных информационных систем*.