**Задание № 3**

**Применение структурного подхода к проектированию ИС.**

**Реализация фазы структурного анализа: анализ поведения системы (построение DFD диаграммы верхнего уровня)**

Весь проект разделяется на 4 фазы: анализ, глобальное проектирование (проектирование архитектуры системы), детальное проектирование и реализация (программирование).

На фазе анализа строится модель среды (Environmental Model). Построение модели среды включает:

* анализ поведения системы (определение назначения ИС, построение начальной контекстной диаграммы потоков данных (DFD) и формирование матрицы списка событий (ELM), построение контекстных диаграмм);
* анализ данных (определение состава потоков данных и построение диаграмм структур данных (DSD), конструирование глобальной модели данных в виде ER-диаграммы).

Для реализации фазы анализа поведения системы используем методологию моделирования потоков данных. В основе данной методологии лежит построение **модели анализируемой ИС** - проектируемой или реально существующей.

В соответствии с методологией **модель системы** определяется как **иерархия диаграмм потоков данных** (ДПД или DFD), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю.

Диаграммы верхних уровней иерархии (**контекстные диаграммы**) определяют основные процессы или подсистемы ИС с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процессы становятся элементарными и детализировать их далее невозможно.

**Диаграммы потоков данных (DFD)** являются основным средством моделирования **функциональных требований** проектируемой системы. С их помощью эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных.

Главная цель таких средств - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Основными компонентами диаграмм потоков данных являются:

* внешние сущности;
* системы/подсистемы;
* процессы;
* накопители данных;
* потоки данных.

**Внешняя сущность** представляет собой материальный предмет или физическое лицо. Это источник или приемник информации, например, заказчики, персонал, поставщики, клиенты, склад.

**Подсистема** – составная часть ИС

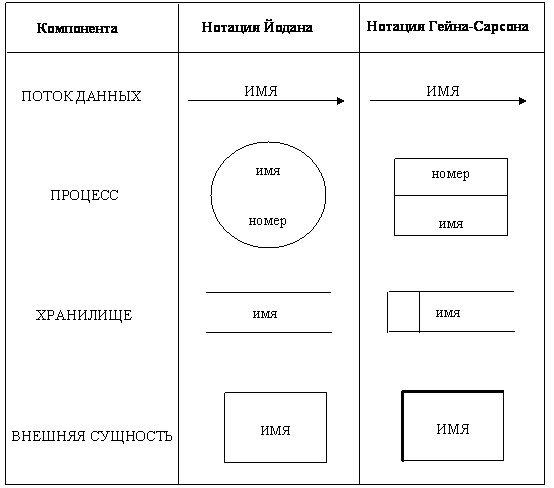
**Процесс** – процесс преобразования входных данных в выходные. Физически процесс может быть реализован различными способами: это может быть:

* подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов,
* программа,
* аппаратно реализованное логическое устройство и т.д.

**Накопитель данных** представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

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

**Основные символы диаграммы**



Применение методологии будем осуществлять на примере построения ИС для видеобиблиотеки, которая получает запросы на фильмы от клиентов и ленты, возвращаемые клиентами. Запросы рассматриваются администрацией видеобиблиотеки с использованием информации о клиентах, фильмах и лентах. При этом проверяется и обновляется список арендованных лент, а также проверяются записи о членстве в библиотеке. Администрация контролирует также возвраты лент, используя информацию о фильмах, лентах и список арендованных лент, который обновляется. Обработка запросов на фильмы и возвратов лент включает следующие действия: если клиент не является членом библиотеки, он не имеет права на аренду. Если требуемый фильм имеется в наличии, администрация информирует клиента об арендной плате. Однако, если клиент просрочил срок возврата имеющихся у него лент, ему не разрешается брать новые фильмы. Когда лента возвращается, администрация рассчитывает арендную плату плюс пени за несвоевременный возврат. Видеобиблиотека получает новые ленты от своих поставщиков. Когда новые ленты поступают в библиотеку, необходимая информация о них фиксируется. Информация о членстве в библиотеке содержится отдельно от записей об аренде лент.

Перед построением контекстной DFD необходимо проанализировать внешние события (внешние объекты), оказывающие влияние на функционирование библиотеки. Эти объекты взаимодействуют с ИС путем информационного обмена с ней.

Для нашего примера из описания предметной области следует, что в процессе работы библиотеки участвуют следующие группы людей**: клиенты, поставщики** и **руководство**. Эти группы являются внешними объектами. Они не только взаимодействуют с системой, но также определяют ее границы и изображаются на начальной контекстной DFD как терминаторы (внешние сущности). В отличие от нотации Гейна-Сарсона (Gane/Sarson) внешние сущности обозначаются обычными прямоугольниками, а процессы – окружностями.

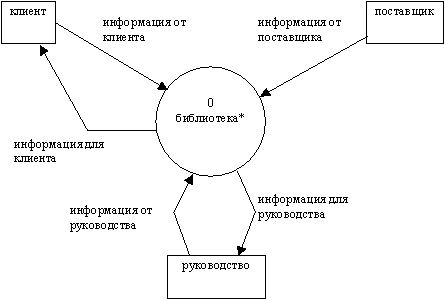


Рис.1. DFD (контекстная) диаграмма верхнего уровня иерархии

**Задание № 3.1**

Построить контекстную диаграмму верхнего уровня иерархии (начальную) для проектируемой ИС на бумаге и оформить с помощью программного средства (Word, Paint, Corel).

**Задание № 3.2**

Построить в файле Word матрицу списка событий.

Список событий строится в виде матрицы (ELM) и описывает различные действия внешних сущностей и реакцию ИС на них. В нашем примере эти действия представляют собой внешние события, воздействующие на библиотеку.

Различают следующие типы событий:

|  |  |
| --- | --- |
| **Аббревиатура** | **Тип** |
| NC | Нормальное управление |
| ND | Нормальные данные |
| NCD | Нормальное управление/данные |
| TC | Временное управление |
| TD | Временные данные |
| TCD | Временное управление/данные |

Все действия помечаются как нормальные данные. Эти данные являются событиями, которые ИС воспринимает непосредственно, например, изменение адреса клиента, которое должно быть сразу зарегистрировано. Они появляются в DFD в качестве содержимого потоков данных.

**Нормальные данные** – это такие данные, которые обеспечивают корректную и быструю обработку запросов системой. Нормальность данных обеспечивается в дальнейшем (при построении логической модели данных) путем приведения отношений к 1-ой. 2-ой и 3-ей нормальной форме.

Таблица. Матрица списка событий

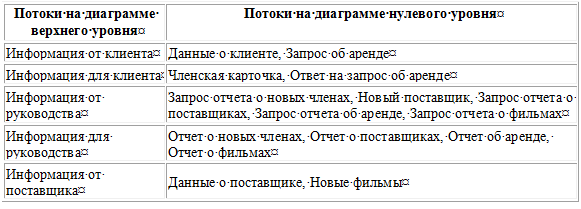
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Описание** | **Тип** | **Реакция** |
| **1** | Клиент желает стать членом библиотеки | ND | Регистрация клиента в качестве члена библиотеки |
| **2** | Клиент сообщает об изменении адреса | ND | Регистрация измененного адреса клиента |
| **3** | Клиент запрашивает аренду фильма | ND | Рассмотрение запроса |
| **4** | Клиент возвращает фильм | ND | Регистрация возврата |
| **5** | Руководство предоставляет полномочия новому поставщику | ND | Регистрация поставщика |
| **6** | Поставщик сообщает об изменении адреса | ND | Регистрация измененного адреса поставщика |
| **7** | Поставщик направляет фильм в библиотеку | ND | Получение нового фильма |
| **8** | Руководство запрашивает новый отчет | ND | Формирование требуемого отчета для руководства |

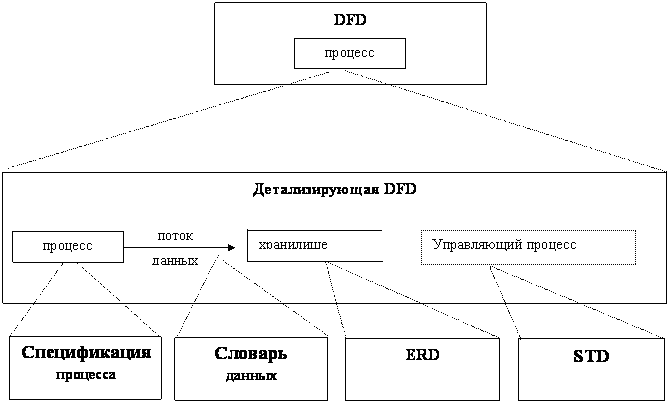
**Задание № 3.3**

Для дальнейшей конкретизации контекстной диаграммы верхнего уровня необходимо конкретизировать абстрактные потоки, отмеченные на диаграмме верхнего уровня. Для этого необходимо построить список потоков.

Для этого в нашем примере процесс "библиотека" декомпозируется на 4 процесса, отражающие основные виды административной деятельности библиотеки. Существующие "абстрактные" потоки данных между терминаторами и процессами трансформируются в потоки, представляющие обмен данными на более конкретном уровне. Список событий показывает, какие потоки существуют на этом уровне: каждое событие из списка должно формировать некоторый поток (событие формирует входной поток, реакция - выходной поток). Один "абстрактный" поток может быть разделен на более чем один "конкретный" поток.

Таблица. Список потоков





Логическая модель проектируемой системы