**ЛЕКЦИЯ 13-15**

**Тема**: DFD методология моделирования. Нотация и принципы работы. Разбор примеров. Знакомство с правила распределения потоков. Определение правил оформления функций

Стандарт описания бизнес-процессов **DFD — Data Flow Diagram** переводится как диаграмма потоков данных и используется для описания процессов верхнего уровня и для описания реально существующих в организации потоков данных.

**Диаграммы потоков данных показывают**, как каждый процесс преобразует свои входные данные в выходные, и выявляют отношения между этими процессами. DFD представляет моделируемую систему как сеть связанных работ.

При построении DFD-схемы бизнес-процесса ***нужно помнить, что данная схема показывает потоки материальных и информационных потоков и ни в коем случае не говорит о временной последовательности работ,*** хотя в большинстве случаев временная последовательность работ и совпадает с направлением движения потоков в бизнес-процессе.

**три уровня проектирования DFD**

Проектирование DFD-диаграмм охватывает 3 уровня абстракции:

**1. Концептуальная диаграмма** показывает движения потоков данных на самом верхнем уровне абстракции. Концептуальная диаграмма детализируется

**логическим** и **физическим** потоками данных.

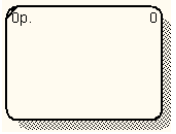
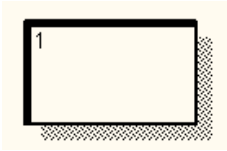
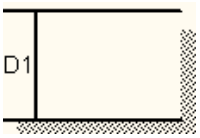
**2. Диаграмма логических потоков данных** отражает будущее или текущее состояние, описывая какие преобразования должны происходить независимо от физических ограничений.

**3. Диаграмма физических потоков данных** моделирует хранилища данных: принтеры, формы, устройства и другие проявления данных.

**Основные компоненты нотации:**

1. **Работы (Activities).** Отображают процессы обработки и изменения информации
2. **Стрелки (Arrows).** Отображают информационные потоки
3. **Хранилища данных (Data Store).** Отображают данные, к которым осуществляется доступ, эти данные используются, создаются или изменяются работами

4. **Внешние сущности (External References).** Отображают объекты, с которыми происходит взаимодействие

| **Описание** | **Графическое представление** | |
|---|---|---|
| **Работа (Activity)** | Объект обозначает функции или процессы, которые обрабатывают и изменяют информацию. |  |
| **Информационный поток (Precedence)** | Объект обозначает информационный поток от объекта-источника к объекту-приемнику. |  |
| **Внешняя ссылка (External reference)** | Указывают на место, организацию или человека, которые участвуют в процессе обмена информацией с системой, но располагаются за рамками этой диаграммы. |  |
| **Хранилище данных (Data store)** | Хранилища данных представляют собой собственно данные, к которым осуществляется доступ, эти данные также могут быть созданы или изменены работами. На одной диаграмме может присутствовать несколько копий одного и того же хранилища данных. |  |

***<u>Требования к оформлению функций:</u>***

**1.** <u>Каждая функция должна иметь идентификатор;</u>

**2.** <u>Названия работы нужно формулировать согласно следующее формуле:</u>

***<u>Название работы = Действие + Объект, над которым действие осуществляется</u>***

Например, если эта работа связана с действием по продаже продукции, то ее нужно назвать Продажа продукции

**3.** <u>Название работы должно быть по возможности кратким (не более 50 символов) и состоять из 2-3 слов. В сложных случаях также рекомендуется для каждого краткого названия работы сделать ее подробное описание, которое поместить в глоссарий.</u>

***<u>Требования к оформлению потока данных:</u>***

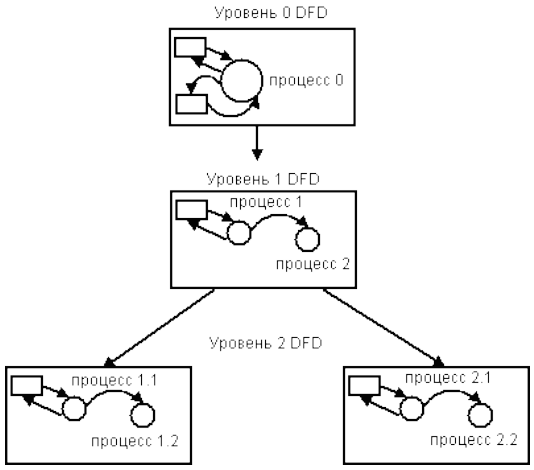
**<u>1.</u>** <u>Название потока нужно формулировать согласно следующей формуле:</u>

***<u>Название потока = Объект, представляющий поток + Статус объекта</u>***

Если речь идет о продукции, которую отгрузили клиенту, то поток можно назвать <Продукция, отгруженная> или <Продукция, отгруженная клиенту>. В данном случае <Продукция> это объект, представляющий поток, а <отгруженная клиенту> — статус объекта.

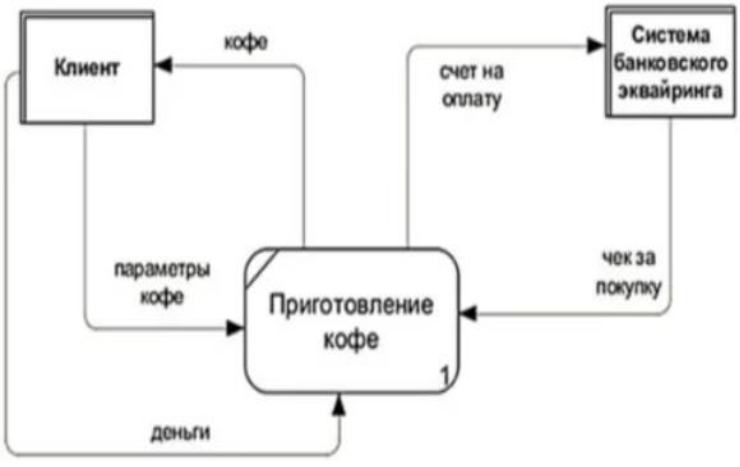
**2.** Название должно быть по возможности кратким и состоять из 2-3 слов.

**При детализации должны выполняться следующие правила:**



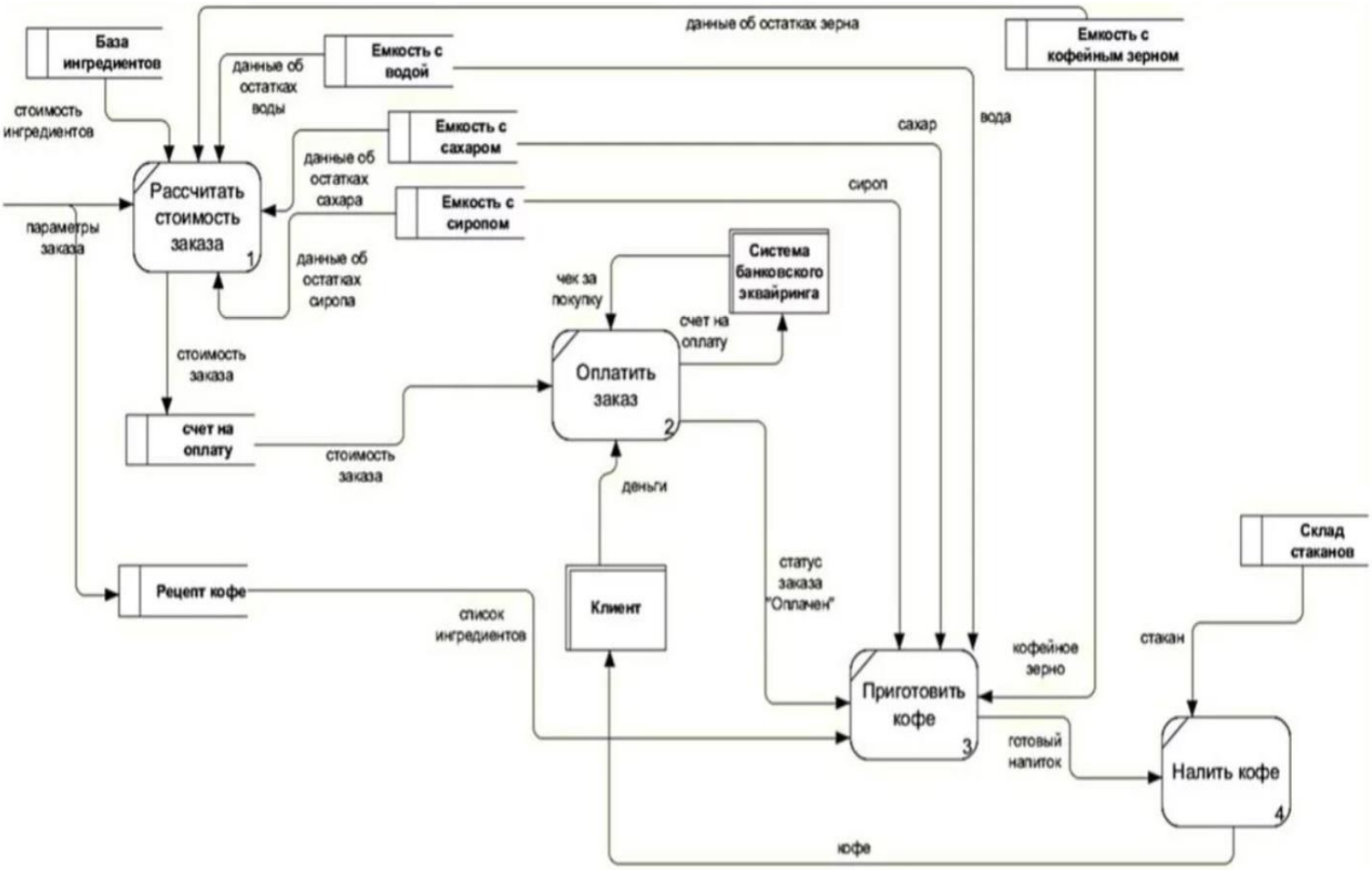

- **правило балансировки** — *при детализации процесса дочерняя диаграмма в качестве внешних источников/приемников данных может* иметь только те компоненты данные с которыми имеет информационную связь соответствующий процесс на родительской диаграмме;
- **правило нумерации** — при детализации процессов должна поддерживаться их иерархическая нумерация.
- **правило семи** — для того, чтобы диаграмма легко читалась, количество функций на диаграмме не должно быть больше семи.

На этапе декомпозиции процесса **Приготовление кофе** важно понять, что в первую очередь необходимо рассчитать стоимость заказа на основании параметров заказа: сколько требуется воды, сиропа, зерен и сахара.

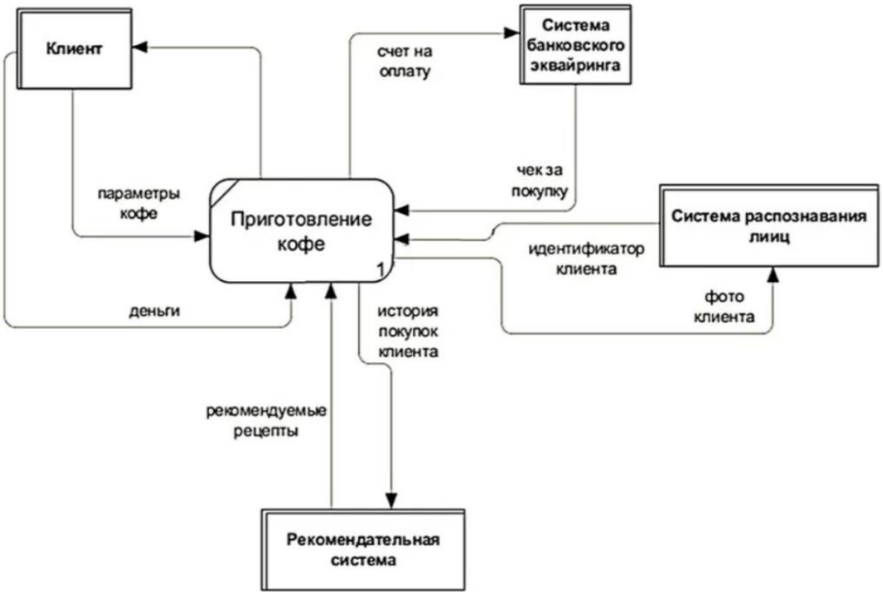



Кроме того, нужно знать, достаточно ли этих ингредиентов в аппарате. Данные об ингредиентах мы обозначим как хранилища, которые передают данные об остатках в процесс **Рассчитать стоимость заказа**. Стоимость ингредиентов передаётся в процесс расчёта из соответствующей базы. Результатом процесса является рассчитанная стоимость заказа, которая попадает в хранилище **Счёт на оплату**. Дальше стоимость заказа поступает в процесс **Оплатить заказ**, клиент вносит деньги, а сам процесс взаимодействует с внешней

сущностью **Система банковского эквайринга**. Процесс направляет счёт на оплату и получает чек за покупку.




**<u>приготовление кофе в кофейном автомате с системой распознавания лиц.</u>**




Когда заказ оплачен, ингредиенты и рецепт передаются в процесс **Приготовить кофе**. Готовый напиток наливается в стаканчик, переданный из **Склада стаканов**. Это происходит в процессе **Налить кофе**. В результате клиент получает готовый напиток.

Давайте добавим в этот пример немного киберпанка. Предположим, что у нас умный кофейный автомат, который

распознает клиента и хранит историю его покупок. Аппарат помнит, что клиент по понедельникам в хорошем настроении пьет сладкий латте, а в плохом настроении предпочитает маленькую чашечку бодрящего эспрессо.

Контекстная диаграмма в этом случае дополняется потоками идентификации клиента и формирования рекомендаций. В диаграмме также появится несколько новых внешних сущностей: **Система распознавания лиц** и **Рекомендательная система**. В качестве тренировки вы можете самостоятельно раскрыть эту контекстную диаграмму. Она будет похожа на предыдущий пример, но будет содержать ещё больше процессов обмена данными.


**Пример DFD: предоставление контента по интересам пользователя**

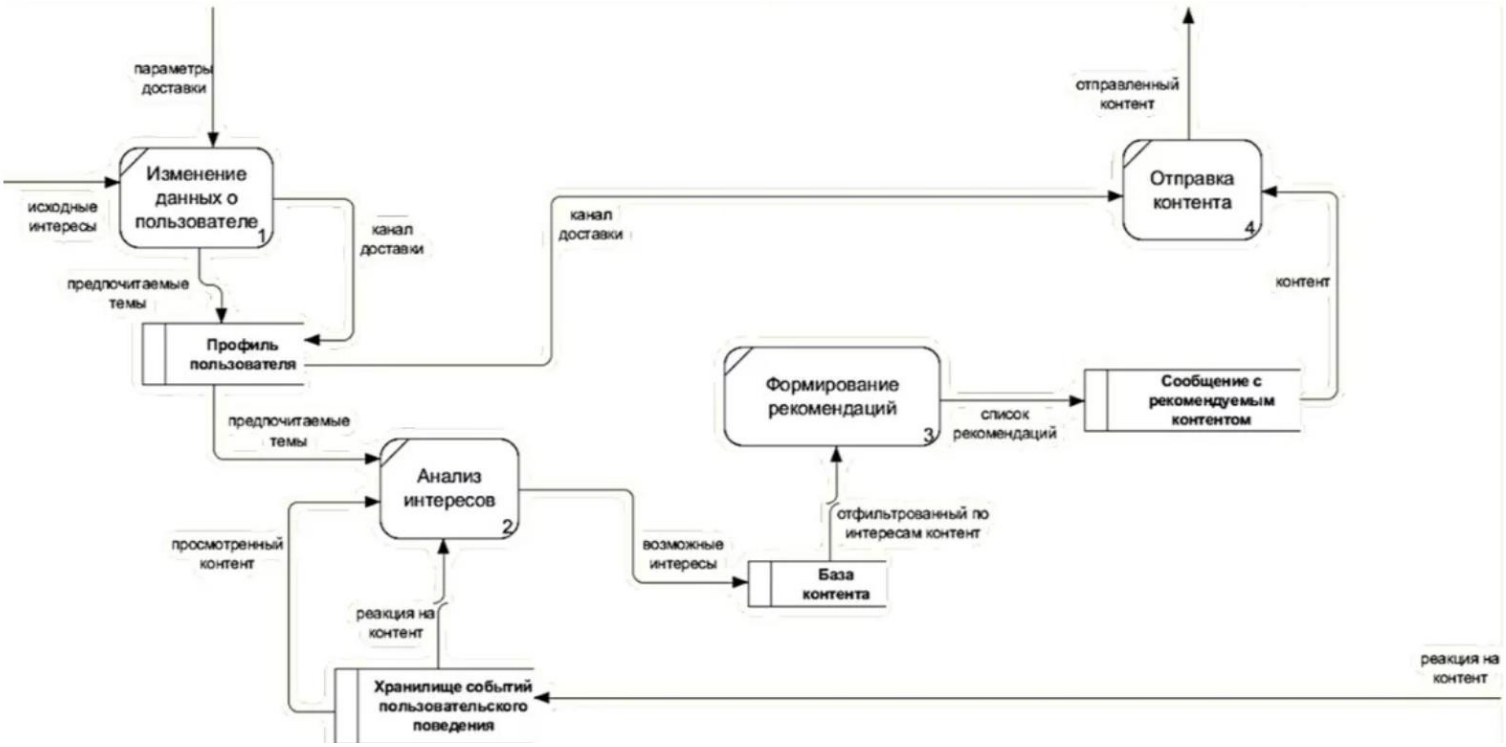
систему по предоставлению пользователю контента с учетом его интересов. Контент поступает по каналам, которые предпочел клиент: это может быть telegram, любой другой мессенджер или просто email.

Как выглядит процесс?

Клиент вводит первичные параметры доставки контента и свои исходные интересы, а система будет направлять ему контент. Поскольку система интеллектуальная, контент будет формироваться с учетом **реакций** и **предпочтений** пользователя (Content-based filtering).

Если пользователю нравится смотреть на котиков, будем показывать ему больше котиков. Если пользователь указал, что подборка котиков ему не очень интересна, будем исключать такой материал из контента.

При этом система сама будет определять, что нравится пользователю, а что не нравится.

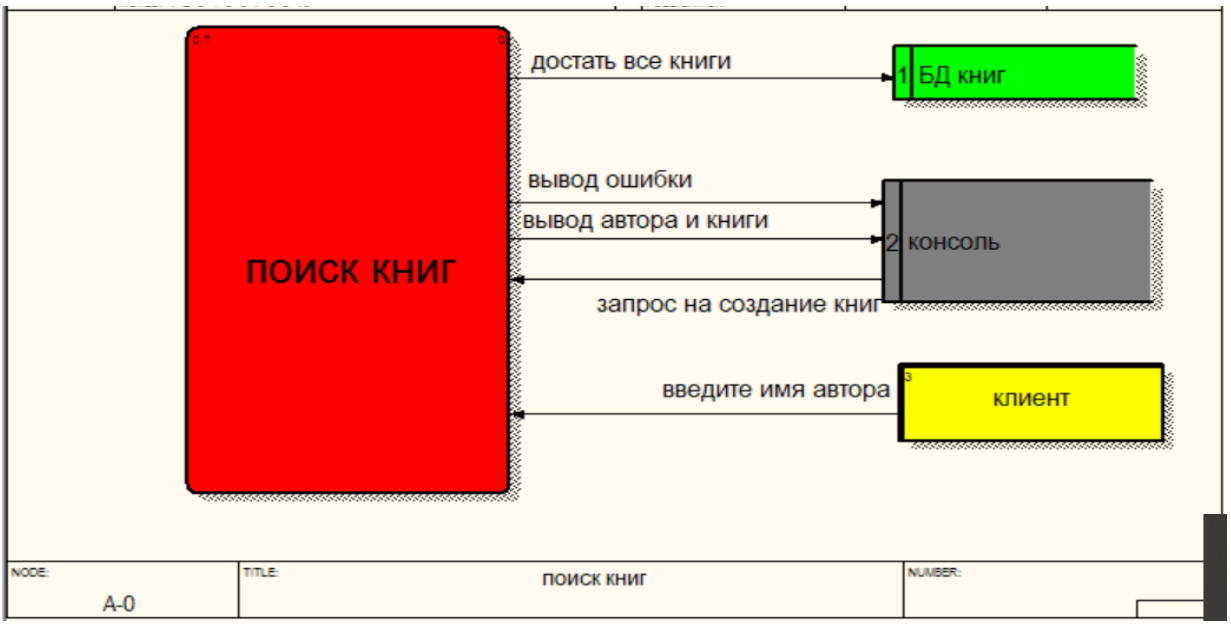



В процессе **Изменение данных пользователем**, куда приходят исходные интересы и параметры доставки, формируются предпочитаемые темы. Они передаются в промежуточное хранилище **Профиль пользователя**.

Просмотренный контент и — главное — реакция на него собираются в **Хранилище событий пользовательского поведения**. Как правило, это NoSQL-хранилища. Для сбора реакций на контент система анализирует поведение пользователя. Например, просмотрел ли пользователь данное видео полностью или быстро закрыл, прочитал статью и оставил комментарий либо поделился ссылкой и т.д.

В процессе **Анализа интересов**, куда попадают предпочитаемые пользователем темы и контент, который он просмотрел, формируется поток возможных интересов. Так система формирует **Базу контента**, откуда отфильтрованный контент попадает в процесс **Формирование рекомендаций**.

Сформированный список рекомендаций передаётся в промежуточное хранилище **сообщения с рекомендуемым контентом**. Процесс **Отправка** с учетом предпочитаемого канала доставляет контент пользователю.


ПОИСК КНИГ
достать все книги
1 БД книг
вывод ошибки
вывод автора и книги
2 консоль
запрос на создание книг
введите имя автора
3 клиент
NODE:
A-0
TITLE:
поиск книг
NUMBER: