Лабораторная работа №1

Разработка технического задания

# Задание на лабораторную работу

Разработать техническое задание автоматизированной информационной системы в соответствии с выбранным вариантом задания.

# Перечень разделов технического задания и их содержание

**1. перечень разделов**

Техническое задание (ТЗ) на программный продукт курсовой работы (автоматизированную систему - АС) выполняется в соответствии с ГОСТ 34.602-89 и должно содержать следующие разделы:

* общие сведения;
* назначение и цели создания (развития) системы;
* характеристика объектов автоматизации;
* требования к системе;
* состав и содержание работ по созданию системы; .
* порядок контроля и приемки системы;
* требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
* требования к документированию;
* источники разработки.

В ТЗ могут включаться приложения.

В зависимости от вида, назначения, специфических особенностей объекта автоматизации и условий (функционирования системы допускается оформлять разделы ТЗ в виде приложений, вводить дополнительные, исключать или объединять подразделы ТЗ).

**2. Содержание разделов.**

2.1. В разделе “Общие сведения” указывают:

* полное наименование системы и ее условное обозначение;
* шифр темы или шифр (номер) договора;
* наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты;
* перечень документов, на основании которых создается система, кем и когда утверждены эти документы;
* плановые сроки начала и окончания работы по созданию системы;
* сведения об источниках и порядке финансирования работ;
* порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке
* отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы.

2.2. Раздел “Назначение и цели создания (развития) системы” состоит из подразделов:

* назначение системы;
* цели создания системы.

2.2.1. В подразделе “Назначение системы” указывают вид автоматизируемой деятельности (управление, проектирование и т. п.) и перечень объектов автоматизации (объектов), на которых предполагается ее использовать.

2.2.2. В подразделе “Цели создания системы” приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть достигнуты в результате создания АС, и указывают критерии оценки достижения целей создания системы.

2.3. В разделе “Характеристики объекта автоматизации” приводят:

* краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию;
* сведения об условиях эксплуатации объекта автоматизация и характеристиках окружающей среды.

2.4. Раздел “Требования к системе” состоит из следующих подразделов:

* требования к системе в целом;
* требования к функциям (задачам), выполняемым системой;
* требования к видам обеспечения.

Состав требований к системе, включаемых в данный раздел ТЗ на АС, устанавливают в зависимости от вида, назначения, специфических особенностей и условий функционирования конкретной системы. В каждом подразделе приводят ссылки на действующие НТД, определяющие требования к системам соответствующего вида.

2.4.1. В подразделе “Требования к системе в целом” указывают:

* требования к структуре и функционированию системы;
* требования к численности и квалификации персонала системы и режиму его работы;
* показатели назначения;
* требования к надежности;
* требования безопасности;
* требования к эргономике и технической эстетике;
* требования к транспортабельности для подвижных АС;
* требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
* требования к защите информации от несанкционированного доступа;
* требования по сохранности информации при авариях;
* требования к защите от влияния внешних воздействий;
* требования к патентной чистоте;
* требования по стандартизации и унификации;
* дополнительные требования.

2.4.1.1. В требованиях к структуре и функционированию системы приводят:

* перечень подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы;
* требования к способам и средствам связи для информационного обмена между компонентами системы;
* требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости, в том числе указания о способах обмена информацией (автоматически, пересылкой документов, по телефону и т. п.);
* требования к режимам функционирования системы;
* требования по диагностированию системы;
* перспективы развития, модернизации системы.

2.4.1.2. В требованиях к численности и квалификации персонала да АС приводят:

* требования к численности персонала (пользователей) АС;
* требования к квалификации персонала, порядку его подготовки я контроля знаний и навыков;
* требуемый режим работы персонала АС.

2.4.1.3. В требованиях к показателям назначения АС приводят значения параметров, характеризующие степень соответствия системы ее назначению.

2.4.1.4. В требования к надежности включают:

* состав и количественные значения показателей надежности для системы в целом или ее подсистем;
* перечень аварийных ситуаций, по которым должны быть регламентированы требования к надежности, и значения соответствующих показателей;
* требования к надежности технических средств и программного обеспечения;
* требования к методам оценки и контроля показателей надежности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами.

2.4.1.5. В требования по безопасности включают требования по обеспечению безопасности при монтаже, наладке, эксплуатации, обслуживании и ремонте технических средств системы (защита от воздействий электрического тока, электромагнитных полей, акустических шумов и т. п.), по допустимым уровням освещенности, вибрационных и шумовых нагрузок.

2.4.1.6. В требования по эргономике и технической эстетике включают показатели АС, задающие необходимое качество взаимодействия человека с машиной и комфортность условий работы персонала.

2.4.1.7. Для подвижных АС в требования к транспортабельности включают конструктивные требования, обеспечивающие транспортабельность технических средств системы, а также требования к транспортным средствам.

2.4.1.8. В требования к эксплуатации, техническому обслуживанию, ремонту и хранению включают:

* условия и регламент (режим) эксплуатации, которые должны обеспечивать использование технических средств (ТС) системы с заданными техническими показателями, в том числе виды и периодичность обслуживания ТС системы или допустимость работы без обслуживания;
* предварительные требования к допустимым площадям для размещения персонала и ТС системы, к параметрам сетей энергоснабжения и т. п.;
* требования по количеству, квалификации обслуживающего персонала и режимам его работы;
* требования к составу, размещению и условиям хранения комплекта запасных изделий и приборов;
* требования к регламенту обслуживания.

2.4.1.9. В требования к защите информации от несанкционированного доступа включают требования, установленные в отрасли (ведомстве) заказчика.

2.4.1.10. В требованиях по сохранности информации приводят перечень событий: аварий, отказов технических средств (в том числе - потеря питания) и т. п., при которых должна быть обеспечена сохранность информации в системе.

2.4.1.11. В требованиях по патентной чистоте указывают перечень стран, в отношении которых должна быть обеспечена патентная чистота системы и ее частей.

2.4.1.12. В требования к стандартизации и унификации включают: показатели, устанавливающие требуемую степень использования стандартных, унифицированных методов реализации функций (задач) системы, поставляемых программных средств, типовых математических методов и моделей, типовых проектных решений, унифицированных форм управленческих документов, требования к использованию типовых автоматизированных рабочих мест, компонентов и комплексов.

2.4.1.13. В дополнительные требования включают:

* требования к оснащению системы устройствами для обучения персонала (тренажерами, другими устройствами аналогичного назначения) и документацией на них;
* требования к сервисной аппаратуре, стендам для проверки элементов системы;
* требования к системе, связанные с особыми условиями эксплуатации;
* специальные требования по усмотрению разработчика или заказчика системы.

2.4.2. В подразделе “Требование к функциям (задачам)”, выполняемым системой, приводят:

* по каждой подсистеме перечень функций, задач или их комплексов (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации;
* при создании системы в две или более очереди - перечень функциональных подсистем, отдельных функций или задач, вводимых в действие в 1-й и последующих очередях;
* временной регламент реализации каждой функции, задачи (или комплекса задач);
* требования к качеству реализации каждой функции (задачи или комплекса задач), к форме представления выходной информации, характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций, достоверности выдачи результатов;
* перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

2.4.3. В подразделе “Требования к видам обеспечения” в зависимости от вида системы приводят требования к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другие видам обеспечения системы.

2.4.3.1. Для математического обеспечения системы приводят требования к составу, области применения (ограничения) и способам, использования в системе математических методов и моделей, типовых алгоритмов и алгоритмов, подлежащих разработке.

2.4.3.2. Для информационного обеспечения: системы приводят требования:

* к составу, структуре и способам организации данных в системе;
* к информационному обмену между компонентами системы;
* к информационной совместимости со смежными системами;
* по использованию общесоюзных и зарегистрированных республиканских, отраслевых классификаторов, унифицированных документов и классификаторов, действующих на данном предприятии;
* по применению систем управления базами данных;
* к структуре процесса сбора, обработки, передачи данных в системе и представлению данных;
* к защите данных от разрушений при авариях и сбоях в электропитании системы;
* к контролю, хранению, обновлению и- восстановлению данных;
* к процедуре придания юридической силы документам, продуцируемым техническими средствами ас (в соответствии с ГОСТ ,6.10.4).

2.4.3.3. Для лингвистического обеспечения системы приводят требования к применению в системе языков программирования высокого уровня, языков взаимодействия пользователей и 'технических средств системы, а также требования к кодированию и декодированию данных, к языкам ввода-вывода данных, языкам манипулирования данными, средствам описания предметной области (объекта автоматизации), к способам организации диалога.

2.4.3.4. Для программного обеспечения системы приводят перечень покупных программных средств, а также требования:

2.4.3.5. Для технического обеспечения системы приводят требования:

* к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию в системе;
* к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения системы.

2.5. Раздел “Состав и содержание работ ;по созданию (развитию) системы” должен содержать перечень стадий и этапов работ по созданию системы в соответствии с ГОСТ 24.601, сроки их выполнения, перечень организаций — исполнителей работ, ссылки на документы, подтверждающие согласие этих организаций на участие в создании системы, или запись, определяющую ответственного (заказчик или разработчик) за проведение этих работ.

В данном разделе также приводят:

* перечень документов, по ГОСТ 34.201, предъявляемых по окончании соответствующих стадий и этапов работ;
* вид и порядок проведения экспертиза технической документации (стадия, этап, объем проверяемой 'документации, организация-эксперт);
* программу работ, направленных на обеспечение требуемого уровня надежности разрабатываемой системы (при необходимости);
* перечень работ по метрологическому обеспечению на всех стадиях создания системы с указанием их сроков выполнения и организаций-исполнителей (при необходимости).

2.6. В разделе “Порядок контроля и приемки системы” указывают:

* виды, состав, объем и методы испытаний системы и ее составных частей (виды испытаний в соответствии с действующими нормами, распространяющимися на разрабатываемую систему);
* общие требования к приемке работ по стадиям (перечень участвующих предприятий и организаций, место и сроки проведения), порядок согласования и утверждения приемочной документации;
* статус приемочной комиссии.

2.7. В разделе “Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие” необходимо привести перечень основных мероприятий и их исполнителей, которые следует выполнить при подготовке объекта автоматизации к вводу АС в действие.

2.8. В разделе “Требования к документированию” приводят:

* согласованный разработчиком и Заказчиком системы перечень подлежащих разработке комплектов и видов документов, соответствующих требованиям ГОСТ 34.201 и НТД отрасли заказчика; перечень документов, выпускаемых на машинных носителях; требования к микрофильмированию документации;
* требования по документированию комплектующих элементов межотраслевого применения в соответствии с требованиями ЕСКД и ЕСПД;
* при отсутствии государственных стандартов, определяющих требования к документированию элементов системы, дополнительно включают требования к составу и содержанию таких документов.

2.9. В разделе “Источники разработки” должны быть перечислены документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги н др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы.

Приложения включают в состав ТЗ на АС по согласованию между разработчиком и заказчиком системы.

**3. ПРАВИЛА ОФОРМЛЕНИЯ ТЗ**

3.1. Разделы и подразделы ТЗ на АС должны быть размещены в порядке, установленном в настоящем стандарте.

3.2. ТЗ должно содержать титульный лист. На титульном листе помещают подписи руководителя и разработчиков.

Лабораторная работа №2

Разработка DFD диаграмм

# Задание на лабораторную работу

Разработать DFD диаграммы для программного проекта в соответствии с выбранным вариантом задания.

Минимальное количество уровней – 3 (контекстная диаграмма, диаграммы 1-го и 2-го уровней).

# Теоретическая часть

# Средства структурного анализа. Диаграммы потоков данных (DFD)

Структурным анализом принято называть метод исследования системы, которое начинается с ее общего обзора и затем детализируется, приобретая иерархическую структуру с все большим числом уровней. Для таких методов характерно: разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 6-7); ограниченный контекст, включающий лишь существенные на каждом уровне детали; дуальность данных и операций над ними; использование строгих формальных правил записи; последовательное приближение к конечному результату.

Все методологии структурного анализа базируются на ряде общих принципов, часть из которых регламентирует организацию работ на начальных этапах ЖЦ, а часть используется при выработке рекомендаций по организации работ. В качестве двух базовых принципов используются: принцип "разделяй и властвуй" и принцип иерархического упорядочивания. Первый принцип является способом решения трудных проблем путем разбиения их на множество меньших независимых задач, легких для понимания и решения. Второй декларирует, что устройство этих частей также существенно для понимания. Понимаемость проблемы резко повышается при организации ее частей в древовидные иерархические структуры, т.е. система может быть понята и построена по уровням, каждый из которых добавляет новые детали.

## 1.Средства структурного анализа и их взаимоотношения

В основе методологии проектирования ПО в настоящее время используются методы и соответствующие им средства структурного анализа. Структурный анализ используется на заключительной стадии этапа определения требований и на протяжении всего этапа проектирования.

Для целей моделирования систем вообще, и структурного анализа в частности, используются три группы средств, иллюстрирующих:

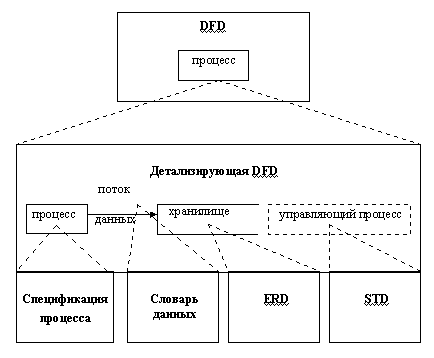
* функции, которые система должна выполнять;
* отношения между данными;
* зависящее от времени поведение системы (аспекты реального времени).

В методологиях структурного анализа наиболее часто и эффективно применяемыми являются следующие:

* DFD (Data Flow Diagrams) — диаграммы потоков данных совместно со словарями данных и спецификациями процессов или миниспецификациями;
* ERD (Entity-Relationship Diagrams) — диаграммы "сущность-связь";
* STD (State Transition Diagrams) — диаграммы переходов состояний.

Все они содержат графические и текстовые средства моделирования: первые — для удобства демонстрирования основных компонентов модели, вторые — для обеспечения точного определения ее компонентов и связей.

Логическая DFD показывает внешние по отношению к системе источники и стоки (адресаты) данных, идентифицирует логические функции (процессы) и группы элементов данных, связывающие одну функцию с другой (потоки), а также идентифицирует хранилища (накопители) данных, к которым осуществляется доступ. Структуры потоков данных и определения их компонентов хранятся и анализируются в словаре данных. Каждая логическая функция (процесс) может быть детализирована с помощью DFD нижнего уровня; когда дальнейшая детализация перестает быть полезной, переходят к выражению логики функции при помощи спецификации процесса (мини-спецификации). Содержимое каждого хранилища также сохраняют в словаре данных, модель данных хранилища раскрывается с помощью ERD. В случае наличия реального времени DFD дополняется средствами описания зависящего от времени поведения системы, раскрывающимися с помощью STD. Эти связи показаны на рис. 1.



***Рис. 1:*** Компоненты логической модели.

Перечисленные средства дают полное описание системы независимо от того, является ли она существующей или разрабатываемой с нуля. Таким образом строится логическая функциональная спецификация — подробное описание того, что должна делать система, освобожденное насколько это возможно от рассмотрения путей реализации. Это дает проектировщику четкое представление о конечных результатах, которые следует достигать.

## 2. Диаграммы потоков данных (DFD)

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.

Для изображения DFD традиционно используются две различные нотации: Йодана (Yourdon) и Гейна-Сарсона (Gane-Sarson). Далее при построении примеров будет использоваться нотация Йодана, все исключения будут предварительно оговариваться.

**Основные символы**

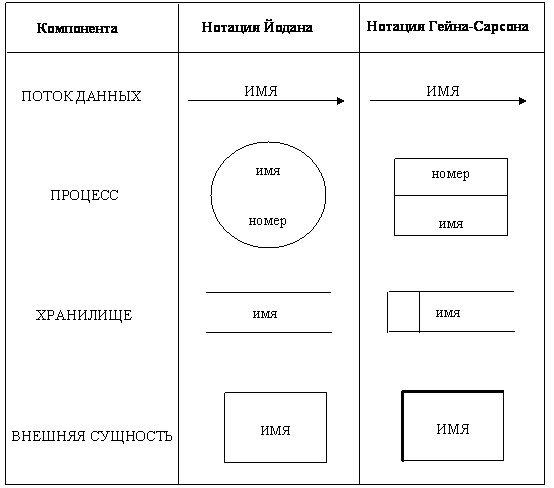
Основные символы DFD изображены на рис.2. Опишем их назначение. На диаграммах функциональные требования представляются с помощью процессов и хранилищ, связанных потоками данных.

**Потоки данных** являются механизмами, использующимися для моделирования передачи информации (или даже физических компонентов) из одной части системы в другую. Потоки на диаграммах обычно изображаются именованными стрелками, ориентация которых указывает направление движения информации. Иногда информация может двигаться в одном направлении, обрабатываться и возвращаться назад в ее источник. Такая ситуация может моделироваться либо двумя различными потоками, либо одним — двунаправленным.

Назначение **процесса** состоит в преобразовании входных потоков из выходных в соответствии с действием, задаваемым именем процесса. Это имя должно содержать глагол в неопределенной форме с последующим дополнением (например, ПРОВЕРИТЬ ДАННЫЕ). Кроме того, каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

**Хранилище (накопитель)** данных позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет "срезы" потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое и быть существительным. В случае, когда поток данных входит или выходит в/из хранилища, и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое нет необходимости отражать на диаграмме.

**Внешняя сущность (терминатор)** представляет сущность вне контекста системы, являющуюся источником или приемником системных данных. Ее имя должно содержать существительное, например, СКЛАД ТОВАРОВ. Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке.



***Рис. 2:*** Основные символы диаграммы потоков данных

**Описание потоков данных и процессов**

Базовые средства диаграммы не обеспечивают полного описания требований к программному изделию. Очевидно, что должны быть описаны стрелки — потоки данных — и преобразователи — процессы. Для этих целей используются словарь требований (данных) и спецификации процессов.

Словарь данных содержит описания потоков данных и хранилищ данных. Словарь данных является неотъемлемым элементом любой CASE-утилиты автоматизации анализа. Структура словаря зависит от особенностей конкретной CASE-утилиты. Тем не менее, можно выделить базисную информацию типового словаря требований.

Большинство словарей содержит следующую информацию.

1. Имя (основное имя элемента данных, хранилища или внешнего объекта).

2. Прозвище (Alias) — другие имена того же объекта.

3. Где и как используется объект — список процессов, которые используют данный элемент, с указанием способа использования (ввод в процесс, вывод из процесса, как внешний объект или как память).

4. Описание содержания — запись для представления содержания.

5. Дополнительная информация — дополнительные сведения о типах данных, допустимых значениях, ограничениях и т. д.

Спецификация процесса — это описание преобразователя. Спецификация поясняет: ввод данных в преобразователь, алгоритм обработки, характеристики производительности преобразователя, формируемые результаты.

Количество спецификаций равно количеству преобразователей диаграммы.

**Контекстная диаграмма и детализация процессов**

Декомпозиция DFD осуществляется на основе процессов. Каждый процесс может раскрываться с помощью DFD нижнего уровня.

Важную специфическую роль в модели играет специальный вид DFD — **контекстная диаграмма**, моделирующая систему наиболее общим образом. Контекстная диаграмма отражает интерфейс системы с внешним миром, а именно, информационные потоки между системой и внешними сущностями, с которыми она должна быть связана. Она идентифицирует эти внешние сущности, а также, как правило, единственный процесс, отражающий главную цель или природу системы насколько это возможно. Каждый проект должен иметь только одну контекстную диаграмму, при этом нет необходимости в нумерации ее единственного процесса.

DFD первого уровня строится как декомпозиция процесса, который присутствует на контекстной диаграмме.

Построенная диаграмма первого уровня также имеет множество процессов, которые в свою очередь могут быть декомпозированы в DFD нижнего уровня. Таким образом, строится иерархия DFD с контекстной диаграммой в корне дерева. Этот процесс декомпозиции продолжается до тех пор, пока процессы могут быть эффективно описаны с помощью коротких (до одной страницы) миниспецификаций обработки (спецификаций процессов).

При таком построении иерархии DFD каждый процесс более низкого уровня необходимо соотнести с процессом верхнего уровня. Обычно для этой цели используются структурированные номера процессов. Так, например, если мы детализируем процесс номер 2 на диаграмме первого уровня, раскрывая его с помощью DFD, содержащей три процесса, то их номера будут иметь следующий вид: 2.1, 2.2 и 2.3. При необходимости можно перейти на следующий уровень, т.е. для процесса 2.2 получим 2.2.1, 2.2.2. и т.д.

**Построение модели**

Главная цель построения иерархического множества DFD заключается в том, чтобы сделать требования ясными и понятными на каждом уровне детализации, а также разбить эти требования на части с точно определенными отношениями между ними. Для достижения этого целесообразно пользоваться следующими рекомендациями:

• Размещать на каждой диаграмме от 3 до 6-7 процессов. Верхняя граница соответствует человеческим возможностям одновременного восприятия и понимания структуры сложной системы с множеством внутренних связей, нижняя граница выбрана по соображениям здравого смысла: нет необходимости детализировать процесс диаграммой, содержащей всего один или два процесса.

• Не загромождать диаграммы несущественными на данном уровне деталями.

• Декомпозицию потоков данных осуществлять параллельно с декомпозицией процессов; эти две работы должны выполняться одновременно, а не одна после завершения другой.

• Выбирать ясные, отражающие суть дела, имена процессов и потоков для улучшения понимаемости диаграмм, при этом стараться не использовать аббревиатуры.

• Однократно определять функционально идентичные процессы на самом верхнем уровне, где такой процесс необходим, и ссылаться на него на нижних уровнях.

• Пользоваться простейшими диаграммными техниками: если что-либо, возможно описать с помощью DFD, то это и необходимо делать, а не использовать для описания более сложные объекты.

• Отделять управляющие структуры от обрабатывающих структур (т.е. процессов), локализовать управляющие структуры.

В соответствии с этими рекомендациями процесс построения модели разбивается на следующие этапы:

1. Расчленение множества требований и организация их в основные функциональные группы.

2. Идентификация внешних объектов, с которыми система должна быть связана.

3. Идентификация основных видов информации, циркулирующей между системой и внешними объектами.

4. Предварительная разработка контекстной диаграммы, на которой основные функциональные группы представляются процессами, внешние объекты — внешними сущностями, основные виды информации — потоками данных между процессами и внешними сущностями.

5. Изучение предварительной контекстной диаграммы и внесение в нее изменений по результатам ответов на возникающие вопросы по всем ее частям.

6. Построение контекстной диаграммы путем объединения всех процессов предварительной диаграммы в один процесс, а также группирования потоков.

7. Формирование DFD первого уровня на базе процессов предварительной контекстной диаграммы.

8. Проверка основных требований по DFD первого уровня.

9. Декомпозиция каждого процесса текущей DFD с помощью детализирующей диаграммы или спецификации процесса.

10. Проверка основных требований по DFD соответствующего уровня.

11. Добавление определений новых потоков в словарь данных при каждом их появлении на диаграммах.

12. Параллельное (с процессом декомпозиции) изучение требований (в том числе и вновь поступающих), разбиение их на элементарные и идентификация процессов или спецификаций процессов, соответствующих этим требованиям.

13. После построения двух-трех уровней проведение ревизии с целью проверки корректности и улучшения понимаемости модели.

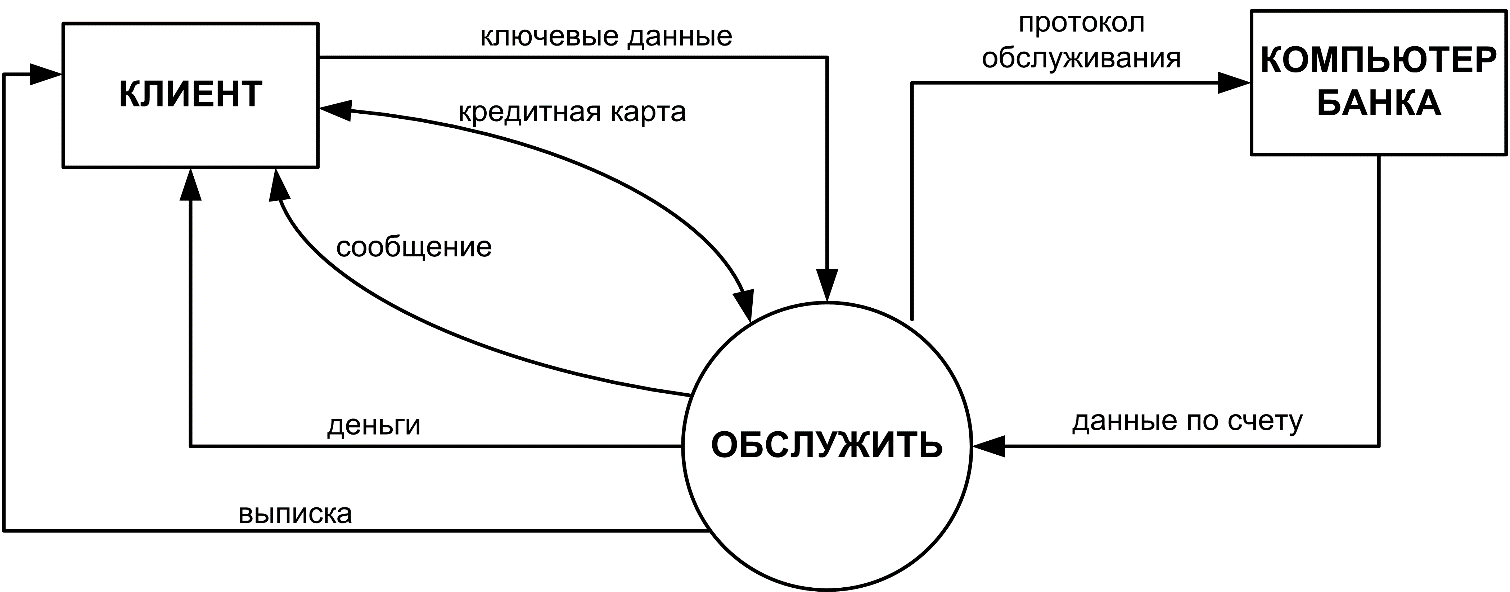
Построение спецификации процесса (а не простейшей диаграммы) в случае, если некоторую функцию сложно или невозможно выразить комбинацией процессов.

**Пример банковской задачи**

В качестве примера создания модели рассмотрим фрагмент проекта системы, организующей работу банкомата по обслуживанию клиента по его кредитной карте. Этот пример будет строиться поэтапно, на нем будут продемонстрированы базовые техники структурного анализа и проектирования по мере их определения.

На рис. 2. приведена контекстная диаграмма системы с единственным процессом ОБСЛУЖИТЬ, идентифицирующая внешние сущности КЛИЕНТ и КОМПЬЮТЕР БАНКА, хранящий информацию о счетах всех клиентов.

Опишем потоки данных, которыми обменивается проектируемая система с внешними объектами.



***Рис.*** *2.* Контекстная диаграмма банковской задачи

Для банковского обслуживания клиенту необходимо предоставить системе свою КРЕДИТНУЮ КАРТУ для автоматического считывания с нее информации (ПАРОЛЬ, ЛИМИТ ДЕНЕГ, ДЕТАЛИ КЛИЕНТА), а также сообщить свои КЛЮЧЕВЫЕ ДАННЫЕ, а именно ПАРОЛЬ и ЗАПРОС НА ОБСЛУЖИВАНИЕ, т. е. требуемую ему услугу (например, снятие со счета наличных денег). Банковское обслуживание с позиций клиента, в свою очередь, должно обеспечить следующее:

• выдать СООБЩЕНИЕ, приглашающее клиента ввести КЛЮЧЕВЫЕ ДАННЫЕ;

• выдать клиенту ДЕНЬГИ;

• выдать клиенту ВЫПИСКУ по проведенному обслуживанию, включающую ВЫПИСКУ О ДЕНЬГАХ, ВЫПИСКУ ПО БАЛАНСУ и ВЫПИСКУ ПО ОПЕРАЦИИ, проведенной банком.

Контекстный процесс и КОМПЬЮТЕР БАНКА должны обмениваться следующей информацией:

• ДАННЫЕ ПО СЧЕТУ клиента в банке;

• ПРОТОКОЛ ОБСЛУЖИВАНИЯ, включающий информацию об ОБРАБОТАННОЙ ДОКУМЕНТАЦИИ, изымаемой ДЕНЕЖНОЙ СУММЕ и ДАННЫЕ ПО ИСТОРИИ ЗАПРОСА.

Контекстный процесс может быть детализирован DFD первого уровня как показано на рис. 2. 3. Эта диаграмма содержит 4 процесса и хранилище ДАННЫЕ КРЕДИТНОЙ КАРТЫ, которое изображено дважды на диаграмме с целью избежания пересечений линий потоков данных.

**Процесс 1.** **1** (ПОЛУЧИТЬ ПАРОЛЬ) осуществляет прием и проверку пароля клиента и имеет на входе/выходе следующие потоки:

• внешний выходной поток СООБЩЕНИЕ для информирования клиента о готовности принять пароль;

• входной поток ВВЕДЕННЫЙ ПАРОЛЬ как элемент внешнего потока КЛЮЧЕВЫЕ ДАННЫЕ;

• входной поток ПАРОЛЬ из хранилища ДАННЫЕ КРЕДИТНОЙ КАРТЫ для проверки вводимого клиентом пароля.

**Процесс 1. 2 (**ПОЛУЧИТЬ ЗАПРОС НА ОБСЛУЖИВАНИЕ) осуществляет прием и проверку запроса клиента на проведение необходимой ему банковской операции и имеет на входе/выходе следующие потоки:

• внешний выходной поток СООБЩЕНИЕ для информирования клиента о своей готовности принять запрос на обслуживание;

• входной поток ЗАПРОС НА ОБСЛУЖИВАНИЕ как элемент внешнего потока КЛЮЧЕВЫЕ ДАННЫЕ;

• входной поток ЛИМИТ ДЕНЕГ из хранилища ДАННЫЕ КРЕДИТНОЙ КАРТЫ для контроля наличия денег на счете клиента.

**Процесс 1. 3** (ОБРАБОТАТЬ ЗАПРОС НА ОБСЛУЖИВАНИЕ) имеет внешний входной поток ДАННЫЕ ПО СЧЕТУ (из внешней сущности КОМПЬЮТЕР БАНКА), входной поток ДЕТАЛИ КЛИЕНТА (из хранилища), а также внешние выходные потоки ВЫПИСКА, ДЕНЬГИ и ПРОТОКОЛ ОБСЛУЖИВАНИЯ.

**Процесс 1. 4** (ОБРАБОТАТЬ КРЕДИТНУЮ КАРТУ) осуществляет считывание информации с кредитной карты и имеет на входе внешний поток КРЕДИТНАЯ КАРТА, а на выходе поток ДАННЫЕ КРЕДИТНОЙ КАРТЫ. Отметим, что нет необходимости в идентификации последнего потока, т. к. идентифицировано соответствующее хранилище.

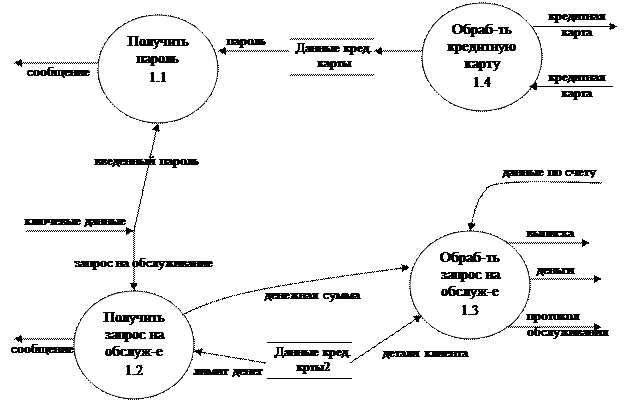
Процессы 1. 1, 1. 2 и 1. 4 являются элементарными, поэтому нет необходимости в их детализации с помощью DFD уровня 2. Процесс 1. 3 может быть детализирован с помощью DFD второго уровня как показано на рис. 4. Эта диаграмма содержит 4 элементарных процесса.

**Процесс 1. 3. 1** (ОБРАБОТАТЬ ДОКУМЕНТАЦИЮ БАНКА) осуществляет обработку внутренней банковской документации по клиенту и имеет входной поток ДЕТАЛИ КЛИЕНТА и выходной поток ОБРАБОТАННАЯ ДОКУМЕНТАЦИЯ (часть внешнего потока ПРОТОКОЛ СДЕЛКИ).

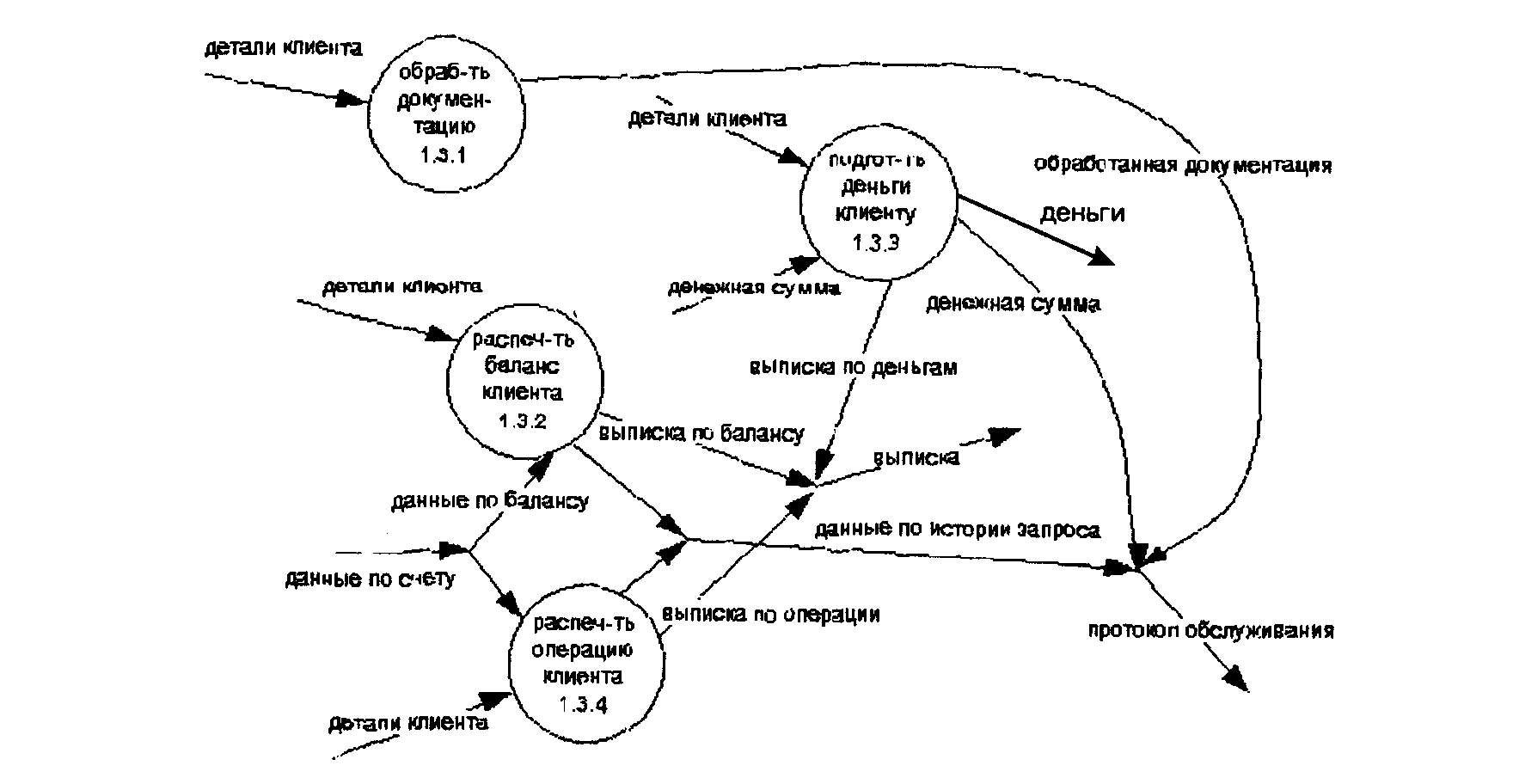
**Процесс 1. 3. 2** (РАСПЕЧАТАТЬ БАЛАНС КЛИЕНТА) выдает справку по истории счета клиента и по балансу клиента. Входные потоки — ДЕТАЛИ КЛИЕНТА и ДАННЫЕ ПО БАЛАНСУ (часть внешнего потока ДАННЫЕ ПО СЧЕТУ), выходные потоки - ВЫПИСКА ПО БАЛАНСУ (часть внешнего потока ВЫПИСКА) и ДАННЫЕ ПО ИСТОРИИ ЗАПРОСА (часть внешнего потока ПРОТОКОЛ ОБСЛУЖИВАНИЯ).

**Процесс 1. 3. 3** (ПРИГОТОВИТЬ ДЕНЬГИ КЛИЕНТУ) обеспечивает выдачу наличных денег и информирование компьютера банка об изъятых из банка деньгах. Он имеет входные потоки ДЕНЕЖНАЯ СУММА и ДЕТАЛИ КЛИЕНТА, и выходные потоки ДЕНЬГИ и ДЕНЕЖНАЯ СУММА (часть потока ПРОТОКОЛ ОБСЛУЖИВАНИЯ).

**Процесс 1. 3. 4** (РАСПЕЧАТАТЬ ОПЕРАЦИЮ КЛИЕ НТА) выдает справку по истории счета и уведомление по проведенной операции. Входные потоки ДАННЫЕ ПО СЧЕТУ и ДЕТАЛИ КЛИЕНТА, выходные потоки -ВЫПИСКА ПО ОПЕРАЦИИ (часть потока ВЫПИСКА) и ДАННЫЕ ПО ИСТОРИИ ЗАПРОСА (часть потока ПРОТОКОЛ ОБСЛУЖИВАНИЯ).



**Рис. 3.:** Детализация процесса ОБСЛУЖИТЬ



**Рис. 4.:** Детализация процесса ОБРАБОТАТЬ ЗАПРОС НА ОБСЛУЖИВАНИЕ

Лабораторная работа №3

Разработка ER-диаграмм

# Задание на лабораторную работу

Разработать ER-диаграммы для программного проекта в соответствии с выбранным вариантом задания.

# Теоретическая часть

# Диаграммы сущность-связь

[http:/m.ru/database/dblearn/dblearn08.shtml](http://citforum.ru/database/dblearn/dblearn08.shtml)

При проектировании структуры базы данных часто применяются метод семантического моделирования. Семантическое моделирование представляет собой моделирование структуры данных, опираясь на смысл этих данных. В качестве инструмента семантического моделирования используются различные варианты диаграмм сущность-связь (ER - Entity-Relationship).

Первый вариант модели сущность-связь был предложен в 1976 г. Питером Пин-Шэн Ченом. В дальнейшем многими авторами были разработаны свои варианты подобных моделей (нотация Мартина, нотация IDEF1X, нотация Баркера и др.). Различные программные средства, реализующие одну и ту же нотацию, могут отличаться своими возможностями. Все варианты диаграмм сущность-связь исходят из одной идеи - рисунок всегда нагляднее текстового описания. ER-диаграммы используют графическое изображение сущностей предметной области, их свойств (атрибутов) и взаимосвязей между сущностями.

В данной работе ER-диаграммамы даны в нотации Баркера, как довольно легкой в понимании основных идей.

**Основные понятия ER-диаграмм**

***Сущность*** - это класс однотипных объектов, информация о которых должна быть учтена в модели.

Каждая сущность должна иметь наименование, выраженное существительным в единственном числе.

Примерами сущностей могут быть такие классы объектов как "Поставщик", "Сотрудник", "Накладная".

Каждая сущность в модели изображается в виде прямоугольника с наименованием:



**Рис. 1**

***Экземпляр сущности*** - это конкретный представитель данной сущности.

Например, представителем сущности "Сотрудник" может быть "Сотрудник Иванов".

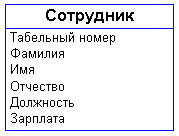
Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

***Атрибут сущности*** - это именованная характеристика, являющаяся некоторым свойством сущности.

Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными).

Примерами атрибутов сущности "Сотрудник" могут быть такие атрибуты как "Табельный номер", "Фамилия", "Имя", "Отчество", "Должность", "Зарплата" и т.п.

Атрибуты изображаются в пределах прямоугольника, определяющего сущность:

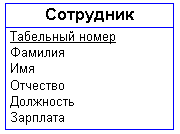


**Рис. 2**

***Ключ сущности*** - это *неизбыточный* набор атрибутов, значения которых в совокупности являются *уникальными* для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность.

Сущность может иметь несколько различных ключей.

Ключевые атрибуты изображаются на диаграмме подчеркиванием:



**Рис. 3**

***Связь*** - это некоторая ассоциация между *двумя* сущностями. Одна сущность может быть связана с другой сущностью или сама с собою.

Связи позволяют по одной сущности находить другие сущности, связанные с нею.

Например, связи между сущностями могут выражаться следующими фразами - "СОТРУДНИК может иметь несколько ДЕТЕЙ", "каждый СОТРУДНИК обязан числиться ровно в одном ОТДЕЛЕ".

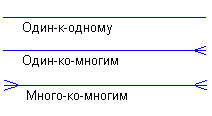
Графически связь изображается линией, соединяющей две сущности:



**Рис. 4**

Каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: "иметь", "принадлежать" и т.п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности.

Каждая связь может иметь один из следующих ***типов связи***:



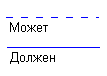
**Рис. 5**

Связь типа ***один-к-одному*** означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

Связь типа ***один-ко-многим*** означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется ***родительской***, правая (со стороны "много") - ***дочерней***. Характерный пример такой связи приведен на Рис. 4.

Связь типа ***много-ко-многим*** означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является *временным* типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

Каждая связь может иметь одну из двух ***модальностей связи***:



**Рис. 6**

Модальность "***может***" означает, что экземпляр одной сущности *может быть связан* с одним или несколькими экземплярами другой сущности, *а может быть и не связан* ни с одним экземпляром.

Модальность "***должен***" означает, что экземпляр одной сущности *обязан быть связан не менее чем с одним* экземпляром другой сущности.

Связь может иметь *разную модальность* с разных концов (как на Рис. 4).

Описанный графический синтаксис позволяет *однозначно* читать диаграммы, пользуясь следующей схемой построения фраз:

<Каждый экземпляр СУЩНОСТИ 1> <МОДАЛЬНОСТЬ СВЯЗИ> <НАИМЕНОВАНИЕ СВЯЗИ> <ТИП СВЯЗИ> <экземпляр СУЩНОСТИ 2>.

Каждая связь может быть прочитана как слева направо, так и справа налево. Связь на Рис. 4 читается так:

Слева направо: "каждый сотрудник может иметь несколько детей".

Справа налево: "Каждый ребенок обязан принадлежать ровно одному сотруднику".

**Пример разработки простой ER-модели**

При разработке ER-моделей мы должны получить следующую информацию о предметной

области:

1. Список сущностей предметной области.
2. Список атрибутов сущностей.
3. Описание взаимосвязей между сущностями.

ER-диаграммы удобны тем, что процесс выделения сущностей, атрибутов и связей является итерационным. Разработав первый приближенный вариант диаграмм, мы уточняем их, опрашивая экспертов предметной области. При этом документацией, в которой фиксируются результаты бесед, являются сами ER-диаграммы.

Предположим, что перед нами стоит задача разработать информационную систему по заказу некоторой оптовой торговой фирмы. В первую очередь мы должны изучить предметную область и процессы, происходящие в ней. Для этого мы опрашиваем сотрудников фирмы, читаем документацию, изучаем формы заказов, накладных и т.п.

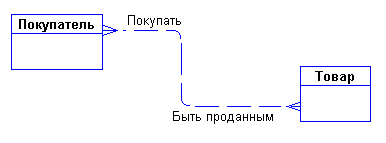
Например, в ходе беседы с менеджером по продажам, выяснилось, что он (менеджер) считает, что проектируемая система должна выполнять следующие действия:

* Хранить информацию о покупателях.
* Печатать накладные на отпущенные товары.
* Следить за наличием товаров на складе.

Выделим все существительные в этих предложениях - это будут потенциальные кандидаты на сущности и атрибуты, и проанализируем их (непонятные термины будем выделять знаком вопроса):

* *Покупатель* - явный кандидат на сущность.
* *Накладная* - явный кандидат на сущность.
* *Товар* - явный кандидат на сущность
* *(?)Склад* - а вообще, сколько складов имеет фирма? Если несколько, то это будет кандидатом на новую сущность.
* *(?)Наличие товара* – это, скорее всего, атрибут, но атрибут какой сущности?

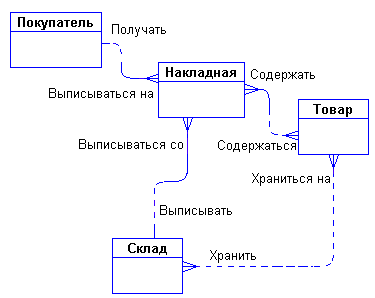
Сразу возникает очевидная связь между сущностями - "покупатели могут покупать много товаров" и "товары могут продаваться многим покупателям". Первый вариант диаграммы выглядит так:



**Рис. 7**

Задав дополнительные вопросы менеджеру, мы выяснили, что фирма имеет несколько складов. Причем, каждый товар может храниться на нескольких складах и быть проданным с любого склада.

Куда поместить сущности "Накладная" и "Склад" и с чем их связать? Спросим себя, как связаны эти сущности между собой и с сущностями "Покупатель" и "Товар"? Покупатели покупают товары, получая при этом накладные, в которые внесены данные о количестве и цене купленного товара. Каждый покупатель может получить несколько накладных. Каждая накладная обязана выписываться на одного покупателя. Каждая накладная обязана содержать несколько товаров (не бывает пустых накладных). Каждый товар, в свою очередь, может быть продан нескольким покупателям через несколько накладных. Кроме того, каждая накладная должна быть выписана с определенного склада, и с любого склада может быть выписано много накладных. Таким образом, после уточнения, диаграмма будет выглядеть следующим образом:



**Рис. 8**

Пора подумать об атрибутах сущностей. Беседуя с сотрудниками фирмы, мы выяснили следующее:

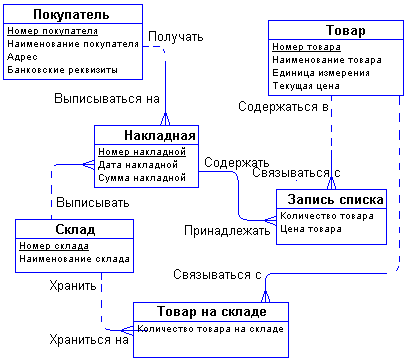
* Каждый покупатель является юридическим лицом и имеет наименование, адрес, банковские реквизиты.
* Каждый товар имеет наименование, цену, а также характеризуется единицами измерения.
* Каждая накладная имеет уникальный номер, дату выписки, список товаров с количествами и ценами, а также общую сумму накладной. Накладная выписывается с определенного склада и на определенного покупателя.
* Каждый склад имеет свое наименование.
* Снова выпишем все существительные, которые будут потенциальными атрибутами, и проанализируем их:
* *Юридическое лицо* - термин риторический, мы не работаем с физическими лицами. Не обращаем внимания.
* *Наименование покупателя* - явная характеристика покупателя.
* *Адрес* - явная характеристика покупателя.
* *Банковские реквизиты* - явная характеристика покупателя.
* *Наименование товара* - явная характеристика товара.
* *(?)Цена товара* - похоже, что это характеристика товара. Отличается ли эта характеристика от цены в накладной?
* *Единица измерения* - явная характеристика товара.
* *Номер накладной* - явная уникальная характеристика накладной.
* *Дата накладной* - явная характеристика накладной.
* *(?)Список товаров в накладной* - список не может быть атрибутом. Вероятно, нужно выделить этот список в отдельную сущность.
* *(?)Количество товара в накладной* - это явная характеристика, но характеристика чего? Это характеристика не просто "товара", а "товара в накладной".
* *(?)Цена товара в накладной* - опять же это должна быть не просто характеристика товара, а характеристика товара в накладной. Но цена товара уже встречалась выше - это одно и то же?
* *Сумма накладной* - явная характеристика накладной. Эта характеристика не является независимой. Сумма накладной равна сумме стоимостей всех товаров, входящих в накладную.
* *Наименование склада* - явная характеристика склада.

В ходе дополнительной беседы с менеджером удалось прояснить различные понятия цен. Оказалось, что каждый товар имеет некоторую текущую цену. Эта цена, по которой товар продается в данный момент. Естественно, что эта цена может меняться со временем. Цена одного и того же товара в разных накладных, выписанных в разное время, может быть различной. Таким образом, имеется *две цены* - цена товара в накладной и текущая цена товара.

С возникающим понятием "Список товаров в накладной" все довольно ясно. Сущности "Накладная" и "Товар" связаны друг с другом отношением типа много-ко-многим. Такая связь, как мы отмечали ранее, должна быть расщеплена на две связи типа один-ко-многим. Для этого требуется дополнительная сущность. Этой сущностью и будет сущность "Список товаров в накладной". Связь ее с сущностями "Накладная" и "Товар" характеризуется следующими фразами - "каждая накладная обязана иметь несколько записей из списка товаров в накладной", "каждая запись из списка товаров в накладной обязана включаться ровно в одну накладную", "каждый товар может включаться в несколько записей из списка товаров в накладной", " каждая запись из списка товаров в накладной обязана быть связана ровно с одним товаром". Атрибуты "Количество товара в накладной" и "Цена товара в накладной" являются атрибутами сущности " Список товаров в накладной".

Точно также поступим со связью, соединяющей сущности "Склад" и "Товар". Введем дополнительную сущность "Товар на складе". Атрибутом этой сущности будет "Количество товара на складе". Таким образом, товар будет числиться на любом складе и количество его на каждом складе будет свое.

Теперь можно внести все это в диаграмму:



**Рис. 9**

Данный пример ER-диаграммы является примером ***концептуальной диаграммы***. Это означает, что диаграмма *не учитывает* особенности конкретной СУБД.

Лабораторная работа №4

Разработка документации на программное обеспечение.

# Задание на лабораторную работу

1. Разработать документацию на созданное в соответствии с заданием ПО.

Состав комплекта документации:

1. Описание программы (см. Приложение 1).
2. Руководство пользователя (см. Приложение 2).

В описание программы необходимо включить DFD иER диаграммы, разработанные на лабораторных работах №2, 3.

1. Продемонстрировать работу ПО разработанного в соответствии с заданием.
2. Сдать отчеты по лабораторным работам в составе:
   1. Документ видение
   2. Описание программы
   3. Руководство пользователя

# Приложение 1.

Описание программы

* **Титульный лист**
* **Содержание**
* **Основная часть**
  + Общие сведения
    - *Обозначение и наименование программы*
    - *Программное обеспечение, необходимое для функционирования программы*
    - *Языки программирования, на которых написана программа*
  + Функциональное назначение
    - *Классы решаемых задач*
    - *Назначение программы*
    - *Сведения о функциональных ограничениях на применение*
  + Описание логической структуры
    - *Алгоритм программы*
    - *Используемые методы*
    - *Структура программы с описанием функций составных частей и связи между ними*
    - *Связи программы с другими программами*
  + Используемые технические средства
  + Вызов и загрузка
    - *Способ вызова программы с соответствующего носителя данных*
    - *Входные точки в программу*
  + Входные данные
    - *Характер, организация и предварительная подготовка входных данных*
    - *Формат, описание и способ кодирования входных данных*
  + Выходные данные
    - *Характер и организация выходных данных*
    - *Формат, описание и способ кодирования выходных данных*
* **Приложения**
  + Текст программы

# Приложение 2.

Руководство пользователя

* **Титульный лист**
* **Содержание**
* **Основная часть**
  + Назначение программы
    - *Виды деятельности и функции, которые автоматизированы:*
* Функция (1);
* Функция (2);
* …
* Функция (N).
  + Условия выполнения программы
    - *Минимальный состав аппаратных средств*
    - *Минимальный состав программных средств*
    - *Требования к персоналу (пользователю)*
  + Выполнение программы
    - *Загрузка и запуск программы*
    - *Выполнение программы (функции программы, процессы работы с данными, выполнение конкретных задач пользователя):*
      * Выполнение функции (1);
      * Выполнение функции (2);
      * …
      * Выполнение функции (N).
    - *Завершение работы программы*
  + Сообщения  оператору
    - *Сообщение (1);*
    - *Сообщение (2);*
    - *…*
    - *Сообщение (N).*
* **Приложения** (необязательны)