Методические указания

по курсовому проектированию

по курсу “ **Базы данных** “

для студентов специальности

***230100.62 Информатика и вычислительная техника***

***2014***

СОДЕРЖАНИЕ

1. Основные понятия ………………………………………………3
2. Этапы разработки информационной системы……………...….6
3. Отчетная документация…………………………………………8
4. Темы курсовых работ (с примерным описанием предметной области)…………………………………………………………16
5. Правила оформления курсовой работы………………………27
6. ЛИТЕРАТУРА……………………………………..…………..29

**1. О С Н О В Н Ы Е П О Н Я Т И Я**

При наличии большого объема перерабатываемой с помощью компью-тера информации возникают задачи обеспечения *наилучшего хранения дан-ных* ( без дублирования ) и *манипулирования данными* ( поиска, сортировки, добавления, изменения, обработки ).

Следовательно, нужно наилучшим образом *организовать данные* и обеспечить наилучшее *управление данными*.

*Данные* – информация, представленная в определенной форме, пригодной для последующей обработки, хранения и передачи.

*Структура данных* – способ объединения нескольких элементов данных в один.

*Предметная область* ( ПО ) – часть реального мира, подлежащая изучению с целью организации управления и последующей автоматизации.

ПО определена, если известны существующие в ней объекты, их свойства и отношения.

*Модель данных* – представление о предметной области в виде данных и связей между ними.

То есть, модель данных – это совокупность взаимосвязанных структур данных и операций над этими структурами.

Понятие “Модель данных” включает три компонента:

1) *организацию данных* ( количество и типы объектов модели данных, ограничения на структуру данных );

2) *множество допустимых операций* над данными:

операции выборки ( поиск ),

операции модификации ( включить, удалить, изменить данные );

3) средства обеспечения логической *целостности и достоверности данных* ( ограничения на значения данных и связи ), с помощью которых достигается непротиворечивость хранимой информации.

Выбор модели данных зависит от объема информации, сложности решаемых задач и имеющегося технического и программного обеспечения.

*База данных* ( БД ) – совокупность данных конкретной предметной области. Они организованы по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, и не зависят от программ обработки.

*Система управления базами данных* ( СУБД ) – набор программных средств ( программная система или пакет ), обеспечивающих создание и обслуживание баз данных и выполнение операций над данными БД ( доступ к ним и обработку ).

СУБД поддерживает один из типов моделей данных *– сетевую, иерархическую* или *реляционную*.

*Реляционная модель* ориентирована на табличное представление данных, т.е. организацию данных в виде двумерных таблиц.

В теории множеств таблице соответствует термин отношение (*relation*), который дал название модели.

*Реляционная база данных* (РБД) – база данных, логически организованная как *набор отношений* ( прямоугольных таблиц ) конкретной предметной области ( ПО ).

*Таблица* соответствует *объекту* ПО; *строка* ( кортеж ) – *запись* об одном экземпляре объекта.

Размещение в одной строке таблицы определенных элементов данных означает установление между ними связи или *отношения* ( *relation* ). Вся таблица в целом как совокупность конкретных экземпляров отношений называется *отношением*.

Значения в *столбце ( поле* ) таблицы определяют характеристику или свойство объекта ( *атрибут отношения* ).

Таблица имеет фиксированное число столбцов, их порядок фиксирован; число строк – произвольное, их порядок безразличен.

Таблица обладает следующими *свойствами:*

– столбцам ( полям ) присвоены уникальные имена;

– элементы каждого столбца имеют одинаковую природу, т.е. столбцы однородные;

– в таблице нет одинаковых строк ( записей ), т.е. любые две строки отличаются хотя бы одним элементом ( *полем записи* );

– строки и столбцы могут обрабатываться в любой последовательности.

Реляционная БД обычно включает несколько таблиц ( отношений ).

*Связи* между таблицами осуществляется с использованием *ключей*.

*Ключ* – атрибут ( поле ) или совокупность атрибутов, значения которых однозначно определяют запись в таблице.

Преимущества хранения данных в РБД:

1) каждый элемент данных хранится только в одной таблице ( *экономия места* );

2) внесение изменений упрощается, *уменьшается риск ошибки* (например, в написании фамилий );

3) наличие связей между таблицами *ускоряет обработку взаимосвязанной информации;*

4) *ошибочные записи* (с некорректными ссылками) должны автоматически *исключаться*.

Техническим возможностям персональных компьютеров в настоящее время лучше всего соответствуют реляционные СУБД.

*Информационная система* (ИС) представляет собой коммуникационную систему по сбору, передаче и обработке информации о заданной *предметной области*, снабжающую всех своих пользователей необходимой информацией.

*Информационную систему* определяют как систему информационных, математических, программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного многоаспектного использования данных для получения необходимой информации.

Основными компонентами ИС являются:

– собственно *база данных* (БД), содержащая необходимую информацию и описание структуры хранимых данных;

– *система управления базой данных* (СУБД), выполняющая типовые процедуры управления данными;

– прикладная программа (*приложение пользователя*), реализующая требуемый алгоритм ведения диалога пользователя с информационной системой для обслуживания БД и решения всего комплекса задач обработки данных.

**2. Э Т А П Ы Р А З Р А Б О Т К И И Н Ф О Р М А Ц И О Н Н О Й**

**С И С Т Е М Ы**

*Автоматизация предметной области* может осуществляться в различ-ных масштабах – от отдельных процедур обработки данных, задач и комплек-сов задач управления до создания функционально полной автоматизирован-ной информационной системы (АИС ).

Процессу проектирования АИС предшествует *этап анализа предметной области*. На этом этапе осуществляется:

– определение, исходя из потребностей *всех* пользователей, *информации*, которая будет храниться в БД, ее источника;

– выявление имеющихся и перспективных *задач* обработки данных;

– *документирование* результатов анализа.

Следовательно, процесс построения (разработки) информационно-программного комплекса АИС можно разбить на две части:

– определение состава решаемых задач;

– создание интегрированной базы данных .

Эти две части взаимосвязаны в том смысле, что структура БД зависит от задач обработки данных заданной предметной области.

Таким образом, *база данных* – это некоторая *модель предметной области*, т.е. в БД находят отражение только те факты о ПО, которые необходимы для функционирования ИС.

*Этап разработки БД* предшествует *этапу разработки приложения пользователя* или проходит параллельно с ним; при этом фазы проектирования и реализации могут перекрываться.

Если БД спроектирована, то алгоритм обработки данных конкретной задачи отражает процесс формирования выходных документов на основании выборок из БД.

*Реализация конкретной задачи* осуществляется с помощью средств СУБД, ориентированных на конечного пользователя: запросы, экранные формы, отчеты, макросы, стандартные программы.

При разработке алгоритмов решения каждой задачи следует ориентироваться на укрупненные операции обработки данных, легко реализуемые запросами или отчетами.

*Диалоговое приложение пользователя* (прикладная программа) объединяет всю технологию обработки данных ПО, включая загрузку, ведение БД и решение всего комплекса задач.

Программа должна иметь “дружественный” интерфейс и содержать набор команд решения конкретных задач:

– создание нового набора (таблицы, списка) данных;

– добавление, удаление, изменение, просмотр данных;

– сортировка данных;

– выбор данных по запросам;

– формирование выходных документов (с возможностью просмотра на экране и вывода на печать).

*Интерфейс пользователя* должен строиться на основе *иерархических меню* с использованием диалоговых окон, кнопок, “горячих клавиш” и т.п.

При реализации ИС следует использовать инструментальные средства, предоставляемые системой управления базами данных (СУБД) :

– *генераторы* ( конструкторы, мастера ) экранных форм и отчетов для создания диалоговых окон для ввода данных и выходных (отчетных) документов;

– *языки запросов* для поиска и фильтрации данных;

– *языки программирования* для реализации алгоритмов обработки информации.

**3. О Т Ч Е Т Н А Я Д О К У М Е Н Т А Ц И Я**

Все этапы создания базы данных и разработки информационной системы должны быть документированы. В ходе проектирования и реализации создается *рабочая* (промежуточная) документация: описания, схемы, тесты, распечатки. Некоторые из рабочих документов в дальнейшем войдут в состав *отчетной* (окончательной) документации.

*Отчетная документация* по курсовому проекту должна включать следующие разделы (далее описано содержание курсовой работы):

***1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ***

Описание предметной области (ПО) должно охватывать *реальные объекты и процессы*, содержать всю необходимую информацию для удовлетворения предполагаемых *запросов пользователя* и определять потребности в обработке данных – *конкретные задачи пользователя*.

Должны быть приведены *ограничения ПО*, касающиеся выполнения конкретного индивидуального задания.

Информация рассматриваемой ПО может представляться входными и выходными документами (с приложением форм документов) и справочниками.

Если пользователи базы данных различаются по уровню компетенции или форме представления запросов, то указываются возможности *доступа пользователя* к тем или иным данным.

***2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ***

База данных – это *датологическое* (в виде данных) представление *информационной модели предметной области*.

Процесс разработки БД представляет собой процесс реализации отображения.

Наибольшее распространение получил трехуровневый подход к проектированию *модели данных*, включающий *внешний, концептуальный* и *внутренний* уровни представления данных.

При таком подходе на *внешнем уровне* реализуются модели предметной области в виде, требуемом для отдельных пользователей. На *концептуальном уровне* поддерживается модель ПО для всех приложений. Хранимые данные также представляют ПО для всех приложений, но они выделены в отдельный – *внутренний уровень*.

В процессе проектирования БД разрабатываются *схемы моделей* названных уровней, проверяется возможность отображения объектов модели одного уровня объектами модели другого уровня.

При такой архитектуре БД обладает высокой способностью адаптации к возможным изменениям как в самих данных, так и в прикладных программах.

Основным уровням абстрагирования предшествует еще один – *информационный*. Модель этого уровня – *инфологическая модель* ПО – должна выражать информацию о ПО в виде, независимом от используемой СУБД и опираться на знания всех пользователей.

***2.1. О п и с а н и е Б Д в т е р м и н а х о б ъ е к т о в П О***

Проектирование БД начинается с предварительной структуризации предметной области: объекты реального мира подвергаются классификации, фиксируется совокупность подлежащих отображению в БД объектов. Для каждого объекта фиксируется совокупность свойств, посредством которых будут описываться конкретные экземпляры объекта, и отношения (взаимосвязи) с другими объектами. Затем решаются вопросы о том, какая информация об объектах должна быть представлена в БД и как ее представить с помощью данных.

*Объектная система* имеет следующие основные составляющие: объект, свойство, связь (объектное отношение).

*Объект –* это то, о чем накапливается информация.

Каждый объект характеризуется определенным состоянием, которое описывается с помощью ограниченного набора свойств и связей (отношений) с другими объектами.

*Свойства* объекта могут не зависеть от его связей с другими объектами, т.е. являются локальными. Если свойства объекта зависят от связей с другими объектами, то они называются реляционными.

*Связь* между объектами в зависимости от числа входящих в нее объектов характеризуется степенью: *n* = 2,3,...k.

На этом этапе проектирования базы данных необходимо определить:

– какие *объекты* важны для применения;

– какие *свойства* могут иметь эти объекты;

– какие *связи* существуют между объектами;

– какие *имена* можно присвоить отдельным составляющим *объектной системы*.

***2.2. П о с т р о е н и е и н ф о р м а ц и о н н о й с т р у к т у р ы П О***

*Концептуальная модель* применяется для структурирования ПО с учетом информационных потребностей самой ПО и информационных интересов пользователей системы и независима от конкретной СУБД.

Для проектирования концептуальной схемы (информационной структуры ПО ) можно использовать различные модели, например, *бинарные модели* и *модели «сущность – связь»*.

Из моделей типа «сущность – связь» наиболее известна модель П.Чена, или ER - модель. Общим для всех моделей этого типа является использование трех основных конструкций: сущность, атрибут и связь.

*Сущность* – собирательное понятие, некоторая абстракция реально существующего объекта, процесса или явления, о котором необходимо хранить информацию.

*Тип сущности* определяет множество подобных экземпляров объекта, а *экземпляр сущности* – конкретный экземпляр объекта. Каждый рассматриваемый в модели *тип* сущности должен быть *поименован*.

*Атрибут* – поименованная характеристика сущности, которая принимает значение из некоторого множества значений. В модели атрибут выступает в качестве средства, с помощью которого моделируются свойства сущностей.

*Связь* – средство представления отношения между сущностями.

Могут встречаться бинарные (между двумя сущностями ) и в общем случае *n* - арные связи.

Для каждой сущности необходимо указать идентификатор, служащий для *однозначного распознавания* экземпляров сущности. В качестве идентификатора служит один атрибут или совокупность атрибутов – составной атрибут, который называют *ключом*. Если совокупность атрибутов, описывающих объект, не содержит ключа, то в состав атрибутов вводится специальный атрибут, выступающий в качестве ключа. Во многих случаях это некоторый последовательный номер.

Один и тот же объект может иметь несколько ключей. Один из них назначается *первичным* ( главным ) *ключом*, все остальные ключи объекта называются *возможными ключами*.

Ключ должен выполнять свою главную задачу – однозначной идентификации экземпляра объекта – и включать в свой состав минимально необходимое количество атрибутов.

На языке ER - модели концептуальная схема может быть представлена ERD ( ER *- диаграммой*), в которой множество сущностей обозначается прямоугольниками, множество связей – ромбами. На ER - диаграмме допустимо обозначать множество атрибутов овалами, соединяя их с соответствующими типами сущностей; идентифицирующие атрибуты подчеркиваются.

***2.3. П р е д с т а в л е н и е Б Д р е л я ц и о н н о й м о д е л ь ю***

Основной задачей *логического проектирования* является разработка *логической схемы,* ориентированной на выбранную СУБД. Так как подавляющее большинство современных СУБД – реляционные, то и концептуальную модель БД следует отображать на *реляционную модель*.

В основе реляционной модели используется понятие “отношения”, которое используется для представления

1) набора экземпляров объекта (сущности),

2) отношений (связей ) между объектами.

Отношение представляется как определенным образом организованная *таблица* (см. раздел 1).

Для отображения информационной структуры ПО на логическую схему реляционной БД следует получить ответы на вопросы:

– сколько *таблиц* и какие должна включать БД;

– каковы *степень* ( число столбцов ) и состав каждой таблицы;

– какие атрибуты ( поля ) используются в качестве *ключей*;

– как устанавливаются *связи* между разными таблицами:

а) использование в разных таблицах одного и того же ключа,

б) помещение ключа одной таблицы в качестве атрибута ( поля ) в записи другой таблицы,

в) создание специальных связующих таблиц;

– как обеспечить полноту, непротиворечивость и согласованность информации, хранящейся в БД.

Для уменьшения избыточности информации и исключения аномалий выполняется *нормализация*  исходных схем отношений проекта БД.

***3. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ***

Дается общая характеристика информационной системы: назначение и круг пользователей системы, основные процедуры обработки данных, компоненты системы с точки зрения их организации.

***3.1. Ф у н к ц и и и н ф о р м а ц и о н н о й с и с т е м ы***

*Поясняются назначение системы и приводится перечень основных процедур и обрабатываемых данных.*

Для каждого объекта БД указывается набор типовых операций обработки данных:

– ввод записей, просмотр записей;

– редактирование (изменение значений полей записей, добавление и удаление записей);

– сортировка;

– поиск записей с указанием ключей поиска.

Отдельно описываются конкретные для заданной ПО задачи пользователя с указанием способа доступа к информации, если ИС предназначена для использования пользователями разного класса.

***3.2. А р х и т е к т у р а и н ф о р м а ц и о н н о й с и с т е м ы***

*Словесное или графическое описание системы.*

*Архитектура системы* включает в себя структуру как всей системы в целом, так и ее подсистем и отдельных элементов. Она описывает компоненты структуры и отношения, в которых они находятся друг к другу.

Для организации эффективной работы пользователя все компоненты системы должны быть *сгруппированы по функциональному назначению*.

Для демонстрации связей, существующих между отдельными компонентами системы, используются различные *графические схемы*. Для представления архитектуры системы в целом чаще всего используется *иерархическая диаграмма* – древоподобная схема, показывающая какие модули ( базовые функциональные элементы, блоки системы ) осуществляют вызовы других модулей, и каких именно.

При этом графические материалы должны быть дополнены словесными пояснениями.

***4. КОНТРОЛЬНЫЙ ПРИМЕР***

*Контрольный (или тестовый) пример* представляет собой упрощенный вариант реальной задачи, просчитываемый вручную вплоть до получения конечного результата. В контрольном примере указываются *требования* к объему и составу данных используемой *исходной информации и результатов решения*. Требования к данным контрольного примера – их представительность, учитывающая особенности информации, указанные в описании ПО.

С помощью контрольного примера проверяют постановку задач обработки данных и работоспособность отдельных программ и информационной системы в целом.

***5. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ***

Выбор программных средств для реализации БД и информационной системы в целом зависит от предметной области и решаемых задач пользователя и при выполнении курсового проекта подлежит согласованию с преподавателем.

***5.1. О п и с а н и е С У Б Д***

*Приводятся основные характеристики и возможности выбранной системы управления базами данных*.

Одним из основных критериев выбора СУБД является оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему. СУБД, ориентированные на персональные компьютеры, как правило, поддерживают реляционную модель данных.

Выбранная модель данных представляет средства для *описания структуры данных*.

***5.2. О п и с а н и е с т р у к т у р ы Б Д***

Структура реляционной базы данных является адекватным отображением логической модели (см. п. 2.3.), не требующим преобразований.

*Для каждой реляционной таблицы БД приводится ее структура: состав полей, их имена, тип данных и размер каждого поля, ключи таблицы и другие свойства полей.*

Каждое *поле* в таблице должно иметь *уникальное имя*, удовлетворяющее соглашениям об именах объектов в выбранной СУБД.

*Тип данных (*текстовый, числовой, дата и т.д.) определяется значениями, которые предполагается вводить в поле, и операциями, которые будут выполняться с этими значениями.

*Размер поля* задает максимальный размер данных, сохраняемых в поле. Для числовых данных задается также *число десятичных знаков* после запятой.

Имена ключевых полей выделяются.

***5.3. О п и с а н и е и н т е р ф е й с а***

*Описывается структура меню информационной системы и средства ее реализации*.

Приложение пользователя образуется объединением множества задач предметной области. *Интерфейс приложения* объединяет всю технологию обработки данных ПО, которая включает загрузку, ведение БД и решение всего комплекса задач.

Интерфейс пользователя реализуется с помощью

– системы *меню* с использованием мыши и клавиатуры;

– *диалоговых окон* для вывода сообщений и выполнения команд меню;

– “*горячих клавиш*” для быстрого вызова команд;

– *контекстно-зависимой помощи*.

Структура меню – дерево меню – организуется на основе объектов (таблиц) БД (например, Товар, Покупатель, Заказ) или функций приложений (Поступление товара, Продажа) и соответствует архитектуре системы (см. п. 3.2.).

Меню реализуется в виде *иерархического меню* или *кнопочной формы*.

Для повышения эффективности текущей работы пользователей могут быть предусмотрены и реализованы

– средства разграничения прав доступа к информации БД и к ИС в целом;

– средства анализа непротиворечивости БД и обеспечения логической целостности данных;

– средства обработка ошибок.

***5.4. Р е а л и з а ц и я а л г о р и т м о в о б р а б о т к и д а н н ы х***

*Описываются обобщенный алгоритм и алгоритмы решения конкретных задач и средства их реализации*.

Алгоритм отражает последовательность и логику выполнения операций обработки информации, способа формирования результатов решения с указанием последовательности вычислений, расчетных и/или логических формул.

*Обобщенный алгоритм* решения комплекса задач сводится к указанию последовательности обращения к таблицам БД для выборки необходимой информации.

Алгоритм может быть представлен графически (в виде блок-схемы), в виде текста или таблиц решений, а также в аналитическом или операторном виде.

Алгоритм решения задачи и его программная реализация тесно взаимосвязаны. Для реализации практических задач пользователя используются встроенные в СУБД *языки программирования* и другие программные средства. Специфика применяемых инструментальных средств разработки программ могут повлиять на форму и содержание алгоритмов обработки и их реализацию.

Решение конкретной задачи обработки данных БД можно реализовать на основе специальных стандартных программных средств СУБД, выполняющих укрупненные операции обработки данных: *запросов, экранных форм, отчетов*.

1) Создание *экранных форм*

В *экранной форме* (Screen или Form) – пользовательском формуляре для ввода, просмотра и редактирования взаимосвязанных данных БД – поля размещаются в удобном для пользователя порядке. Макет экранной формы соответствует структуре входного документа. В форме могут быть представлены поля данных разных таблиц и добавлены необходимые надписи, иллюстрации, кнопки и другие объекты. Разработка экранной формы осуществляется инструментальными средствами СУБД: *генераторами* (конструкторами, построителями) или *мастерами экранов* (*форм*).

2) Разработка *отчетов*

*Отчет* ( Report ) – специальное средство для вывода данных на экран, принтер, в файл, позволяющее придать данным определенную форму. В отчете можно разместить и оформить извлеченные из БД записи; подсчитать суммы, проценты; включить графики.

Для создания стандартного отчета или отчета в свободной форме используются *генераторы* ( конструкторы) или *мастера отчетов*.

3) Выбор данных из таблиц с помощью *запросов*

*Запрос* ( Query ) – специальное средство отбора записей – содержит точную формулировку критерия отбора, заданного пользователем. Можно ограничить сферу поиска, отсортировать отобранные записи, связать несколько таблиц. СУБД, выполняя запрос, перебирает записи в таблицах БД и отбирает нужные.

Система управления РБД должна для выполнения запросов на выборку информации

– извлекать подмножество столбцов таблицы,

– объединять столбцы разных таблиц,

– включать в выходные таблицы записи, удовлетворяющие определенным условиям.

Для формирования запросов к БД используются “ Запрос по образцу ” (Query by Example QBE ) или язык запросов SQL ( Structured Query Language).

***ЛИТЕРАТУРА***

Приводится список использованной при выполнении курсовой работы литературы в общепринятой форме.

***ПРИЛОЖЕНИЯ***

***П1.*** Руководство пользователя информационной системы Детальное описание функциональных возможностей и технологии работы с программным продуктом для *конечного пользователя*.

***П2***. Руководство программиста, оператора или администратора БД Описание особенностей установки, а также внутренней и внешней структуры программного продукта; требования к базовому программному обеспечению; правила эксплуатации и т.п.

***П3.*** Формы входных и выходных документов

***П4.*** Результаты решения задачи по данным контрольного примера

***П5.*** Тексты программ и процедур обработки данных

##### **4. Т Е М Ы К У Р С О В Ы Х Р А Б О Т**

***4.1. Т е л е ф о н н ы й с п р а в о ч н и к***

Телефонный справочник – информационная система (и база данных), содержащая информацию об абонентах телефонной сети. Телефонный справочник может быть использован как в сфере работы предприятий, предоставляющих услуги телефонной связи, так и в других целях. Например, его можно использовать в работе справочной службы, когда любой человек по фамилии или по номеру телефона абонента может узнать его адрес (если информация об этом абоненте имеется в системе).

В телефонном справочнике хранится следующая информация об *абоненте*: номер телефона, фамилия, имя, отчество, адрес (улица, номер дома, номер квартиры), номер почтового отделения.

Создаваемая система должна автоматизировать процесс ведения телефонного справочника: ввод новых и редактирование уже имеющихся записей об абонентах (обновление и удаление данных). Система также должна обеспечивать поиск по различным критериям, даже в том случае, если часть символов в полях, входящих в условия поиска, неизвестна (в этом случае допустимо использование символов: “\*” – вместо нескольких неизвестных символов и “?” – вместо одного неизвестного символа). Конечно, в этом случае результат поиска может быть неоднозначным.

Рекомендация: в целях исключения дублирования информации при хранении адреса рекомендуется выделить в отдельную таблицу *список улиц* (понятие улицы включает и другие указатели адреса: переулок, тупик, проспект, площадь и другие возможные).

***4.2. С п е ц и а л и з и р о в а н н а я б и б л и о т е к а***

Рассмотрим специализированную библиотеку, которая располагает *книжным фондом* определенной тематической направленности. Предполагается, что каждая книга фонда может быть как в одном экземпляре, так и в нескольких. Поэтому каждой книге соответствует уникальный инвентарный номер и библиотечный код. Данные о книге содержатся в библиографической карточке, карточки объединяются в *каталоги*. Существует два вида каталогов: алфавитный и тематический; в алфавитном каталоге карточки отсортированы по фамилии автора, а в тематическом – сначала по темам, а в пределах каждой темы – по фамилии автора.

Библиотека *выдает книги* читателям во временное пользование. При записи в библиотеку каждому *читателю* присваивается порядковый номер, ему выдается читательский билет и для него заводится учетная карточка. Учетная карточка кроме данных о читателе в дальнейшем будет содержать информацию о выданных и возвращенных книгах.

Данные, характеризующие работу библиотеки с книгами и читателями, можно сгруппировать следующим образом:

– *книжный фонд* (инвентарный номер книги, библиотечный код книги, отметка о выдаче/возвращении);

– *каталог* (библиотечный код книги, автор, название, издательство, год издания, количество страниц, тема, цена);

– *читатели* (номер читательского билета, фамилия, имя, отчество, домашний адрес, домашний телефон, рабочий телефон);

– *выдача книг* (инвентарный номер книги, номер читательского билета, дата выдачи, запланированная дата возврата).

Создаваемая информационная система предназначена, прежде всего, для ведения данных: о книгах (регистрация новых поступлений, списание литературы), о читателях (регистрация новых читателей, удаление информации о выбывших читателях), а также о перемещении книг между библиотекой и читателями, что должно найти отражение в таблицах *книжный фонд* и *выдача книг*. Кроме того, в системе должны быть реализованы возможности просмотра и поиска как среди книг, так и среди читателей.

***4.3. И з д а т е л ь с т в о***

Издательство – предприятие, занимающееся выпуском разнообразной печатной продукции. Издательство заключает договор с *заказчиком* (клиентом) на выполнение *заказа*. Заказчиком может выступать частное лицо или организация. Частное лицо может быть автором издания (или одним из авторов, если их несколько) или представителем автора. Организация для контактов с издательством также имеет своего представителя – контактную персону.

Заказ может быть книгой, брошюрой, рекламным проспектом, буклетом, бюллетенем для голосования или каким-либо другим видом издательской продукции. Подготовленные издательством материалы заказчика печатаются в *типографиях*, где издательство размещает свои заказы.

Информацию о работе издательства можно сгруппировать следующим образом:

– *сведения о заказчиках* (частное лицо или организация, личные данные контактной персоны, адрес, телефон, факс);

– *сведения о заказах* (номер заказа, заказчик, вид печатной продукции, издание, типография, дата приема заказа, отметка о выполнении, дата выполнения заказа);

– *сведения об изданиях* (код издания, автор и название, объем в печатных листах, тираж, номер заказа);

– *сведения об авторах* (личные данные: Ф.И.О., домашний адрес, телефон; дополнительные сведения);

– *сведения о типографиях* (название, адрес, телефон).

***4.4. Х о з р а с ч е т н а я п о л и к л и н и к а***

Рассмотрим хозрасчетную поликлинику, например, стоматологическую. Поликлиника ведет прием и учет пациентов, учет их посещений (визитов) и учет обслуживания пациентов специалистами (врачами) поликлиники. Существует необходимость в хранении информации обо всех посещениях поликлиники пациентами и о том, на приеме у каких специалистов они находились.

Хранимую информацию о деятельности хозрасчетной поликлиники и ее пациентах можно сгруппировать так:

– *пациент* (номер истории болезни, Ф.И.О. пациента, домашний адрес пациента, телефон);

– *специалист* (личный номер специалиста, Ф.И.О. специалиста, специальность, домашний адрес, телефон);

– *визиты* (пациент, специалист, визит первый или повторный, дата визита, анамнез, диагноз, лечение, стоимость лекарств, стоимость услуг).

Необходимо обеспечить ввод, хранение и, возможно, редактирование данных. В определенных случаях необходимо выполнять удаление данных. Например, можно удалить информацию обо всех визитах некоторого пациента, если после его последнего визита прошел определенный срок (например, 3 года), а данные о самом пациенте перенести в архив (или также удалить).

Необходимо предусмотреть поиск сведений о пациентах как по фамилии, так и по номеру истории болезни.

Так как поликлиника хозрасчетная, то за каждое посещение пациенту выписывается счет, который он должен оплатить.

Кроме задач, перечисленных выше, могут быть также решены и другие задачи, например:

– подсчет выручки каждого специалиста за определенный период (день, месяц);

– подсчет выручки поликлиники в целом за определенный период (день, месяц);

– подсчет оплаченной суммы за лекарства за определенный период (день, месяц).

Также могут решаться задачи статистической обработки данных. Например, подсчет количества посещений поликлиники за месяц в целом и по каждой группе специалистов. А затем на основании этих данных можно построить графики посещений поликлиники по месяцам.

***4.5. А т е л ь е м о д***

Ателье мод выполняет *заказы клиентов* на индивидуальный пошив одежды. В ателье существует *каталог моделей* и *каталог тканей*. По каталогу моделей клиент выбирает модель, а по каталогу тканей – ткань, из которой будет выполнена модель, и заказывает ее пошив в ателье.

*Заказ* каждого клиента содержит: Ф.И.О. клиента, информацию о модели (ее номер из каталога моделей), информацию о ткани (номер из каталога тканей), Ф.И.О. закройщика (исполнителя заказа), дату приема заказа, дату примерки, отметку о выполнении заказа, дату выполнения заказа.

В *каталоге моделей* каждая модель имеет уникальный номер, для каждой модели указывается рекомендуемая ткань, необходимый расход ткани для данной модели с учетом ширины ткани, цена готовой модели, включающая цену ткани и стоимость пошива изделия.

В *каталоге тканей* каждая ткань имеет уникальный номер, название, а также указываются ее ширина и цена за 1 метр.

В ателье может быть еще и *склад тканей.* В *книге учета тканей* на складе для каждой ткани указывается общий метраж, который изменяется, если принимается заказ на изготовление модели из данной ткани.

***4.6. О п т о в ы й с к л а д***

Склад осуществляет продажу товаров оптом. Любая фирма, занимающаяся продажей товаров в розницу, закупает необходимые ей товары на складе, который служит посредником между производителями и продавцами.

На склад *товар* поступает от некоторой *фирмы-поставщика*, в свою очередь склад продает товар *фирме-покупателю*, заключая с ним *сделку о продаже* товара.

Деятельность оптового склада характеризуется следующей информацией, которую можно объединить в группы следующим образом:

– *поставщики* (код поставщика, название фирмы-поставщика, адрес, телефон);

– *покупатели* (код покупателя, название фирмы-покупателя, адрес, телефон);

– *товар на складе* (код товара, поставщик, название товара, единицы измерения, количество, цена покупки за единицу товара, цена продажи за единицу товара);

– *сделки о продаже* (код товара, поставщик, покупатель, количество проданного товара).

На основании описанных данных необходимо вести учет поставщиков, покупателей, продаж, движения товара на складе. Кроме того, можно делать выводы о работе склада, спросе на определенные товары, выгодности работы с некоторыми поставщиками и покупателями.

# ***4.7. Т о р г о в о - з а к у п о ч н о е п р е д п р и я т и е***

Торгово-закупочное предприятие имеет склад, содержащий определенные виды товаров, например, продовольственные товары. Предприятие имеет штат сотрудников, являющихся *агентами-реализаторами*. Предприятие выдает агенту *товар*, устанавливая цену его продажи. Агент-реализатор оплачивает выданный товар не сразу, а по мере его реализации, оформляя *приходные кассовые ордера*. С каждой единицы проданного товара агент получает оплату, установленную предприятием.

Данные, характеризующие торгово-закупочное предприятие, могут быть сгруппированы следующим образом:

– *агенты-реализаторы* (код агента, Ф.И.О. агента, адрес, телефон);

– *товары* (код товара, название товара, отпускная цена);

– *товары на складе* (код товара, количество);

– *товары у агента* (код агента, код товара, количество);

– *приходный ордер* (код агента, код товара, количество единиц реализованного товара, общая стоимость реализованного товара).

Необходимо вести учет движения товаров как на складе, так и у агентов-реализаторов. Кроме того, предприятие проводит операции: по новым поступлениям товара, по выдаче товара агенту, по расчету с агентом за реализованный товар.

# ***4.8. А в т о с а л о н***

Существует некоторая фирма, торгующая автомобилями. Автомобиль выступает в качестве *товара* и как товар имеет определенные характеристики. Кроме того, на каждый автомобиль имеются исчерпывающие *технические данные*. Фирма имеет своих *клиентов* – покупателей автомобилей, сведения о которых хранит в течение определенного времени.

Деятельность фирмы может быть описана данными, сгруппированными следующим образом:

– *товар* (код товара, страна-изготовитель, марка автомобиля, модель, наличие на складе (да, нет, когда будет), цена);

– *технические данные* (код товара, тип кузова, количество дверей, количество мест, тип двигателя, расположение двигателя, рабочий объем двигателя);

– *клиенты* (код товара, ФИО клиента, паспортные данные (серия, но-мер), домашний адрес, телефон, доставка (да, нет), вид оплаты (пере-числение или наличные: кредит или сразу)).

Необходимо обеспечить ввод, редактирование и просмотр данных в удобной для пользователя форме.

Предполагается также решение следующих задач:

– выдать информацию о наличии автомобилей определенной марки и модели;

– выдать технические данные заданной модели;

– выдать информацию обо всех проданных моделях некоторой марки, значение которой вводится в качестве параметра;

– посчитать сумму продаж моделей каждой марки и общую сумму про-даж;

– выдать полную или частичную информацию о клиентах фирмы;

– выдать списки клиентов и автомобилей по виду оплаты.

# ***4.9. П р о д а ж а п о д е р ж а н н ы х а в т о м о б и л е й***

Фирма по продаже подержанных автомобилей работает с физическими лицами – *клиентами* фирмы, имеющими подержанный автомобиль или автомобили и желающими продать их через фирму. Непосредственной продажей автомобилей занимаются сотрудники фирмы – *дилеры*. На каждый предлагаемый в продажу автомобиль фирма заключает с клиентом *договор*, содержащий данные о клиенте, необходимые сведения об автомобиле, а также данные о дилере, обслуживающем этот договор.

Данные, характеризующие деятельность фирмы, могут быть сгруппированы следующим образом:

– *клиенты* (код клиента, фамилия, имя, отчество, город, адрес, контактный телефон);

– *дилеры* (код дилера, фамилия, имя, отчество, фотография, домашний адрес, телефон);

– *договоры* (код договора, код клиента, код дилера, дата заключения договора, марка автомобиля, фото автомобиля, дата выпуска, пробег, дата продажи, цена продажи, размер комиссионных, примечание).

В создаваемой информационной системе необходимо обеспечить ввод и редактирование данных. Кроме того, необходимо выдавать информацию о клиентах и предлагаемых ими автомобилях, а также информацию о деятельности дилеров (с перечислением договоров) и клиентах, которые они обслуживают. Могут быть выполнены разнообразные запросы, например:

– посчитать количество договоров, заключенных с каждым клиентом;

– посчитать количество договоров, обслуживаемых каждым дилером;

– выдать некоторую информацию (например: данные дилера, дата заключения договора, данные клиента, отметка о продаже) обо всех договорах, договорах за некоторый промежуток времени или договорах, удовлетворяющих определенному условию.

***4.10.*** ***А с с о ц и а ц и я к р е с т ь я н с к и х ф е р м е р с к и х***

***х о з я й с т в***

Предполагается, что существует некая региональная организация, назовем ее условно ассоциацией, которая является организационным объединением крестьянских фермерских хозяйств (КФХ). Ассоциация ведет учет зарегистрированных *фермерских хозяйств*, собирает информацию о видах их деятельности, а также о предлагаемой хозяйствами *продукции* и ее цене, ведет статистический учет. Вид деятельности хозяйства определяет его *специализацию*, например: овощеводство, животноводство, виноградарство и другие. В каждой специализации имеются виды производимых товаров.

Ассоциация располагает следующей информацией о хозяйствах:

– *хозяйство* (код КФХ, название хозяйства, *специализация*, личные данные фермера, регион, адрес, телефон);

– *продукция* (код КФХ, произведенный товар, единицы измерения, цена за единицу товара, предлагаемое количество).

Необходимо обеспечить ввод и обновление данных, возможности анализа товаров и цен. Необходимо также предусмотреть возможность получение информации о деятельности конкретных хозяйств: их продукции и ценах, а также получение информации о конкретных видах товаров: их производителях и ценах.

# ***4.11. П а с с а ж и р с к о е а в т о п р е д п р и я т и е***

Муниципальное автопредприятие осуществляет пассажирские перевозки на внутригородских маршрутах.

Автопредприятие имеет *парк автобусов*, которые работают на определенных *маршрутах*. Работу автопредприятия обеспечивает *персонал* предприятия, который можно разделить по категориям занимаемых должностей на администрацию, инженерно-технический персонал и персонал, обслуживающий маршруты (водители, кондукторы). Выезжая на маршрут, водитель автобуса получает *маршрутный лист* (или путевой лист), содержащий данные об автобусе, маршруте, режиме работы, водителе, кондукторе.

Автопредприятие описывается данными, которые могут быть сгруппированы следующим образом:

– *автобусы* (бортовой номер автобуса, гос.номер автобуса, марка, год выпуска, пробег);

– *маршруты* (номер маршрута, маршрут, протяженность маршрута в км, среднее время одного рейса, плановое количество рейсов за смену);

– *личные данные персонала* (табельный номер, Ф.И.О. сотрудника, дата рождения, домашний адрес, домашний телефон, рабочий телефон);

– *учетные данные персонала* ( табельный номер, категория, должность, дата приема на работу, номер автобуса ( для водителей и кондукторов));

– *маршрутные листы* (номер маршрута, бортовой номер автобуса, дата, количество выполненных рейсов, водитель, кондуктор).

Возможно, что в реальном автопредприятии учитывается большее коли-чество данных, однако исполнителю задания можно ограничиться перечис-ленными. При желании в учетных данных персонала можно ввести данные для оплаты труда, если предполагается автоматизация начисления зарплаты. В маршрутных листах можно ввести плановую и фактическую выручки за смену соответственно.

Создаваемая информационная система, прежде всего, должна обеспечивать ввод и редактирование данных в удобной для пользователя форме. Кроме того, можно сформулировать разнообразные запросы как по кадровому составу предприятия, так и по характеристике и техническому обеспечению маршрутов. Например,

– выдать полную или частичную информацию по персоналу;

– выдать полную или частичную информацию по автобусному парку;

– выдать полную или частичную информацию по маршрутам.

В частности, запросы могут быть такими:

– выдать список сотрудников администрации с указанием должности;

– на определенную дату для всех номеров маршрутов выдать информацию о количестве автобусов, обслуживающих каждый маршрут;

– по каждому номеру маршрута и дате (параметры запроса) выдать информацию об автобусах, обслуживающих маршрут: бортовой номер, марка, гос.номер автобуса.

Могут решаться и другие задачи. Например, по итогам работы за месяц посчитать количество рейсов, выполненных каждым автобусом или на каждом маршруте. По итогам работы за месяц посчитать количество смен, отработанных каждым водителем и кондуктором.

# ***4.12. М е ж д у г о р о д н ы е п а с с а ж и р с к и е п е р е в о з к и***

Рассмотрим автовокзал, который занимается обслуживанием и учетом пассажиров на междугородных автобусных маршрутах. На автовокзале имеется *расписание движения автобусов*, содержащее информацию о маршрутах и рейсах. Кроме того, на автовокзале имеется *справочное бюро*, в котором можно получить информацию о наличии мест на определенный рейс конкретной даты. И, наконец, на автовокзале есть *кассы*, в которых *пассажир* может приобрести билет. Кассы начинают *предварительную продажу* билетов за определенный промежуток времени до дня отправления автобуса (например, за 10 дней).

Необходимо построить такую базу данных, в которой хранится информация как о технических характеристиках маршрутов, содержащаяся в расписании, так и информация о наличии мест на рейсы, и информация о пассажирах, купивших билеты на определенный рейс.

Ниже предлагается вариант организации информации о рейсах и пассажирах (однако исполнитель задания может предложить собственный вариант организации данных).

Администратор базы данных к началу продажи билетов на рейс, т.е., например, за 10 дней до дня отправления, создает таблицу, соответствующую рейсу и дате отправления. Назовем эту таблицу *схемой рейса*, в момент создания схема рейса содержит только номера мест, а в процессе продажи билетов схема будет заполняться информацией о пассажире, которому продан билет на соответствующее место. Причем, в схеме может находиться собственно информация о пассажире (фамилия, имя, отчество) или ссылка на нее.

После того как рейс выполнен, таблица со схемой рейса удаляется, но перед этим список пассажиров этого рейса можно отправить в *архив*, который хранится в течение определенного срока, а информацию о рейсе – в таблицу *выполненных рейсов*. Если по какой-либо причине рейс отменяется, то информация об этом помещается в таблицу *отмененных рейсов*.

При описанной организации данные можно сгруппировать следующим образом:

– *расписание рейсов* (номер рейса, маршрут, тип автобуса, расстояние в км, дни отправления, время отправления, время прибытия, цена билета);

– *схема рейса* (номер места, информация о пассажире: фамилия, имя, отчество);

– *адреса схем рейсов* (номер рейса, дата отправления, ссылка на схему рейса);

– *архив пассажиров* (номер рейса, дата отправления, номер места, фамилия, имя, отчество);

– *выполненные рейсы* (номер рейса, дата отправления, количество проданных мест, полученная сумма);

– *отмененные рейсы* (номер рейса, дата, причина отмены рейса).

В создаваемой системе, прежде всего, необходимо обеспечить ведение данных: организацию таблиц для схем рейсов и ссылок на них, ввод и редактирование данных в таблицах, архивирование данных, удаление таблиц.

Кроме того, в рассматриваемой задаче представляют интерес запросы. Например, следующие:

– наличие свободных мест на рейс;

– количество пассажиров уже выполненного рейса, доходность рейса;

– список всех пассажиров определенного рейса (выполненного или того, на который идет продажа билетов);

– определить, покупал ли билет человек с заданной фамилией и, если покупал, то на какой рейс.

Количественные данные рассматриваемой задачи позволяют также вести их статистический учет, можно для наглядности использовать графическое представление данных. Например, определять количество перевезенных пассажиров и объем перевозок (в денежном выражении) по дням, по месяцам в целом по всем направлениям или по определенному маршруту.

***4.13. А г е н т с т в о п о п р о д а ж е а в и а б и л е т о в***

Агентство занимается продажей авиабилетов на различные рейсы, ведет учет проданных билетов и учет пассажиров, купивших билеты.

Поэтому возникает потребность в хранении и обработке данных, сгруппированных следующим образом:

– *информация о расписании рейсов* (номер рейса, тип самолета, пункт отправления, пункт назначения, дата вылета, время вылета, время полета, цена билета);

– *информация о свободных местах на рейс* (номер рейса, дата вылета, общее количество мест, количество свободных мест);

– *информация о пассажирах*, заказавших билет (фамилия, имя, отчество, предъявленный документ, его серия и номер, номер рейса, дата вылета).

Особенность данной задачи состоит в том, что информация в базе данных может использоваться как пассажирами (например, для получения сведений о расписании и наличии свободных мест на рейс), так и служащими агентства: кассирами и диспетчерами (администраторами). Т.е. выделяются три группы пользователей базы данных: *пассажир*, *кассир*, *диспетчер*. В связи с этим возникает проблема разграничения прав доступа этих пользователей к информации базы данных для защиты ее от несанкционированного использования.

Другая особенность рассматриваемой задачи: удаление в архив информации о выполненном рейсе (см. п. 4.12.) и ввод данных о рейсе с таким же номером на очередную дату, отстоящую от даты удаленного в архив рейса на определенный промежуток, например, на 30 дней вперед.

***4.14. Г о с т и н и ц а***

Рассмотрим возможную модель организации размещения и учета проживающих в некоторой гостинице граждан (клиентов).

Предполагаемая гостиница располагает *номерами* с разным уровнем сервиса, комфортности и, соответственно, оплаты. Одной из характеристик номера является его тип. Предположим, что существуют следующие типы номеров: люкс – многокомнатный номер с высоким уровнем сервиса, комфортности и обслуживания; полулюкс – номер меньшей, чем люкс, площади, но с достаточным уровнем сервиса и комфортности; одноместный или двухместный номер с минимальным уровнем сервиса; многоместный номер, также с некоторым уровнем сервиса. Стоимость для номеров типа люкс и полулюкс устанавливается как стоимость всего номера (в сутки), независимо от количества проживающих в номере. Стоимость проживания в одно-, двух- и многоместных номерах устанавливается для одного человека (в сутки). Номера и места в номерах могут бронироваться. При наличии телефона в номере пользование междугородным телефоном оплачивается отдельно по фактическим счетам.

Все прибывающие и размещаемые в гостинице граждане при вселении должны заполнить *карточку регистрации*. Кроме того, для расчетов с клиентами администрация гостиницы заводит *расчетные карточки*, которые содержат данные о размещении клиента, оказанных услугах и всех расчетах с ним. Для номеров типа люкс и полулюкс расчетная карточка может заводиться только для одного клиента, оплачивающего номер.

Любой номер гостиницы имеет *номер*, по которому ведется учет проживающих в гостинице. Это свойство номера в последующих описаниях будем называть *номером комнаты* (независимо от фактического количества комнат в номере).

При выбытии клиента данные о нем сохраняются в *архиве*.

Можно предположить использование следующих свойств (данных) о номерах гостиницы и ее клиентах, которые можно сгруппировать описанным ниже способом:

– *сведения о номерах типа люкс и полулюкс* (номер комнаты, тип номера, занят/свободен, количество комнат, этаж, телефон, стоимость номера в сутки, сведения о бронировании, количество фактически проживаюших);

– *сведения о прочих номерах* (номер комнаты, тип номера, количество мест, этаж, телефон, стоимость проживания одного человека в сутки, количество свободных мест);

– *карточки регистрации* (номер регистрации клиента, номер комнаты, дата прибытия, фамилия, имя, отчество, предъявленный документ, серия и номер документа, дата рождения, пол, домашний адрес, домашний телефон);

– *расчетные карточки* (номер регистрации клиента, номер комнаты, дата и время прибытия, оплата брони, предполагаемая дата убытия, количество оплаченных дней, сумма оплаты, окончательный расчет);

– *архив* (номер регистрации клиента, фамилия, имя, отчество, документ, серия и номер, дата рождения, пол, домашний адрес, домашний телефон, номер комнаты, дата прибытия, дата убытия).

По усмотрению исполнителя можно расширить или, наоборот, уменьшить количество свойств (данных) рассматриваемой задачи.

Создаваемая информационная система предназначена для администрации гостиницы, которая на основании информации о номерах занимается размещением клиентов в соответствии с их запросами. При выбытии клиента информация о номере, в котором он проживал, должна обновляться, а информация о клиенте должна удаляться из рабочих таблиц (карточки регистрации клиентов и карточки учета) и помещаться в архивную таблицу.

Кроме перечисленных задач ведения данных, в системе могут решаться задачи поиска, например, поиск номера или места в номере в соответствии с некоторыми критериями поиска. Другая задача поиска – это поиск клиента, проживающего в гостинице в данный момент или проживавшего в ней ранее.

Кроме того, в рассматриваемой задаче можно организовать статистическую обработку данных; например, учет количества проживающих в гостинице по определенным периодам.

1. **П Р А В И Л А О Ф О Р М Л Е Н И Я К У Р С О В О Й Р А Б О Т Ы**

По объему курсовые проекты должны быть не менее 30 страниц печатного текста.

Курсовая работа должна быть сдана в машинописном виде (на компьютере). Текст должен быть напечатан через 1,5 интервала шрифтом Times New Roman стандартного размера (14 шрифт) с соблюдением установленных размеров отступа от края листа:

- левое поле - 30 мм;

- правое поле - 10 мм;

- верхнее и нижнее поля - по 20 мм.

Титульный лист и содержание не нумеруются, но включены в общую нумерацию.

Содержание проекта, введение, название параграфов, заключение, список литературы, приложение начинается с новой страницы. Каждый самостоятельный раздел нумеруется и озаглавливается аналогично тому, как это сделано в "Содержании". Названия параграфов печатаются симметрично тексту (по центру), 16 шрифтом, жирным. Названия пунктов печатаются с красной строки строчными буквами (кроме первой), 14 шрифтом, жирным.

Расстояние между заголовком параграфа и основным текстом должно быть равно трем (одинарным) межстрочным интервалам.

Рисунки (схемы, графики) должны быть выполнены разборчиво и иметь поясняющие надписи, которые расположены под рисунком (примеры смотрите выше). В текст должна быть ссылка на каждый рисунок (Например, данные… см. на рис.1).

При оформлении таблицы слово «Таблица» пишется в левом верхнем углу (без кавычек) и ставится е порядковый номер. Нумерация может быть сквозной через всю работу или по параграфам. Таблица должна содержать название (табл. 3 - Таблица входных данных и их файлов).

При упоминании о таблице в тексте делается ссылка (табл. 1.1).

Список литературы должен содержать перечень всех источников (не менее 5), использованных при написании курсовой работы. Источники располагаются в алфавитном порядке (по первой букве первого слова. В авторских источниках первым словом считается фамилия автора).

- все источники в перечне нумеруются. Для каждого источника указываются: фамилия и инициалы автора (авторов); полное название книги, статьи; название журнала или сборника статей (для статей);

- название города (Москва и Санкт-Петербург - сокращенно, соответственно М. и СПб., остальные полностью);

- название издательства (если имеется в выходных данных) - для книг;

- год издания (для статей - N журнала и год).

Например:

1. Стуков, С.А. Экономический анализ и рынок / С.А. Стуков. // Бухгалтерский учет. -1991.- №3.- C. 106-111.

2. Ткач, В.И. Международная система учета и отчетности / В.И. Ткач. - М.: Финансы и статистика, 1992. – 528с.

3. Управление по результатам. Пер. с финск. / Под ред. А.Я. Лейтмана. - М.: Прогресс, 1993. – 157с.

**Примеры описаний электронных ресурсов:**

- *Библиографические записи Интернет-ресурсов.*

Исследовано в России [Электронный ресурс]: многопредмет. науч. журн. / Моск. физ.- техн.ин-т. – Электрон. журн. – Долгопрудный: МФТИ, 1998. Режим доступа к журн.:http://zhurnal.mipt.rssi.ru.

Российская государственная библиотека [Электронный ресурс] / Центр информ. технологий РГБ; ред. Власенко Т.В.; Web-мастер Козлова Н.В. – Электрон. дан. – М.: Рос. гос. б-ка, 1997. –Режим доступа: http://www.rsl.ru, свободный. – Загл. с экрана. – Яз. рус., англ.

- *Библиографические записи сборников электронных ресурсов без общего заглавия:*

Английский для бизнесменов [Электронный ресурс]; Английский технический;

Английский для чтения газет и журналов: [к сб. в целом]: курс изучения иностр. яз. Intell. –Электрон. дан. и прогр. – М.: сор. Квант, 1994–1997. – 1 электрон., опт. диск (CD-ROM): зв., цв. –(25 кадр).

- *Библиографические записи электронных ресурсов, состоящие из нескольких отдельных частей (выпусков):*

Большая автомобильная энциклопедия [Электронный ресурс]. – Электрон. дан. – М.: Xelana Media Group, 1998.

Помимо основного текста курсовой проект может содержать приложения. В качестве приложений целесообразно оформлять код программного продукта и основные экраны интерфейса.

В этом случае в основном тексте в необходимых местах должны содержаться ссылки: «см. приложение». Приложение располагается непосредственно за списком литературы, на первом листе приложенных документов в верхнем правом углу пишется слово "ПРИЛОЖЕНИЕ" (а затем его порядковый номер) то же пишется в последней строке содержания.

Каждое приложение должно иметь заголовок, который выравнивается по центру страницы. Приложения располагаются в порядке ссылке на них в основном тексте.

##### **Л И Т Е Р А Т У Р А**

1. Голицина О.Л. и др. Базы данных – М.: Форум, Изд-во Проспект, 2006
2. Залогова Н.П. Microsoft Access.– М. Интеллект-Центр, 2005г
3. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2006
4. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию. – М.: Финансы и статистика, 2005
5. Михеева Е.В. Практикум по информационным технологиям в проф.деятельности. – М.:ТК Велби. Изд. Проспект, 2009
6. Петкович Д. Microsoft SQL Server 2012. Руководство для начинающих. – СПб.: БВХ-Петербург, 2013.
7. Фуфаев Э.В. Базы данных. – М.: Издательский центр «Академия», 2009