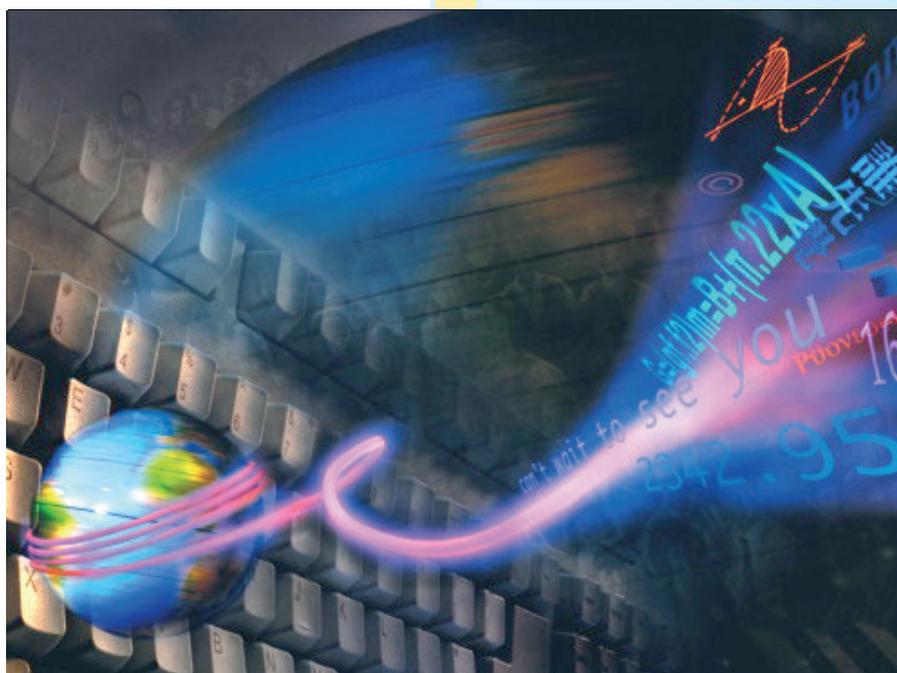


В.Ф. Глазова
А.В. Богданова
А.П. Тонких

ИНФОРМАТИКА

Учебно-методическое
пособие

Часть 2



Министерство образования и науки Российской Федерации
Тольяттинский государственный университет
Институт математики, физики
и информационных технологий
Кафедра «Информатика и вычислительная техника»

В.Ф. Глазова, А.В. Богданова, А.П. Тонких

ИНФОРМАТИКА

Учебно-методическое пособие

В двух частях

Часть 2

Тольятти
Издательство ТГУ
2013

УДК 004(075.8)

ББК 32.81я73

Г524

Рецензенты:

канд. пед. наук, доцент Института коммерции и права
(Тольяттинский филиал) *С.А. Гудкова*;
канд. техн. наук, доцент Тольяттинского государственного
университета *Э.В. Егорова*.

Г524 Глазова, В.Ф. Информатика : учеб.-метод. пособие :
в 2 ч. / В.Ф. Глазова, А.В. Богданова, А.П. Тонких. — Тольят-
ти : Изд-во ТГУ, 2013. — Ч. 2. — 223 с. : обл.

Учебно-методическое пособие (в двух частях) предназна-
чено для обучения дисциплине «Информатика». Вторая
часть состоит из трех разделов, каждый содержит изложение
теоретического материала, вопросы для самоконтроля, зада-
ния для выполнения практических работ, тестовые задания.
Пособие может быть использовано на аудиторных занятиях
и для самостоятельной работы студентов.

Предназначено для студентов системы высшего профес-
сионального образования, обучающихся очно по направле-
ниям подготовки бакалавров 080200 «Менеджмент», 080400
«Управление персоналом», 221400 «Управление качеством».

УДК 004 (075.8)

ББК 32.81я73

Рекомендовано к изданию научно-методическим советом
Тольяттинского государственного университета.

© ФГБОУ ВПО «Тольяттинский
государственный университет», 2013

Предисловие

Вторая часть учебно-методического пособия по дисциплине «Информатика» предназначена для студентов очной формы обучения. Учебные материалы и связанные с ними учебные мероприятия в рамках изучения дисциплины организованы в обучающие модули. Каждый раздел данного пособия соответствует отдельному обучающему модулю дисциплины и имеет следующую структуру: цели изучения; методическая, теоретическая, практическая и контролирующая части.

Методическая часть включает структурную схему работы с учебными материалами модуля, а также ссылки на дополнительные источники информации. Теоретическая часть представляет краткое изложение учебных материалов. В конце каждого параграфа приводятся вопросы и упражнения для самоконтроля.

Практическая часть содержит описание практических работ. Каждая практическая работа включает блок целеполагания, методический блок с рекомендациями по выполнению работы, упражнения или практические задания, пример выполнения задания или описание технологии выполнения задания, варианты заданий. В конце каждого раздела приводятся тестовые задания по изучаемым темам.

Приложение 1 содержит краткий словарь ключевых терминов по всем темам (глоссарий), приложение 2 – ответы на тестовые задания. В приложении 3 приведены вопросы для подготовки к экзамену (зачету), в приложении 4 – варианты для выполнения индивидуального домашнего задания по теме «Базы данных». Описание основных тегов языка HTML дано в приложении 5.

В данном пособии изложены учебные материалы по темам, изучаемым **во втором семестре** в рамках дисциплины «Информатика».

Цель изучения дисциплины – формирование системы знаний и умений в области информационных технологий, составляющих основу базовой информационной компетентности выпускника вуза.

В результате изучения дисциплины во втором семестре студент должен **знать**:

- понятия модели и моделирования, типы и классификацию моделей, виды моделирования;
- разновидности математического моделирования;
- модели организации данных;
- понятие базы данных, системы управления базами данных;
- принципы работы с базами данных в среде СУБД Microsoft Access;
- основы организации компьютерных сетей;
- методы защиты информации;

уметь:

- разрабатывать информационные компьютерные модели решения функциональных и вычислительных задач;
- использовать электронные таблицы для моделирования экономических процессов;
- работать с базами данных в среде СУБД Microsoft Access;
- работать в компьютерных сетях;
- защищать документы от несанкционированного доступа и изменения;

владеть:

- основными методами, способами и средствами компьютерного моделирования;
- навыками работы с базами данных;
- средствами защиты информации;
- навыками работы в локальных и глобальных компьютерных сетях.

Содержание разделов дисциплины «Информатика», относящихся ко второму семестру обучения, представлено в таблице.

Раздел 1. Модели решения функциональных и вычислительных задач	
Тема	Учебные вопросы темы
1.1. Понятия «модель» и «моделирование». Классификация моделей	1. Понятие и свойства моделей. 2. Классификация моделей по различным признакам
1.2. Компьютерное математическое моделирование	1. Этапы математического моделирования. 2. Типы математических моделей. 3. Математическое моделирование стохастических процессов
1.3. Использование в моделировании средств компьютерной графики	1. Разновидности компьютерной графики. 2. Применение в моделировании средств компьютерной графики
1.4. Моделирование средствами табличного процессора Microsoft Excel	1. Построение графиков функций и линий тренда. 2. Использование в моделировании инструментов «Подбор параметра» и «Поиск решения»
Раздел 2. Базы данных	
Тема	Учебные вопросы темы
2.1. Базы данных и модели данных. Основы проектирования реляционных баз данных	1. Базы данных и системы управления базами данных. Классические модели данных: иерархическая, сетевая, реляционная. 2. Основы проектирования реляционных баз данных. Нормализация реляционных таблиц
2.2. Основы работы с СУБД Microsoft Access. Основные объекты базы данных. Работа с таблицами	1. Объекты базы данных Microsoft Access. 2. Работа с таблицами в среде СУБД Microsoft Access
2.3. Основы работы с СУБД Microsoft Access. Работа с запросами, формами и отчетами	1. Типы запросов. 2. Создание запросов к базе данных в среде СУБД Microsoft Access. 3. Создание форм и отчетов в среде СУБД Microsoft Access

Раздел 3. Локальные и глобальные компьютерные сети. Защита информации в сетях	
Тема	Учебные вопросы темы
3.1. Компьютерные сети	1. Понятие, классификация и виды компьютерных сетей. 2. Аппаратные компоненты компьютерных сетей
3.2. Интернет	1. Организация сети Интернет. 2. Сервисы Интернета
3.3. Организация защиты информации	1. Угрозы безопасности информации, их виды. 2. Методы и средства обеспечения безопасности информации
3.4. Антивирусная защита компьютера. Защита информации в локальных и глобальных компьютерных сетях	1. Понятие и виды вредоносных программ. 2. Виды компьютерных вирусов, их классификация. 3. Защита от компьютерных вирусов. 4. Средства защиты информации от несанкционированного доступа. 5. Криптографическая защита информации. 6. Электронная цифровая подпись

При работе с пособием предлагается учитывать следующие соглашения:

- впервые используемые в рамках пособия термины и определения выделены *жирным курсивным* шрифтом;
- команды меню, названия диалоговых полей, кнопок и прочих элементов интерфейса пользователя программ пакета Microsoft Office выделены **жирным** шрифтом;
- команды меню программы, выполняемые последовательно, разделяются тире, например, строка **Файл – Сохранить** предполагает следующую последовательность операций: открыть пункт меню **Файл**, затем выбрать команду **Сохранить**;

- фрагменты текста пособия, на которые рекомендуется обратить особое внимание, отмечены специальным значком ☑.

Описание работы с табличным процессором и системами управления базами данных выполнено для соответствующих приложений пакета Microsoft Office 2003.

Над пособием работал авторский коллектив:

В.Ф. Глазова — раздел «Модели решения функциональных и вычислительных задач»;

А.В. Богданова — раздел «Базы данных»;
и В.Ф. Глазова

А.П. Тонких — раздел «Локальные и глобальные компьютерные сети. Защита информации в сетях».

1. МОДЕЛИ РЕШЕНИЯ ФУНКЦИОНАЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

Изучив материалы этого раздела, вы **узнаете**:

- способы представления моделей, классификацию моделей по различным признакам;
- виды моделирования, способы математического моделирования;
- возможности применения компьютерной графики в моделировании;
- средства моделирования, предоставляемые табличным процессором Microsoft Excel.

Выполнив практические работы, вы **научитесь**:

- аппроксимировать ряд числовых данных аналитической функцией и исследовать параметры математической модели, представленной нелинейным алгебраическим уравнением, в среде табличного процессора;
- обрабатывать матричные математические модели, используя операции с массивами электронной таблицы;
- использовать инструменты «Подбор параметра» и «Поиск решения» Microsoft Excel для моделирования.

Методические рекомендации

Схема работы с учебными материалами данного раздела представлена на рис. 1.1. Теоретическая часть предполагает изучение четырех тем. Каждая тема заканчивается перечнем вопросов и упражнений для самоконтроля. Практическая часть содержит четыре практические работы. Изучение раздела заканчивается выполнением тестовых заданий, ключ к которым можно найти в прил. 2. С дополнительными учебными материалами по данному разделу можно ознакомиться в рекомендованных литературных источниках [2], [7], [8] и интернет-ресурсах [11], [13], список которых представлен в конце пособия.

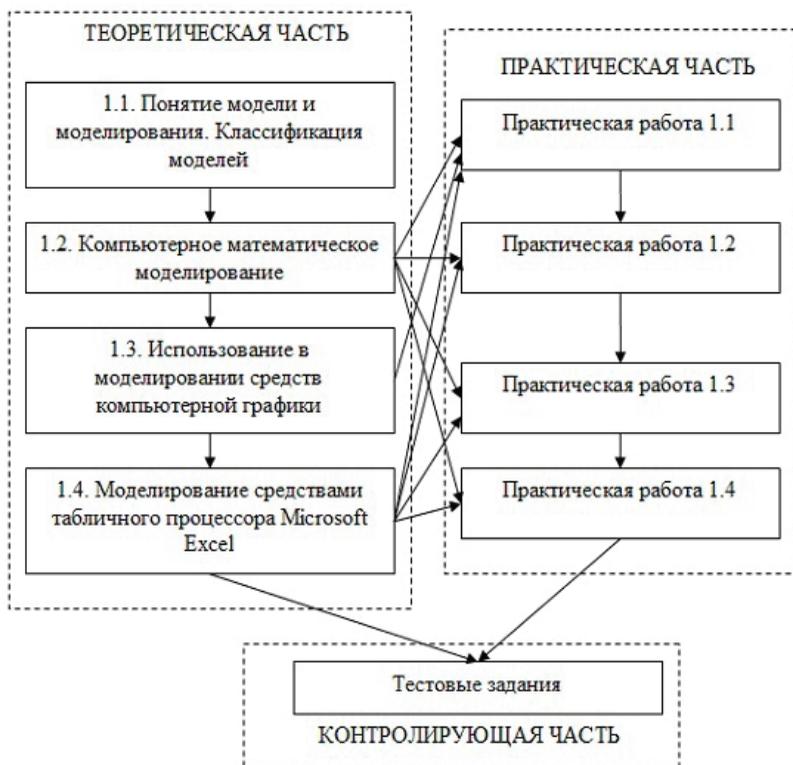


Рис. 1.1. Схема работы с учебными материалами раздела 1

1.1. Понятия «модель» и «моделирование». Классификация моделей

Задачи, которые решает человек в своей образовательной, научно-исследовательской и профессиональной деятельности, делятся на две категории: *вычислительные* и *функциональные*. Цель *вычислительных задач* – расчет параметров, характеристик, обработка данных. *Функциональные задачи* требуют решения при реализации функций управления, проектирования. Например, управление деятельностью торгового предприятия, планирование выпуска продукции, управление перевозкой грузов и т. п.

Процесс решения задачи средствами моделирования отображает схема, показанная на рис. 1.2. Под реальным объектом подразумевается исследуемый объект (система, явление, процесс). *Модель* – это материальный или воображаемый объект, который в процессе познания замещает реальный объект, сохраняя его существенные свойства. *Моделирование* – это процесс исследования реального объекта с помощью модели. Исходный объект называется *прототипом* или *оригиналом*.

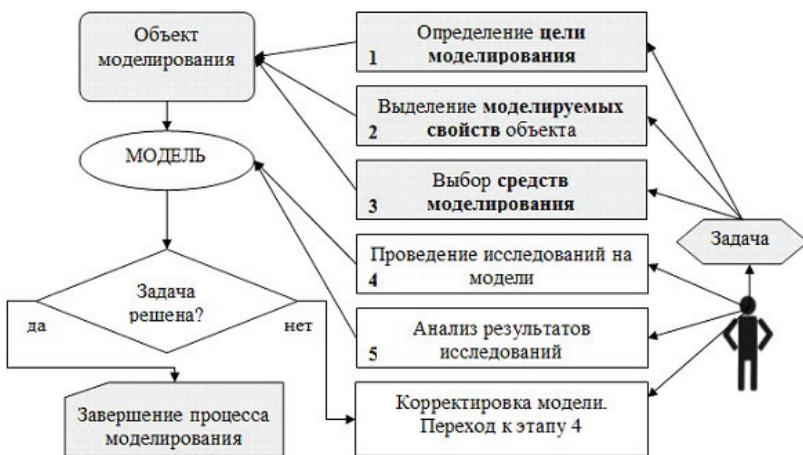


Рис. 1.2. Схема процедуры решения задачи посредством моделирования

Моделировать можно не только материальные объекты, но и процессы. Например, конструкторы используют аэродинамическую трубу для воспроизведения на земле условий полета самолета.

☑ В дальнейшем термин «объект моделирования» будем понимать в широком смысле: это может быть как некоторый вещественный объект (предмет, система), так и реальный процесс.

Модель повторяет не все свойства реального объекта, а только те, которые требуются для ее будущего применения. Поэтому важнейшим в моделировании является понятие цели. *Цель моделирования* – это назначение будущей модели.

Цель определяет те свойства объекта-оригинала, которые должны быть воспроизведены в модели. Иначе говоря, *модель* — это *упрощенное* подобие реального объекта, отражающее *существенные особенности* (свойства) изучаемого реального объекта, отвечающие *цели моделирования*.

К построению модели прибегают в тех случаях, когда использование объекта-оригинала по каким-либо причинам затруднено или невозможно. Такими причинами могут быть, например:

- слишком большой (Солнечная система) или слишком маленький размер объекта (молекула или атом);
- моделируемый процесс протекает слишком быстро (сгорание топлива в двигателе внутреннего сгорания) или слишком медленно (процесс возникновения жизни на Земле);
- исследование объекта может оказаться опасным для окружающих (атомный взрыв);
- объект-оригинал может быть разрушен в процессе исследования (исследование прочностных характеристик конструкции самолета).

Для одного и того же объекта можно создать множество различных моделей. Какую модель выбрать — зависит от цели моделирования, определяемой в соответствии с решаемой задачей. С другой стороны, одна и та же модель может представлять разные объекты. Например, математические модели процесса распространения инфекционной болезни и процесса радиоактивного распада являются одинаковыми с точки зрения их математического описания.

В модели сконцентрированы те свойства реального объекта, которые подлежат изучению в конкретном исследовании. Исключение несущественных свойств при построении модели очень важно, так как их наличие в реальном объекте создает помехи, на фоне которых труднее выявить изучаемые свойства и закономерности.

Существует ряд общих требований к свойствам, которым должны удовлетворять модели:

- *адекватность* — достаточно точное отображение свойств объекта;

- **конечность** – модель отображает оригинал лишь в конечном числе его отношений и свойств;
- **полнота (информативность)** – предоставление исследователю всей необходимой информации об объекте в рамках гипотез, принятых при построении модели;
- **упрощенность** – модель отображает только существенные стороны объекта;
- **гибкость** – возможность воспроизведения различных ситуаций во всем диапазоне изменения условий и параметров;
- приемлемая для имеющегося времени и программных средств **трудоемкость разработки модели**.

Классификация моделей

Классификация – это разделение объектов на группы, имеющие один или несколько общих признаков. В зависимости от признака классификации одни и те же модели могут быть отнесены к разным классам.

Классификация по области использования модели представлена на рис. 1.3.



Рис. 1.3. Классификация моделей по области использования

Учебные модели – наглядные пособия, тренажеры, обучающие программы.

Игровые модели – это экономические, военные, деловые игры. Они репетируют поведение объекта в различных ситуациях.

Исследовательские модели – создаются для исследования процессов или явлений, например, стенды для проверки электронной аппаратуры.

Опытные модели – это уменьшенные или увеличенные копии объектов. Их используют для исследования объекта и прогнозирования его будущих характеристик (например, опытная модель проектируемого автомобиля).

Имитационные модели – имитируют реальность, при этом, как правило, эксперимент многократно повторяется.

Классификация по отрасли представленных в модели знаний разделяет все модели на *физические, биологические, социальные, экономические* и т. д.

Классификация по способу представления модели

Отразить в модели признаки оригинала можно разными способами. Можно скопировать признаки, построив *натурную (материальную) модель*. Примерами натуральных моделей являются макеты и муляжи – уменьшенные или увеличенные копии, воспроизводящие либо внешний вид объекта (например, глобус), либо его структуру (например, модель Солнечной системы), либо поведение (например, радиоуправляемая модель автомобиля).

Можно построить модель объекта, описав его свойства на одном из языков кодирования информации – дать словесное описание, привести формулу, чертеж, рисунок. Такая модель называется *информационной*. Замена реального объекта его формальным описанием, т. е. его информационной моделью, называется *формализацией*. Существуют разные формы представления информационных моделей: словесные (вербальные), графические, математические, табличные и др. (рис. 1.4).

Вербальная модель – информационная модель в мысленной или разговорной форме.

Знаковая модель – информационная модель, выраженная знаками, т. е. средствами любого формального языка.

Математическая модель – модель, представленная с помощью математических формул.

Логическая модель – это модель, в которой представлены различные варианты выбора действий на основе умозаключений и анализа условий.

Специальные модели – это, например, химические формулы, ноты и т. д.

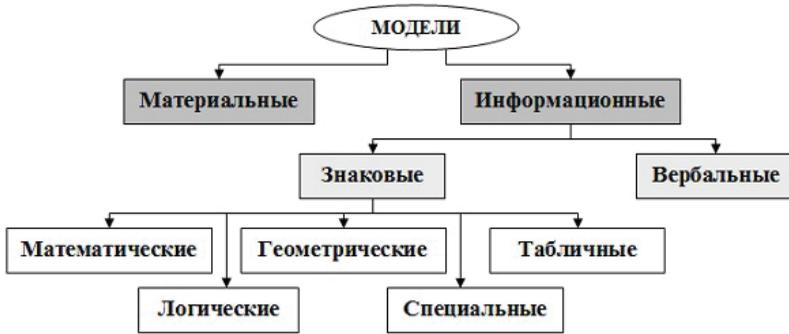


Рис. 1.4. Классификация моделей по способу представления

Геометрическая модель – модель, представленная с помощью графических форм (граф, блок-схема алгоритма решения задачи, диаграмма).

Граф – это множество *вершин* и множество *ребер*, соединяющих между собой все или часть этих вершин. На рис. 1.5,*а* показана геометрическая модель в виде графа, представляющая схему дорог, соединяющих населенные пункты. Вершины графа – это населенные пункты, ребра – дороги. Построенная модель позволяет, например, ответить на вопрос: через какие населенные пункты нужно проехать, чтобы добраться из пункта **А** в пункт **В**. Однако эта модель не позволяет ответить на вопрос, сколько составит расстояние от одного населенного пункта до другого. На этот вопрос можно ответить с помощью модели в виде **взвешенного графа**, каждое ребро которого отмечено числом, равным по значению расстоянию между соответствующими населенными пунктами (рис. 1.5,*б*).

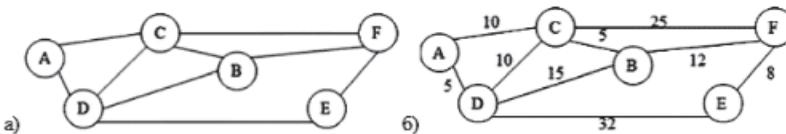


Рис. 1.5. Модель в виде графа

Табличная модель – это информация о моделируемом объекте, структурированная в виде таблицы. Различают следующие типы табличных моделей:

- таблица типа **«объект-свойство»** в одной строке содержит информацию об одном объекте в виде заданного набора его свойств:

ФИО студента	Номер зачетки	Направление подготовки	Шифр группы	Военнообязанный (да/нет)
Андреев А.В.	100050	080100.62	Экб-101	да
Борисов Е.В.	100121	080200.62	Менб-102	нет
...	...			

- таблица типа **«объект-объект»** отражает взаимосвязи между разными объектами по какому-либо свойству (связь между объектами **Студент** и **Экзамен** через свойство **Оценка за экзамен**):

Экзамен \ Студент	Информатика	История	Математика	Иностранный язык	Физика
Иванов А.А.	5	4	–	5	–
Борисов Е.И.	4	5	3	–	–
Гаврилов П.П.	–	–	4	3	4
...

- таблица типа **«двоичная матрица»** является частным случаем таблицы **«объект-объект»** и отражает наличие или отсутствие связи между объектами (1 – связь присутствует; 0 – отсутствует). Далее показана табличная модель типа **«двоичная матрица»**, соответствующая графу на рис. 1.5,б).

Населенный пункт	A	B	C	D	E	F
A	1	0	1	1	0	0
B	0	1	1	1	0	1
C	1	1	1	1	0	1
D	1	1	1	1	1	0
E	0	0	0	1	1	1
F	0	1	1	0	1	1

Классификация по характеру отображаемых свойств объекта моделирования

По характеру отображаемых свойств выделяют два типа моделей:

1) **структурные** – отражают структуру (устройство) моделируемого объекта, существенные для целей исследования свойства и взаимосвязи компонентов этого объекта;

2) **функциональные** – отражают внешне воспринимаемое поведение (функционирование) объекта.

Функциональные модели часто строятся как **модели черного ящика**. В такой модели задаются только входные и выходные связи моделируемого объекта со средой (рис. 1.6). Название «черный ящик» образно подчеркивает отсутствие сведений о внутреннем содержании объекта.

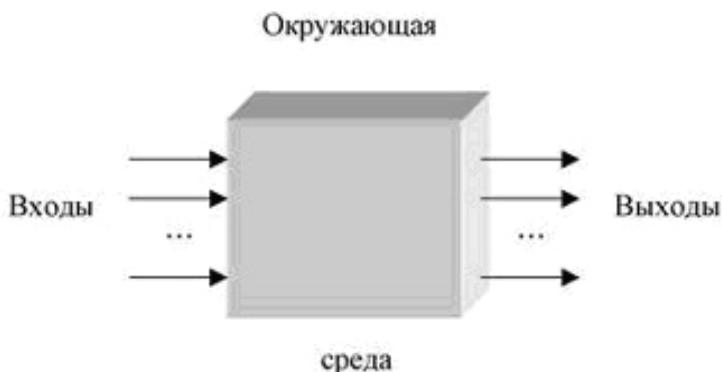


Рис. 1.6. Модель черного ящика

Наряду с моделью черного ящика по степени информированности исследователя о моделируемом объекте выделяют еще два вида моделей:

1) **«белый ящик»** – известно все о внутреннем содержании объекта;

2) **«серый ящик»** – известна структура объекта, неизвестны количественные значения параметров.

Классификация с учетом фактора времени

С учетом фактора времени модели можно разделить на два класса:

1) **статические модели** – это одномоментный срез информации об объекте;

2) **динамические модели** – позволяют увидеть изменение объекта во времени.

Например, медицинская карта состояния здоровья пациента в поликлинике отражает изменение состояния здоровья человека за некоторый период времени (динамическая модель), а медицинское обследование при поступлении на работу дает картину состояния здоровья на данный момент времени (статическая модель).

Классификация по характеру изменения модели во времени охватывает динамические модели и делит их на два типа:

1) **непрерывные** – изменяют свое состояние во времени за сколь угодно малое приращение времени;

2) **дискретные** – изменяют свое состояние во времени дискретно, через определенный временной интервал.

Классификация по признаку причинной обусловленности выполняется в зависимости от возможности или невозможности учета в рассматриваемой модели одного или нескольких случайных факторов, при этом выделяют два вида моделей:

1) **детерминированные** – модели, в которых все воздействия и факторы определены и известны заранее;

2) **стохастические (вероятностные)** – модели, в которых хотя бы один из факторов носит случайный характер.

По способу реализации информационные модели делятся на компьютерные и некомпьютерные. **Компьютерная** – модель, реализованная с помощью программных средств на компьютере. Программное обеспечение, средствами которого может осуществляться компьютерное моделирование, может быть как универсальным (например, текстовые или табличные процессоры), так и специализированным, предназначенным лишь для определенного вида моделирования.

? Вопросы и упражнения для самоконтроля

1. Объясните смысл понятий «модель» и «моделирование».

2. Каким требованиям должны удовлетворять модели?

3. На какие классы разделяются модели по области использования?

4. Опишите классификацию моделей по способу их представления.
5. Поясните термины «материальная (натурная) модель», «информационная модель». Приведите примеры моделей такого рода.
6. Объясните понятие «вербальная модель». Приведите примеры.
7. Назовите и охарактеризуйте типы табличных моделей.
8. На какие группы можно разделить динамические модели в зависимости от характера изменения модели во времени?
9. Опишите назначение структурных и функциональных моделей.
10. Каковы особенности моделей черного ящика?
11. Опишите классификацию моделей с учетом фактора времени.
12. В чем заключается различие между детерминированными и стохастическими моделями?

1.2. Компьютерное математическое моделирование

Математическое моделирование не связано напрямую с компьютером. Аналитические решения (математические формулы, выражающие зависимость результата от исходных данных) удобнее и информативнее численных решений. *Аналитическая модель* — математическая модель, представляющая собой совокупность аналитических выражений и зависимостей, позволяющих оценивать те или иные свойства моделируемого объекта.

Аналитические модели позволяют быстро и точно объяснить процессы, происходящие в системах и предсказать их возможное поведение в различных условиях. Однако возможности аналитических методов решения сложных математических задач весьма ограничены, поэтому исследователи часто прибегают к построению моделей, основанных на численных методах решения математических задач. Получаемые решения являются приближенными, допускающими некоторую заранее заданную погрешность.

Этапы математического моделирования

Построение математической модели начинается с описания исходных данных и результатов. Затем на основании изучения реальной системы устанавливают виды взаимосвязи между исходными данными и результатами. Формальная запись этих зависимостей дает математическую модель. Рассмотрим *этапы компьютерного математического моделирования*, включающего численный эксперимент с моделью.

Первый этап математического моделирования — *определение целей моделирования*. Основные цели моделирования:

- 1) понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействия с окружающим миром;
- 2) научиться управлять объектом;
- 3) прогнозировать последствия тех или иных способов и форм воздействия на объект.

Второй этап моделирования — *ранжирование параметров* — разделение входных параметров по степени важности их влияния на результаты моделирования.

Третий этап — *выбор математического описания*. На этом этапе необходимо перейти от абстрактной формулировки модели к математическому описанию в виде уравнения, системы уравнений, системы неравенств и т. д.

Следующий необходимый этап — *выбор метода исследования*. Если выбранный метод использует компьютер, то необходимо подобрать программное средство из числа имеющихся или разработать соответствующую программу на одном из доступных языков программирования.

Проведение исследования — выполнение эксперимента с моделью (изменение входных данных с последующей фиксацией значений на выходе модели, изменение параметров в описании модели и т. д.).

На этапе *анализа результатов* выясняется, соответствует ли модель реальному объекту или процессу. Модель адекватна реальному процессу, если изучаемые характеристики процесса, полученные в ходе моделирования, совпадают с заданной степенью точности с экспериментальными. В случае несоответствия модели реальному процессу возвращаются к одному из предыдущих этапов.

Типы математических моделей

С точки зрения *целей моделирования* можно выделить следующие типы математических моделей: описательные, оптимизационные, игровые, имитационные.

Описательные математические модели используются для описания объекта моделирования с помощью математических формул. Такое описание позволяет применить для исследования модели математические методы. Например, в решении экономических задач широко используются матричные математические модели, для исследования которых применяются методы линейной алгебры.

Пример 1.1. Предприятие производит n типов продукции, используя m видов ресурсов. Нормы затрат i -го ресурса ($i = 1, 2, \dots, m$) на производство единицы продукции j -го типа ($j = 1, 2, \dots, n$) заданы матрицей затрат

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

где a_{ij} — количество i -го ресурса, затраченное на производство единицы продукции j -го типа. Пусть за определённый отрезок времени предприятие выпустило количество продукции каждого типа x_j ($j = 1, 2, \dots, n$). Представим выпуск продукции с помощью вектора-столбца

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

Построенная описательная математическая модель производства продукции позволит решать вычислительные задачи. Например, можно определить полные затраты ресурсов каждого вида s_i ($i = 1, 2, \dots, m$) на производство всей продукции за данный период времени как произведение матрицы A на вектор-столбец X , т. е.

$$S = \begin{pmatrix} s_1 \\ s_2 \\ \dots \\ s_m \end{pmatrix} = A \times X = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

Оптимизационные модели. Возможны случаи, когда, моделируя те или иные процессы, можно воздействовать на них, пытаясь добиться какой-либо цели. В этом случае в модель входит один или несколько параметров, значения которых можно варьировать.

Пример 1.2. Расширим матричную модель производства продукции, введя в рассмотрение вектор-столбец значений цены реализации единицы продукции каждого вида

$$C = \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix}.$$

Кроме того, в модели должны быть использованы ограничения на ресурсы, которые не бесконечны, и ограничения на переменные, которые должны быть положительными. Тогда можно поставить задачу определения такого плана выпуска продукции (вектора X), при котором значение суммарного дохода от реализации всей произведенной продукции будет наибольшим. Изменяемыми параметрами при этом будут элементы вектора X , а целевой функцией – скалярное произведение $F = C \cdot X = \sum_{i=1}^n c_i x_i = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n$, определяющее доход от реализации всей произведенной продукции.

Игровые модели предназначены для обоснования решений в условиях неопределенности (неполноты информации) и связанного с этим риска. Рассматриваются ситуации, в которых сталкиваются противоборствующие стороны, каждая из которых преследует свою цель. Достижение цели каждой из сторон (*выигрыш*) зависит от того, какие действия предпримет

противник. Такие ситуации называются **конфликтными**. Игровые модели находят применение при обосновании управленческих решений в условиях политических, социальных, производственных, трудовых и других конфликтов.

Теория игр – это раздел математики, изучающий методы разрешения конфликтных ситуаций, характеризующихся неопределенностью возможных действий конфликтующих сторон. Под **игрой** понимается взаимодействие нескольких **игроков**, каждый из которых стремится добиться выигрыша. **Стратегия** – это реализуемый игроком метод выбора ходов в течение игры. Если рассматривать игру двух участников, то совокупность выигрышей можно представить в виде **матрицы выигрышей**. Матрица строится с позиции одного из игроков. Каждый элемент матрицы соответствует величине выигрыша этого игрока в зависимости от выбранной стратегии. Обычно строки матрицы соответствуют стратегиям первого игрока, а столбцы – стратегиям второго. Первый игрок выбирает строку, второй игрок – столбец, при этом на пересечении находится выигрыш (или проигрыш, если значение отрицательное) первого игрока. Рассмотрим построение матрицы выигрышей на простейшем **примере** анализа военных действий.

Описание ситуации. В город можно войти только по двум мостам. Город обороняют 3 роты, контингент нападающих – 2 роты. Считается, что город будет взят нападающими, если на каком-либо мосту они окажутся в численном превосходстве. Требуется построить матрицу выигрышей для обороняющихся, приняв значение выигрыша равным 1 (успешная оборона) или -1 (потеря города).

Стратегии обороняющихся (строки матрицы): выделить на оборону первого моста 0, 1, 2, 3 роты (соответственно, оставив защищать второй мост 3, 2, 1, 0 рот). Стратегии наступающих (столбцы): атаковать первый мост силами 0, 1, 2 рот (остальные направить на второй мост). Сформируем матрицу выигрышей (рис. 1.7).

Слева от матрицы указаны стратегии защитников города, над матрицей – стратегии нападающих. Получение элементов матрицы выигрышей рассмотрим на примере,

соответствующем выборе защитниками третьей стратегии, если нападающие выберут первую стратегию (элемент матрицы, находящийся на пересечении третьей строки и первого столбца). В этом случае первый мост будут оборонять 2 роты, притом что нападающие этот мост не атакуют. Однако на втором мосту окажется 1 рота защитников против 3 рот нападающих, которые одержат победу. В этом случае выигрыш равен -1. Аналогично получены значения других элементов матрицы.

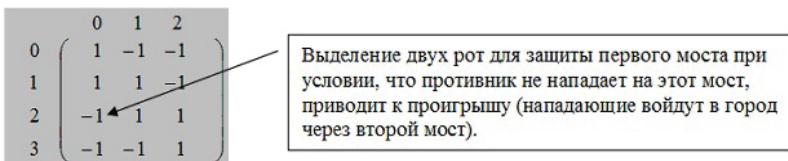


Рис. 1.7. Матрица выигрышей

Анализ матрицы выигрышей показывает, что для защитников города предпочтительными являются стратегии, соответствующие защите первого моста силами 1 или 2 рот (вторая или третья строки матрицы). При выборе этих стратегий в двух случаях из трех нападающие не войдут в город.

Имитационные модели. Имитационное моделирование — это метод исследования, при котором изучаемый объект заменяется компьютерной математической моделью, с достаточной точностью описывающей реальный объект. С полученной моделью проводятся эксперименты с целью получения информации об объекте. Часто имитационные модели строятся как статистические модели на основе метода Монте-Карло.

Например, при оценке риска инвестиционных проектов используют прогнозные данные об объемах продаж, затратах, ценах и т. д. Чтобы адекватно оценить риск, необходимо иметь достаточное количество информации для формулировки правдоподобных гипотез о вероятностных распределениях ключевых параметров процесса. Отсутствующие фактические данные заменяются величинами, полученными в процессе имитационного эксперимента, т. е. сгенерированными компьютером.

Математическое моделирование стохастических процессов

При моделировании стохастических (случайных, вероятностных) процессов, которые зависят от некоторых случайных факторов, приходится строить модели случайных величин, а также зависимостей между ними. **Случайная величина** — это величина, которая в зависимости от исхода испытания принимает одно из множества возможных значений. Предсказать заранее, какое значение будет принято случайной величиной в результате опыта или наблюдения, невозможно. Примеры случайных величин: количество посетителей магазина в течение дня; количество оценок «отлично» на экзамене и др.

В качестве примера статистического моделирования рассмотрим построение **регрессионной модели** статистической зависимости одной случайной величины (Y) от другой (X). Установлено, например, что рост доходов населения ведет к увеличению потребления, рост цены на какой-либо товар — к снижению спроса на этот товар. При этом зависимость не является однозначной, т. е. одному значению независимой величины (X) может соответствовать множество значений зависимой величины (Y).

Для решения задачи построения **уравнения регрессии** по известным выборочным значениям наблюдаемых случайных величин X и Y используется **метод наименьших квадратов**. Рассмотрим применение метода наименьших квадратов для линейной регрессии, когда зависимость Y от X является линейной. В этом случае метод наименьших квадратов позволяет найти уравнение прямой $y = b_0 + b_1x$, которая в некотором смысле «ближе» расположена к точкам, построенным по наблюдаемым парам значений случайных величин (x_i, y_i) .

Критерием того, насколько близко исходные точки лежат к искомой прямой, является сумма квадратов отклонений наблюдаемых значений y_i от теоретических y_i^T , соответствующих x_i и лежащих на прямой $y = b_0 + b_1x$, т. е.

$$S = \sum_{i=1}^n (y_i^T - y_i)^2 = \sum_{i=1}^n (b_0 + b_1x_i - y_i)^2 \rightarrow \min,$$

где n — количество пар исходных данных.

Рассматривая S как функцию двух переменных $S(b_0, b_1)$, определим такие значения b_0 и b_1 , при которых функция S принимает минимальное значение. Для этого частные производные функции S по b_0 и b_1 приравняем к нулю (условие экстремума функции). Получим систему двух уравнений, линейных относительно b_0 и b_1 . Решив систему, найдем формулы для вычисления b_0 и b_1 через наблюдаемые значения исследуемых случайных величин X и Y :

$$b_0 = \frac{\sum_{i=1}^n y_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i y_i \sum_{i=1}^n x_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}; \quad b_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n y_i \sum_{i=1}^n x_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}. \quad (1.1)$$

Пример 1.3. Определим характер зависимости выпуска продукции (Y) от количества потребляемой электроэнергии (X). Результаты наблюдения за 12 месяцев ($n = 12$) представлены в табл. 1.1.

Таблица 1.1

Данные наблюдений

Потребление электроэнергии x_i	50	70	100	120	150	170	180	200	220	230	240	250
Выпуск продукции y_i	120	220	260	380	420	450	580	620	640	700	750	760

Отобразим результаты наблюдений из табл. 1.1 на графике (рис. 1.8). Предположим, что экспериментальные данные подчиняются гипотезе о линейной зависимости Y от X , т. е. примем в качестве модели зависимость $y = b_0 + b_1 x$. Выполнив расчеты с помощью формул (1.1), получим в результате уравнение $y = -30,56 + 3,17x$.

Построенная регрессионная кривая может использоваться для вычисления прогнозных значений случайной величины Y . Например, можно построить прогноз относительно выпуска продукции для значения потребления электроэнергии, равного 270. Подставим в уравнение $y = -30,56 + 3,17x$ значение $x = 270$ и получим $y = 824$.

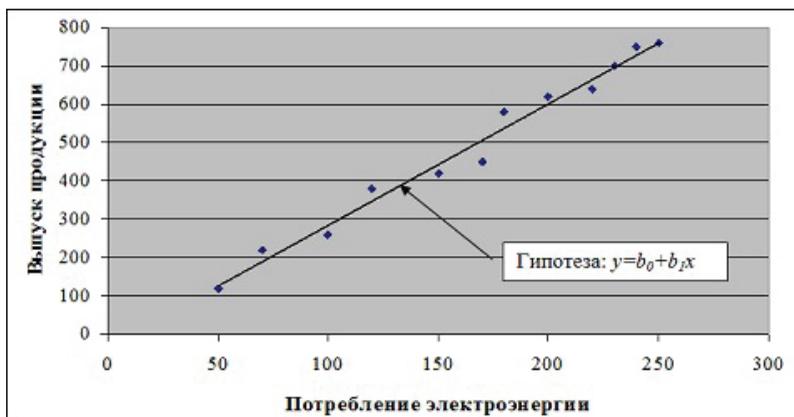


Рис. 1.8. Графический вид представления результатов наблюдения

☑ **Примечание.** В данном случае мы оставили без рассмотрения вопрос о правомерности гипотезы о линейной зависимости Y от X . Этот вопрос требует отдельного исследования в каждом конкретном случае с использованием методов математической статистики.

Статистические модели на основе метода Монте-Карло. Метод Монте-Карло (метод статистических испытаний) — один из методов имитационного моделирования, применяемый при решении задач, связанных с изучением случайных процессов. Его смысл состоит в том, что исследуемый процесс моделируется путем многократных повторений его случайных реализаций. Механизм случайного выбора реализуется методом генерации случайных чисел на компьютере, для чего применяются специальные программы, которые называются *генераторами случайных чисел*.

☑ Определяющим условием применения метода Монте-Карло является то, что статистические характеристики генерируемой случайной величины должны совпадать с соответствующими характеристиками исследуемой случайной величины.

Пример 1.4. Объект моделирования — очередь покупателей в магазине. Моделируемая характеристика — время ожидания

покупателем своей очереди. Цель моделирования – сокращение времени пребывания покупателя в очереди до некоторого заданного значения. Моделируемые параметры: n – количество покупателей, приходящих в магазин за единицу времени, t – время обслуживания продавцом одного покупателя.

Приход покупателей и время обслуживания одного покупателя носят случайный характер. Взаимодействие этих случайных процессов создает очередь. Если генерировать с помощью компьютера все возможные значения n и t , можно искусственно воссоздать картину процесса обслуживания покупателей в магазине. Повторяя компьютерный эксперимент с разными параметрами, можно изучать получаемые статистические данные так, как если бы они были получены при наблюдении над реальным потоком покупателей.

Примечание. Важным моментом является необходимость соблюдения условия совпадения законов распределения случайной величины, генерируемой компьютером, и исследуемой случайной величины, представленной выборкой ее наблюдаемых значений.

? Вопросы и упражнения для самоконтроля

1. Какие модели называются математическими моделями?
2. Перечислите и опишите этапы математического моделирования.
3. Опишите основные типы математических моделей.
4. Каково назначение оптимизационных математических моделей?
5. Каково назначение описательных моделей? Приведите примеры такого типа моделей.
6. Каково назначение игровых моделей? Назовите сферы применения моделей этого типа.
7. Объясните смысл терминов теории игр, используемых при описании игровых моделей (игра, стратегия, игрок, выигрыш).
8. По каким правилам строится матрица выигрышей?
9. Каково назначение имитационного моделирования?

10. В каких случаях идет речь о статистическом моделировании?

11. Опишите процедуру определения коэффициентов линейной регрессионной модели по методу наименьших квадратов.

12. Для построения какого типа моделей используется метод Монте-Карло?

1.3. Использование в моделировании средств компьютерной графики

С помощью компьютерной графики можно моделировать реальные явления и процессы, которые невозможно увидеть в действительности, например, эффект теории относительности. Можно выполнить графическую имитацию реальности в компьютерных играх, тренажерах, системах архитектурного проектирования. Прежде чем говорить об использовании компьютерной графики в моделировании, рассмотрим принципы обработки графической информации компьютером.

1.3.1. Разновидности компьютерной графики

Обычно двумерную графику по типу представления графической информации и алгоритмам обработки изображений разделяют на векторную, растровую и фрактальную.

Растровая графика использует представление изображения с помощью набора пикселей, имеющих различные характеристики (рис. 1.9,а). *Пиксель* – это наименьший неделимый элемент изображения. Любое изображение, созданное с помощью растровой графики, характеризуется количеством пикселей и используемых цветов. Файл изображения растровой графики хранит информацию о каждом пикселе в виде последовательности двоичных кодов. Программа (графический редактор или программа просмотра изображений) прочитывает эти коды, формируя «попиксельно» изображение на экране монитора. Для хранения растровых изображений требуется много памяти, так как компьютер должен хранить информацию о каждом пикселе.

Изображение, полученное с помощью растровой графики, невозможно увеличить (масштабировать) без потери качества. Это объясняется тем, что при увеличении должно быть увеличено число пикселей, формирующих это изображение. Компьютер сам не в состоянии определить, каким цветом нужно заполнить новые пиксели, сделав переходы между цветами плавными. Поэтому при увеличении получаются видимые глазом квадраты, каждый из которых закрашен в один цвет.

Основным преимуществом растровых изображений является их реалистичность, а недостатками – большой объем файлов изображений и проблемы с масштабированием.

К программам работы с растровой графикой относятся Adobe Photoshop, Corel PhotoPaint, Microsoft Paint. Форматы файлов растровых графических изображений: **BMP** – поддерживается всеми графическими редакторами, работающими под управлением Windows; **PSD** – формат файлов изображений, созданных в программе Adobe Photoshop; **PCX** – первый графический формат для IBM PC; **GIF, TIFF, JPEG, PNG** – аппаратно-независимые форматы для сжатия и передачи растровых изображений по сетям.

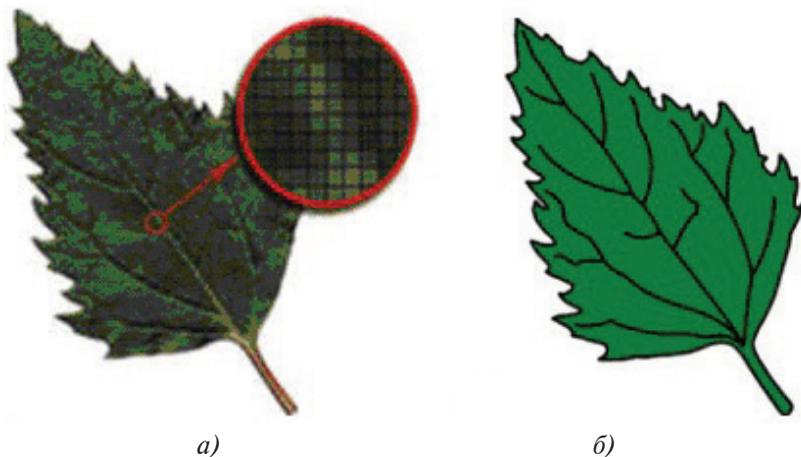


Рис. 1.9. Примеры *a* – растрового и *б* – векторного изображений

Векторная графика. При использовании векторной графики изображение формируется из совокупности отдельных геометрических фигур – **геометрических примитивов**, каждый из которых задается с помощью математического описания. Такими примитивами могут быть точка, линия, прямоугольник, окружность. У каждого примитива есть свои параметры, например, толщина линии или цвет заполнения фигуры. Все графические примитивы описываются с помощью математических выражений, которые при работе с векторным изображением или при его выводе на экран рассчитываются компьютером. Графический файл содержит набор координат и параметры каждого из графических примитивов, образующих рисунок.

Файл векторного изображения оказывается значительно меньшего размера, чем файл аналогичного растрового изображения. При этом размер файла векторного изображения не зависит от размера самого изображения. Поскольку векторное изображение строится на математическом описании геометрических фигур, то его можно масштабировать без какой-либо потери качества. При увеличении компьютер рассчитывает заново параметры каждой фигуры в зависимости от пропорции увеличения и выведет эту же фигуру с другими параметрами.

Существенный недостаток векторной графики – условность (низкая реалистичность) изображений (рис. 1.9,б). Главные достоинства – отсутствие проблемы с масштабированием изображений и небольшой объем файлов. Программы для работы с векторными изображениями: CorelDraw, AutoCad, Arhcad. Основные форматы файлов векторных графических изображений: **WMF (Windows Metafile)** – встроенный векторный формат Windows; **CDR** – формат векторных изображений, созданных с помощью программы CorelDraw; **DXF** – открытый формат для обмена графическими изображениями между программами САПР.

Следует отметить, что все современные профессиональные графические редакторы совмещают элементы как растрового, так и векторного вида графики. Например,

векторный редактор CorelDraw имеет как собственные, так и подключаемые инструменты для редактирования растровых изображений, а новые версии растрового редактора Adobe Photoshop расширены инструментальными возможностями для работы с векторными изображениями.

Фрактальная графика. Математической основой фрактальной графики является фрактальная геометрия. В основу построения изображений положен принцип наследования геометрических свойств «объектами-наследниками» от «объектов-родителей». **Фракталом** называется структура, состоящая из частей, которые в каком-то смысле подобны целому, когда небольшая часть фрактала содержит информацию обо всем фрактале.

Простой пример геометрического фрактала — «кривая Коха». Процесс её построения выглядит следующим образом: берется отрезок прямой линии, делится на три равные части и средний интервал заменяется равносторонним треугольником без основания. В результате образуется ломаная, состоящая из четырех отрезков, длина каждого из которых равна $1/3$ длины исходного отрезка. На следующем шаге операция повторяется для каждого из четырёх получившихся отрезков и т. д. На рис. 1.10,*а* показаны первые четыре итерации построения «кривой Коха». Применение описанной процедуры к каждой стороне равностороннего треугольника дает фрактал, получивший название «снежинка Коха» (рис. 1.10,*б*). Изменяя и комбинируя окраску фрактальных фигур, можно моделировать образы живой и неживой природы, например ветви дерева (рис. 1.10,*в*).

Фрактальная графика, как и векторная, является вычисляемой. Изображение строится по уравнению или системе уравнений, поэтому в памяти компьютера для выполнения всех вычислений требуется хранить только формулы. Изменив коэффициенты уравнения, можно получить совершенно другое изображение. Среди программ, используемых для генерации фрактальных изображений, можно назвать следующие: Art Dabbler, Ultra Fractal, Fractal Explorer, ChaosPro, Mystica.

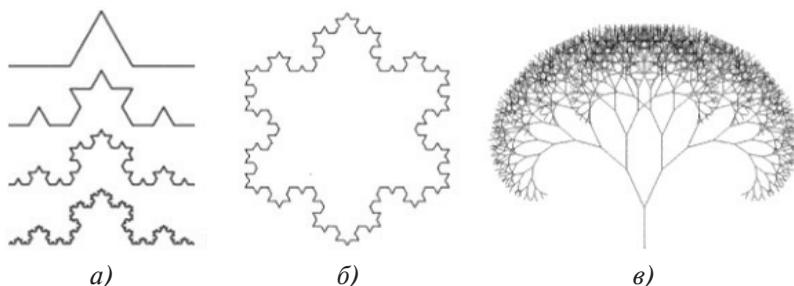


Рис. 1.10. Примеры фрактальных фигур: *а* – «кривая Коха»;
б – «снежинка Коха»; *в* – «дерево Серпинского»

1.3.2. Применение компьютерной графики в моделировании

В компьютерной графике выделяют несколько разделов. Рассмотрим и охарактеризуем эти разделы с точки зрения возможностей их использования для моделирования.

Иллюстративная графика, простейшими программными средствами которой являются графические редакторы (MS Paint, Adobe Photoshop, CorelDRAW и др.), служит для создания и редактирования изображений. В моделировании средства иллюстративной графики могут быть использованы для наглядного представления объектов моделирования. Например, средствами компьютерной графики может быть составлен фоторобот – графическая модель внешности преступника, создаваемая по показаниям свидетелей посредством подбора отдельных частей лица.

Инженерная графика, известная также как САПР (системы автоматизированного проектирования, английская аббревиатура CAD – Computer Aided Design), предназначена для автоматизации процесса проектирования технических объектов и создания проектных документов с учетом существующих норм и стандартов. САПР были исторически первыми интерактивными системами. В системе интерактивной компьютерной графики пользователь воспринимает на дисплее изображение, представляющее некоторый объект (модель объекта), и может вносить изменения в эту модель. Это может быть ввод и редактирование отдельных элементов,

задание числовых значений для любых параметров, а также другие операции по вводу информации на основе восприятия изображений. Системы типа САПР активно используются во многих областях, например, в машиностроении, электронике, архитектуре и строительстве.

Деловая графика применяется, если нужно представить числовые данные в виде диаграммы (графика, гистограммы, круговой, биржевой или какого-либо иного типа). Диаграмма позволяет визуализировать числовые данные, сделать их более наглядными. Это способствует их лучшему восприятию человеком для дальнейшего анализа этих данных. Моделирование числовых данных графическими средствами находит широкое применение в экономике, статистике, менеджменте.

Научная графика применяется для наглядного представления решения научной задачи (обычно по итогам математического моделирования). В задачах моделирования часто применяется построение линий (поверхностей), вдоль которых некоторая функция имеет одинаковое значение, называемых *изолиниями* (изоповерхностями). Это могут быть изотермы – линии равной температуры, изобары – линии равного давления, изолинии концентрации вредных примесей в окружающей среде и т. д. Например, газеты или телепрограммы, давая прогноз погоды, часто используют изотермы.

В математическом моделировании для графического решения уравнений, полученных при математическом описании моделей тех или иных объектов, используется **построение графиков функций**. Для этой цели могут быть использованы как специализированные программные средства, так и универсальные, например, инструмент построения диаграмм в программе MS Excel.

Трёхмерная графика (3D-графика) использует методы создания объёмных моделей объектов, которые максимально соответствуют реальным. Такие объёмные изображения можно вращать и рассматривать со всех сторон. Для создания объёмных изображений используют разные графические фигуры и гладкие поверхности. При помощи их сначала

создаётся каркас объекта, потом его поверхность покрывают материалами, визуально похожими на реальные.

Широкое применение 3D-графика находит в индустрии компьютерных игр: анимационные заставки, интерфейсы и персонажи компьютерных игр создаются в программах 3D-графики. Архитекторы и дизайнеры используют 3D-графику для построения макетов зданий и трёхмерных моделей архитектурных памятников, которых ещё не существует в реальности.

Наиболее известными редакторами 3D-графики являются пакеты программ 3D Studio Max и Autodesk Maya. Распространяются бесплатно программы Blender, Google SketchUp и др.

Виртуальная реальность — это высокотехнологичная форма компьютерного моделирования, которая позволяет пользователю погрузиться в модельный мир и непосредственно действовать в нем. Зрительные, слуховые, осязательные и моторные ощущения пользователя при этом заменяются их имитацией, генерируемой компьютером. Признаки устройств виртуальной реальности:

- моделирование в реальном масштабе времени;
- имитация окружающей обстановки с высокой степенью реализма;
- возможность воздействовать на имитируемую окружающую обстановку и иметь при этом обратную связь.

Геоинформационные системы (ГИС). Это относительно новая для массовых пользователей разновидность систем интерактивной компьютерной графики. Они интегрируют методы и технологии из разных сфер науки и техники — баз данных, геодезии, картографии, космонавтики, навигации, компьютерной графики. Известны такие системы, как ArcGIS, AutoCAD Map 3D, MapInfo. Для любой ГИС типичны следующие операции: ввод и редактирование объектов с учетом их расположения на поверхности Земли, формирование разнообразных цифровых моделей, запись в базы данных, выполнение разнообразных запросов к базам данных. Важной функцией ГИС является анализ пространственных отношений множества объектов, расположенных на какой-то территории.

? Вопросы и упражнения для самоконтроля

1. Перечислите разновидности компьютерной графики.
2. Что является наименьшим неделимым элементом растрового изображения?
3. Каковы основные достоинства и недостатки растровых изображений?
4. Назовите основные программы, используемые для работы с растровыми изображениями.
5. Какая информация сохраняется в файле растрового графического изображения? Перечислите форматы файлов растровых графических изображений.
6. Опишите принципы представления векторных графических изображений. Что понимается под графическим примитивом?
7. Каковы основные достоинства и недостатки векторных изображений?
8. Какая информация сохраняется в файле векторного графического изображения? Перечислите форматы файлов векторных графических изображений.
9. Опишите принципы формирования изображений средствами фрактальной графики. Что такое фрактал?
10. Приведите примеры использования в моделировании программ иллюстративной графики.
11. Каково назначение программных средств инженерной графики? Приведите примеры использования инженерной графики в моделировании.
12. Охарактеризуйте назначение и возможности программных средств деловой графики.
13. Каковы сферы применения научной графики?
14. Назовите особенности и области применения 3D-графики.
15. Охарактеризуйте геоинформационные системы с точки зрения возможностей моделирования в их среде, а также сферы их практического использования.

1.4. Моделирование средствами табличного процессора Microsoft Excel

Построение линий тренда

В практике научных исследований часто приходится иметь дело с таблицей численных значений экспериментальных данных, отражающих некоторую функциональную зависимость $y_i = f(x_i)$. Замена табличной функции $y_i = f(x_i)$ аналитической функцией $y = \varphi(x)$ называется *аппроксимацией*, а функция $\varphi(x)$ – *аппроксимирующей функцией*.

Получение аппроксимирующей функции – это математическое моделирование того или иного параметра исследуемого процесса или объекта по результатам наблюдений или измерений значений этого параметра.

В качестве простейших аппроксимирующих функций используют линейную $y = \varphi(x) = ax + b$, степенную $y = \varphi(x) = ax^b$, логарифмическую $y = \varphi(x) = a \ln(x) + b$, экспоненциальную $y = \varphi(x) = ae^{bx}$.

Все эти функции содержат два коэффициента аппроксимации a и b . Если эти функции не обеспечивают необходимой точности аппроксимации, количество коэффициентов можно увеличить и использовать более сложные функции. Например, можно использовать полиномиальную аппроксимацию функцией $y = \varphi(x) = a + bx + cx^2 + \dots$, которая позволяет путем увеличения степени полинома и, соответственно, количества коэффициентов обеспечить необходимую точность аппроксимации. Задача аппроксимации может быть решена в MS Excel с использованием инструмента добавления линии тренда.

Линия тренда – это графическое представление общей закономерности изменения ряда данных. Она может быть добавлена для любого ряда данных на диаграмме с областями, линейчатой диаграмме, гистограмме, графике или точечной диаграмме. Для построения линии тренда необходимо выполнить следующие действия:

- построить диаграмму для таблицы данных;
- выделить на диаграмме любой элемент данных и открыть контекстное меню щелчком правой кнопки;

- выбрать команду **Добавить линию тренда**, при этом появляется окно **Линия тренда**;
- на вкладке **Тип** выбрать нужный тип линии тренда, на вкладке **Параметры** при необходимости установить флажок **показывать уравнение на диаграмме** и флажок **поместить на диаграмму величину достоверности аппроксимации (R^2)**.

Значение коэффициента достоверности аппроксимации (R^2) может находиться в диапазоне от 0 до 1. Чем ближе R^2 к 1, тем точнее выбранная функция аппроксимирует экспериментальные данные.

Линии тренда позволяют графически отображать тенденции данных и прогнозировать их дальнейшие изменения. Можно продлить линию тренда в диаграмме за пределы реальных данных для предсказания будущих значений. Чем точнее она будет выбрана, тем точнее будет прогноз. Добавление линии тренда к экспериментальным данным позволяет, в частности, построить линейную регрессионную модель (см. пункт 1.2) по результатам измерений или наблюдений.

Использование для моделирования инструмента «Подбор параметра»

Решение многих практических задач требует нахождения численных значений параметров функции, обеспечивающих получение некоторого заранее заданного или экстремального значения этой функции. Функция, значение которой должно достигнуть искомого значения, называется **целевой функцией**. В MS Excel эта функция записывается в виде формулы, расположенной в определенной ячейке таблицы, которая называется **целевой ячейкой**.

Подбор параметра в Microsoft Excel — это инструмент, позволяющий целенаправленно перебрать множество значений *одиночного параметра* с одновременным контролем результирующего значения целевой функции. При этом очередное значение параметра подставляется в формулу, выполняется вычисление, и полученный результат сравнивается с искомым (целевым) значением. Если на очередном шаге вычислений разница между текущим результатом и целевым

значением стала меньше заданной величины погрешности, то вычисления прекращаются, и текущее значение предлагается как итоговый результат подбора.

При подборе параметра пользователь должен определить три обязательные составляющие:

- 1) местоположение **целевой ячейки** (*содержимое которой обязательно должно быть формулой*);
- 2) **значение, которое должно быть достигнуто** в целевой ячейке при изменении параметра;
- 3) ячейку, содержимое которой (параметр) будет меняться для достижения целевой ячейкой искомого значения (*изменяемая ячейка*). Эта ячейка *не может содержать формулу, в ней должно быть числовое значение*.

Все три названные составляющие поиска объединены в диалоговом окне **Подбор параметра** (рис. 1.12), которое вызывается из меню **Сервис**.

Первое поле **Установить в ячейке** должно содержать адрес целевой ячейки с формулой, для которой следует подобрать параметр, второе поле **Значение** – искомое значение целевой ячейки, третье поле **Изменяя значение ячейки** – адрес ячейки, значение которой будет изменяться в процессе подбора параметра. Эта ячейка должна влиять на значение целевой ячейки.

Пример 1.5. Необходимо распределить премию среди сотрудников. В качестве премии всем сотрудникам начисляется один и тот же процент от начисленной суммы заработной платы. Требуется определить процент премии, который позволит не выйти за пределы премиального фонда 30000 руб.

Первоначально установим премию равной 10% и выполним необходимые вычисления. Заполним расчетную таблицу (рис. 1.11). При вычислениях используются следующие формулы:

*Премия = Начислено * Процент премии;*

*Удержано = (Начислено + Премия) * Подоходный налог;*

Сумма к выдаче = Начислено + Премия – Удержано.

	A	B	C	D	E
2	Премимальный фонд		30000		
3	Процент премии		0,1		
4	Подходный налог		0,13		
5	ФИО	Начислено	Премия	Удержано	Сумма к выдаче
6	Алексеева А.И.	10000	=B6*\$C\$3	=(B6+C6)*\$C\$4	=B6+C6-D6
7	Богданов О.К.	12000	=B7*\$C\$3	=(B7+C7)*\$C\$4	=B7+C7-D7
8	Волков С.С.	16000	=B8*\$C\$3	=(B8+C8)*\$C\$4	=B8+C8-D8
9	Демин А.К.	15000	=B9*\$C\$3	=(B9+C9)*\$C\$4	=B9+C9-D9
10	Иванов А.И.	20000	=B10*\$C\$3	=(B10+C10)*\$C\$4	=B10+C10-D10
11	Яковлева А.Ф.	12000	=B11*\$C\$3	=(B11+C11)*\$C\$4	=B11+C11-D11
12	Итого	=СУММ(B6:B11)	=СУММ(C6:C11)	=СУММ(D6:D11)	=СУММ(E6:E11)

Рис. 1.11. Расчетная таблица в режиме показа формул для примера 1.5

	A	B	C	D	E
2	Премимальный фонд		30000		
3	Процент премии		0,1		
4	Подходный налог		0,13		
5	ФИО	Начислено	Премия		
6	Алексеева А.И.	10000	=B6*\$C\$3		
7	Богданов О.К.	12000	=B7*\$C\$3		
8	Волков С.С.	16000	=B8*\$C\$3		
9	Демин А.К.	15000	=B9*\$C\$3		
10	Иванов А.И.	20000	=B10*\$C\$3		
11	Яковлева А.Ф.	12000	=B11*\$C\$3		
12	Итого	=СУММ(B6:B11)	=СУММ(C6:C11)	=СУММ(D6:D11)	=СУММ(E6:E11)

Подбор параметра

Установить в ячейке: C12

Значение: 30000

Изменяя значение ячейки: \$C\$3

OK Отмена

Рис. 1.12. Процедура подбора процента премии для примера 1.5

Для решения поставленной задачи используем процедуру **Подбор параметра**. В качестве целевой следует указать ячейку C12, содержащую формулу расчета суммарной премии всех сотрудников. В этой ячейке потребуем установить целевое значение 30000. В качестве изменяемой укажем ячейку C3, содержащую процент премии (рис. 1.12). Результат решения задачи показан на рис. 1.13.

	A	B	C	D	E
2	Премиаальный фонд		30 000,00р.		
3	Процент премии		35,29%		
4	Подоходный налог		13,00%		
5	ФИО	Начислено	Премия	Удержано	Сумма к выдаче
6	Алексеева А.И.	10 000,00р.	3 529,41р.	1 758,82р.	11 770,59р.
7	Богданов О.К.	12 000,00р.	4 235,29р.	2 110,59р.	14 124,71р.
8	Волков С.С.	16 000,00р.	5 647,06р.	2 814,12р.	18 832,94р.
9	Демин А.К.	15 000,00р.	5 294,12р.	2 638,24р.	17 655,88р.
10	Иванов А.И.	20 000,00р.	7 058,82р.	3 517,65р.	23 541,18р.
11	Яковлева А.Ф.	12 000,00р.	4 235,29р.	2 110,59р.	14 124,71р.
12	Итого	85 000,00р.	30 000,00р.	14 950,00р.	100 050,00р.

Рис. 1.13. Результаты подбора параметра для примера 1.5

Использование в моделировании процедуры «Поиск решения»

Поиск решения в Microsoft Excel – это набор управляемых пользователем алгоритмов решения оптимизационных задач. При поиске решения целенаправленному изменению подвергаются, как правило, не один, а несколько параметров. Кроме того, на диапазоны изменения параметров могут быть наложены ограничения. Значения целевой функции получаются в результате вычисления по формуле, расположенной в целевой ячейке, которая обязательно должна быть связана с изменяемыми параметрами. Рассмотрим использование процедуры поиска решения на примере.

Пример 1.6. Для нужд частного кафе требуется закупить крупы трех видов (гречневая, рисовая, пшенная). Необходимо подобрать объемы закупки круп таким образом, чтобы при известных ценах за 1 кг общая сумма затрат составила 5000 руб. При этом рисовой крупы необходимо купить не менее 100 кг, а гречневой крупы – в два раза больше, чем пшенной.

Создадим расчетную таблицу (рис. 1.14):

- 1) введем значения исходных данных в ячейки **C2:C4**;
- 2) в ячейку **D2** введем формулу **=B2*C2**, которую скопируем на ячейки **D3** и **D4**;
- 3) в ячейку **D5** введем формулу **=СУММ(D2:D4)**.

D5		fx =СУММ(D2:D4)		
	A	B	C	D
1	<i>Крупа</i>	<i>Количество</i>	<i>Цена</i>	<i>Стоимость</i>
2	Гречневая	0,00	30,50р.	- р.
3	Рисовая	0,00	26,00р.	- р.
4	Пшеничная	0,00	16,70р.	- р.
5	Итого			- р.
6				

Рис. 1.14. Расчетная таблица для примера 1.6

Перейдем к поиску решения. В данном случае целью будет достижение заданного значения в ячейке **D5** (общая сумма затрат составляет 5000 руб.). При этом должны быть выполнены условия: **B3** \geq **100** (рисовой крупы необходимо купить не менее 100 кг), **B2** $= 2 * B4$ (гречневой крупы необходимо купить в два раза больше, чем пшеничной).

Выделим ячейку **D5**, выполним команду **Сервис – Поиск решения**. В открывшемся окне **Поиск решения** (рис. 1.15) в поле **Установить целевую ячейку** укажем **D5**. Установим параметр **Равной** в положение **значению** и введем число 5000. В поле **Изменяя ячейки** следует указать, какие ячейки влияют на результат вычислений в целевой ячейке, в нашем примере это ячейки **B2:B4**.

Для ввода ограничений используется диалоговое окно **Добавление ограничения**, которое открывается кнопкой **Добавить** в окне **Поиск решения**. Запуск процедуры поиска решения выполняется нажатием кнопки **Выполнить** в окне **Поиск решения**. В окне **Результаты поиска решения** (рис. 1.16) для сохранения найденного решения следует установить переключатель **Сохранить найденное решение** и нажать кнопку **ОК**.

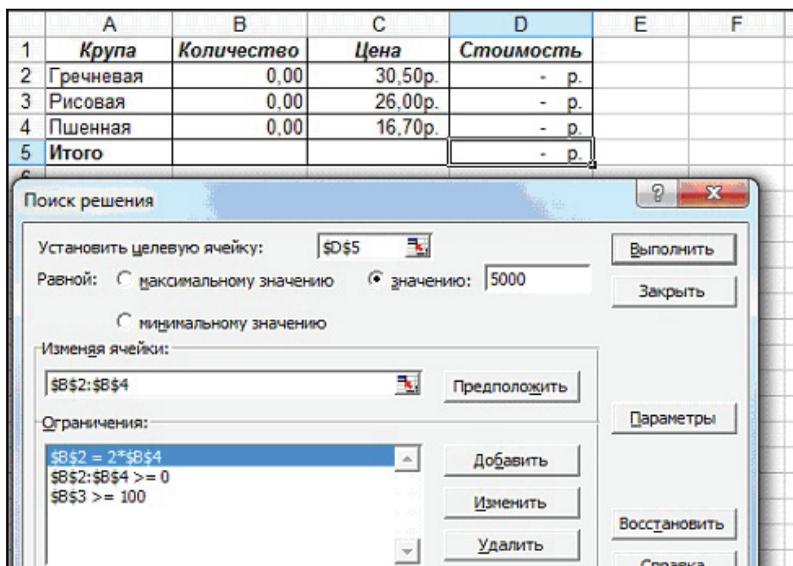


Рис. 1.15. Диалоговое окно Поиск решения для примера 1.6

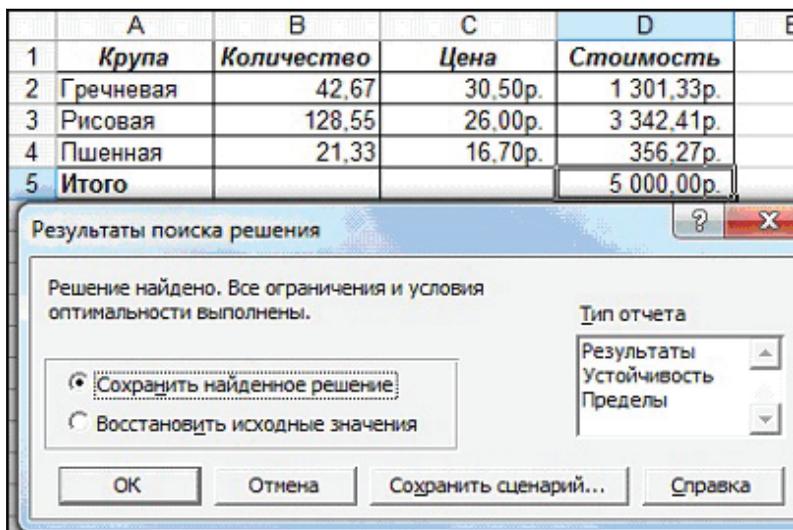


Рис. 1.16. Диалоговое окно Результаты поиска решения

Анализ полученного решения показывает, что на выделенные деньги в соответствии с заданными ограничениями можно будет купить около 42 кг гречневой, примерно 128,5 кг рисовой и около 21 кг пшеничной крупы.

? Вопросы и упражнения для самоконтроля

1. Каково назначение линий тренда?
2. Укажите типы диаграмм, которые можно использовать для построения линий тренда.
3. Как оценить достоверность аппроксимации данных при построении линии тренда?
4. Каково назначение процедуры **Подбор параметра** в программе MS Excel?
5. Каким должно быть содержимое целевой ячейки при выполнении процедуры **Подбор параметра**?
6. Можно ли в качестве изменяемой ячейки в процедуре **Подбор параметра** указать ячейку, которая не влияет на результат вычисления по формуле, содержащейся в целевой ячейке?
7. Можно ли при подборе параметра указать несколько изменяемых ячеек?
8. Может ли ячейка, влияющая на целевую, содержать формулу при подборе параметра?
9. Каково назначение процедуры **Поиск решения** в программе MS Excel?
10. Каким должно быть содержимое целевой ячейки, указываемой в диалоговом окне **Поиск решения**?
11. Можно ли при поиске решения указать несколько изменяемых ячеек?
12. Опишите процедуру ввода ограничений в окне **Поиск решения**.

1.5. Практические работы

Практическая работа 1.1. Применение в моделировании деловой графики и инструмента «Подбор параметра» MS Excel

Цель работы – научиться:

- использовать линии тренда для аппроксимации экспериментальных значений и прогнозирования;
- выполнять табулирование функций, заданных аналитически, и строить графики функций;
- использовать инструмент **Подбор параметра** для исследования математических моделей, представленных в виде нелинейных алгебраических уравнений.

Порядок выполнения работы

1. Выполнить задания 1 и 2.

Задание 1

По данным наблюдений, представленных в виде таблицы значений x и y , выполнить моделирование в среде табличного процессора MS Excel:

- найти коэффициенты b_0 и b_1 уравнения линейной регрессии $y = b_0 + b_1x$;
- оценить соответствие полученного уравнения экспериментальным значениям с помощью коэффициента достоверности аппроксимации (R^2);
- выполнить прогноз по значению y для заданного значения x , используя полученное уравнение линейной регрессии;
- построить аппроксимирующий полином второго порядка по данным таблицы;
- оценить соответствие полученного уравнения экспериментальным значениям с помощью коэффициента достоверности аппроксимации (R^2);
- выполнить прогноз по значению y для заданного значения x , используя полученный полином;
- сравнить результаты прогноза для линейной и полиномиальной аппроксимации.

Данные для расчетов выбираются в соответствии с номером варианта из табл. 1.2.

Пример выполнения задания

Выполним задание для данных, представленных в следующей таблице:

x_i	1	3	4	5	6	8	10	11	12	13	14
y_i	11	21	27	45	55	68	78	83	86	89	92

Для построения прогноза будем использовать $x = 16$.

Оформим таблицу с данными на рабочем листе электронной таблицы так, как показано на рис. 1.17. Построим по данным таблицы диаграмму точечного типа. Наведем курсор мыши на любую точку построенного графика и нажмем правую клавишу мыши. В появившемся контекстном меню следует выбрать пункт **Добавить линию тренда...** Откроется диалоговое окно **Линия тренда** (рис. 1.18), в котором на вкладке **Тип** выберем тип линии тренда **Линейная**, а на вкладке **Параметры** установим флажки **показывать уравнение на диаграмме** и **поместить на диаграмму величину достоверности аппроксимации (R^2)**.

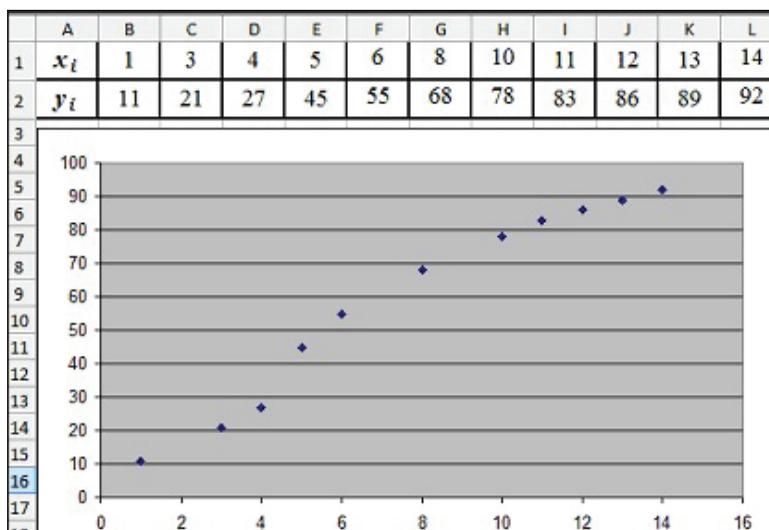


Рис. 1.17. Точечная диаграмма, соответствующая таблице экспериментальных данных

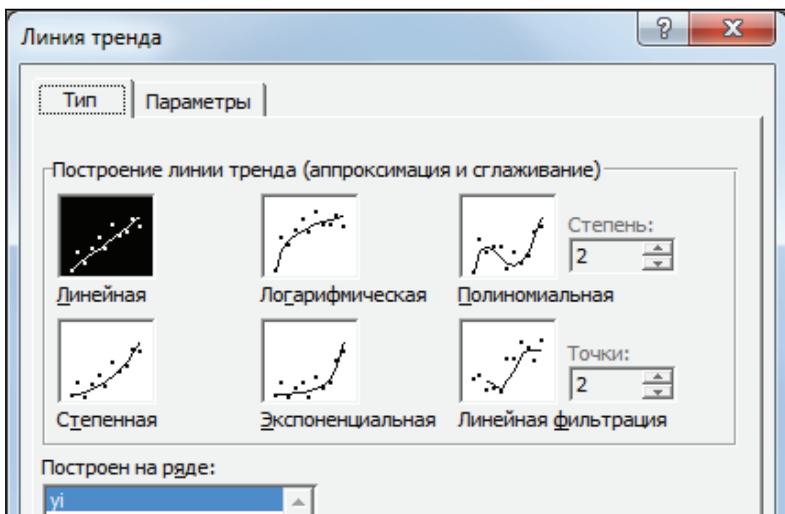


Рис. 1.18. Выбор типа линии тренда

Линия тренда, соответствующая уравнению линейной регрессии, построенной по методу наименьших квадратов, будет добавлена к диаграмме. Кроме того, будет выведено уравнение регрессии и коэффициент достоверности аппроксимации (рис. 1.19). Уравнение регрессии будет иметь вид $y = 6,5862x + 7,4543$, оценка достоверности аппроксимации $R^2 = 0,96$. Определим прогноз по y для $x = 16$, выполнив вычисление y в ячейке N2 по формуле $= 6,5862*N1 + 7,4543$, получим $y = 112, 83$.

На той же диаграмме построим еще одну линию тренда, выбрав тип **Полиномиальная** степени **2** и установив параметры для вывода уравнения и величины достоверности аппроксимации. На этот раз уравнение примет вид $y = -0,3077 x^2 + 11,335x - 5,4593$, оценка достоверности аппроксимации $R^2 = 0,984$. Определим прогноз по y для $x = 16$, выполнив вычисление y в ячейке N3 по формуле $= -0,3077*N1*N1 + 11,335*N1 - 5,4593$, получим $y = 97,13$.

Анализ результатов позволяет предположить, что прогноз с использованием полиномиальной аппроксимации является более точным, так как степень достоверности аппроксимации в этом случае выше.

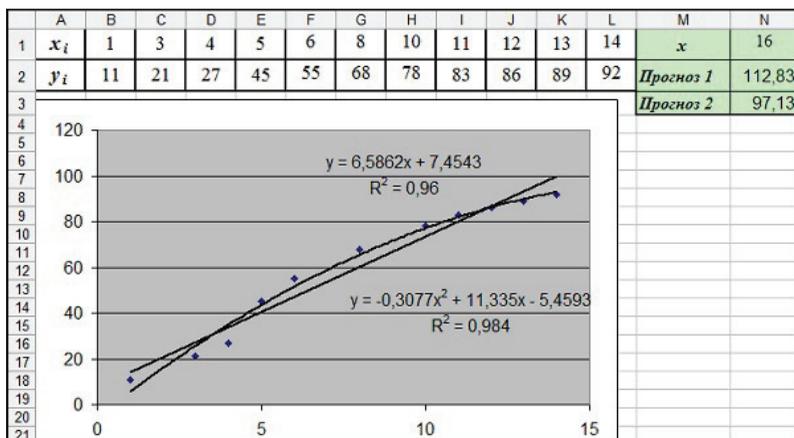


Рис. 1.19. Результат выполнения задания 1

Таблица 1.2

Варианты для задания 1

№ варианта	x_i	2	4	5	6	8	9	11	12	13	15	16	18	Значение x для прогноза
1	y_i	3	6	6	6	10	9	12	14	15	16	17	19	21
2	y_i	5	9	10	13	17	19	23	26	28	28	29	30	20
3	y_i	7	13	16	19	25	28	34	38	41	45	47	50	22
4	y_i	9	15	20	22	33	37	45	50	54	56	60	63	23
5	y_i	11	18	20	31	40	44	56	62	67	70	76	78	24
6	y_i	11	26	31	40	49	58	68	74	80	85	89	93	25
7	y_i	15	29	36	43	57	64	78	86	92	98	103	106	20
8	y_i	17	33	40	48	65	73	88	104	106	114	116	122	21
9	y_i	6	10	12	17	18	20	22	28	29	30	33	35	22
10	y_i	8	14	17	20	26	29	36	39	42	46	49	55	23

Задание 2

При математическом моделировании часто приходится решать уравнения вида

$$f(x, p_1, p_2, \dots, p_n) = 0, \quad (1.2)$$

здесь f – заданная функция; x – неизвестная переменная; p_1, p_2, \dots, p_n – параметры модели. Как правило, исследователя интересует поведение решения уравнения в зависимости от параметров $p_k, k = 1, \dots, n$. Только для линейных или простейших нелинейных уравнений удается найти решение в аналитической форме, т. е. записать формулу, выражающую искомую величину x в явном виде через параметры. В большинстве случаев приходится решать уравнение (1.2) численными методами.

Предположим, что в результате математического моделирования была получена модель вида $p_1 \cdot x \cdot \sin(p_2 \cdot x^2) + p_3 = 0$. Для заданных значений p_1, p_2, p_3 требуется определить корни уравнения на заданном интервале изменения переменной x с погрешностью, не превышающей $\varepsilon = 0,00001$. Значения p_1, p_2, p_3 , а также $x_{нач}$ и $x_{кон}$ (границы интервала изменения x) выбрать из табл. 1.3 в соответствии с номером варианта. При построении графика функции использовать 20 точек.

Пример выполнения задания

Исследуем уравнение при $p_1 = 3; p_2 = 1; p_3 = -1; x_{нач} = -2; x_{кон} = 2$. Требуется найти все корни уравнения $3 \cdot x \cdot \sin(x^2) - 1 = 0$ на отрезке изменения $x \in [-2, 2]$ с погрешностью, не превышающей $\varepsilon = 0,00001$.

Для того чтобы определить количество корней уравнения на заданном интервале и найти начальные приближения для выполнения процедуры **Подбор параметра**, построим график функции $y = 3 \cdot x \cdot \sin(x^2) - 1$ на интервале изменения x от -2 до $+2$. С этой целью сначала выполним табуляцию функции, вычислив значение y для каждого x , изменяющегося от -2 до $+2$.

В ячейки **C2** и **D2** введем левую (-2) и правую ($+2$) границы интервала изменения x (рис. 1.20). В ячейку **E2** – число точек для построения графика (в нашем случае 20), в ячейку **F2** – формулу для вычисления шага изменения переменной x : $x = (D2 - C2) / (E2 - 1)$.

В ячейки **A2:A21** введем формулы для вычисления значений x , как показано на рис. 1.22. Используем автозаполнение ячеек, начиная с ячейки **A3**. В ячейку **B2** введем формулу для вычисления значения функции $y = 3 \cdot x \cdot \sin(x^2) - 1$: $=3*A2*SIN(A2*A2)-1$, которую затем скопируем на ячейки **B3:B21**, используя автозаполнение.

Выделив прямоугольный диапазон **A1:B21**, активируем **Мастер диаграмм** (в меню **Вставка – Диаграмма**). При построении используем тип диаграммы **Точечная**. Результат построения показан на рис. 1.20.

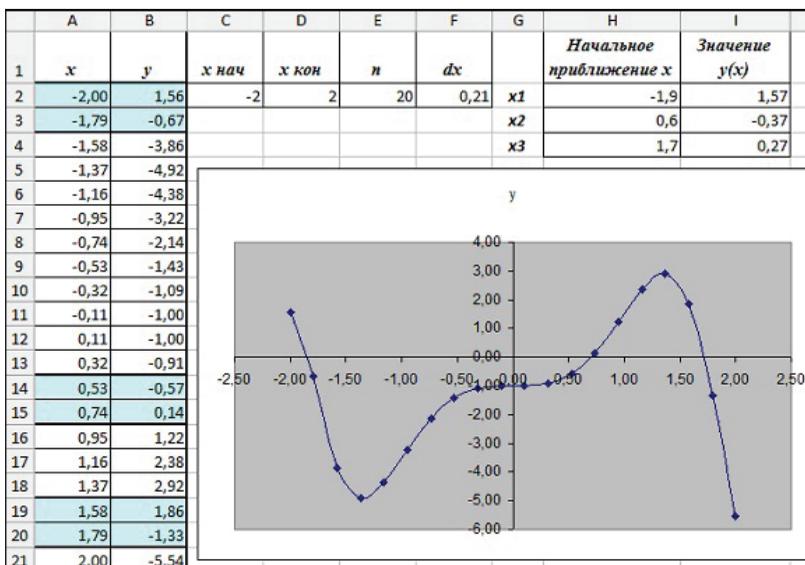


Рис. 1.20. Построение графика функции $y = 3 \cdot x \cdot \sin(x^2) - 1$

Анализ графика позволяет утверждать, что на отрезке изменения x от -2 до $+2$ рассматриваемое уравнение имеет **три корня**. Выполнив визуальный анализ графика, а также анализ таблицы значений функции с целью определения значений x , при которых функция изменяет знак на противоположный, определим начальные (грубые) приближения для корней уравнения: $x1 = -1,9$; $x2 = 0,6$; $x3 = 1,7$.

Запишем эти значения в ячейки **H2**, **H3**, **H4**, с тем чтобы в дальнейшем использовать их для уточнения значений корней. В ячейку **I2** введем формулу для вычисления значения функции $=3*H2*SIN(H2*H2)-1$. Скопируем формулу в ячейки **I3**, **I4** (рис. 1.21).

	A	B	H	I
			<i>Начальное приближение x</i>	<i>Значение y(x)</i>
1	x	y		
2	=C2	=3*A2*SIN(A2*A2)-1	-1,9	=3*H2*SIN(H2*H2)-1
3	=A2+\$F\$2	=3*A3*SIN(A3*A3)-1	0,6	=3*H3*SIN(H3*H3)-1
4	=A3+\$F\$2	=3*A4*SIN(A4*A4)-1	1,7	=3*H4*SIN(H4*H4)-1
5	=A4+\$F\$2	=3*A5*SIN(A5*A5)-1		
6	=A5+\$F\$2	=3*A6*SIN(A6*A6)-1		
7	=A6+\$F\$2	=3*A7*SIN(A7*A7)-1		

Рис. 1.21. Фрагменты таблицы в режиме просмотра формул

Для обеспечения заданной точности получаемых решений ($\varepsilon=0,00001$) в меню **Сервис – Параметры – Вычисления** установим флажок **Итерации** в положение **включено**, в поле **Предельное число итераций** укажем значение **100**, а в поле **Относительная погрешность** – значение относительной погрешности **0,00001** (рис. 1.22).

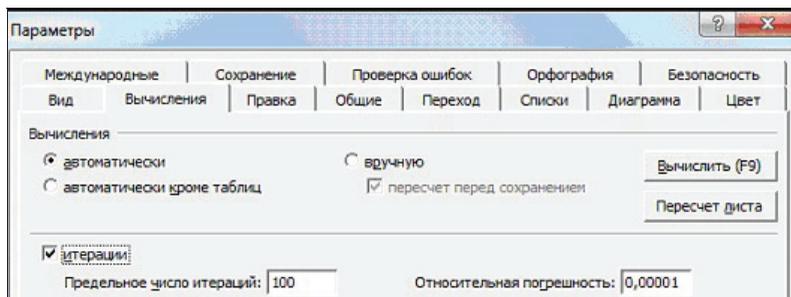


Рис. 1.22. Задание предельного числа итераций и относительной погрешности вычислений

Проведем уточнение корня x_1 . Установим курсор в ячейку **I2**, выполним в меню **Сервис – Подбор параметра**. В диалоге определим условия подбора параметра: установить в ячейке **I2** значение, равное 0, изменяя число в ячейке **H2** (рис. 1.23). Значение в ячейке **H2** будет изменяться согласно алгоритму, реализуемому процедурой **Подбор параметра**. При каждом значении будет вычисляться функция $y = 3 \cdot x \cdot \sin(x^2) - 1$. Процедура будет остановлена, когда значение в ячейке **I2** по абсолютной величине станет меньшим либо равным 0,00001.

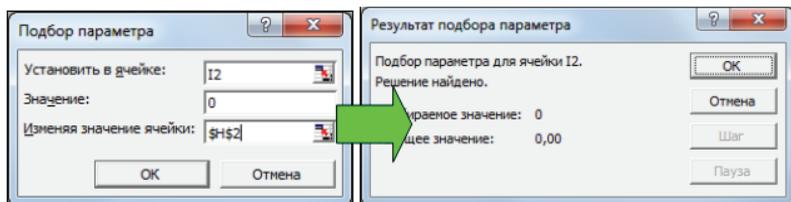


Рис. 1.23. Этапы выполнения подбора параметра

В диалоговом окне **Результат подбора параметра** убедимся, что решение найдено и нулевое значение функции достигнуто. Примем полученное решение, нажав кнопку **ОК**. В ячейке **H2** вместо начального приближения будет записано найденное значение первого корня уравнения x_1 . Аналогично выполним уточнение корней x_2 и x_3 . Ограничим для наглядности число разрядов в дробной части чисел для значений x двумя знаками после запятой, для значений y – шестью знаками после запятой. Число знаков после запятой можно регулировать с помощью инструментов **Увеличить разрядность**  и **Уменьшить разрядность**  на панели инструментов **Форматирование**. Результат представлен на рис. 1.24.

Анализ результатов показывает, что значение функции y для каждого из найденных значений x отличается от нуля на величину, меньшую $\varepsilon = 0,00001$. Это означает, что найдены все три корня уравнения с погрешностью, не превышающей 0,00001.

G	H	I
	<i>Начальное приближение x</i>	<i>Значение y(x)</i>
x1	-1,82	0,000000
x2	0,70	-0,000005
x3	1,72	0,000000

Рис. 1.24. Результат выполнения процедуры **Подбор параметра**

Примечание. При выполнении процедуры подбора параметра возможна ситуация, когда будет выполнено заданное число итераций, однако заданная точность не будет достигнута (решение не найдено). В этом случае следует выбрать другое начальное приближение для x и вновь повторить процедуру подбора параметра. В некоторых случаях может оказаться необходимым увеличить число итераций.

Таблица 1.3

Варианты для задания 2

№ варианта	p_1	p_2	p_3	$x_{нач}$	$x_{кон}$
1	1	3	0,5	-2	-1
2	5	2	-2	-2	0
3	2	2	0	-2	-1
4	1	2	0,5	-2	0
5	1	1	1	-2	2
6	-2	0,7	1	0	3
7	-3	1	2	-3	0
8	-1	1	-0,5	-3	0
9	1	3	0,5	-1	0
10	5	2	-2	0	2

Практическая работа 1.2. Использование функций работы с массивами Microsoft Excel в моделировании (на примере упрощенной модели межотраслевого баланса)

Цель работы — научиться использовать функции обработки массивов MS Excel в решении задач моделирования экономических процессов.

Порядок выполнения работы

1. Ознакомиться с постановкой задачи и математической моделью межотраслевого баланса.
2. Выбрать исходные данные в соответствии с номером варианта из табл. 1.5.
3. Заполнить расчетную таблицу MS Excel по аналогии с примером выполнения задания.

Описание модели межотраслевого баланса [7]. Предположим, что производственный сектор народного хозяйства разбит на n отраслей (энергетика, сельское хозяйство и т. д.). Рассмотрим отрасль с номером i ($i = 1, 2, \dots, n$), которая выпускает продукцию за некоторый промежуток времени (например, за год) в объеме x_i , который **называют валовым выпуском**. Часть продукции x_i используется для собственного производства в объеме x_{ii} , другая часть поступает в остальные отрасли для потребления при производстве в объемах x_{ij} ($j = 1, 2, \dots, i-1, i+1, \dots, n$), и некоторая часть объемом y_i — для потребления в непромышленной сфере (так называемый **конечный продукт**). Получаем соотношение:

$$x_i = x_{i1} + x_{i2} + \dots + x_{in} + y_i = \sum_{j=1}^n x_{ij} + y_i, \quad i = 1, 2, \dots, n.$$

Введем **коэффициенты прямых затрат** a_{ij} , которые показывают, сколько единиц продукции i -й отрасли затрачивается на производство одной единицы продукции в отрасли с номером j . Общее количество продукции i -й отрасли, поступающей в отрасль с номером j , равно $x_{ij} = a_{ij} \cdot x_j$. Получаем следующее соотношение баланса:

$$x_i = \sum_{j=1}^n a_{ij} \cdot x_j + y_i, \quad i = 1, 2, \dots, n. \quad (1.3)$$

Введя **вектор валового выпуска** X , **матрицу прямых затрат** A и **вектор конечного продукта** Y , модель (1.3) можно записать в матричном виде:

$$X = A \cdot X + Y, \quad (1.4)$$

где
$$X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}; A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}; Y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}.$$

Основная задача балансового анализа состоит в отыскании значений вектора валового выпуска X , которые при известной матрице прямых затрат A обеспечат заданные значения вектора конечного продукта Y . Иначе говоря, требуется решить матричное уравнение $X = A \cdot X + Y$ относительно X при известных A и Y .

Перенесем произведение $A \cdot X$ в левую сторону равенства и учтем, что

$$X = E \cdot X,$$

где
$$E = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$
 — единичная матрица.

Тогда уравнение $X = A \cdot X + Y$ перепишется в виде $(E - A) \cdot X = Y$. Если определитель матрицы $(E - A)$ не равен нулю, то решением уравнения является

$$X = (E - A)^{-1} \cdot Y. \quad (1.5)$$

Примечание. Существование решения уравнения (1.5) зависит от выполнения условия продуктивности матрицы A . Этот вопрос требует отдельного рассмотрения. Чтобы не усложнять задачу, в дальнейшем будем считать, что матрица A является продуктивной.

Пример выполнения задания

Экономическая система состоит из трех отраслей, для которых известны матрица прямых затрат A и вектор конечного продукта Y :

$$A = \begin{pmatrix} 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.0 \\ 0.3 & 0.1 & 0.2 \end{pmatrix}; Y = \begin{pmatrix} 200 \\ 100 \\ 300 \end{pmatrix}.$$

Требуется определить вектор валового выпуска X и межотраслевые поставки продукции x_{ij} . Считаем, что матрица A является продуктивной.

Вектор валового выпуска X рассчитывается по формуле $X = (E - A)^{-1} \cdot Y$. Межотраслевые поставки продукции x_{ij} вычисляются по формуле $x_{ij} = a_{ij} \cdot x_j$.

Заполнение расчетной таблицы MS Excel

В расчетах будут использоваться стандартные функции MS Excel, краткое описание которых приведено в табл. 1.4. Результатом использования каждой из функций является массив (матрица или вектор). Формулы, использующие эти функции, будут вводиться как формулы массива.

Формула массива, в отличие от обычной формулы, позволяет вычислить сразу несколько значений (массив). Такая формула заключается в фигурные скобки, ее ввод завершается нажатием комбинации клавиш <Ctrl>+<Shift>+<Enter> (в то время как ввод обычной формулы завершается нажатием клавиши <Enter>).

Таблица 1.4

Функции, применяемые для работы с массивами

Функция	Реализуемые действия
ТРАНСП(Массив)	Преобразует вертикальный диапазон ячеек в горизонтальный или наоборот
МОБР(Массив)	Вычисляет матрицу, обратную к матрице, которая задана в диапазоне ячеек Массив
МУМНОЖ(Массив1, Массив2)	Вычисляет произведение матриц, которые хранятся в диапазонах ячеек Массив1 и Массив2

Введем коэффициенты матрицы A в ячейки с адресами **A2:C4**, вектор Y – в ячейки **E2:E4**, единичную матрицу – в ячейки **G2:I4**. Подготовим шаблоны для размещения данных, получаемых в результате вычислений (рис. 1.25).

	A	B	C	D	E	F	G	H	I
1	Матрица A				Вектор Y		Матрица E		
2	0,3	0,1	0,4		200		1	0	0
3	0,2	0,5	0		100		0	1	0
4	0,3	0,1	0,2		300		0	0	1
5									
6	Матрица E-A				Матрица (E-A)⁻¹			Вектор X	
7									
8									
9									
10									
11	Транспонированный вектор X				Межотраслевые поставки				
12									
13									
14									

Рис. 1.25. Таблица-шаблон для ввода расчетных формул

Вычислим матрицу $E-A$, которая является разностью матриц E и A . Для этого выполним следующие действия:

- установим курсор мыши в левый верхний угол (это ячейка с адресом **A7**) результирующей матрицы, которая будет расположена в ячейках с адресами **A7:C9**;
- введем формулу **=G2-A2** для вычисления первого элемента результирующей матрицы $E-A$;
- выполним автозаполнение остальных ячеек, протянув выделение ячейки **A7** сначала вправо на ячейки **B7:C7**, а затем вниз на две следующие строки.

Вычислим матрицу $(E-A)^{-1}$, являющуюся обратной по отношению к матрице $E-A$. Для вычисления матрицы выполним следующие действия:

- выделим диапазон ячеек **E7:G9** для размещения матрицы;
- выполнив в меню **Вставка – Функция**, в появившемся окне выберем категорию **Математические**, затем – функцию **МОБР**;

- в диалоговом окне **Аргументы функции** введем в поле **Массив** диапазон ячеек **A7:C9** (протянув указатель мыши при нажатой левой кнопке от ячейки **A7** до ячейки **C9**);
- для закрытия окна нажмем комбинацию клавиш **<Ctrl>+<Shift>+<Enter>** (нажимать клавишу **OK** не нужно!).

В диапазоне ячеек **E7:G9** появится обратная матрица $(E-A)^{-1}$ (рис. 1.26).

E7		fx {=МОБР(A7:C9)}					
	A	B	C	D	E	F	G
1	Матрица A				Вектор Y		Матрица E
2	0,3	0,1	0,4		200		1
3	0,2	0,5	0		100		0
4	0,3	0,1	0,2		300		0
5							
6	Матрица E-A				Матрица (E-A) ⁻¹		
7	=G2-A2	=H2-B2	=I2-C2		=МОБР(A7:C9)	=МОБР(A7:C9)	=МОБР(A7:C9)
8	=G3-A3	=H3-B3	=I3-C3		=МОБР(A7:C9)	=МОБР(A7:C9)	=МОБР(A7:C9)
9	=G4-A4	=H4-B4	=I4-C4		=МОБР(A7:C9)	=МОБР(A7:C9)	=МОБР(A7:C9)

Рис. 1.26. Промежуточные результаты расчетов

Вычисление вектора $X = (E - A)^{-1} \cdot Y$ производится с помощью операции умножения матриц. Для вычисления X выполним следующие действия:

- выделим диапазон ячеек **I7:I9**, где будет расположен вектор X ;
- выполнив в меню **Вставка — Функция**, в появившемся окне выберем категорию **Математические**, затем — функцию **МУМНОЖ**;
- в диалоговом окне **Аргументы функции** в поле **Массив1** введем диапазон ячеек **E7:G9** (протянув указатель мыши при нажатой левой кнопке от ячейки **E7** до ячейки **G9**);
- в поле **Массив2** введем диапазон ячеек **E2:E4**;
- для закрытия окна нажмем комбинацию клавиш **<Ctrl>+<Shift>+<Enter>**.

В диапазоне ячеек **I7:I9** появится искомый вектор X (рис. 1.27).

	A	B	C	D	E	F	G	H	I
1	Матрица A				Вектор Y		Матрица E		
2	0,3	0,1	0,4		200		1	0	0
3	0,2	0,5	0		100		0	1	0
4	0,3	0,1	0,2		300		0	0	1
5									
6	Матрица E-A				Матрица (E-A) ⁻¹				Вектор X
7	0,7	-0,1	-0,4		2,04	0,61	1,02		775,51
8	-0,2	0,5	0		0,82	2,24	0,41		510,20
9	-0,3	-0,1	0,8		0,87	0,51	1,68		729,59
10									
11	Транспонированный вектор X				Межотраслевые поставки				
12	775,51	510,20	729,59		232,65	51,02	291,84		
13					155,10	255,10	0,00		
14					232,65	51,02	145,92		

Рис. 1.27. Результаты решения задачи

Межотраслевые поставки продукции x_{ij} вычисляются по формуле $x_{ij} = a_{ij} \cdot x_j$, где a_{ij} – элементы матрицы *A*, расположенной в ячейках **A2:C4**; x_j – элементы вектора *X*, расположенного в ячейках **I7:I9**.

Для определения x_{ij} сначала *вычислим транспонированный вектор для вектора X*. Это необходимо для согласования размерностей при последующих операциях. С этой целью выполним следующие действия:

- выделим ячейки **A12:C12**, в которых будет располагаться транспонированный вектор;
- выполнив в меню **Вставка – Функция**, в появившемся окне выберем категорию **Ссылки и массивы**, затем – функцию **ТРАНСП**;
- в диалоговом окне **Аргументы функции** в поле **Массив** диапазон ячеек **I7:I9**;
- нажмем сочетание клавиш **<Ctrl>+<Shift>+<Enter>**.

Межотраслевые		
=A2*A12	=B2*B12	=C2*C12
=A3*A12	=B3*B12	=C3*C12
=A4*A12	=B4*B12	=C4*C12

Рис. 1.28. Формулы для расчета межотраслевых поставок

Вычислим межотраслевые поставки продукции x_{ij} . Для этого выполним следующие действия:

- в ячейку **E12**, в которой будет расположено значение x_{11} , введем формулу =A2*A12, которая означает, что $x_{11} = a_{11} \cdot x_1$;
- формулу из ячейки **E12** скопируем в остальные ячейки первой строки (**F12:G12**), используя автозаполнение, при этом будут вычислены $x_{12} = a_{12} \cdot x_2$ и $x_{13} = a_{13} \cdot x_3$;
- в ячейку **E13** введем формулу =A3*A12 и аналогично предыдущему пункту получим значения $x_{21} = a_{21} \cdot x_1$, $x_{22} = a_{22} \cdot x_2$ и $x_{23} = a_{23} \cdot x_3$;
- повторим аналогичные действия для ячеек **E14:G14**.

В результате все межотраслевые поставки продукции будут найдены в ячейках **E12:G14** (рис. 1.27). На рис. 1.28 показан фрагмент таблицы с расчетом межотраслевых поставок в режиме показа формул.

Таблица 1.5

Варианты заданий для практической работы 1.2

№ варианта	Исходные данные для расчетов
1	$A = \begin{pmatrix} 0.33 & 0.32 & 0.21 \\ 0.22 & 0.31 & 0.0 \\ 0.11 & 0.25 & 0.35 \end{pmatrix}, Y = \begin{pmatrix} 200 \\ 150 \\ 250 \end{pmatrix}$
2	$A = \begin{pmatrix} 0.12 & 0.20 & 0.3 \\ 0.25 & 0.35 & 0.15 \\ 0.33 & 0.00 & 0.45 \end{pmatrix}, Y = \begin{pmatrix} 300 \\ 150 \\ 450 \end{pmatrix}$
3	$A = \begin{pmatrix} 0.55 & 0.20 & 0.15 \\ 0.15 & 0.35 & 0.25 \\ 0.00 & 0.25 & 0.15 \end{pmatrix}, Y = \begin{pmatrix} 400 \\ 150 \\ 550 \end{pmatrix}$
4	$A = \begin{pmatrix} 0.40 & 0.25 & 0.00 \\ 0.14 & 0.52 & 0.15 \\ 0.17 & 0.20 & 0.3 \end{pmatrix}, Y = \begin{pmatrix} 830 \\ 620 \\ 280 \end{pmatrix}$

№ варианта	Исходные данные для расчетов
5	$A = \begin{pmatrix} 0.05 & 0.22 & 0.26 \\ 0.58 & 0.11 & 0.0 \\ 0.17 & 0.34 & 0.37 \end{pmatrix}, Y = \begin{pmatrix} 430 \\ 650 \\ 910 \end{pmatrix}$
6	$A = \begin{pmatrix} 0.00 & 0.52 & 0.00 \\ 0.27 & 0.31 & 0.33 \\ 0.63 & 0.12 & 0.00 \end{pmatrix}, Y = \begin{pmatrix} 9800 \\ 450 \\ 150 \end{pmatrix}$
7	$A = \begin{pmatrix} 0.21 & 0.42 & 0.00 \\ 0.32 & 0.31 & 0.20 \\ 0.41 & 0.21 & 0.23 \end{pmatrix}, Y = \begin{pmatrix} 1200 \\ 6150 \\ 7250 \end{pmatrix}$
8	$A = \begin{pmatrix} 0.17 & 0.12 & 0.19 \\ 0.28 & 0.21 & 0.08 \\ 0.19 & 0.32 & 0.35 \end{pmatrix}, Y = \begin{pmatrix} 500 \\ 750 \\ 350 \end{pmatrix}$
9	$A = \begin{pmatrix} 0.21 & 0.42 & 0.21 \\ 0.32 & 0.31 & 0.10 \\ 0.41 & 0.12 & 0.13 \end{pmatrix}, Y = \begin{pmatrix} 2900 \\ 1950 \\ 2950 \end{pmatrix}$
10	$A = \begin{pmatrix} 0.31 & 0.22 & 0.11 \\ 0.23 & 0.31 & 0.10 \\ 0.41 & 0.20 & 0.13 \end{pmatrix}, Y = \begin{pmatrix} 6600 \\ 3150 \\ 3950 \end{pmatrix}$

Практическая работа 1.3. Моделирование на основе метода Монте-Карло

Цель работы – научиться:

- строить простейшие модели стохастических процессов с использованием программирования на алгоритмическом языке и средствами электронных таблиц;
- использовать генераторы псевдослучайных чисел для имитации стохастических процессов.

Порядок выполнения работы

1. Ознакомиться с постановкой задачи.
2. Выбрать условие задачи в соответствии с номером варианта из табл. 1.8.
3. Заполнить расчетную таблицу MS Excel по аналогии с примером выполнения задания 1.
4. Составить программу на языке Паскаль, имитирующую статистический эксперимент, по аналогии с примером выполнения задания 2.
5. Выполнить расчеты с помощью программы.
6. Сравнить результаты расчетов в MS Excel с результатами расчетов с помощью Паскаль-программы и точным значением вероятности отказа системы.

Условие задачи

Задана схема соединения приборов, составляющих систему контроля качества продукции. Вероятность отказа каждого из приборов в течение некоторого интервала времени одинакова и равна $P1$. Приборы выходят из строя независимо друг от друга. Используя метод статистических испытаний (метод Монте-Карло), найти вероятность P того, что система откажет в течение рассматриваемого периода времени.

Задание 1

Выполнить имитационное моделирование эксперимента средствами табличного процессора MS Excel, приняв число экспериментов равным **10**, число испытаний в каждом эксперименте – равным **1000**. Для имитации состояния прибора в испытании использовать стандартную функцию СЛЧИС. Искомую вероятность отказа системы определить как среднее значение по результатам 10 экспериментов.

Задание 2

Выполнить имитационное моделирование эксперимента средствами языка программирования Паскаль. Для имитации состояния прибора в испытании использовать стандартную функцию *Random*. Предусмотреть в программе ввод значений числа экспериментов и числа испытаний системы в одном эксперименте. Выполнить расчеты для разных значений числа экспериментов (m) и числа испытаний в эксперименте (n). Заполнить таблицу с результатами расчетов в документе MS Word (табл. 1.7)

Описание процедуры моделирования

Обычно числа, полученные с помощью программного генератора случайных чисел, подчиняются равномерному закону распределения. На практике это означает, что **вероятности появления любого из чисел из некоторого заданного интервала одинаковы**.

Допустим, все случайные числа принадлежат интервалу $[0,1)$. Тогда попадание сгенерированного числа в интервал $[0, 0.1)$ будет соответствовать выходу из строя испытуемого прибора, если вероятность этого события равна 0.1. Попадание числа в интервал $[0.1, 1)$ будет соответствовать безотказной работе прибора (рис. 1.29).



Рис. 1.29. Моделирование работы прибора

Примечание. В случае если вероятность отказа каждого прибора равна 0.2, отказ прибора имитируется попаданием случайного числа в интервал $[0, 0.2)$. Если вероятность отказа каждого прибора равна 0.3, отказ прибора имитируется попаданием случайного числа в интервал $[0, 0.3)$ и т. д.

Так будет моделироваться работа любого из приборов, составляющих систему контроля качества. Компьютерная генерация случайных чисел, количество которых равно количеству приборов в системе, с проверкой условия, соответствующего схеме соединения приборов, будет соответствовать **одному испытанию** со всей системой. Рассмотрим пример имитации испытаний системы.

Пример 1.7. Система проверки качества состоит из 6 приборов, схема соединения которых представлена на рис. 1.30. Вероятность выхода из строя каждого отдельного прибора известна и равна $P1=0.2$. Отказ прибора имитируется выполнением условия $\Pi_i < 0.2$, где Π_i – значение сгенерированного случайного числа, соответствующего прибору с номером i ($i = 1, 2, \dots, 6$).

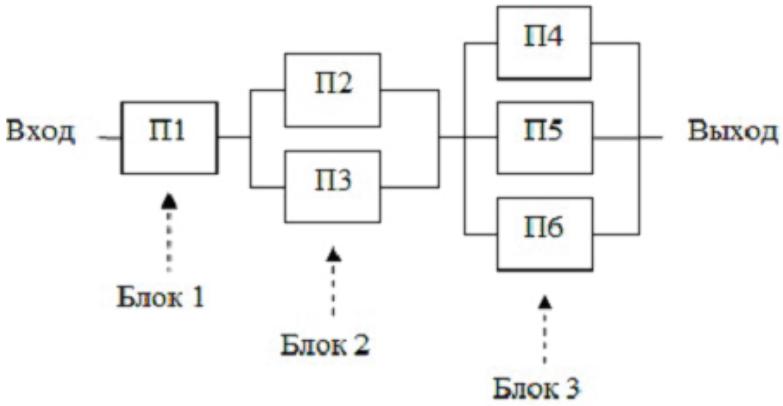


Рис. 1.30. Схема соединения приборов в системе для примера 1.7

Анализ рис. 1.30 позволяет сделать вывод, что сигнал от входа системы не поступит на выход (фиксируется отказ системы), если выйдет из строя хотя бы один из блоков, обозначенных номерами 1, 2, 3. Отказ первого блока описывается выполнением условия $\Pi_1 < 0.2$, отказ второго – одновременным выполнением двух условий: $\Pi_2 < 0.2$ и $\Pi_3 < 0.2$, отказ третьего – одновременным выполнением трех условий: $\Pi_4 < 0.2$ и $\Pi_5 < 0.2$ и $\Pi_6 < 0.2$. Для всей системы в целом можно записать условие отказа в следующем виде:

если ($\Pi_1 < 0.2$) или ($\Pi_2 < 0.2$ и $\Pi_3 < 0.2$) или ($\Pi_4 < 0.2$ и $\Pi_5 < 0.2$ и $\Pi_6 < 0.2$), то фиксируется отказ системы.

Теперь, если предположить, что в ячейках электронной таблицы $A1, B1, C1, D1, E1, F1$ сгенерированы случайные числа, соответствующие состоянию приборов $\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5$ и Π_6 соответственно, то функция для проверки

состояния всей системы может быть записана так, как показано на рис. 1.31. Результатом вычисления по формуле будет ноль (система отказала) либо единица (система исправна).

Под *экспериментом* будем понимать серию последовательных испытаний. Построенная модель позволит многократно имитировать испытания, фиксируя всякий раз результат (безотказная работа или отказ системы). При большом количестве испытаний отношение числа отказов системы к общему числу испытаний даст приближенное значение вероятности P выхода из строя системы за рассматриваемый временной интервал.

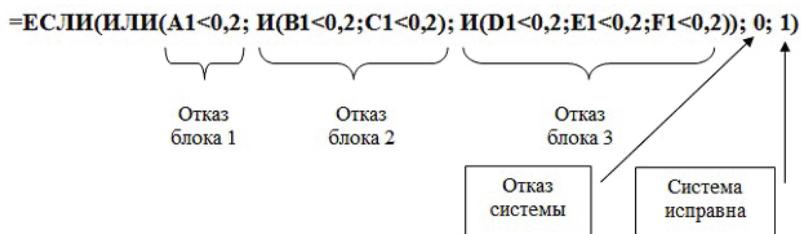


Рис. 1.31. Формула в MS Excel для проверки состояния системы в результате испытания

Пример выполнения практической работы

Выполним расчеты для схемы соединения приборов, показанной на рис. 1.32, при $P1 = 0.1$. Анализ схемы соединения приборов позволяет сделать вывод, что вся система выйдет из строя, если хотя бы один из приборов выйдет из строя. В условиях имитационной модели этот факт формулируется следующим образом: *если хотя бы одно из трех случайных чисел, полученных в результате имитации испытания системы, попадает в интервал $[0, 0.1)$, фиксируется отказ всей системы в данном испытании.*

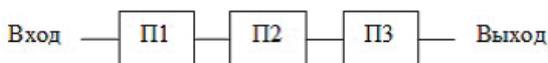


Рис. 1.32. Схема соединения приборов в системе контроля качества

Задание 1

Выполним имитационное моделирование эксперимента средствами табличного процессора MS Excel.

Для генерации случайных чисел в электронной таблице MS Excel используется стандартная функция **СЛЧИС()**. Функция возвращает равномерно распределенное случайное число, большее либо равное 0 и меньшее 1.

Для имитации испытаний системы заполним таблицу, показанную на рис. 1.33. При заполнении столбца **Номер испытания** введем значения 1 и 2 в ячейки **A2** и **A3** соответственно. Затем выделим эти ячейки и «протянем» вниз до ячейки **A1001**, используя маркер заполнения. В ячейку **B2** введем формулу **=СЛЧИС()** и «протянем» сначала вправо на ячейки **C2** и **D2**, а затем вниз до ячеек **B1001:D1001**.

	A	B	C	D	E
	Номер испытания				
1		Прибор 1	Прибор 2	Прибор 3	Состояние системы
2	1	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B2<0,1;C2<0,1;D2<0,1);0;1)
3	2	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B3<0,1;C3<0,1;D3<0,1);0;1)
4	3	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B4<0,1;C4<0,1;D4<0,1);0;1)
5	4	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B5<0,1;C5<0,1;D5<0,1);0;1)
6	5	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B6<0,1;C6<0,1;D6<0,1);0;1)
7	6	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B7<0,1;C7<0,1;D7<0,1);0;1)
8	7	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B8<0,1;C8<0,1;D8<0,1);0;1)
9	8	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B9<0,1;C9<0,1;D9<0,1);0;1)
10	9	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B10<0,1;C10<0,1;D10<0,1);0;1)
11	10	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B11<0,1;C11<0,1;D11<0,1);0;1)
12	11	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B12<0,1;C12<0,1;D12<0,1);0;1)
1000	999	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B1000<0,1;C1000<0,1;D1000<0,1);0;1)
1001	1000	=СЛЧИС()	=СЛЧИС()	=СЛЧИС()	=ЕСЛИ(ИЛИ(B1001<0,1;C1001<0,1;D1001<0,1);0;1)

Рис. 1.33. Таблица имитации испытаний в режиме показа формул

В ячейку **E2** введем формулу **=ЕСЛИ(ИЛИ(B2<0,1;C2<0,1;D2<0,1);0;1)**. Эта формула фиксирует состояние системы, полученное в испытании с номером 1. Если хотя бы один прибор выходит из строя (**ИЛИ(B2<0,1;C2<0,1;D2<0,1)**), то фиксируется отказ всей системы (значение в ячейке **E2** равно 0), в противном случае система работоспособна (значение в ячейке **E2** равно 1). Используя маркер заполнения, распространим формулу на ячейки диапазона **E3:E1001**.

Для удобства работы с таблицей скроем строки с 13 по 999, выделив их и выбрав из контекстного меню (открывается щелчком правой кнопки мыши) команду **Скрыть**. В ячейку **F2** введем формулу для вычисления вероятности отказа системы по результатам 1000 испытаний. За вероятность принимается отношение числа зафиксированных отказов к общему числу испытаний: **=СЧЁТЕСЛИ(E2:E1001;0)/1000**.

С целью возможности управления экспериментом установим пересчет формул на рабочем листе в ручном режиме, выполнив в меню **Сервис – Параметры – вкладка Вычисления – вручную**. В этом случае для имитации серии испытаний (пересчета значений рабочего листа таблицы) потребуется нажимать на клавиатуре клавишу <F9>.

В ячейки диапазона **G2:G11** введем номера экспериментов от 1 до 10 (рис. 1.34). В ячейки **H2:H11** будем заносить вычисленные в каждом эксперименте значения из ячейки **F2**. В ячейку **H12** внесем формулу для вычисления среднего арифметического значения вероятности отказа системы по результатам 10 экспериментов: **=СРЗНАЧ(H2:H11)**.

	A	B	C	D	E	F	G	H
	Номер испытания	Прибор 1	Прибор 2	Прибор 3	Состояние системы	Вероятность отказа системы в текущем эксперименте	Номер эксперимента	Зафиксированная вероятность отказа системы
1								
2	1	0,0972	0,3252	0,5998	0	0,261	1	0,278
3	2	0,2551	0,0893	0,5421	0		2	0,250
4	3	0,5277	0,1100	0,4013	1		3	0,263
5	4	0,9842	0,9320	0,8107	1		4	0,261
6	5	0,8710	0,7206	0,1392	1		5	
7	6	0,8665	0,0949	0,3259	0		6	
8	7	0,0492	0,4134	0,6870	0		7	
9	8	0,2037	0,9354	0,5590	1		8	
10	9	0,1445	0,0050	0,3610	0		9	
11	10	0,3791	0,4699	0,8709	1		10	
12	11	0,1643	0,9000	0,7776	1	Среднее значение		0,2637
1000	999	0,8499	0,7428	0,6525	1			
1001	1000	0,4105	0,8463	0,7776	1			

Рис. 1.34. Таблица имитации испытаний в режиме показа результатов расчета

Опишем последовательность действий, имитирующих **один эксперимент**:

- нажимаем на клавиатуре клавишу <F9>;
- выделив ячейку F2, выполняем **Правка – Копировать**;
- выделив ячейку столбца H, соответствующую номеру эксперимента, выполняем **Правка – Специальная вставка... – значения** (в ячейку будет вставлено вычисленное значение вероятности).

Полученное в результате расчетов значение в ячейке **H12** даст нам приближенное значение вероятности отказа системы. Для оценки точности полученного с помощью имитационной модели значения вероятности можно сравнить его с точным значением вероятности отказа системы, вычисленным по формулам теории вероятностей, которое для рассматриваемого случая равно **0.271**.

Задание 2

Выполним имитационное моделирование эксперимента средствами языка программирования Паскаль. В программе на языке Паскаль для включения генератора случайных чисел используется процедура **Randomize**. Функция **Random** возвращает случайное число типа **Real** из диапазона [0, 1). Введем соответствие обозначений в программе и задаче (табл. 1.6).

Таблица 1.6

Соответствие обозначений в задаче и программе

В задаче	В программе
Количество экспериментов	<i>m</i>
Количество испытаний в каждом эксперименте	<i>n</i>
Текущий номер эксперимента	<i>j</i>
Текущий номер испытания в рамках эксперимента	<i>i</i>
Состояние прибора 1 в испытании с номером <i>i</i>	<i>k1 [i]</i>
Состояние прибора 2 в испытании с номером <i>i</i>	<i>k2 [i]</i>
Состояние прибора 3 в испытании с номером <i>i</i>	<i>k3 [i]</i>
Количество отказов системы в каждом эксперименте	<i>kol</i>
Вероятность отказа системы в каждом эксперименте	<i>p</i>
Сумма вероятностей отказа системы по всем экспериментам	<i>sum</i>
Среднее значение вероятности отказа системы по результатам всех экспериментов	<i>sred</i>

В окне редактора Turbo Pascal введем с клавиатуры **текст программы**.

```
Program MonteCarlo; {1}
Var i, j, n, m, kol: word; {2}
k1, k2, k3: array [ 1.. 2000 ] of real; {3}
p, sred, sum: real; {4}
begin {5}
Randomize; {Запуск генератора случайных чисел} {6}
Repeat {7}
    Writeln ( ' Введите число экспериментов >=1 и <=1000' ); {8}
    Readln ( m ); {9}
until ( m >= 1 ) and ( m <= 1000 ); {10}
Repeat {11}
    Writeln ( ' Введите число испытаний >=100 и <=2000' ); {12}
    Readln ( n ); {13}
until ( n >= 100 ) and ( n <= 2000 ); {14}
sum:= 0; {15}
for j:= 1 to m do {Цикл по номеру эксперимента} {16}
    begin {17}
        kol:= 0; {18}
        for i:= 1 to n do {Цикл по номеру испытания} {19}
            begin {20}
                k1 [ i ]:= random; k2 [ i ]:= random; k3 [ i ]:= random; {21}
                if ( k1 [ i ] < 0.1 ) or ( k2 [ i ] < 0.1 ) or ( k3 [ i ] < 0.1 ) then kol:=
                    kol + 1; {22}
            end; {23}
            p:= kol / n; writeln ( ' Вероятность отказа системы = ', p:
                8: 4 ); {24}
            sum:= sum + p; {25}
        end; {26}
    sred:= sum / m; {27}
    writeln ( ' Среднее значение вероятности отказа системы = ',
        sred: 8: 4 ); {28}
    readln {29}
end. {30}
```

Комментарий к программе

Строка 6 – запуск генератора случайных чисел.

Строки 7, 8, 9, 10 – ввод числа экспериментов с контролем ввода данных.

Строки 11, 12, 13, 14 – ввод числа испытаний с контролем ввода данных.

Строки 16, 17, 26 – границы цикла (начало и конец) по номеру эксперимента.

Строка 18 – обнуление счетчика числа отказов системы в каждом эксперименте.

Строки 19, 20, 23 – границы цикла (начало и конец) по номеру испытания в рамках одного эксперимента.

Строка 21 – имитация одного испытания системы (присваивание элементам массивов $k1[i]$, $k2[i]$, $k3[i]$ случайных значений из интервала $[0, 1)$, имитирующих состояние прибора в испытании с номером i).

Строка 22 – анализ состояния системы (отказ или исправное состояние) в зависимости от значений $k1[i]$, $k2[i]$, $k3[i]$ и изменение значения счетчика (переменная kol) в случае отказа системы в текущем испытании.

Строка 24 – вычисление и вывод значения вероятности отказа системы по результатам текущего эксперимента.

Строка 25 – вычисление суммы вероятностей отказа системы по результатам всех экспериментов.

Строки 27, 28 – вычисление и вывод среднего значения вероятности отказа системы по результатам всех экспериментов.

Выполним расчеты по программе для разных значений m и n и зафиксируем эти результаты в табл. 1.7. Анализ результатов расчетов позволяет принять за приближенное значение вероятности отказа значение $P = 0,27$.

Таблица 1.7

Результаты расчетов по методу Монте-Карло

Номер расчета	Число экспериментов (m)	Число испытаний в эксперименте (n)	Вероятность отказа системы
1	10	100	0,2670
2	10	1000	0,2665
3	10	2000	0,2720

Окончание табл. 1.7

Номер расчета	Число экспериментов (m)	Число испытаний в эксперименте (n)	Вероятность отказа системы
4	100	100	0,2744
5	100	1000	0,2715
6	100	2000	0,2709
7	1000	100	0,2704
8	1000	1000	0,2712
9	1000	2000	0,2709

Таблица 1.8

Варианты заданий

№ варианта	Вид схемы соединения приборов в системе	Значение P_I	Точное значение вероятности отказа системы (для контроля правильности расчетов)
1		0,2	0,232
2		0,3	0,363
3		0,4	0,064
4		0,5	0,125
5		0,3	0,1719
6		0,4	0,2944

№ варианта	Вид схемы соединения приборов в системе	Значение P_I	Точное значение вероятности отказа системы (для контроля правильности расчетов)
7		0,3	0,3189
8		0,2	0,2064
9		0,1	0,1981
10		0,1	0,3439

Практическая работа 1.4. Решение задачи об оптимальном распределении ресурсов при выпуске продукции

Цель работы – научиться использовать инструмент **Поиск решения** программы Microsoft Excel для построения оптимизационных математических моделей.

Порядок выполнения работы

1. Ознакомиться с постановкой задачи и математической моделью задачи об оптимальном распределении ресурсов при выпуске продукции.
2. Выбрать условие задачи в соответствии с номером варианта.
3. Заполнить расчетную таблицу MS Excel по аналогии с примером выполнения задания.

Условие задачи об оптимальном распределении ресурсов при выпуске продукции [7]. Предположим, что предприятие выпускает n различных типов продукции. Для производства

требуются m различных видов ресурсов (сырья, вспомогательных материалов, рабочего и машинного времени и т. д.). Эти ресурсы ограничены и составляют в планируемый период b_1, b_2, \dots, b_m условных единиц. Известны также коэффициенты a_{ij} , которые показывают, сколько единиц i -го ресурса требуется для производства единицы продукции j -го типа ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$). Задана цена c_j ($j = 1, 2, \dots, n$) реализации единицы продукции j -го типа. В планируемый период все показатели b_i, a_{ij} и c_j предполагаются постоянными. Допустим, что предприятие будет выпускать продукцию каждого типа в количестве $x_j, j = 1, 2, \dots, n$.

Требуется составить оптимальный план выпуска продукции, т. е. найти такие значения переменных x_1, x_2, \dots, x_n , чтобы обеспечить предприятию получение максимального общего дохода от реализации всей продукции и чтобы на ее производство хватило имеющихся в распоряжении ресурсов. Предполагается, что рынок сбыта продукции неограничен.

Сформулированная задача является задачей линейного программирования, для решения которой разработаны специальные методы. Мы применим для решения задачи процедуру **Поиск решения**.

Пример выполнения задания

Фирма выпускает продукцию четырех типов — *Продукт1*, *Продукт2*, *Продукт3*, *Продукт4*, для изготовления которой требуются ресурсы трех видов: *трудовые*, *сырьевые*, *финансовые*. Количество ресурса каждого вида, необходимое для выпуска единицы продукции данного типа, а также цена реализации единицы каждого типа продукции приведены в табл. 1.9. Требуется определить, в каком количестве следует выпускать продукцию каждого типа, чтобы доход от реализации всей произведенной продукции был максимальным.

Процесс решения задачи включает следующие этапы:

- 1) формулировка математической модели как задачи линейного программирования;
- 2) создание расчетной таблицы в MS Excel;
- 3) использование процедуры **Поиск решения** для нахождения оптимального решения.

Таблица 1.9

Виды ресурсов	Тип продукции				Наличие ресурса
	Продукт1	Продукт2	Продукт3	Продукт4	
Трудовые	1	1	1	1	16
Сырьевые	6	5	4	3	110
Финансовые	4	6	10	13	100
Цена реализации	60	70	120	130	

Формулировка математической модели

Обозначим через x_1, x_2, x_3, x_4 количество выпускаемой продукции каждого типа. Общий доход от реализации четырех типов продукции вычисляется как $60 \cdot x_1 + 70 \cdot x_2 + 120 \cdot x_3 + 130 \cdot x_4$. **Целевая функция** примет вид:

$$60 \cdot x_1 + 70 \cdot x_2 + 120 \cdot x_3 + 130 \cdot x_4 \rightarrow \max. \quad (1.6)$$

Запишем **ограничения на ресурсы**. Если для выпуска единицы продукции **Продукт1** требуется 6 единиц сырья, тогда для выпуска всей продукции первого типа требуется $6 \cdot x_1$ единиц сырья, где x_1 – количество выпускаемой продукции **Продукт1**. С учетом того, что для других типов продукции зависимости будут аналогичны, ограничение для первого ресурса (сырье) запишется в виде: $6 \cdot x_1 + 5 \cdot x_2 + 4 \cdot x_3 + 3 \cdot x_4 \leq 110$. В этом ограничении левая часть равна величине требуемого ресурса, а правая показывает количество имеющегося ресурса. Аналогично можно составить ограничения для других видов ресурсов (трудовые и финансовые). Таким образом, **ограничения на ресурсы** запишутся в виде:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &\leq 16; \\ 6 \cdot x_1 + 5 \cdot x_2 + 4 \cdot x_3 + 3 \cdot x_4 &\leq 110; \\ 4 \cdot x_1 + 6 \cdot x_2 + 10 \cdot x_3 + 13 \cdot x_4 &\leq 100. \end{aligned} \quad (1.7)$$

Необходимо учесть, что количество выпускаемой продукции не может быть отрицательным. Отсюда появляются **ограничения на переменные**:

$$x_1 \geq 0; x_2 \geq 0; x_3 \geq 0; x_4 \geq 0. \quad (1.8)$$

Примечание. В тех случаях, когда производимая продукция представляет собой штучные изделия, т. е. количество продукции каждого типа должно быть целым положительным числом, необходимо добавить дополнительные ограничения:

$$x_j - \text{целое}, j = 1, 2, \dots, n \quad (1.9)$$

Заполнение расчетной таблицы MS Excel

Форма для ввода исходных данных и расчетных формул представлена на рис. 1.36. В ячейках **B3:E3** должны находиться искомые значения x_1, x_2, x_3, x_4 . Запишем в эти ячейки нулевые значения. В ячейки **B5:E5** запишем правые части ограничений на переменные из математической модели (1.8), т. е. также нулевые значения. В ячейки **B7:E7** запишем коэффициенты целевой функции (1.6). Затраты ресурсов на производство каждого продукта запишем в ячейки диапазона **B12:E14**. Правые части ограничений на ресурсы (1.7) запишем в ячейки **C17:C19**.

В ячейку **B8** введем формулу для вычисления значения целевой функции **=СУММПРОИЗВ (B7:E7; B3:E3)**. Функция **СУММПРОИЗВ** перемножает соответствующие элементы заданных массивов и возвращает сумму произведений. В нашем случае значение в ячейке **B8** будет вычисляться по формуле **=B7*B3+C7*C3+D7*D3+E7*E3**.

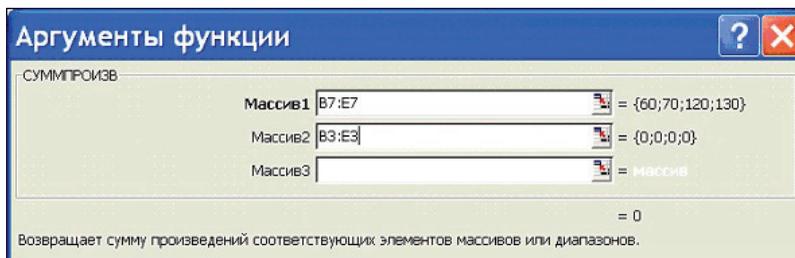


Рис. 1.35. Задание параметров функции **СУММПРОИЗВ**

	A	B	C	D	E	F
1	<i>Переменные</i>					<i>Комментарии</i>
2		Продукт 1	Продукт 2	Продукт 3	Продукт 4	
3	Значения переменных x_1, x_2, x_3, x_4	0	0	0	0	$x_1 >= 0; x_2 >= 0;$ $x_3 >= 0; x_4 >= 0$
4	Характер ограничений на переменные	$>=$	$>=$	$>=$	$>=$	
5	Ограничения на значения переменных	0	0	0	0	
6	<i>Целевая функция</i>					
7	Коэффициенты целевой функции	60	70	120	130	
8	Значение целевой функции	=СУММПРОИЗВ(В7;Е7;Е3;Е3)				$60x_1 + 70x_2 + 120x_3 + 130x_4 \rightarrow \max$
9	Характер оптимизации	Максимизация				
10	<i>Затраты ресурсов на производство</i>					
11	Ресурсы	Продукт 1	Продукт 2	Продукт 3	Продукт 4	
12	Трудовые	1	1	1	1	
13	Сырье	6	5	4	3	
14	Финансы	4	6	10	13	
15	<i>Ограничения на ресурсы</i>					
16	Левая часть	Знак сравнения	Правая часть			
17	=СУММПРОИЗВ(\$B\$3:\$E\$3;B12:E12)	$<=$	16			$x_1 + x_2 + x_3 + x_4 \leq 16$
18	=СУММПРОИЗВ(\$B\$3:\$E\$3;B13:E13)	$<=$	110			$6x_1 + 5x_2 + 4x_3 + 3x_4 \leq 110$
19	=СУММПРОИЗВ(\$B\$3:\$E\$3;B14:E14)	$<=$	100			$4x_1 + 6x_2 + 10x_3 + 13x_4 \leq 100$

Рис. 1.36. Таблица с формулами

При вводе формулы в диалоговом окне **Мастера функций** следует выбрать категорию **Математические**, функцию **СУММПРОИЗВ**, на экране появится диалоговое окно для ввода аргументов функции (рис. 1.35). Ввод формулы в ячейку **A17** осуществляется аналогично, с той лишь разницей, что в формуле используются абсолютные адреса **\$B\$3:\$E\$3**, для того чтобы можно было копировать формулу из ячейки **A17** в ячейки **A18** и **A19**, используя маркер заполнения.

Использование процедуры «Поиск решения»

Процедура **Поиск решения** вызывается из меню **Сервис**. Если в меню **Сервис** отсутствует команда **Поиск решения**, следует выбрать команду **Надстройки**. На экран выводится диалоговое окно для установки дополнительных надстроек программы MS Excel (рис. 1.37).

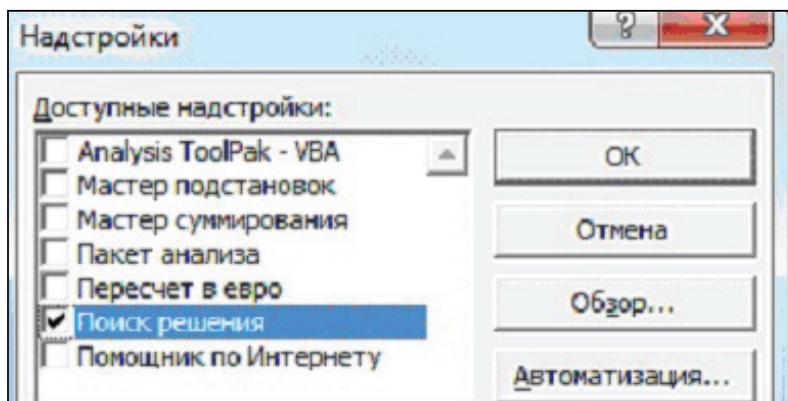


Рис. 1.37. Диалоговое окно для установки надстройки **Поиск решения**

Параметры поиска решения определяются в диалоговом окне **Поиск решения** (рис. 1.38). Рассмотрим подробнее ввод ограничений на переменные и ресурсы.

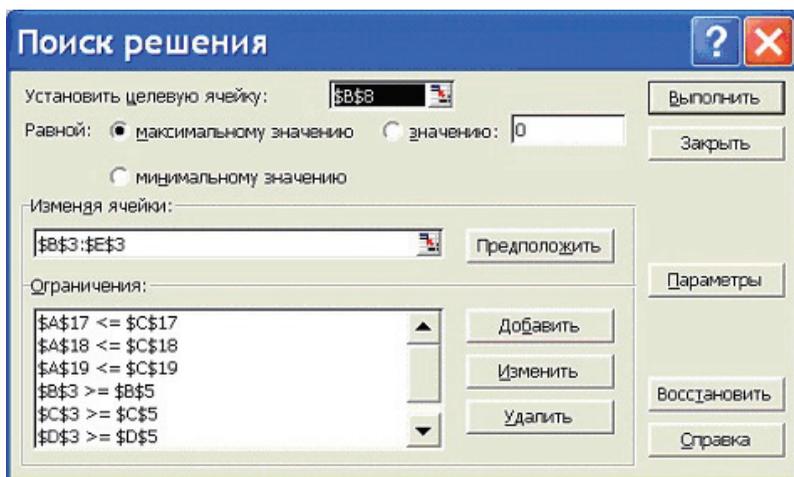


Рис. 1.38. Диалоговое окно **Поиск решения**

Щелчком по кнопке **Добавить** открывается окно **Добавление ограничения** (рис. 1.39). Для добавления ограничения $x_1 \geq 0$ в поле **Ссылка на ячейку** вводим адрес ячейки **\$B\$3** (щелкаем ячейку на рабочем листе таблицы). Для ввода значения в поле **Ограничение** щелкаем ячейку **\$B\$5**. Выбираем знак отношения \geq . Аналогично вводим ограничения на другие переменные и ограничения на ресурсы. Всего должно быть введено семь ограничений (3 ограничения на ресурсы и 4 ограничения на переменные):

$$\left\{ \begin{array}{l} \$A\$17 \leq \$C\$17 \\ \$A\$18 \leq \$C\$18 \\ \$A\$19 \leq \$C\$19 \end{array} \right. \text{ Ограничения на ресурсы}$$

$$\left\{ \begin{array}{l} \$B\$3 \geq \$B\$5 \\ \$C\$3 \geq \$C\$5 \\ \$D\$3 \geq \$D\$5 \\ \$E\$3 \geq \$E\$5 \end{array} \right. \text{ Ограничения на переменные}$$

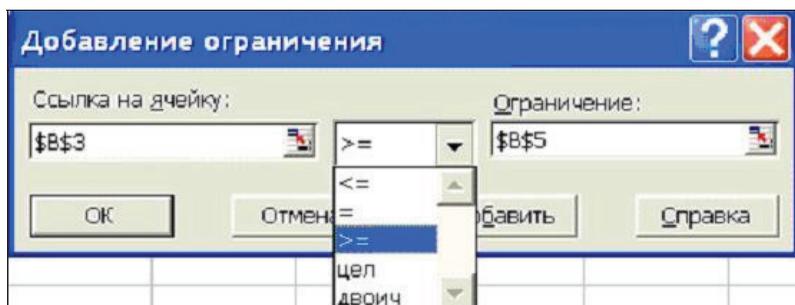


Рис. 1.39. Ввод ограничений задачи

При вводе ограничений можно указывать не отдельные ячейки, а диапазоны ячеек. В этом случае ограничения на ресурсы примут вид: $\$A\$17:\$A\$19 \leq \$C\$17:\$C\19 . Аналогично ограничения на переменные запишутся в виде: $B\$3:E\$3 \leq B\$5:E\5 .

Примечание. Если количество производимой продукции каждого вида должно быть целым числом, необходимо добавить ограничения (1.9): $B\$3:E\3 – целое.

Запуск процедуры **Поиск решения** производится щелчком по кнопке **Выполнить** в окне **Поиск решения**. На экране появится окно **Результаты поиска решения** (рис. 1.40). В диалоговом окне можно **сохранить найденное решение** или **восстановить исходные значения**.

На рис. 1.40 можно видеть, что оптимальное решение $Продукт1=B3=10$; $Продукт2=C3=0$; $Продукт3=D3=6$; $Продукт4=E3=0$. Максимальный общий доход будет составлять $B8=1320$ (условных денежных единиц), а количество использованных ресурсов равно: $трудо\ вых=A17=16$, $сырье\ вых=A18=84$, $финансо\ вых=A19=100$. Сохраним найденное значение.

	A	B	C	D	E
1	<i>Переменные</i>				
2		Продукт 1	Продукт 2	Продукт 3	Продукт 4
3	Значения переменных x_1, x_2, x_3, x_4	10	0	6	0
4	Характер ограничений на переменные	>=	>=	>=	>=
5	Ограничения на значения переменных	0	0	0	0
6	<i>Целевая функция</i>				
7	Коэффициенты целевой функции	60	70	120	130
8	Значение целевой функции	1320			
9	Характер ограничений	Максимизация			
10					
11	Ресурсы				ст 4
12	Трудовые				
13	Сырье				
14	Финансы				
15					
16	<i>Левая часть</i>	<i>сравнения</i>	<i>правая часть</i>		
17	16	<=	16		
18	84	<=	110		
19	100	<=	100		

Результаты поиска решения

Решение найдено. Все ограничения и условия оптимальности выполнены.

Сохранить найденное решение!
 Восстановить исходные значения

Тип отчета
Результаты
Устойчивость
Пределы

ОК Отмена Сохранить сценарий... Справка

Рис. 1.40. Результат выполнения процедуры Поиск решения

Варианты заданий

Вариант 1. На заводе используется сталь трех марок: А, В и С. Запасы стали каждой из марок равны соответственно 1000, 1600 и 1200 единиц. Завод выпускает два вида изделий. Для выпуска каждого изделия первого вида используется по одной единице стали всех марок. Для выпуска каждого изделия второго вида используется две единицы стали марки В, одна единица стали марки С и не используется сталь марки А. Цена реализации одного изделия первого вида составляет 600 руб., одного изделия второго вида – 400 руб. Определить, в каком количестве следует произвести изделия каждого вида, чтобы суммарный доход от реализации всех произведенных изделий был максимальным.

Вариант 2. Фабрика выпускает три вида тканей. Суточные ресурсы фабрики следующие: 700 единиц производственного оборудования, 800 единиц сырья и 900 единиц

электроэнергии. Суточный расход каждого вида ресурсов на 1 кв. м производимой ткани представлен в табл. 1.10. Цена реализации одного кв. м ткани вида А равна 180 руб., ткани вида В – 170 руб., ткани вида С – 160 руб. Определить объем суточного производства ткани каждого вида, при котором суммарный доход от реализации всей произведенной ткани будет максимальным.

Таблица 1.10

Ресурсы	Вид ткани		
	А	В	С
Оборудование	2	3	4
Сырье	1	4	5
Электроэнергия	3	4	2

Вариант 3. Компания производит полки для ваннных комнат двух типов. Для каждой полки типа А требуется 1 м² материала, для полки типа В – 2 м² материала. Недельный ресурс материала составляет 100 м². Для изготовления одной полки типа А требуется 20 мин машинного времени, для изготовления одной полки типа В – 30 мин. Можно использовать 100 часов машинного времени в неделю. Цена реализации полок типа А составляет 100 руб., полок типа В – 120 руб. Определить, каким должен быть недельный объем производства полок каждого типа, чтобы суммарный доход от их полной реализации был максимальным.

Вариант 4. Четыре станка обрабатывают два вида деталей: А и В. Известно время обработки детали на каждом станке, время работы станков в течение одного цикла производства и цена реализации одной произведенной детали каждого вида. Эти данные приведены в табл. 1.11. Определить, сколько деталей каждого вида следует произвести, чтобы суммарный доход от реализации произведенной партии деталей был максимальным.

Вариант 5. Фирма производит два вида изделий: А и В. Каждое изделие должно быть обработано на станках I, II, III. Время обработки в часах для каждого из изделий приведено

в табл. 1.12. Время работы станков I, II, III равно соответственно 40, 36 и 36 часов в неделю. Цена реализации одной единицы изделий **A** и **B** составляет соответственно 5 денежных единиц и 3 денежные единицы. Определить, каким должен быть недельный объем производства изделий каждого вида, чтобы суммарный доход от их полной реализации был максимальным.

Таблица 1.11

Станки	Время обработки одной детали (мин)		Резерв времени станка (мин)
	A	B	
I	2	2	16
II	2	2	25
III	1	2	10
IV	2	2	24
Цена реализации детали (руб.)	110	120	

Таблица 1.12

	Время обработки изделия (час)		
	Станок I	Станок II	Станок III
Изделие A	0,5	0,4	0,2
Изделие B	0,25	0,3	0,4

Вариант 6. Завод выпускает изделия двух типов: **A** и **B**. При этом используется сырье четырех видов. Расход сырья каждого вида на изготовление единицы продукции каждого типа и запасы сырья заданы в табл. 1.13. Предполагаемая цена реализации одного изделия типа **A** составляет 3 денежные единицы, цена реализации одного изделия типа **B** – 2 денежные единицы. Определить, в каком количестве следует произвести изделия каждого типа, чтобы суммарный доход от реализации всех произведенных изделий был максимальным.

Таблица 1.13

Тип изделия	Сырье			
	I	II	III	IV
A	2	1	2	2
B	3	1	1	1
Запасы сырья	40	40	60	20

Вариант 7. Процесс изготовления двух видов изделий состоит в последовательной обработке каждого из них на трех станках. Время использования каждого из этих станков для производства данных изделий ограничено 10 часами в сутки. Время обработки и предполагаемая цена реализации одного изделия приведены в табл. 1.14. Определить, сколько следует произвести изделий каждого вида, чтобы суммарный доход от их полной реализации был максимальным.

Таблица 1.14

Изделие	Время обработки 1 изделия (мин)			Цена реализации одного изделия
	Станок 1	Станок 2	Станок 3	
Изделие 1	10	6	6	2 ден. ед.
Изделие 2	5	20	15	3 ден. ед.

Вариант 8. Фирма производит три вида продукции: **A**, **B** и **C**. Для выпуска каждого из видов продукции требуется определенное время обработки на каждом из четырех станков **I**, **II**, **III**, **IV**. Требуемое время, недельный резерв времени для каждого устройства и цена реализации единицы продукции каждого вида приведены в табл. 1.15. Определить, каким должен быть недельный объем производства каждого вида продукции, чтобы суммарный доход от реализации всей произведенной продукции был максимальным.

Вариант 9. Намечается выпуск двух видов костюмов. Для пошива костюма первого вида требуется 1,5 м шерсти, 2 м лавсана и 1 человеко-день трудозатрат, для пошива костюма второго вида — 3,5 м шерсти, 0,5 м лавсана и 1 человеко-день

трудозатрат. Предприятие располагает следующими ресурсами: 350 м шерсти, 240 м лавсана и 150 человеко-дней трудозатрат. Предполагаемая цена реализации одного костюма первого вида составляет 1500 руб., цена реализации одного костюма второго вида – 2000 руб. Определить, каким должен быть объем выпуска костюмов каждого вида, чтобы суммарный доход от их полной реализации был максимальным.

Таблица 1.15

Вид продукции	Время обработки (час)				Цена (ден. ед.)
	Станок I	Станок II	Станок III	Станок IV	
А	1	3	1	2	3
В	6	1	3	3	6
С	3	3	2	4	4
Резерв времени (ч)	84	42	21	42	

Вариант 10. Некоторое предприятие планирует организовать производство четырех видов продукции при потреблении трех видов материалов. Нормы расхода материалов, объемы материальных ресурсов и цена реализации единицы продукции каждого вида приведены в табл. 1.16. Определить, каким должен быть объем выпуска продукции каждого вида, чтобы суммарный доход от реализации всей продукции был максимальным.

Таблица 1.16

Вид продукции	Норма расхода материалов на единицу продукции			Цена реализации единицы продукции
	Материал I	Материал II	Материал III	
А	7	5	2	3
Б	2	8	4	4
В	2	4	1	3
Г	6	3	8	1
Ресурсы материалов	80	430	130	

1.6. Тестовые задания

1. Мысленно представляемая или материально реализованная система, которая, отображая или воспроизводя объект исследования, способна замещать его так, что ее изучение дает новую информацию об этом объекте, называется...

- a) копией объекта;
- b) оригиналом объекта;
- c) моделью объекта;
- d) изображением объекта.

2. Модель отражает...

- a) свойства оригинала, являющиеся существенными с точки зрения целей моделирования;
- b) все свойства оригинала;
- c) физические свойства оригинала;
- d) главные свойства оригинала.

3. Формализованное описание объекта на некотором языке, адекватно отражающее свойства объекта, существенные с точки зрения целей моделирования, представляет модель объекта...

- a) предметную;
- b) информационную;
- c) натурную;
- d) физическую.

4. Компьютерная программа, моделирующая возможные случайные исходы какого-либо процесса или явления, представляет модель...

- a) опытную;
- b) игровую;
- c) учебную;
- d) имитационную.

5. Модель, отражающая только входные и выходные связи моделируемого объекта со средой, – это модель типа...

- a) «белого ящика»;
- b) «черного ящика»;
- c) «серого ящика».

6. Изменяют свое состояние за сколь угодно малое приращение времени модели...

- а) непрерывные;
- б) дискретные;
- с) статические.

7. Математические модели, используемые для планирования возможных действий в условиях неполной информации, называются моделями...

- а) имитационными;
- б) игровыми;
- с) описательными.

8. Метод Монте-Карло используется при построении моделей...

- а) графических;
- б) предметных;
- с) статистических;
- д) вербальных.

9. Метод исследования математической модели, позволяющий получить зависимость результата от исходных данных в виде математической формулы, называется...

- а) аналитическим;
- б) числовым;
- с) дискретным;
- д) физическим.

10. Двумерная компьютерная графика, работающая с изображениями, базовым элементом которых является пиксель, называется графикой...

- а) векторной;
- б) фрактальной;
- с) растровой.

11. Масштабирование изображения без потери качества относится к преимуществам графики...

- а) растровой;
- б) векторной;
- с) фрактальной.

2. БАЗЫ ДАННЫХ

Изучив материалы этого раздела, вы **узнаете**:

- основные понятия теории баз данных;
- основные типы моделей данных;
- основные принципы и приемы проектирования реляционных баз данных;
- типы и назначение объектов базы данных Microsoft Access;
- свойства полей таблицы базы данных Microsoft Access.

Выполнив практические работы, вы **научитесь**:

- создавать, сохранять, открывать файл базы данных;
- создавать таблицы, заполнять их данными, изменять структуру и содержимое таблиц;
- создавать, сохранять, выполнять запросы разных типов;
- создавать формы и отчеты для базовых таблиц и запросов.

Методические рекомендации

Схема работы с учебными материалами данного раздела представлена на рис. 2.1. Теоретическая часть предполагает изучение трех тем. Каждая тема заканчивается перечнем вопросов и упражнений для самоконтроля. Практическая часть содержит три практические работы и индивидуальное домашнее задание. Варианты для выполнения индивидуального домашнего задания представлены в прил. 4. Изучение раздела заканчивается выполнением тестовых заданий, ключ к которым можно найти в прил. 2. С дополнительными учебными материалами по данному разделу можно ознакомиться в рекомендованных литературных источниках [2], [3], [4], [6], [8] и интернет-ресурсах [11], [12], список которых представлен в конце пособия.

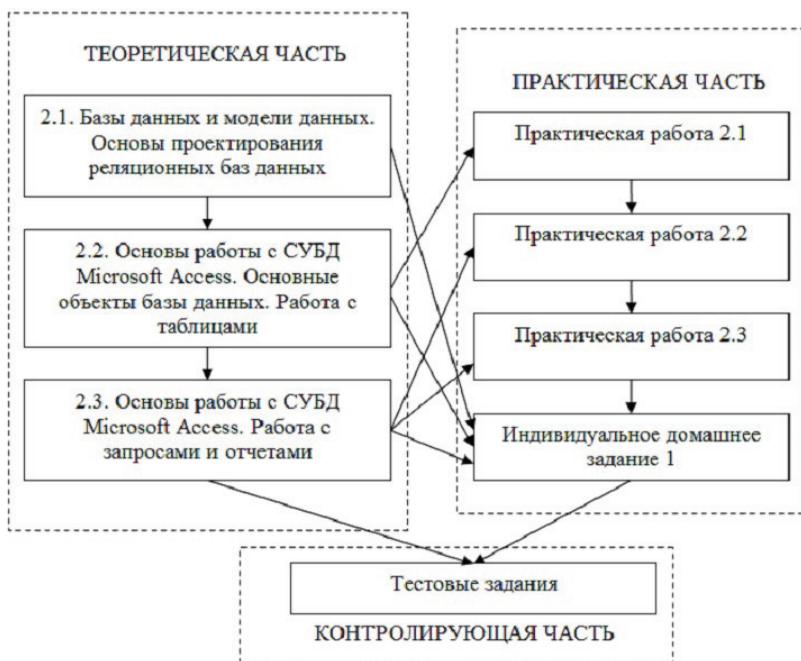


Рис. 2.1. Схема работы с учебными материалами раздела 2

2.1. Базы данных и модели данных. Основы проектирования реляционных баз данных

База данных (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области. Под **предметной областью** будем понимать некоторую часть реального мира, информация о которой представлена в базе данных. **Система управления базами данных (СУБД)** – это комплекс программных средств, предназначенных для создания и сопровождения базы данных.

По технологии обработки данных базы данных подразделяются на **централизованные** и **распределенные**. Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом компьютерной сети, то возможен доступ к такой базе с других компьютеров, подключенных к сети.

Распределенная база данных состоит из нескольких частей, которые хранятся на разных компьютерах, подключенных к сети. Пользователь работает с такой базой данных точно так же, как с централизованной.

По способу доступа к данным различают базы данных с *локальным* и *удаленным (сетевым) доступом*. При локальном доступе база данных доступна только пользователю компьютера, на котором хранится база данных. При удаленном доступе база данных доступна с любого компьютера, находящегося в одной компьютерной сети с данным.

Системы централизованных баз данных с удаленным доступом предполагают различные архитектуры подобных систем. Архитектура *файл-сервер* систем баз данных с сетевым доступом предполагает выделение одного из компьютеров сети в качестве центрального сервера файлов, на котором хранится совместно используемая централизованная база данных. Все другие компьютеры сети выполняют функции *рабочих станций*, с помощью которых поддерживается доступ пользователей к базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность такой информационной системы падает.

В архитектуре *клиент-сервер* сервер базы данных, помимо хранения централизованной базы данных, должен обеспечивать выполнение основного объема обработки данных. Запрос на данные формируется рабочей станцией и передается на сервер. Процесс обработки и выполнения запроса происходит на сервере. Результат выполнения запроса (извлеченные из базы данные, соответствующие запросу) транспортируется по сети от сервера к клиенту (рабочей станции).

Базы данных различаются по используемой для их реализации модели данных. *Модель данных* – это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними. *Информационный объект* – это описание некоторой *сущности* (реального объекта, явления, процесса, события) в виде совокупности логически

связанных *атрибутов* (отличительных свойств, характеристик). Например, сущность **СТУДЕНТ** может быть представлена в базе данных с помощью следующих атрибутов: **Код студента, Фамилия, Имя, Отчество, Год рождения**. Это можно записать так: **СТУДЕНТ (Код студента, Фамилия, Имя, Отчество, Год рождения)**. Указание конкретных значений атрибутов дает нам *экземпляр* информационного объекта. Например, экземпляром информационного объекта **СТУДЕНТ** является набор атрибутов (**21345, Сергеев, Игорь, Андреевич, 1990**).

К числу классических относятся иерархическая, сетевая и реляционная модели данных. В последние годы появились и получили широкое применение постреляционная, многомерная и объектно ориентированная модели.

Иерархическая модель данных – это совокупность элементов данных, расположенных в порядке их подчинения и образующих по структуре перевернутое дерево. Применительно к базам данных элементами данных являются экземпляры информационных объектов, представленных в базе.

Иллюстрация использования иерархической модели для построения базы данных «Институт» представлена на рис. 2.2. Информация базы данных структурирована в виде иерархических деревьев, количество которых равно количеству специальностей, по которым проводится подготовка специалистов в институте. На первом уровне находится информационный объект **СПЕЦИАЛЬНОСТЬ (номер, название)**. На втором уровне – **ГРУППА (шифр, староста)**, на третьем – **СТУДЕНТ (зачетка, ФИО)**. Подчеркиванием выделен атрибут, который однозначно определяет каждый экземпляр информационного объекта.

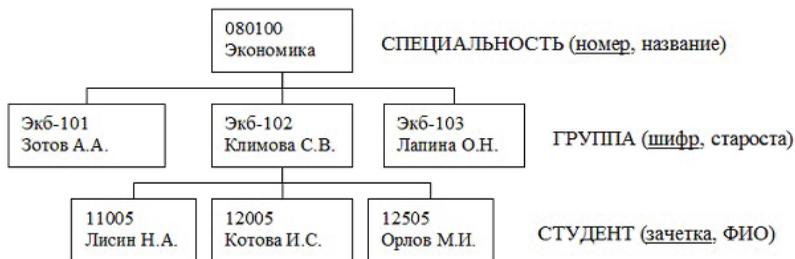


Рис. 2.2. Пример иерархической структуры данных

Сетевая модель данных отличается от иерархической тем, что в ней каждый элемент данных может быть связан с любым другим элементом данных. На рис. 2.3 показана сетевая структура базы данных, содержащей сведения о студентах, занимающихся в спортивных секциях.

СТУДЕНТ (номер зачетной книжки, фамилия, шифр группы)



СЕКЦИЯ (шифр, тренер, вид спорта)

Рис. 2.3. Пример сетевой структуры данных

Реляционная модель данных использует организацию данных в виде двумерных таблиц. Каждая такая таблица, называемая **реляционной таблицей** или **отношением**, обладает следующими свойствами:

- каждый столбец имеет уникальное имя;
- все элементы в одном столбце таблицы имеют одинаковый тип и максимально допустимый размер;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись. **Поле** (столбец реляционной таблицы) соответствует конкретному атрибуту, а **запись** (строка реляционной таблицы) – конкретному экземпляру информационного объекта. Например, в виде реляционной таблицы можно представить информацию о студентах, обучающихся в вузе (рис. 2.4).

Номер личного дела	Фамилия	Имя	Отчество	Пол	Дата рождения	Специальность
16493	Сергеев	Петр	Михайлович	м	01.01.86	080104
16593	Петрова	Анна	Владимировна	ж	15.03.85	080102
16693	Анохин	Андрей	Борисович	м	14.04.86	080104

Рис. 2.4. Пример реляционной таблицы

Основы проектирования реляционных баз данных

Основными объектами реляционной базы данных являются таблицы. Таблицы базы данных создаются таким образом, чтобы каждая из них содержала сведения об одном информационном объекте. Между таблицами должны быть установлены реляционные связи. Установка таких связей делает возможным выполнение одновременной обработки данных из нескольких таблиц.

Первичный ключ реляционной таблицы – это поле или группа полей, которые позволяют однозначно определить каждую запись (строку) в таблице. Первичный ключ должен обладать двумя свойствами:

- 1) **однозначная идентификация записи** – запись должна однозначно определяться значением ключа;
- 2) **отсутствие избыточности** – никакое поле нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации записи.

Если первичный ключ состоит из одного поля, то он называется **простым ключом** или **ключевым полем**. Таблица может иметь **составной ключ**, состоящий из нескольких полей. Можно связать две реляционные таблицы, если ключ одной связываемой таблицы ввести в состав ключа другой таблицы (возможно совпадение ключей). Можно ключевое поле одной связываемой таблицы ввести в структуру другой таблицы так, что оно в этой таблице не будет ключевым. В этом случае это поле называется **внешним ключом**.

Пример 2.1 [3]. Рассмотрим базу данных «Деканат», состоящую из трех таблиц: **СТУДЕНТ**, **СЕССИЯ**, **СТИПЕНДИЯ** (рис. 2.5). В таблицах **СТУДЕНТ** и **СЕССИЯ** ключевым

является поле **Номер личного дела**. В столбце **Результат** таблицы **СЕССИЯ** содержатся числа 0, 1 или 2 (0 – студент не получает стипендию, 1 – обычная стипендия, 2 – повышенная стипендия). Таблица **СТИПЕНДИЯ** содержит информацию о проценте начисляемой студенту стипендии в зависимости от результата сдачи сессии. В этой таблице ключевым является поле **Результат**.

Таблица СТУДЕНТ

Номер личного дела	Фамилия	Имя	Отчество	Группа
120603	Свиридова	Анна	Алексеевна	ЭТ-101
120608	Коробов	Игорь	Андреевич	ЭТ-102
...

Таблица СЕССИЯ

Номер личного дела	Экзамен 1	Экзамен 2	Экзамен 3	Результат
120603	3	4	4	0
120608	5	5	5	2
120634	4	5	4	1
...

Таблица СТИПЕНДИЯ

Результат	Процент
0	0
1	100
2	200

Рис. 2.5. Реляционные таблицы базы данных «Деканат»

Таблицы **СТУДЕНТ** и **СЕССИЯ** имеют совпадающие первичные ключи, что позволяет организовать связь между ними по полю **Номер личного дела**. Таблица **СЕССИЯ** имеет первичный ключ **Номер личного дела** и содержит внешний ключ **Результат**, который позволяет обеспечить ее связь с таблицей **СТИПЕНДИЯ**. Для наглядности представления связей между таблицами используется представление таблиц в виде структур этих таблиц (рис. 2.6).

В зависимости от того, какие ключи (первичные или внешние) используются для установки связи между таблицами, различают три типа связей между таблицами реляционной базы данных: «один-к-одному» (1–1), «один-ко-многим» (1–М), «многие-ко-многим» (М–М)

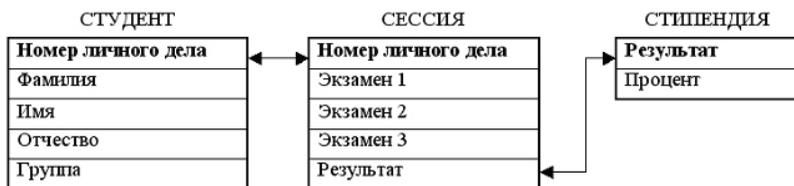


Рис. 2.6. Структура таблиц базы данных «Деканат»

Будем считать, что *запись таблицы А связана с записью таблицы В*, если в обеих таблицах эти записи содержат одно и то же значение в поле, по которому установлена связь между таблицами.

Между таблицами **А** и **В** установлена *связь «один-к-одному»*, если каждая запись в таблице **А** может иметь не более одной связанной с ней записи в таблице **В** и наоборот, каждая запись в таблице **В** может иметь не более одной связанной с ней записи в таблице **А**. В этом случае для связи используются первичные ключи связываемых таблиц. Рассмотренные ранее таблицы **СТУДЕНТ** и **СЕССИЯ** связаны по типу «один-к-одному».

Между таблицами **А** и **В** установлена *связь «один-ко-многим»*, если каждая запись в таблице **А** может быть связана с несколькими записями таблицы **В**, но каждая запись в таблице **В** не может быть связана более чем с одной записью таблицы **А**. Таблица **А** в этом случае называется *главной* таблицей, а таблица **В** — *подчиненной*. В этом случае для связи используется поле, которое является первичным ключом таблицы, находящейся на стороне отношения «один» (таблица **А**), и внешним ключом таблицы, находящейся на стороне отношения «многие» (таблица **В**).

Например, рассмотренные ранее таблицы **СТИПЕНДИЯ** и **СЕССИЯ** связаны по типу «один-ко-многим». При этом на стороне «один» находится таблица **СТИПЕНДИЯ**, а на стороне «многие» — таблица **СЕССИЯ**. Связь устанавливается по полю **Результат**.

Между таблицами **А** и **В** установлена *связь «многие-ко-многим»*, если каждой записи таблицы **А** может соответствовать несколько записей в таблице **В**, и наоборот, каждой

записи таблицы **В** может соответствовать несколько записей в таблице **А**. Такая связь всегда реализуется с помощью третьей связующей таблицы (назовем ее – таблица **С**). Связь «многие-ко-многим» представляет комбинацию двух связей типа «один-ко-многим»: между таблицами **А** и **С** (**А** – главная, **С** – подчиненная) и между таблицами **В** и **С** (**В** – главная, **С** – подчиненная).

Примером могут служить таблицы **ЧИТАТЕЛИ** и **КНИГИ** базы данных «Библиотека» (рис. 2.7). Связь между ними организуется с помощью таблицы **АБОНЕМЕНТ**. При этом между таблицами **ЧИТАТЕЛИ** и **АБОНЕМЕНТ** установлена связь «один-ко-многим», в которой таблица **ЧИТАТЕЛИ** является главной. Аналогично между таблицами **КНИГИ** и **АБОНЕМЕНТ** установлена связь «один-ко-многим», в которой таблица **КНИГИ** является главной.



Рис. 2.7. Пример таблиц, находящихся в отношении «многие-ко-многим»

В теории реляционных баз данных вводятся требования к организации данных в таблицах, формулируемые с помощью *нормальных форм*. Нормальные формы нумеруются последовательно, по мере ужесточения требований.

Первая нормальная форма (1НФ) предписывает, чтобы все данные, содержащиеся в таблице, были атомарными (неделимыми). Это требование означает, что в каждой ячейке таблицы должна находиться только одна величина, но не массив и не какая-либо другая структура данных. Пример таблицы, которая не находится в 1НФ, содержащей данные о сдаче сессии студентами, показан на рис. 2.8.

Номер зачетки	ФИО	Оценки
12005	Зотов Иван Сергеевич	3, 5
12012	Ильина Инна Андреевна	4, 5
...

Рис. 2.8. Пример таблицы, не удовлетворяющей требованиям 1НФ

В поле **Оценки** содержатся массивы данных, что противоречит требованиям 1НФ. С целью приведения таблицы к 1НФ выделим каждую оценку в отдельное хранимое значение (рис. 2.9). Данные в поле **ФИО** могли бы рассматриваться как атомарные в том случае, если в дальнейшем предполагается оперировать этими данными как с неделимыми значениями. Если же потребуется выделять и анализировать отдельные составляющие (фамилия, имя, отчество), то целесообразно выделить их в отдельные поля: **Фамилия**, **Имя**, **Отчество**. Теперь таблица находится в 1НФ.

Номер зачетки	Фамилия	Имя	Отчество	Оценка
12005	Зотов	Иван	Сергеевич	3
12005	Зотов	Иван	Сергеевич	5
12012	Ильина	Инна	Андреевна	4
12012	Ильина	Инна	Андреевна	5
...		...		

Рис. 2.9. Приведение таблицы к 1НФ

Вторая нормальная форма (2НФ). Говорят, что таблица находится во второй нормальной форме, если она находится в 1НФ и каждый неключевой атрибут полностью зависит от первичного ключа. Зависимость от первичного ключа понимается как зависимость от ключа целиком, а не от отдельной его составляющей (в случае составного ключа). В частности, если таблица находится в 1НФ и имеет простой первичный ключ, то она также находится и в 2НФ.

Приведем пример таблицы, не находящейся в 2НФ, которая содержит данные измерения температуры воздуха

(рис. 2.10). В таблице определен первичный ключ, состоящий из двух полей: **Дата+Время суток**. Задав значения даты и времени суток, мы тем самым определим ровно одну строку таблицы. При этом температура воздуха зависит как от даты, так и от времени суток, т. е. поле **Температура воздуха** зависит от первичного ключа целиком. Поле **Долгота дня** определяется только датой и не зависит от времени суток. Это является нарушением требований 2НФ.

Дата	Долгота дня	Время суток	Температура воздуха
26.02.12	10.36	12.00	-2
26.02.12	10.36	18.00	-6
27.02.12	10.38	12.00	-8
...

Рис. 2.10. Пример таблицы, не удовлетворяющей требованиям 2НФ

Несоблюдение требований 2НФ приводит, с одной стороны, к хранению в таблице избыточных данных (долгота дня повторяется для каждого значения времени суток в пределах одной даты), с другой стороны, к возможности ввода в таблицу противоречивых данных (можно ввести ошибочно разные значения долготы дня для одной и той же даты). Избежать возможных осложнений позволит выделение поля **Долгота дня** вместе с полем **Дата** в отдельную таблицу с установлением связи между таблицами по полю **Дата** (рис. 2.11).

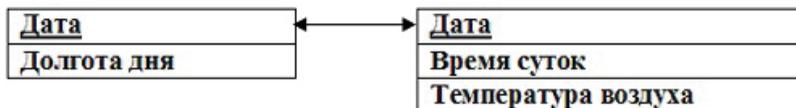


Рис. 2.11. Приведение таблицы к 2НФ

Третья нормальная форма (3НФ). Говорят, что таблица находится в 3НФ, если она соответствует 2НФ и все неключевые атрибуты (столбцы) взаимно независимы. Взаимную зависимость столбцов будем понимать следующим образом:

столбцы являются взаимно зависимыми, если нельзя изменить один из них, не изменяя другой.

Приведем пример таблицы, которая не находится в 3НФ. Рассмотрим таблицу, содержащую данные о студентах (рис. 2.12). Эта таблица находится в 2НФ (находится в 1НФ и имеет простой первичный ключ **Номер зачетки**). Однако значения в поле **Возраст** зависят от значений в поле **Дата рождения**, что противоречит требованиям 3НФ. Для приведения таблицы к 3НФ достаточно исключить из нее поле **Возраст**. Возраст студента в любой момент может быть вычислен как разница между текущей датой и датой рождения.

Номер зачетки	Фамилия	Имя	Отчество	Дата рождения	Возраст
12005	Зотов	Иван	Сергеевич	02.11.1991	20
12015	Орлова	Ольга	Ивановна	12.10.1992	19
...

Рис. 2.12. Пример таблицы, не удовлетворяющей требованиям 3НФ

Нормализация – это процесс приведения таблиц реляционной базы данных к выбранной нормальной форме. Нормализация до 2НФ обычно сводится к декомпозиции – разбиению одной таблицы на несколько других. Нормализация до 3НФ может быть выполнена удалением из таблицы зависимых (вычисляемых) столбцов, либо декомпозицией, как в случае с 2НФ. В теории реляционных баз данных рассматриваются формы порядков выше 3НФ: нормальная форма Бойса-Кодда, 4НФ, 5НФ и др. Однако эти формы не имеют большого практического значения.

Постреляционная модель данных представляет расширенную реляционную модель, снимающую ограничение неделимости данных, допускающую поля, значения которых состоят из подзначений. В постреляционной модели набор значений таких полей считается самостоятельной таблицей, встроенной в основную таблицу. Поскольку постреляционная модель допускает хранение в таблицах ненормализованных данных, возникает проблема обеспечения целостности и непротиворечивости данных. Эта проблема решается включением в СУБД встроенных программных механизмов для решения этой проблемы.

Основными понятиями *многомерной модели данных* являются гиперкуб, измерение и ячейка. *Гиперкуб* – это логическая структура, состоящая из ячеек, адресация которых выполняется с помощью измерений. *Измерение* – множество однотипных данных, образующих одну из граней гиперкуба. *Ячейка* или *показатель* – это поле, значение которого однозначно определяется фиксированным набором измерений. На рис. 2.13,б показан пример двухмерного гиперкуба, в котором измерениями являются **Модель** и **Месяц**, а показателем – **Объем продаж**. Соответствующая гиперкубу реляционная модель показана на рис. 2.13,а.

Модель	Месяц	Объем продаж
Acer	июнь	12
Acer	июль	24
Acer	август	5
Samsung	июнь	2
Samsung	июль	18
Toshiba	июль	19

а)

	Июнь	Июль	Август
Acer	12	24	5
Samsung	2	18	0
Toshiba	0	19	0

б)

Рис. 2.13. Реляционное (а) и многомерное (б) представление данных

В *объектно-ориентированной модели* при представлении данных между записями базы данных и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектно ориентированных языках программирования.

Этапы проектирования реляционной базы данных

Создание базы данных можно рассматривать как процесс компьютерного моделирования заданной предметной области. В процессе моделирования выделяется несколько этапов, для которых получаются разные по природе и назначению модели (рис. 2.14).

На первом этапе выполняется исследование предметной области, которое заканчивается построением *инфологической (концептуальной) модели* предметной области, отражающей

существенные для целей моделирования объекты, их характеристики и связи между объектами. Пример инфологической модели показан на рис. 2.15.



Рис. 2.14. Последовательность этапов моделирования при создании базы данных



Рис. 2.15. Пример инфологической (концептуальной) модели предметной области

По инфологической модели строится *даталогическая модель*, учитывающая реальные возможности организации данных средствами выбранной СУБД. Даталогическая модель реляционной базы данных, называемая *схемой данных*, содержит структуру таблиц с указанием типов данных и связей между таблицами. Пример схемы данных показан на рис. 2.6.

На *физическом уровне* выполняется компьютерная реализация базы данных, которая предполагает выполнение типовых операций: создание файла базы данных, базовых таблиц,

экранных форм; заполнение таблиц данными, работа с созданной базой данных.

Язык SQL (Structured Query Language – язык структурированных запросов) является информационно-логическим языком для описания, изменения и извлечения данных, хранимых в реляционных базах данных. SQL – механизм связи между прикладным программным обеспечением и базой данных. Изначально SQL был единственным способом работы пользователя с базой данных. Современные СУБД предоставляют пользователю развитые средства визуального интерактивного взаимодействия с базой данных.

Каждое предложение SQL – это либо запрос данных из базы, либо обращение к базе данных, которое приводит к изменению данных в ней. Язык SQL содержит некоторую совокупность операторов, инструкций и вычисляемых функций.

Операторы SQL делятся на следующие типы:

- операторы определения данных (CREATE – создает объект базы данных, ALTER – изменяет объект, DROP – удаляет объект);
- операторы манипулирования данными (SELECT – считывает данные, удовлетворяющие заданным условиям, INSERT – добавляет данные и др.);
- операторы определения доступа к данным;
- операторы управления транзакциями (под *транзакцией* понимается неделимая последовательность операций, переводящая базу данных из одного состояния в другое таким образом, что, если по каким-либо причинам одно из действий транзакции невыполнимо, база данных возвращается в исходное состояние, которое было до начала транзакции).

Пример записи инструкции для выбора данных из таблицы **СТУДЕНТ** базы данных «Деканат» (рис. 2.5):

```
SELECT *  
FROM СТУДЕНТ  
WHERE Группа LIKE “ЭТ-101”;
```

Данная инструкция предполагает формирование запроса на выборку из таблицы **СТУДЕНТ** с включением в результирующую таблицу записей, для которых значение в поле

Группа равно **ЭТ-101**. Символ * после слова SELECT означает, что результирующая таблица запроса должна содержать все поля таблицы **СТУДЕНТ**.

? Вопросы и упражнения для самоконтроля

1. Что такое база данных, система управления базами данных, предметная область?
2. Укажите различия между централизованными и распределенными базами данных, между локальными и базами данных с сетевым доступом.
3. Назовите и охарактеризуйте основные архитектуры централизованных баз данных с сетевым доступом.
4. Что такое информационный объект?
5. Что такое модель данных? Назовите основные модели данных, используемые при реализации баз данных.
6. Укажите характеристики и объекты обработки реляционной модели данных.
7. Какими свойствами обладает первичный ключ реляционной таблицы?
8. Что такое внешний ключ реляционной таблицы?
9. Какого типа связи (отношения) могут быть установлены между таблицами реляционной базы данных? Охарактеризуйте каждый тип связи.
10. В чем смысл процедуры нормализации таблиц реляционной базы данных?
11. Опишите требования, которым должна удовлетворять таблица, приведенная к первой нормальной форме, второй нормальной форме, третьей нормальной форме. Приведите примеры.
12. Что такое «схема данных» реляционной базы данных?
13. Перечислите и охарактеризуйте основные модели, получаемые в процессе разработки реляционной базы данных.
14. Укажите назначение и особенности языка SQL.

2.2. Основы работы с СУБД Microsoft Access. Основные объекты базы данных. Работа с таблицами

Объекты базы данных Microsoft Access

База данных в Microsoft Access (MS Access) — это один файл с расширением **.mdb**, содержащий таблицы, запросы и другие **объекты** базы данных. Рассмотрим эти объекты более подробно.

Таблицы — основные объекты любой базы данных, предназначенные для хранения данных.

Запросы — это специальные средства для отбора и анализа данных.

Формы используются для ввода новых данных в таблицы базы данных, а также для просмотра имеющихся данных.

Отчеты предназначены для вывода данных на экран или на принтер. В них предусмотрены средства для вывода специальных элементов оформления, характерных для печатных документов: верхний и нижний колонтитулы, номера страниц и т. п.

Страницы доступа к данным служат для обеспечения удаленного доступа к данным, содержащимся в базе данных. Их можно встроить в любой web-документ. Страницы доступа выполняют связующую функцию между базой данных и web-страницами Интернета.

Макросы предназначены для автоматизации выполнения повторяющихся операций при работе с СУБД. Несколько команд управления базой данных могут быть сохранены в виде макроса, чтобы в дальнейшем можно было воспроизводить эти команды, запуская сохраненный макрос.

Модули создаются средствами языка программирования Visual Basic for Applications. С их помощью разработчик базы данных может заложить в нее нестандартные функциональные возможности.

Работа с таблицами

Структура таблицы определяется набором и свойствами полей (столбцов таблицы), а информация хранится в виде записей (строк таблицы). Далее перечислены некоторые свойства полей:

- **имя поля** (должно быть уникальным в рамках таблицы) — определяет, как следует обращаться к данным этого поля при операциях с базой данных;
- **тип поля** — определяет тип данных, которые могут содержаться в данном поле;
- **размер поля** — определяет предельные значения данных, которые могут размещаться в данном поле;
- **формат поля** — определяет способ форматирования данных в ячейках таблицы, принадлежащих полю (например, значение даты **19 января 2004 года** может быть представлено в разных форматах: **19 января 2004 г., 19.06.2004, 19-январ-2004**);
- **условие на значение** — ограничение, используемое для проверки правильности ввода данных.

Свойство **тип поля** может принимать одно из следующих значений:

- **текстовый** — используется для хранения текста ограниченного размера (до 255 символов);
- **поле Мемо** — специальный тип для хранения больших объемов текста (до 65 535 символов);
- **числовой** — для хранения чисел;
- **дата/время** — для хранения календарных дат и текущего времени;
- **денежный** — для хранения денежных сумм;
- **счетчик** — содержит натуральные числа 1, 2, 3, ... (при этом значения *вводятся в поле автоматически* при создании новой записи);
- **логический** — для хранения логических данных (**Истина** или **Ложь**);
- **поле объекта OLE** — для хранения графических, аудио- и видеоданных, фрагментов электронных таблиц и др.;
- **гиперссылка** — специальное поле для хранения адресов web-страниц Интернета.

Создание таблиц

Создание таблицы выполняется после выбора объекта **Таблицы** в окне базы данных. Чтобы создать таблицу, следует использовать значок **Создание таблицы в режиме**

конструктора или выбрать пункт **Конструктор** в управляющем меню окна.

Окно **Конструктора таблиц** представлено на рис. 2.16. В столбце **Имя поля** вводятся имена полей будущей таблицы. Тип каждого поля выбирается из раскрывающегося списка. Нижняя часть окна содержит список свойств поля, выделенного в верхней части. Закончив создание структуры таблицы, нужно закрыть окно **Конструктора**, сохранив созданную таблицу.

Заполнение таблицы данными производится с клавиатуры в режиме просмотра данных таблицы. Сохранение вводимых в таблицу данных выполняется автоматически в режиме реального времени.

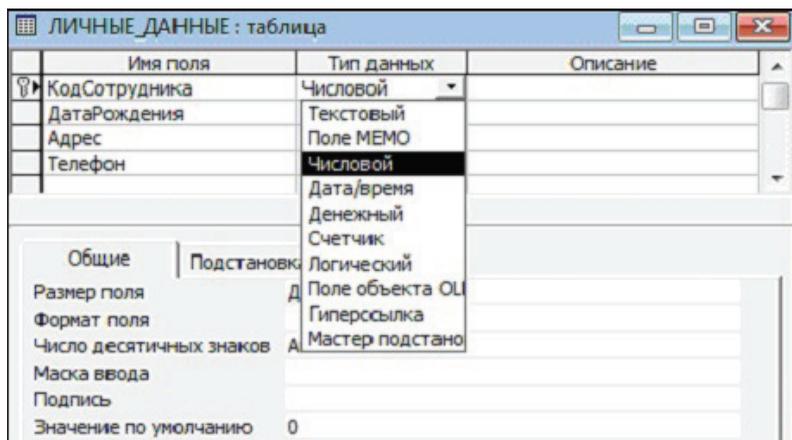


Рис. 2.16. Таблица в режиме **Конструктор**

Создание межтабличных связей

Эта операция выполняется в окне **Схема данных** (рис. 2.17), которое открывается кнопкой  в окне базы данных или командой **Сервис – Схема данных**. Связь между двумя таблицами устанавливается перетаскиванием мышью поля одной таблицы на соответствующее поле другой таблицы. Образовавшаяся связь отображается в виде линии, соединяющей два поля разных таблиц.

Настройка связи с целью защиты целостности данных выполняется в окне **Изменение связей**, которое открывается в окне **Схема данных** выбором команды **Изменить связь** из контекстного меню связи.

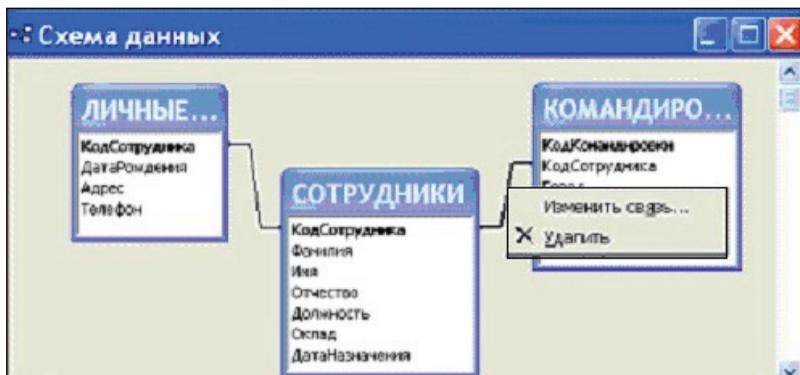


Рис. 2.17. Связанные таблицы в окне **Схема данных**

? Вопросы и упражнения для самоконтроля

1. Какие объекты базы данных MS Access используются для хранения данных?
2. Какие объекты базы данных MS Access используются для отбора данных из базовых таблиц в соответствии с какими-либо критериями отбора?
3. Каково назначение экранных форм и отчетов базы данных MS Access?
4. Каково назначение страниц доступа к данным?
5. Для чего используются макросы и модули базы данных MS Access?
6. Укажите основные свойства полей таблицы базы данных MS Access.
7. Какого типа данные может содержать таблица базы данных MS Access? Охарактеризуйте основные типы данных.
8. В каком режиме работы с таблицей может быть изменена структура таблицы?
9. Для чего используется окно **Схема данных** при работе с таблицами базы данных MS Access?

2.3. Основы работы с СУБД Microsoft Access. Работа с запросами, формами и отчетами

Запросы на выборку

Для получения данных из таблиц базы данных пользователь должен использовать запросы. Запрос извлекает данные из базовых таблиц и создает на их основе временную результирующую таблицу. При сохранении запроса сохраняется не результирующая таблица, а только алгоритм ее получения. Если запрос подготовлен и сохранен в базе данных, то для его выполнения следует открыть панель **Запросы** в окне базы данных, выбрать запрос и открыть его двойным щелчком на значке запроса.

Значок **Создание запроса в режиме конструктора** в окне базы данных открывает окно, называемое **бланком запроса по образцу** (рис. 2.18). В верхней части бланка отображается структура таблиц, к которым применяется запрос, а нижняя часть разбита на столбцы, соответствующие полям будущей результирующей таблицы запроса. Для включения поля базовой таблицы в результирующую таблицу запроса следует дважды щелкнуть соответствующее поле в верхней части бланка.

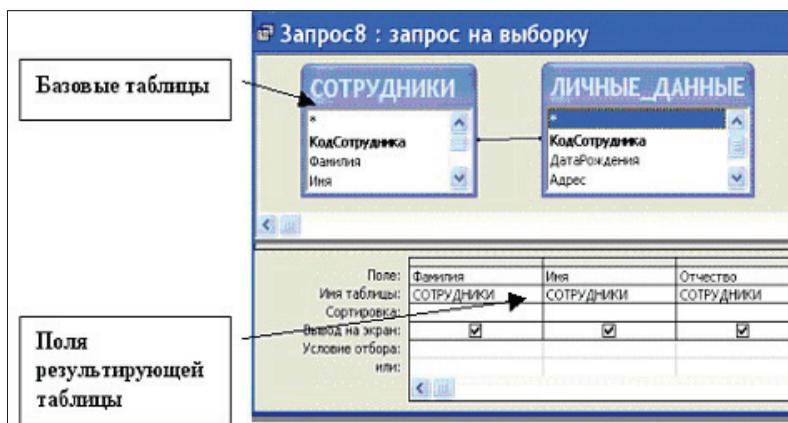


Рис. 2.18. Запрос в режиме Конструктор (фрагмент окна)

Упорядочение записей и управление отображением данных в результирующей таблице запроса

В нижней части бланка конструктора запроса имеется специальная строка **Сортировка**. При щелчке на этой строке в любом столбце таблицы появляется раскрывающийся список, в котором можно выбрать метод сортировки: **по возрастанию** или **по убыванию**. На рис. 2.19 задана сортировка по полю **Фамилия**.

Поле:	КодСотрудника	Фамилия	Город
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	КОМАНДИРОВКИ
Сортировка:		по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора: или:			"Самара"

Рис. 2.19. Задание условий сортировки записей и отображения данных в запросе

Возможна ситуация, когда какое-то поле необходимо включить в запрос, но при этом запретить вывод на экран его содержимого. В таких случаях нужно сбросить флажок **Вывод на экран** в соответствующем поле. На рис. 2.19 поле **Город** в результирующей таблице не будет показываться, хотя оно содержится в этой таблице. Необходимость включения этого поля в запрос обусловлена тем, что по нему задано условие отбора данных.

Простые запросы на выборку используются для отбора записей из одной или нескольких взаимосвязанных таблиц в соответствии с заданными *критериями отбора*. Критерии отбора записей записываются в строке **Условие отбора** бланка запроса по образцу и расположенных под ней строках. В табл. 2.1 приведены примеры записи условий отбора.

Таблица 2.1

Примеры записи условий отбора записей в запросе

Критерий отбора записей	Запись условия отбора
Значение в текстовом поле начинается с символа “А”, остальные символы могут быть любыми	A*
Значение в числовом поле ≥ 60	≥ 60
Значение в текстовом поле НЕ равно “Инженер”	NOT “Инженер”
Значения в поле типа дата в диапазоне от 1.12.96 до 15.12.96	BETWEEN 1.12.96 AND 15.12.96
Значение в числовом поле в диапазоне от 10 до 100	BETWEEN 10 AND 100
Значение в текстовом поле равно “инженер” или “бухгалтер”	“инженер” OR “бухгалтер”

Условия отбора для нескольких полей могут быть записаны в одной строке или в разных строках. В первом случае при выполнении запроса будут выбираться те записи из таблиц, для которых выполнены все условия отбора. Если условия отбора находятся на разных строках бланка, то будут отбираться те записи, для которых выполняется хотя бы одно из условий.

Например, в случае записи условий отбора так, как показано на рис. 2.20, будут отбираться те записи из таблицы **СОТРУДНИКИ**, для которых значение в поле **КодСотрудника** больше 4 и при этом значение в поле **Оклад** меньше 3000.

Поле:	КодСотрудника	Фамилия	Оклад
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ
Сортировка:		по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	>4		<3000
или:			

Рис. 2.20. Пример записи условий отбора в запросе

В случае записи условий отбора так, как показано на рис. 2.21, будут отбираться записи из таблицы **СОТРУДНИКИ**,

для которых значение в поле **КодСотрудника** больше 4, или записи, для которых значение в поле **Оклад** меньше 3000, а также записи, для которых справедливы оба условия отбора.

Поле:	КодСотрудника	Фамилия	Оклад
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ
Сортировка:		по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	>4		
или:			<3000

Рис. 2.21. Пример записи условий отбора в запросе

Вычисления в запросах на выборку. Результирующие таблицы запросов на выборку могут включать не только поля базовых таблиц, но и новые поля, значения в которых получаются как результат вычислений. Такие поля называются *вычисляемыми полями*. Для создания вычисляемого поля вместо имени записывается формула в следующем формате: **Имя поля: выражение.**

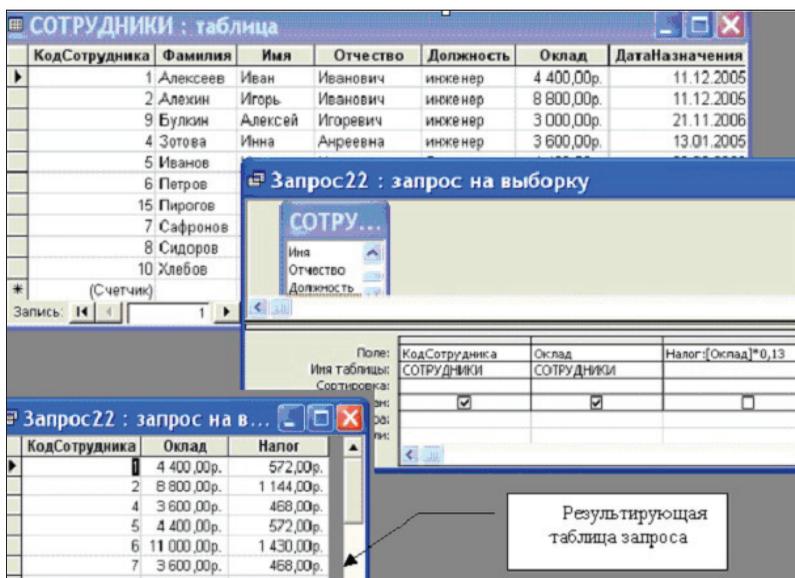


Рис. 2.22. Запрос с вычисляемым полем

На рис. 2.22 показан пример запроса с вычисляемым полем **Налог**. В базовой таблице присутствует поле **Оклад**. При конструировании запроса в заголовке поля вместо имени записывается формула **Налог: [Оклад]*0,13**. При выполнении запроса в результирующей таблице появится поле с именем **Налог**, значения в котором будут получены как результат умножения соответствующих значений из поля **Оклад** на число **0,13**.

Запросы с параметром позволяют пользователю ввести критерий отбора данных при выполнении запроса. При конструировании запроса с параметром в строке **Условие отбора** в квадратных скобках указывается текст, который будет выведен в диалоговом окне при выполнении этого запроса (рис. 2.23). Пользователь должен будет ввести с клавиатуры значение запрашиваемого параметра. Это значение будет использовано для отбора записей из базовых таблиц и включения их в результирующую таблицу запроса.

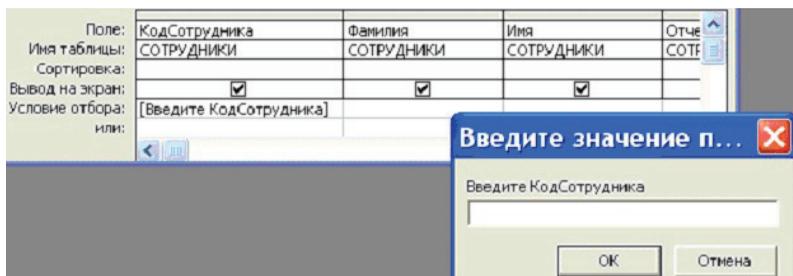


Рис. 2.23. Запрос с параметром-значением

Если в качестве параметров отбора нужно указать верхнюю и нижнюю границы какого-либо диапазона, то следует ввести условие отбора в виде:

BETWEEN [Введите нижнюю границу] **AND** [Введите верхнюю границу].

В этом случае при выполнении запроса будут поочередно запрашиваться значения нижней и верхней границ параметра-диапазона, которые будут использованы для отбора записей из таблиц.

Итоговые запросы и запросы на изменение данных

Итоговый запрос — это запрос, в котором выводятся результаты статистических расчетов по какой-либо группе записей из одной или нескольких таблиц. Можно находить сумму (функция **Sum**), среднее значение (функция **Avg**), наибольшее значение (функция **Max**) или наименьшее значение (функция **Min**), количество значений в группе (функция **Count**).

При создании запроса нужно щелкнуть по кнопке **Групповые операции**  на панели инструментов. В бланке запроса по образцу появляется дополнительная строка **Групповая операция**. В полях, по которым проводится группировка, следует установить функцию **Группировка**. В полях, где выполняются итоговые операции, нужно в строке **Групповая операция** раскрыть список и выбрать одну из функций (**Sum**, **Avg**, **Max**, **Min** и т. д.)

Запрос на изменение данных — это запрос, который за одну операцию изменяет сразу несколько записей таблицы. Существует четыре типа запросов на изменение данных: на удаление записей, на обновление записей, на добавление записей, на создание таблицы.

Запрос на удаление записей удаляет группу записей из одной или нескольких связанных таблиц. **Запрос на обновление записей** вносит общие изменения в группу записей одной или нескольких базовых таблиц базы данных. **Запрос на создание таблицы** создает новую таблицу на основе данных из одной или нескольких существующих таблиц.

Формы и отчеты

Формы позволяют пользователям вводить данные в таблицы базы данных, просматривать и редактировать данные без непосредственного доступа к самим таблицам. **Автоформа** всегда создается на базе только одной таблицы и содержит все поля этой таблицы. Существует три вида автоформ: **в столбец**, **ленточные** и **табличные** (рис. 2.24). С помощью **Мастера форм** можно создать форму на базе одной или нескольких связанных таблиц.

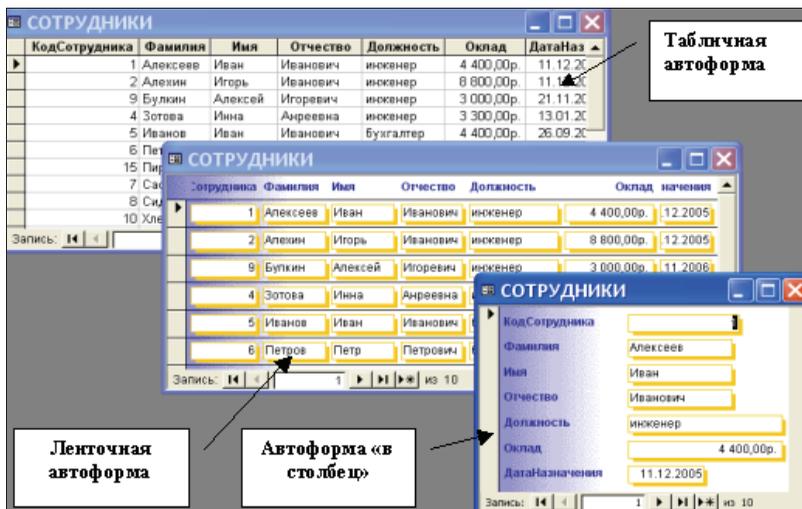


Рис. 2.24. Три вида автоформ

Отчеты служат для вывода данных на печатающие устройства и должны учитывать параметры принтера и параметры используемой бумаги. Средства автоматического проектирования отчетов реализованы автоотчетами. В режиме **Автоотчет** можно создать только отчет на базе одной таблицы или запроса, в котором *выводятся все поля и все записи из базовой таблицы или запроса*. Средством автоматизированного создания отчетов является **Мастер отчетов**, позволяющий включить в отчет поля из нескольких связанных таблиц.

? Вопросы и упражнения для самоконтроля

1. Какого типа запросы используются для отбора записей из одной или нескольких взаимосвязанных таблиц базы данных MS Access?
2. Приведите примеры записи условий отбора в запросах на выборку.
3. В чем заключается отличие запросов с параметром от простых запросов на выборку?
4. Укажите назначение вычисляемых полей запроса. Как включить в запрос вычисляемое поле?

5. Каково назначение итоговых запросов базы данных MS Access?
6. Какие функции могут быть использованы для обработки сгруппированных данных в итоговых запросах?
7. Укажите типы запросов на изменение записей. Поясните назначение каждого из типов.
8. Какой режим следует использовать для создания отчета, содержащего данные из нескольких связанных таблиц?

2.4. Практические работы

Практическая работа 2.1. Работа с таблицами и формами базы данных Microsoft Access

Цель работы – научиться:

- создавать таблицы базы данных, заполнять их информацией, устанавливать связи между таблицами;
- создавать формы и использовать их для работы с таблицами.

Порядок выполнения работы

1. Изучить теоретический материал из раздела 2.2.
2. Выполнить практическое задание.

Практическое задание

Задана схема данных базы данных «Отдел кадров» (рис. 2.25), содержащая информацию о сотрудниках некоторой фирмы. По заданной схеме данных требуется создать компьютерную реализацию базы данных, выполнив следующие этапы работы: создать базовые таблицы, установить связи между таблицами, заполнить таблицы данными, создать формы для просмотра и ввода данных в таблицы.



Рис. 2.25. Схема данных для базы данных «Отдел кадров»

Технология выполнения задания

1. Запустите программу Microsoft Access (**Пуск – Программы – Microsoft Access**). Выполните в меню **Файл – Создать** и затем выберите в области задач окна программы пункт **Новая база данных**.

2. В открывшемся окне сохранения файла базы данных выберите папку для сохранения, введите имя файла **Кадры**, убедитесь в том, что установлен тип файла **Базы данных Microsoft Access (*.mdb)**. Щелкните кнопку **Создать**. Откроется окно **База данных**.

3. В окне **База данных** на панели **Таблицы** дважды щелкните значок **Создание таблицы в режиме конструктора** – откроется бланк создания структуры таблицы (рис. 2.26).

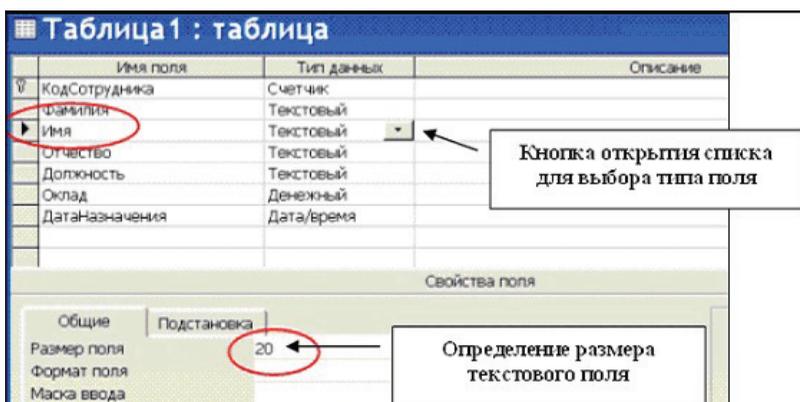


Рис. 2.26. Пример создания структуры таблицы в режиме **Конструктор**

4. Заполните бланк таблицы, введя с клавиатуры имена полей таблицы **СОТРУДНИКИ**, выбрав для каждого поля тип данных и приняв предлагаемые программой или введя с клавиатуры (для полей текстового типа) размеры полей (рис. 2.27).

5. Определите поле **КодСотрудника** как ключевое поле таблицы: выделите это поле, щелчком правой кнопки мыши откройте контекстное меню и выберите пункт **Ключевое поле**.

Имя поля	Тип поля	Размер поля
КодСотрудника	Счётчик	Длинное целое
Фамилия	Текстовый	20
Имя	Текстовый	20
Отчество	Текстовый	20
Должность	Текстовый	30
Оклад	Денежный	
ДатаНазначения	Дата/время	

Рис. 2.27. Структура таблицы **СОТРУДНИКИ**

6. Закройте окно создания структуры таблицы в режиме **Конструктор**. При закрытии окна присвойте таблице имя **СОТРУДНИКИ**.

7. Аналогично пунктам 3–6 создайте таблицу **ЛИЧНЫЕ_ДАННЫЕ**, в которую включите поля, указанные на рис. 2.28. Определите поле **КодСотрудника** как ключевое поле таблицы.

Имя поля	Тип поля	Размер поля
КодСотрудника	Числовой	Длинное целое
ДатаРождения	Дата/время	
Адрес	Текстовый	50
Телефон	Текстовый	20

Рис. 2.28. Структура таблицы **ЛИЧНЫЕ_ДАННЫЕ**

8. Создайте таблицу **КОМАНДИРОВКИ**, структура которой показана на рис. 2.29. Определите поле **КодКомандировки** как ключевое поле таблицы.

Имя поля	Тип поля	Размер поля
КодКомандировки	Счётчик	Длинное целое
КодСотрудника	Числовой	Длинное целое
Город	Текстовый	30
ДатаОтъезда	Дата/время	
ДатаПриезда	Дата/время	
РазмерСуточных	Денежный	

Рис. 2.29. Структура таблицы **КОМАНДИРОВКИ**

9. Откройте таблицу **СОТРУДНИКИ** и введите в нее произвольные данные (3–5 строк). Пример заполненной таблицы показан на рис. 2.30.

КодСотруд	Фамилия	Имя	Отчество	Должность	Оклад	ДатаНазначе
1	Иванов	Иван	Петрович	бухгалтер	40 000,00р.	11.12.2003
2	Алехин	Игорь	Иванович	инженер	25 000,00р.	11.10.2005
3	Зотова	Инна	Андреевна	менеджер	30 000,00р.	13.01.2009
(Счетчик)					0,00р.	

Рис. 2.30. Пример заполнения таблицы **СОТРУДНИКИ**

При вводе данных учитывайте следующие рекомендации:

- в поле **Должность** обязательно введите хотя бы по одному значению **Инженер, Бухгалтер, Менеджер**;
- в поле **ДатаНазначения** введите значения дат назначения сотрудников на должность в формате **дд.мм.гг** (например: 21.03.02) в диапазоне дат 2000–2010 годы. *При вводе дат используйте правый числовой блок клавиатуры.*

10. Откройте таблицу **ЛИЧНЫЕ_ДАННЫЕ** и введите в нее произвольные данные (количество заполненных строк должно быть равно количеству строк в таблице **СОТРУДНИКИ**). Для каждого сотрудника в поле **КодСотрудника** вводите его код из таблицы **СОТРУДНИКИ** (в нашем примере код Иванова равен 1, код Зотовой равен 3). Пример заполненной таблицы показан на рис. 2.31.

КодСотрудни	ДатаРождения	Адрес	Телефон
1	01.02.1980	ул. Мира 45 - 59	24-68-86
2	12.12.1981	ул. Лесная 56 - 98	34-68-87
3	12.12.1976	ул. Лесная 34 - 21	24-68-88

Рис. 2.31. Пример заполнения таблицы **ЛИЧНЫЕ_ДАННЫЕ**

11. Откройте таблицу **КОМАНДИРОВКИ** и введите в нее произвольные данные. Пример заполненной таблицы показан

на рис. 2.32. При заполнении таблицы **КОМАНДИРОВКИ** обязательно используйте названия городов **Самара** и **Москва**.



КодКоманд	КодСотру	Город	ДатаОтъезд	ДатаПриезда	РазмерСуточн
	1	2 Москва	21.03.2008	26.03.2008	100,00р.
	2	1 Самара	10.05.2008	20.05.2008	200,00р.
	3	2 Самара	06.07.2007	16.07.2007	100,00р.
	4	4 Москва	05.03.2008	15.03.2008	200,00р.
▶	Счетчик	0			0,00р.

Рис. 2.32. Пример заполнения таблицы **КОМАНДИРОВКИ**

12. Установите межтабличные связи. Для этого нажмите на панели инструментов кнопку **Схема данных**  или выполните в меню команду **Сервис – Схема данных**. Должно открыться окно **Схема данных**. Одновременно с открытием этого окна открывается диалоговое окно **Добавление таблицы**. Щелчком на кнопке **Добавить** выберите таблицы **СОТРУДНИКИ**, **ЛИЧНЫЕ_ДАННЫЕ** и **КОМАНДИРОВКИ** (рис. 2.33). Закройте окно **Добавление таблицы**.

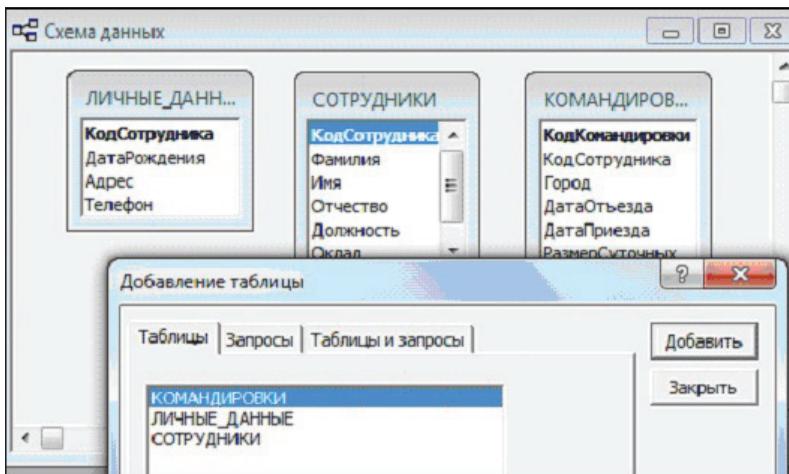


Рис. 2.33. Добавление таблиц к схеме данных

13. Перетащите мышкой поле **КодСотрудника** из таблицы **СОТРУДНИКИ** на поле **КодСотрудника** таблицы **ЛИЧНЫЕ_ДАННЫЕ**. При отпускании кнопки мыши автоматически откроется диалоговое окно **Изменение связей**. В окне **Изменение связей** убедитесь, что поля для связи выбраны правильно и что между таблицами установлено отношение «**один-к-одному**». Щелкните на кнопке **Создать**.

14. Перетащите мышкой поле **КодСотрудника** из таблицы **СОТРУДНИКИ** на поле **КодСотрудника** таблицы **КОМАНДИРОВКИ**. В окне **Изменение связей** убедитесь, что поля для связи выбраны правильно и что между таблицами установлено отношение «**один-ко-многим**». Щелкните на кнопке **Создать**.

15. В окне **Схема данных** рассмотрите образовавшиеся связи. Убедитесь в том, что линию связи можно выделить щелчком левой кнопки мыши, а щелчком правой кнопки мыши открывается контекстное меню, позволяющее удалить или отредактировать связь (рис. 2.34). Закройте окно **Схема данных**.

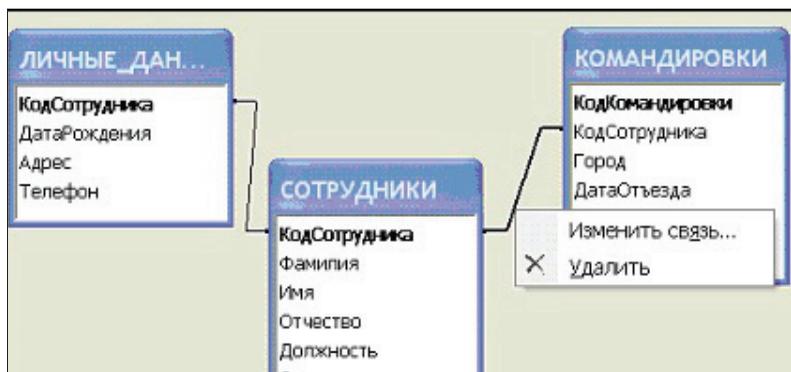


Рис. 2.34. Контекстное меню для изменения или удаления связи

16. Создайте автоформу **в столбец** на основе таблицы **СОТРУДНИКИ**. Для этого откройте панель **Формы** в окне **База данных**. Щелкните кнопку **Создать** – откроется окно для выбора способа создания формы и выбора базовой

таблицы (рис. 2.35). После щелчка по кнопке **ОК** будет создана автоформа.

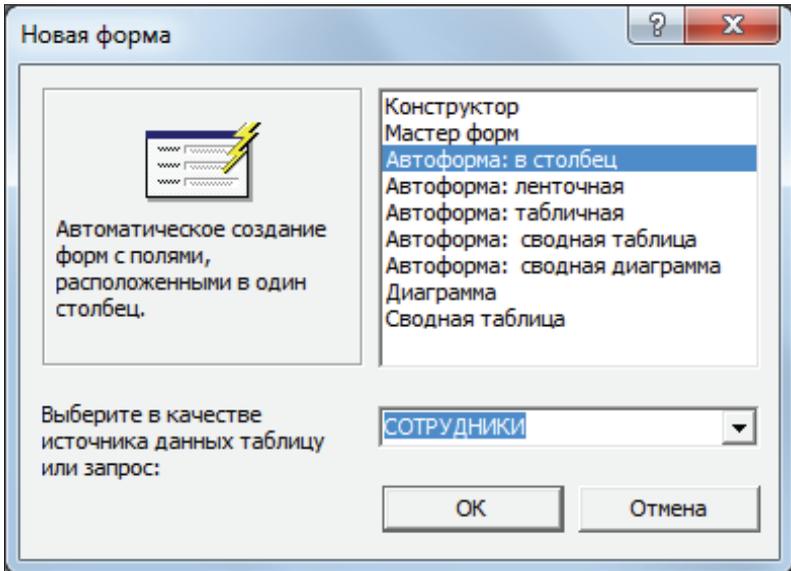


Рис. 2.35. Создание автоформы в столбец для таблицы **СОТРУДНИКИ**

17. Закройте форму. При закрытии сохраните форму с именем **СОТРУДНИКИ**. Значок формы появится в окне **База данных** на вкладке **Формы**. Откройте форму, просмотрите записи таблицы **СОТРУДНИКИ**.

18. Создайте с помощью **Мастера форм** форму, базирующуюся на таблицах **СОТРУДНИКИ** и **ЛИЧНЫЕ_ДАННЫЕ**. С помощью этой формы можно просматривать данные сразу двух таблиц или вводить данные в эти таблицы. Для создания формы выполните следующие действия:

- активируйте **Мастер форм** с помощью значка **Создание формы с помощью мастера** на панели **Формы** в окне **База данных**;
- в диалоговом окне **Создание форм** (рис. 2.36) выбирается сначала таблица **СОТРУДНИКИ**, и все поля таблицы включаются в форму (перемещаются из поля **Доступные поля** в поле **Выбранные поля**);

- затем в этом же окне выбирается таблица **ЛИЧНЫЕ_ДАННЫЕ**, из которой включаются все поля, кроме **КодСотрудника**.

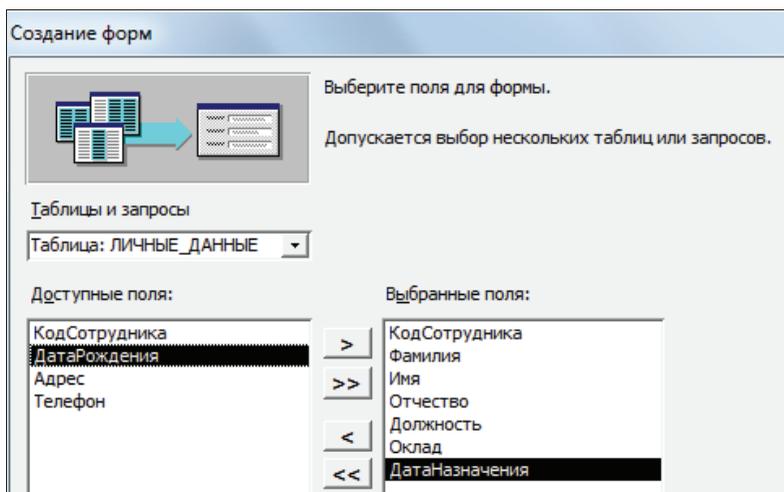


Рис. 2.36. Работа с Мастером форм

19. Выполните все этапы создания формы, приняв предлагаемые по умолчанию параметры, завершите процедуру сохранением формы с именем **ОБЩАЯ_ИНФОРМАЦИЯ**. В результате должна получиться форма, показанная на рис. 2.37.

20. Введите с помощью формы информацию о новом сотруднике фирмы. Открыв таблицы **СОТРУДНИКИ** и **ЛИЧНЫЕ_ДАННЫЕ**, убедитесь, что в таблицах появились новые записи.

21. Создайте с помощью **Мастера форм** форму с подчиненной формой для просмотра и ввода данных в таблицы **СОТРУДНИКИ** и **КОМАНДИРОВКИ**. Процедура создания формы аналогична процедуре, описанной в пунктах 18–19. Из таблицы **СОТРУДНИКИ** следует включить все поля, а из таблицы **КОМАНДИРОВКИ** – все, кроме поля **КодСотрудника**.

Важно правильно выбрать параметры на втором шаге процедуры создания формы: вид представления данных – **по СОТРУДНИКИ**, тип формы – **Подчиненные формы**. На следующем шаге выбирается вид подчиненной формы – **табличный**.

ОБЩАЯ_ИНФОРМАЦИЯ

КодСотрудника: 1

Фамилия: Иванов

Имя: Иван

Отчество: Петрович

Должность: бухгалтер

Оклад: 40 000,00р.

ДатаНазначения: 11.12.2003

ДатаРождения: 01.02.1980

Адрес: ул. Мира 45 - 59

Телефон: 24-68-86

Запись: 1 из 3

Рис. 2.37. Форма для таблиц СОТРУДНИКИ и ЛИЧНЫЕ_ДАННЫЕ

22. Выполнив все этапы создания формы, завершите процедуру сохранением формы с именем **КОМАНДИРОВКИ**. В результате получится форма, показанная на рис. 2.38.

КОМАНДИРОВКИ

КодСотрудника: 1 ДатаНазначения: 11.12.2003

Фамилия: Иванов

Имя: Иван

Отчество: Петрович

Должность: бухгалтер

Оклад: 40 000,00р.

КОМАНДИРОВКИ

	КодКом	Город	ДатаОтъезд	ДатаПриезд	РазмерСуточных
▶	2	Самара	10.05.2008	20.05.2008	200,00р.
* >	Четчик				0,00р.

Запись: 1 из 3

Рис. 2.38. Форма с подчиненной формой

23. Используя форму, введите информацию о новой командировке. Открыв таблицу **КОМАНДИРОВКИ**, убедитесь в том, что в ней появилась новая запись.

24. Завершите работу с базой данных.

Практическая работа 2.2. Работа с запросами на выборку в базе данных Microsoft Access

Цель работы – научиться создавать простые запросы на выборку и запросы с параметром, выполнять вычисления с помощью запросов.

Порядок выполнения работы

1. Изучить теоретический материал из раздела 2.3.
2. Выполнить практическое задание.

Практическое задание

Создать запросы к базе данных **Кадры** следующих типов:

- простые запросы на выборку **Запрос1 – Запрос7**;
- запросы с параметром **Запрос8 – Запрос11**;
- запросы с вычисляемыми полями **Запрос12, Запрос13**.

Технология выполнения задания

1. Запустите программу **Microsoft Access**. Откройте файл базы данных (**Файл – Открыть – Кадры.mdb**).
2. В окне базы данных **Кадры** откройте панель **Запросы** (рис. 2.39). Процедура создания каждого запроса активируется с помощью значка **Создание запроса в режиме конструктора**.

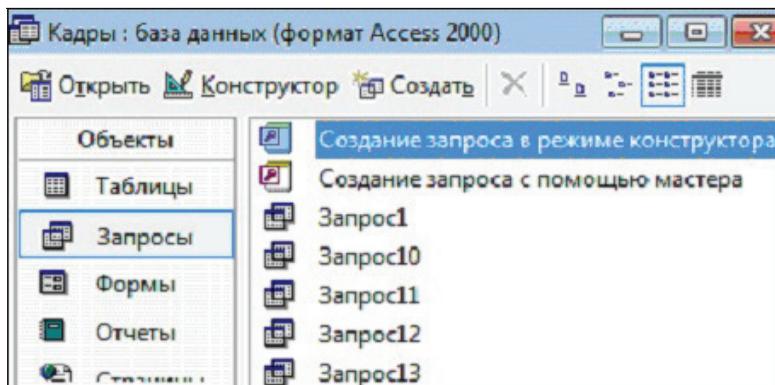


Рис. 2.39. Панель **Запросы** в окне **База данных**

3. Создайте запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос1	КодСотрудника Фамилия Должность	Выбирает из таблицы СОТРУДНИКИ информацию о сотрудниках, работающих в должности менеджер . Выполняет сортировку записей по полю Фамилия по возрастанию

Для создания запроса выполните следующие действия:

- дважды щелкните на значке **Создание запроса в режиме Конструктора** – откроется бланк запроса по образцу, одновременно с ним откроется диалоговое окно **Добавление таблицы** (рис. 2.40);
- в окне **Добавление таблицы** выберите таблицу **СОТРУДНИКИ**, щелкните на кнопке **Добавить**, закройте окно **Добавление таблицы**;
- в списке полей таблицы **СОТРУДНИКИ** выберите поля, включаемые в результирующую таблицу запроса: **КодСотрудника**, **Фамилия**, **Должность**, **Оклад** (выбор производится двойным щелчком на имени поля);
- задайте условие отбора для поля **Должность**, введя текст «**менеджер**» в строку **Условие отбора** для поля **Должность** (рис. 2.41);
- задайте условие сортировки по полю **Фамилия**, выбрав из списка в строке **Сортировка** значение **по возрастанию**;
- закройте бланк запроса по образцу, при закрытии сохраните запрос с именем **Запрос1**;
- в окне **Кадры: база данных** двойным щелчком мыши по имени запроса откройте только что созданный запрос, проанализируйте результирующую таблицу. Ее содержание зависит от того, что было введено в таблицу **СОТРУДНИКИ** при ее заполнении данными.

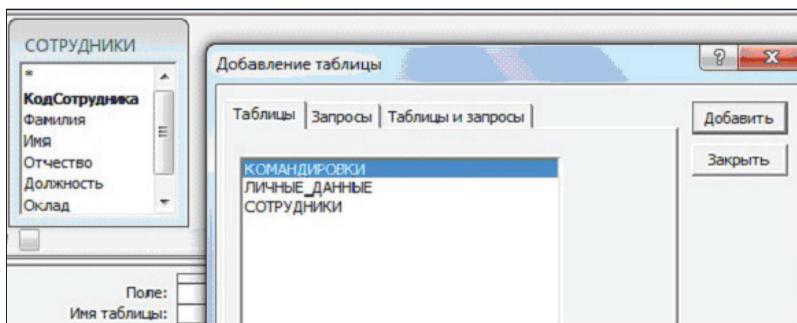


Рис. 2.40. Добавление таблиц для создания запроса

Поле:	КодСотрудника	Фамилия	Должность
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ
Сортировка:		по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	по возрастанию	<input checked="" type="checkbox"/>
Условие отбора:		по убыванию	"менеджер"
или:		(отсутствует)	

Рис. 2.41. Определение условий отбора записей в запросе

4. Создайте запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос2	КодСотрудника Фамилия Должность	Выбирает из таблицы СОТРУДНИКИ информацию о сотрудниках, работающих в должности «бухгалтер», фамилия которых начинается с буквы «И»

При создании запроса в строке **Условие отбора** для поля **Фамилия** введите с клавиатуры **И*** (это означает, что первая буква фамилии равна **И**, а далее могут следовать любые символы).

5. Сохраните запрос с именем **Запрос2**, выполните запрос. Проанализируйте полученную результирующую таблицу.

6. Создайте запрос по данным из двух взаимосвязанных таблиц **СОТРУДНИКИ** и **ЛИЧНЫЕ_ДАННЫЕ** в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос3	Фамилия Имя Отчество Адрес Телефон ДатаРождения	Выбирает из таблиц СОТРУДНИКИ и ЛИЧНЫЕ_ДАННЫЕ информацию о сотрудниках, дата рождения которых находится в диапазоне от 01.01.80 до 31.12.89

Для создания запроса выполните следующие действия:

- дважды щелкните на значке **Создание запроса в режиме Конструктора**, в окне **Добавление таблицы** выберите таблицы **СОТРУДНИКИ** и **ЛИЧНЫЕ_ДАННЫЕ**;
- выберите поля, включаемые в запрос, и задайте условия отбора записей: в строке **Условие отбора** для поля **ДатаРождения** введите условие **Between 01.01.1980 And 31.12.89** (после завершения ввода условие примет вид: **Between #01.01.1980# And #31.12.1989#**);
- закройте бланк запроса по образцу, сохранив запрос с именем **Запрос3**.

7. Выполните запрос (двойной щелчок на имени запроса в окне базы данных). Проанализируйте полученную результирующую таблицу.

8. Создайте и выполните запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос4	Фамилия Имя Отчество Должность Оклад	Выбирает из таблицы СОТРУДНИКИ информацию о сотрудниках, работающих в должности «бухгалтер» или «менеджер», имеющих оклад больше 5000 рублей

При создании запроса воспользуйтесь следующими рекомендациями:

- в строке **Условие отбора** для поля **Должность** введите условие **бухгалтер OR менеджер**;
- в строке **Условие отбора** для поля **Оклад** введите условие **>5000**.

9. Сохраните и выполните запрос. Проанализируйте полученную результирующую таблицу.

10. Создайте и выполните запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос5	Фамилия Имя Отчество Адрес Телефон ДатаРождения	Выбирает из таблиц СОТРУДНИКИ и ЛИЧНЫЕ_ДАННЫЕ информацию о сотрудниках, номера телефонов которых начинаются с цифры «2», родившихся после 01.01.80

При создании запроса воспользуйтесь следующими рекомендациями:

- в строке **Условие отбора** для поля **Телефон** введите условие **2***, что означает «*первый символ 2, остальные символы могут быть любыми*»;
- в строке **Условие отбора** для поля **ДатаРождения** введите условие **>01.01.80**;

11. Создайте и выполните запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос6	КодСотрудника Фамилия Должность ДатаНазначения	Выбирает из таблицы СОТРУДНИКИ информацию о сотрудниках, либо работающих в должности менеджер , либо принятых на работу до 01.01.90 , либо оба эти условия могут быть выполнены одновременно

При создании запроса воспользуйтесь следующими рекомендациями:

- в строке **Условие отбора** для поля **Должность** введите условие **менеджер**;
- в строке **или** для поля **ДатаНазначения** введите условие **<01.01.90**. Условия, введенные на разных строках, приводят к тому, что из таблицы **СОТРУДНИКИ** будут выбраны строки, которые удовлетворяют *хотя бы одному из двух условий*.

12. Создайте и выполните запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос7	Фамилия Имя Отчество Должность Город	Выбирает из таблиц СОТРУДНИКИ и КОМАНДИРОВКИ информацию о сотрудниках, работающих в должности инженер , которые выезжали в командировку в город Москва

13. Создайте и выполните *запрос с параметром* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос8	КодСотрудника Фамилия Имя Отчество Должность	Выбирает из таблицы СОТРУДНИКИ информацию о сотрудниках, работающих в некоторой должности . При этом <i>конкретное значение должности вводится при выполнении запроса</i>

Для поля **Должность** в строке **Условие отбора** введите условие на ввод параметра в квадратных скобках: **[Введите название должности]** (рис. 2.42).

14. Выполните запрос. При выполнении откроется диалоговое окно для ввода значения параметра (рис. 2.43). Введите одно из названий должностей, которые есть в вашей

таблице **СОТРУДНИКИ** (например, **инженер**). Введенное значение будет использовано в качестве критерия отбора записей из таблицы.

Поле:	КодСотруд-	Фамилия	Имя	Отчество	Должность
Имя таблицы:	СОТРУДНИК	СОТРУДНИК	СОТРУДНИК	СОТРУДНИК	СОТРУДНИКИ
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>				
Условие отбора: или:					[Введите название должности]

Должность
СОТРУДНИКИ
<input type="checkbox"/>
<input checked="" type="checkbox"/>
[Введите название должности]

Рис. 2.42. Запрос с параметром в режиме **Конструктор**

Введите значение параметра ✖

Введите название должности

инженер

ОК

Отмена

Рис. 2.43. Диалоговое окно для ввода значения параметра **Должность**

15. Создайте и выполните запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос9	КодСотрудника Фамилия Имя Отчество Город	Выбирает из таблиц СОТРУДНИКИ и КОМАНДИРОВКИ информацию о сотрудниках, которые были в командировке в некотором городе . <i>Название города вводится при выполнении запроса</i>

16. Создайте и выполните *запрос с параметром-диапазоном* для таблиц **СОТРУДНИКИ** и **ЛИЧНЫЕ_ДАННЫЕ** в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос10	Фамилия Имя Отчество ДатаРождения	Выбирает из таблиц СОТРУДНИКИ и ЛИЧНЫЕ_ДАННЫЕ информацию о сотрудниках, родившихся в некоторый период времени . <i>Нижняя и верхняя границы периода вводятся при выполнении запроса</i>

Выберите поля, включаемые в запрос. В строке **Условие отбора** для поля **ДатаРождения** введите условие: **Between [Нижняя граница] And [Верхняя граница]** (рис. 2.44).

Поле:	Фамилия	Имя	Отчество	ДатаРождения
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ	ЛИЧНЫЕ_ДАННЫЕ
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				Between [Нижняя граница] And [Верхняя граница]

ДатаРождения
ЛИЧНЫЕ_ДАННЫЕ
<input checked="" type="checkbox"/>
Between [Нижняя граница] And [Верхняя граница]

Рис. 2.44. Запрос с параметром-диапазоном

17. Выполните запрос. При выполнении запроса сначала появится диалоговое окно для ввода нижней границы диапазона значений поля **ДатаРождения**, затем — для ввода верхней границы. Эти значения будут использованы для определения кодов сотрудников в таблице **ЛИЧНЫЕ_ДАННЫЕ**, даты рождения которых попадают в заданный диапазон. По этим кодам будут отбираться данные из таблицы **СОТРУДНИКИ**.

18. Создайте и выполните запрос в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос11	КодСотрудника Фамилия Должность Оклад	Выбирает из таблицы СОТРУДНИКИ информацию о сотрудниках, работающих в должности менеджер и имеющих Оклад в пределах некоторого диапазона . При этом конкретные значения границ диапазона <i>вводятся при выполнении запроса</i>

19. Создайте и выполните *запрос с вычисляемыми полями* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос12	КодСотрудника Фамилия Оклад Налог СуммаКвыдаче	Вычисляет для каждого сотрудника Налог и СуммуКвыдаче по формулам: Налог: $[\text{Оклад}] * 0,13$ СуммаКвыдаче: $[\text{Оклад}] - [\text{Налог}]$

Для создания запроса выполните следующие действия:

- в окне **Добавление таблицы** выберите таблицу **СОТРУДНИКИ**. Выберите поля, включаемые в запрос;
- в новом поле бланка запроса в строке **Поле** введите формулу **Налог:** $[\text{Оклад}] * 0,13$ (рис. 2.45);
- в следующем поле введите формулу **СуммаКвыдаче:** $[\text{Оклад}] - [\text{Налог}]$.

Поле:	КодСотрудника	Фамилия	Оклад	Налог: $[\text{Оклад}] * 0,13$	СуммаКвыдаче: $[\text{Оклад}] - [\text{Налог}]$
Имя таблицы:	СОТРУДНИКИ	СОТРУДН	СОТРУД		
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>				
Условие отбора:					
или:					

Налог: $[\text{Оклад}] * 0,13$	СуммаКвыдаче: $[\text{Оклад}] - [\text{Налог}]$
--------------------------------	---



Рис. 2.45. Запрос с вычисляемыми полями (фрагмент)

20. Сохраните и выполните запрос.

21. Создайте и выполните *запрос с вычисляемым полем* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос13	КодСотрудника Фамилия Должность Оклад Премия	Вычисляет для сотрудников, работающих в должности «менеджер», размер премии по формуле Премия: [Оклад]*0,5

22. Завершите работу с базой данных.

Практическая работа 2.3. Работа с итоговыми запросами и запросами на изменение таблиц в базе данных Microsoft Access. Создание отчетов

Цель работы – научиться:

- создавать итоговые запросы;
- выполнять автоматическое изменение данных в таблицах с помощью запросов на изменение;
- создавать отчеты на основе базовых таблиц и результирующих таблиц запросов.

Порядок выполнения работы

1. Изучить теоретический материал из раздела 2.3.
2. Выполнить практическое задание.

Практическое задание

Создать запросы к базе данных следующих типов:

- *итоговые* запросы **Запрос14**, **Запрос15**;
- запрос на *создание новой базовой таблицы* **Запрос16**;
- на *обновление данных* **Запрос17**, **Запрос18**;
- на *удаление данных из таблиц* **Запрос19**, **Запрос20**.

Технология выполнения задания

1. Запустите программу **Microsoft Access**. Откройте файл базы данных. В окне **Кадры: база данных** откройте панель **Запросы**.

2. Создайте *итоговый запрос* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос14	Должность Оклад (трижды)	Вычисляет среднее, наибольшее и наименьшее значение по полю Оклад для каждой должности

Для создания запроса выполните следующие действия:

- дважды щелкните на значке **Создание запроса в режиме Конструктора** – откроется бланк запроса по образцу, в окне **Добавление таблицы** выберите таблицу **СОТРУДНИКИ**;
- включите в запрос поле **Должность** и *трижды* включите поле **Оклад**;
- на панели инструментов в окне программы Microsoft

Access щелкните на кнопке **Групповые операции**  или воспользуйтесь командами меню (**Вид – Групповые операции**), в нижней части бланка появится строка **Групповые операции** (рис. 2.46);

- для поля, по которому производится группировка записей (в нашем случае – поле **Должность**), оставьте в строке **Групповые операции** значение **Группировка**, для остальных полей щелкните в этой строке – появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле;
- для первого поля **Оклад** выберите итоговую функцию **Avg** для определения среднего значения оклада по каждой должности, для второго поля **Оклад** – итоговую функцию **Max**, для третьего поля **Оклад** – итоговую функцию **Min**;
- закройте бланк запроса по образцу, присвоив запросу имя **Запрос14**.

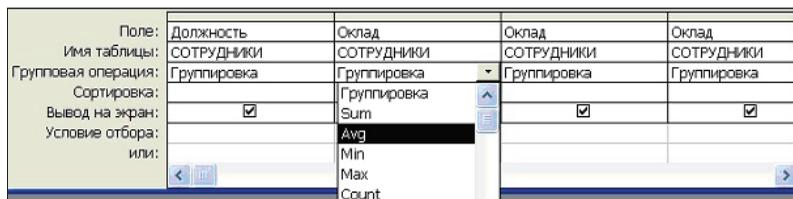


Рис. 2.46. Формирование итогового запроса

3. Выполните запрос и проанализируйте результирующую таблицу.

4. Создайте и выполните *итоговый запрос* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос15	Город РазмерСуточных (трижды)	Вычисляет среднее, наибольшее и наименьшее значение по полю РазмерСуточных для каждого города при поездках в командировки

Запрос15 создается на базе таблицы **КОМАНДИРОВКИ** по аналогии с предыдущим запросом (**Запрос14**).

5. Создайте запрос *на создание базовой таблицы*. В результате выполнения этого запроса в базе данных будет создана новая базовая таблица **АДРЕСА_СОТРУДНИКОВ**.

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос16	КодСотрудника Фамилия Имя Отчество Адрес Телефон	Создает новую базовую таблицу АДРЕСА_СОТРУДНИКОВ

Для создания запроса выполните следующие действия:

- дважды щелкните на значке **Создание запроса в режиме Конструктора** – откроется бланк запроса по образцу, в окне **Добавление таблицы** выберите таблицы **СОТРУДНИКИ** и **ЛИЧНЫЕ_ДАННЫЕ**;
- включите в запрос поля **КодСотрудника**, **Фамилия**, **Имя**, **Отчество** из таблицы **СОТРУДНИКИ** и поля **Адрес** и **Телефон** из таблицы **ЛИЧНЫЕ_ДАННЫЕ**;
- откройте список инструмента **Тип запроса** (рис. 2.47) и выберите **Создание таблицы**;
- в открывшемся диалоговом окне **Создание таблицы** введите имя новой таблицы **АДРЕСА_СОТРУДНИКОВ**,

установите переключатель **в текущей базе данных** и нажмите кнопку **ОК**;

- закройте бланк запроса по образцу, сохранив запрос с именем **Запрос16**.

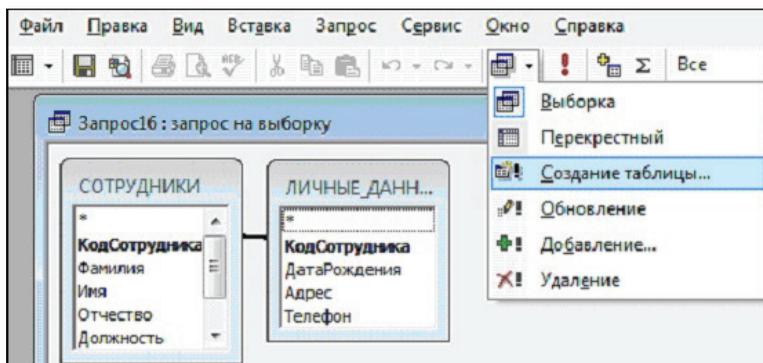


Рис. 2.47. Выбор типа запроса

6. Выполните запрос. Ответьте «Да» на все выводимые программой вопросы. Переключитесь в окне базы данных на панель **Таблицы**. В списке таблиц должна появиться новая таблица **АДРЕСА_СОТРУДНИКОВ**. Откройте таблицу **АДРЕСА_СОТРУДНИКОВ**, просмотрите ее содержимое.

7. Создайте *запрос на обновление данных* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос17	Должность Оклад	Увеличивает на 20% значения в поле Оклад таблицы СОТРУДНИКИ для сотрудников, работающих в должности « инженер »

Для создания запроса выполните следующие действия:

- дважды щелкните на значке **Создание запроса в режиме Конструктора** – откроется бланк запроса по образцу, в окне **Добавление таблицы** выберите таблицу **СОТРУДНИКИ**, включите в запрос поля **Оклад**, **Должность**;
- откройте список инструмента **Тип запроса** и выберите **Обновление**;

- в строке **Условие отбора** для поля **Должность** введите «**инженер**». В строке **Обновление** для поля **Оклад** введите формулу для обновления значений: $[Оклад]*1,2$ (рис. 2.48);
- закройте бланк запроса по образцу, сохраните запрос с именем **Запрос18**.

Поле:	Оклад	Должность
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ
Обновление:	$[Оклад]*1,2$	
Условие отбора:		"инженер"
или:		

Рис. 2.48. Запрос на обновление значений таблицы **СОТРУДНИКИ** в режиме **Конструктор**

8. Выполните запрос. Ответьте «Да» на все выводимые программой вопросы. Переключитесь в окне базы данных на панель **Таблицы**. Откройте таблицу **СОТРУДНИКИ**. Убедитесь, что в строках, соответствующих должности «**инженер**», изменились значения окладов.

9. Создайте и выполните *запрос на обновление данных* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос18	Оклад ДатаНазначения	Увеличивает на 30% значения в поле Оклад таблицы СОТРУДНИКИ для сотрудников, назначенных на должность до 01.01.2009

10. Создайте *запрос на удаление данных* **Запрос19** в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос19	Поле ДатаНазначения из таблицы КОПИЯ_СОТРУДНИКИ , которую нужно предварительно создать	Удаляет из таблицы КОПИЯ_СОТРУДНИКИ записи, для которых значение в поле ДатаНазначения больше 01.01.2008

Для создания таблицы **КОПИЯ СОТРУДНИКИ** щелкните правой кнопкой имя таблицы **СОТРУДНИКИ** на вкладке **Таблицы** в окне **Базы данных**. В открывшемся контекстном меню выберите команду **Копировать**. Затем щелкните правой кнопкой в свободном пространстве вкладки **Таблицы** окна **Базы данных** и выберите команду **Вставить**. Откроется диалоговое окно, в которое нужно ввести имя таблицы **КОПИЯ_СОТРУДНИКИ**, включить опцию **структура и данные** и нажать кнопку **ОК**.

Для создания запроса **Запрос19** выполните следующие действия:

- дважды щелкните на значке **Создание запроса в режиме Конструктора** – откроется бланк запроса по образцу, в окне **Добавление таблицы** выберите таблицу **КОПИЯ_СОТРУДНИКИ**;
- выберите тип запроса **Удаление**;
- переместите из списка полей таблицы **КОПИЯ_СОТРУДНИКИ** на бланке запроса символ «*» в первый столбец бланка;
- включите в запрос поле **ДатаНазначения**, в строке **Условие отбора** для поля **ДатаНазначения** введите условие: **>01.01.2008** (рис. 2.49);
- закройте бланк запроса по образцу, сохраните запрос с именем **Запрос19**.

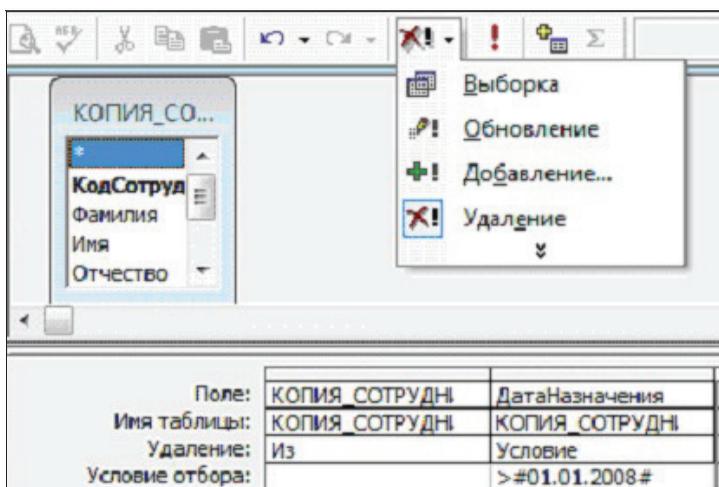


Рис. 2.49. Запрос на удаление записей в режиме **Конструктор**

11. Выполните запрос. Ответьте «Да» на все выводимые программой вопросы. Переключитесь в окне базы данных на панель **Таблицы**. Откройте таблицу **КОПИЯ_СОТРУДНИКИ**. Убедитесь, что произошло удаление строк в соответствии с заданным условием.

12. Создайте и выполните *запрос на удаление данных* в соответствии со следующими требованиями:

Имя запроса	Включаемые в запрос поля	Какую задачу решает запрос
Запрос20	Поле Должность из таблицы КОПИЯ_СОТРУДНИКИ	Удаляет из таблицы КОПИЯ_СОТРУДНИКИ записи, для которых значение в поле Должность равно «бухгалтер» или «менеджер»

13. Создайте автоотчет «**в столбец**» на базе таблицы **СОТРУДНИКИ** в соответствии со следующими требованиями:

Имя отчета	Включаемые в отчет поля	Представленные в отчете данные и тип отчета
Отчет1	Все поля таблицы СОТРУДНИКИ	Автоотчёт « в столбец » отображает данные из таблицы СОТРУДНИКИ

Для создания отчета выполните следующие действия:

- в окне **Кадры: база данных** откройте панель **Отчеты**;
- для создания автоотчета **Отчет1** щелкните кнопку **Создать** и выберите в открывшемся окне **Новый отчет** опцию **Автоотчет: в столбец**;
- выберите таблицу **СОТРУДНИКИ** и щелкните кнопку **ОК**. Закройте отчет, при закрытии сохраните его, присвоив имя **Отчет1**.

14. Завершите работу с базой данных.

2.5. Индивидуальное домашнее задание 1.

Проектирование базы данных для заданной предметной области

Цель работы – научиться проектировать простейшие базы данных, выполнять компьютерную реализацию спроектированной базы данных.

Общие положения

В процессе выполнения индивидуального домашнего задания (ИДЗ) можно условно выделить следующие этапы:

- проектирование схемы данных;
- создание компьютерной реализации базы данных, включающей базовые таблицы и экранные формы для работы с таблицами;
- создание запросов к базе данных и отчетов в соответствии с требованиями задания по ИДЗ.

Представление результатов выполнения ИДЗ

Результатом выполнения ИДЗ является файл базы данных в формате MS Access. Файл базы данных представляется преподавателю на последнем аудиторном занятии в рамках учебной темы «Базы данных». Защита ИДЗ проходит *в форме собеседования* с преподавателем по результатам выполненной работы.

Задание на выполнение

Задание 1

Выполнить проектирование базы данных в соответствии с номером варианта (прил. 4):

- 1) выполнив анализ предметной области, выделить информационные объекты для представления их в базе данных;
- 2) описать информационные объекты с помощью заданных атрибутов (можно ввести в рассмотрение дополнительные атрибуты);
- 3) перейти к представлению информационных объектов в виде реляционных таблиц: определить состав полей базовых таблиц; указать свойства каждого поля в таблице; в каждой таблице найти ключевое поле; установить тип связей между таблицами базы данных.

Пример выполнения задания 1

Описание предметной области

Фирма выполняет ремонт компьютеров. Требуется разработать базу данных для хранения информации о выполнении ремонтных работ сотрудниками фирмы. При оформлении заказа фиксируется дата его выполнения, вид выполненной работы, исполнитель работы. Каждый исполнитель получает фиксированный процент вознаграждения от стоимости выполнения работы. Этот процент устанавливается персонально каждому исполнителю при заключении трудового договора между фирмой и работником.

Анализ описания предметной области позволяет выделить набор данных, которые должны храниться в проектируемой базе данных.

1. **Фамилия исполнителя работы.**
2. **Имя исполнителя работы.**
3. **Отчество исполнителя работы.**
4. **Процент вознаграждения (может различаться для разных исполнителей).**
5. **Наименование работы.**
6. **Стоимость работы (фиксированная для каждого наименования работы).**
7. **Дата исполнения работы.**

Исходя из набора данных, которые должны храниться в БД, можно выделить два информационных объекта: **Исполнитель (Фамилия, Имя, Отчество, Процент вознаграждения)** и **Работа (Наименование, Стоимость работы)**. Создадим соответствующие таблицы **ИСПОЛНИТЕЛИ** и **РАБОТЫ** (рис. 2.50). Ни одно из первоначально заданных полей таблицы **ИСПОЛНИТЕЛИ** не определяет однозначно каждую запись таблицы, поэтому в таблицу введено поле **Код исполнителя**, значения в котором будут уникальными для каждого исполнителя. Это поле является первичным ключом таблицы **ИСПОЛНИТЕЛИ** и будет задано в Microsoft Access как ключевое поле. С этой же целью в таблицу **РАБОТЫ** введен первичный ключ **Код работы**.

ИСПОЛНИТЕЛИ		РАБОТЫ	
Код исполнителя		Код работы	
Фамилия		Наименование	
Имя		Стоимость работы	
Отчество			
Процент вознаграждения			

Рис. 2.50. Таблицы **ИСПОЛНИТЕЛИ** и **РАБОТЫ** проектируемой базы данных

В таблице **ИСПОЛНИТЕЛИ** будут храниться записи следующего вида:

1	Иванов	Андрей	Петрович	20
2	Алексеев	Игорь	Андреевич	25

В таблице **РАБОТЫ** будут храниться записи следующего вида:

1	Установка микропроцессора	100.00 р.
2	Замена вентилятора	50.00 р.

Один исполнитель может выполнять различные работы. Одна и та же работа может быть выполнена разными исполнителями. Между таблицами **ИСПОЛНИТЕЛИ** и **РАБОТЫ** должна быть установлена связь типа «многие-ко-многим». Для организации такой связи потребуется новая таблица, в которую ключевые поля связываемых таблиц войдут как внешние ключи. Такой таблицей в нашем случае может быть таблица **ЗАКАЗЫ**. Данные в нее будут заноситься при оформлении заказа на выполнение работы исполнителем. Из первоначального перечня атрибутов в эту таблицу войдет атрибут **Дата исполнения работы**. В таблице **ЗАКАЗЫ** также определим ключевое поле **Код заказа**.

Установим связь между таблицами посредством совпадающих полей в связываемых таблицах (рис. 2.51). Тип связи между таблицами **ИСПОЛНИТЕЛИ** и **ЗАКАЗЫ** – «один-ко-многим». Здесь таблица **ИСПОЛНИТЕЛИ** находится

на стороне отношения «один», она является главной таблицей. Таблица **ЗАКАЗЫ** находится на стороне отношения «многие» и является подчиненной таблицей. Таблица **РАБОТЫ** – главная для связи между таблицами **РАБОТЫ** и **ЗАКАЗЫ** (связь «один-ко-многим»). А связь между таблицами **ИСПОЛНИТЕЛИ** и **РАБОТЫ** относится к типу «многие-ко-многим».



Рис. 2.51. Связи между таблицами проектируемой базы данных

Определим свойства полей в каждой из табл. 2.2, 2.3, 2.4. Рекомендуется выбирать короткие имена полей, а типы и размеры полей – исходя из характера информации, которую предполагается хранить в поле.

Таблица 2.2

Свойства полей таблицы ИСПОЛНИТЕЛИ проектируемой БД

Имя поля в схеме данных	Имя поля в компьютерной БД	Тип поля	Размер поля
Код исполнителя	Код_исполнителя	счетчик	устанавливается автоматически
Фамилия	Фамилия	текстовый	30 байт
Имя	Имя	текстовый	20 байт
Отчество	Отчество	текстовый	30 байт
Процент вознаграждения	Процент	числовой	длинное целое

Таблица 2.3

Свойства полей таблицы РАБОТЫ проектируемой БД

Имя поля в схеме данных	Имя поля в компьютерной БД	Тип поля	Размер поля
Код работы	Код_работы	счетчик	устанавливается автоматически
Стоимость работы	Стоимость	денежный	устанавливается автоматически

Таблица 2.4

Свойства полей таблицы ЗАКАЗЫ проектируемой БД

Имя поля в схеме данных	Имя поля в компьютерной БД	Тип поля	Размер поля
Код заказа	Код заказа	счетчик	устанавливается автоматически
Код работы	Код_работы	числовой	длинное целое
Код исполнителя	Код_исполнителя	числовой	длинное целое
Дата исполнения	Дата_исполнения	дата/время	—

Теперь наша база данных подготовлена к компьютерной реализации.

Задание 2

Выполнить компьютерную реализацию спроектированной базы данных.

1. Создать таблицы базы данных.
2. Установить связи между таблицами.
3. Заполнить таблицы данными (каждая таблица должна содержать не менее 10 записей).
4. Создать две формы для работы с таблицами базы данных:

- 1) автоформа в столбец для одной из главных таблиц;
- 2) форма с подчиненной формой для главной и подчиненной таблиц.

5. Сформулировать задачи (не менее четырех) для отбора данных из таблиц и создать **запросы на выборку** для их решения.

Внимание! Запросы обязательно должны содержать условия отбора.

6. Сформулировать задачи (не менее двух) и создать **запросы на выборку с параметром** для их решения.

7. Сформулировать задачу и создать **запрос с вычисляемым полем** для ее решения.

8. Сформулировать задачи (не менее двух) для статистической обработки данных из таблиц и создать **итоговые запросы** для их решения.

9. Создать копию одной из таблиц базы данных.

10. Сформулировать задачу и создать не менее одного **запроса на изменение данных** в созданной копии таблицы (удаление или обновление данных).

Примечание. Вопросы компьютерной реализации базы данных с помощью СУБД MS Access были подробно рассмотрены при описании технологии выполнения практических работ в рамках данного пособия.

2.6. Тестовые задания

1. База данных – это...
 - a) именованная совокупность данных на внешнем носителе информации;
 - b) программа, используемая для обработки информации, структурированной в виде таблиц;
 - c) именованная совокупность структурированных данных, относящихся к определенной предметной области;
 - d) программа, используемая для поиска информации.
2. К классическим моделям данных относятся:
 - a) реляционная;
 - b) объектно ориентированная;
 - c) иерархическая;

- d) постреляционная;
 - e) многомерная;
 - f) сетевая.
3. В реляционной базе данных информация структурирована в виде...
- a) узлов многоуровневой сети;
 - b) иерархических древовидных структур;
 - c) списков;
 - d) двумерных таблиц.
4. Столбец таблицы реляционной базы данных называется...
- a) шаблоном;
 - b) полем;
 - c) структурой;
 - d) записью.
5. Строка таблицы реляционной базы данных называется...
- a) шаблоном;
 - b) полем;
 - c) структурой;
 - d) записью.
6. База данных Microsoft Access сохраняется в файле с расширением...
- a) .XLS
 - b) .MDB
 - c) .LNK
 - d) .PPT
7. Для отбора данных из одной или нескольких таблиц в соответствии с заданными критериями отбора в базе данных MS Access используются объекты...
- a) формы;
 - b) модули;
 - c) отчеты;
 - d) таблицы;
 - e) запросы;
 - f) страницы.

8. Изменить ранее созданный объект базы данных MS Access можно в режиме...

- a) конструктора;
- b) мастера;
- c) автоматическом;
- d) форматирования.

9. Окно «Схема данных» в программе MS Access используется для...

- a) вывода на экран содержимого активной таблицы;
- b) изменения структуры таблиц;
- c) просмотра и установки связей между таблицами базы данных;
- d) вывода на экран списка имен полей активной таблицы.

10. Проектируется таблица «СТУДЕНТЫ» базы данных MS Access. Для хранения в таблице данных о фамилиях студентов следует определить поле типа...

- a) логический;
- b) объект OLE;
- c) счетчик;
- d) текстовый;
- e) числовой;
- f) дата/время.

11. Результирующая таблица запроса на выборку, показанного на рис. 2.52, будет содержать сведения о сотрудниках, которые...

- a) работают в должности «инженер» и при этом имеют оклад 5000 или 8000;
- b) работают в должности «инженер» или имеют оклад 5000 или 8000 (или то и другое одновременно);
- c) работают в должности «инженер» и при этом имеют оклад в диапазоне от 5000 до 8000;
- d) работают в должности «инженер» или имеют оклад в диапазоне от 5000 до 8000 (или то и другое одновременно).

Поле:	Фамилия	Должность	Оклад
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"инженер"	Between 5000 and 8000
или:			

Рис. 2.52. Запрос в режиме конструктора для тестового задания 11

12. Структура таблицы реляционной базы данных изменится, если...

- а) добавить/удалить поле;
- б) добавить/удалить запись;
- с) поменять местами записи;
- д) поменять местами поля.

3. ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ. ЗАЩИТА ИНФОРМАЦИИ В СЕТЯХ

Изучив материалы этого раздела, вы **узнаете**:

- понятие компьютерной сети и разновидности компьютерных сетей;
- основные протоколы обмена данными, типы сетевых соединений и основные службы Интернета;
- структуру web-страницы и основы языка HTML;
- виды угроз информации и способы ее защиты.

Выполнив практические работы, вы **научитесь**:

- создавать простейшие web-страницы;
- осуществлять защиту документов средствами пакета Microsoft Office.

Методические рекомендации

Схема работы с учебными материалами данного раздела представлена на рис. 3.1. Теоретическая часть предполагает изучение четырех тем. После каждой темы дается перечень вопросов и упражнений для самоконтроля. Практическая часть содержит четыре практические работы и индивидуальное домашнее задание. Изучение раздела заканчивается выполнением тестовых заданий, ключ к которым можно найти в прил. 2. С дополнительными учебными материалами по данному разделу можно ознакомиться в рекомендованных литературных источниках [1], [2], [4], [5], [9], [10] и интернет-ресурсах [11], [14], список которых представлен в конце пособия.



Рис. 3.1. Схема работы с учебными материалами раздела 3

3.1. Компьютерные сети

Понятие, классификация и виды компьютерных сетей

Компьютерные сети (вычислительные, сети передачи данных) позволяют получать и передавать данные, решать задачи общения, совместно использовать компьютерные ресурсы. **Компьютерная сеть** – это комплекс аппаратных и программных средств, позволяющих осуществлять связь между компьютерами.

Несколько компьютеров, имеющих сетевые платы и расположенных на небольшом удалении друг от друга, соединяют специальными кабелями, и таким образом формируется **локальная компьютерная сеть** (англ. Local Area Network, LAN). Такие сети широко используются на мелких и средних предприятиях. Компьютеры, удаленные на большие расстояния, соединяет **глобальная компьютерная сеть** или

сеть дальней связи (англ. Wide Area Network, WAN). **Городская вычислительная сеть** (англ. Metropolitan Area Network, MAN) объединяет компьютеры в пределах города, представляет сеть по размерам меньшую, чем WAN, но большую, чем LAN. Сети могут соединяться между собой.

Одной из характеристик локальной вычислительной сети является топология сети. **Топология сети** – геометрическая форма и физическое расположение компьютеров по отношению друг к другу. Топология сети позволяет сравнивать и классифицировать различные сети. Различают следующие основные виды топологии: звезда, кольцо, шина.

При построении сети по **шинной схеме** каждый компьютер присоединяется к общему кабелю (рис. 3.2). Данные в виде электрических сигналов передаются всем компьютерам сети, но информацию принимает только тот, адрес которого соответствует адресу получателя, зашифрованному в этих сигналах. Причем в каждый момент времени только один компьютер может вести передачу. После того как данные достигнут адресата, электрические сигналы необходимо погасить. С этой целью на каждом конце кабеля устанавливают **терминаторы**, поглощающие эти сигналы. Шина – пассивная топология. Это значит, что компьютеры только «слушают» передаваемые по сети данные, но не перемещают их от отправителя к получателю. Поэтому если один из компьютеров выйдет из строя, это не скажется на работе остальных.

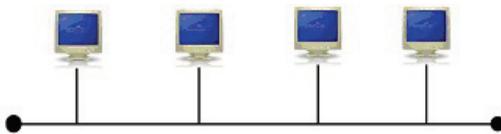


Рис. 3.2. Топология «общая шина»

Топология **«кольцо»** представляет последовательное соединение компьютеров, когда последний соединен с первым (рис. 3.3,а). Сигнал проходит по кольцу от компьютера к компьютеру в одном направлении. Каждый компьютер работает как повторитель, усиливая сигнал и передавая его дальше.

Поскольку сигнал проходит через каждый компьютер, сбой одного из них приводит к нарушению работы всей сети.

Топология «звезда» (рис. 3.3,б) – схема соединения, при которой каждый компьютер подсоединяется к сети при помощи отдельного соединительного кабеля. Один конец кабеля соединяется с гнездом сетевого адаптера, другой подсоединяется к центральному устройству, называемому концентратором.

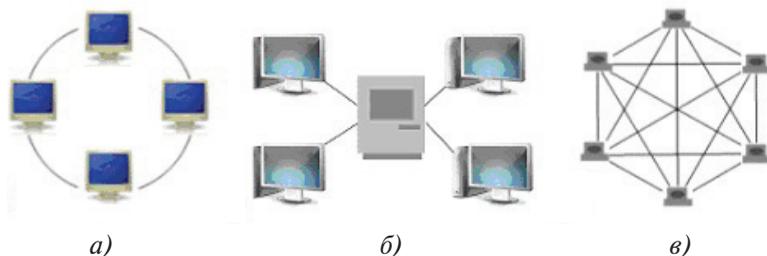


Рис. 3.3. Топология «кольцо» (а), «звезда» (б), «полносвязная» (в)

Полносвязная топология (рис. 3.3,в) – топология компьютерной сети, в которой каждая рабочая станция подключена ко всем остальным. Этот вариант является громоздким и неэффективным, несмотря на свою логическую простоту. Для каждой пары должна быть выделена независимая линия, каждый компьютер должен иметь столько коммуникационных портов, сколько компьютеров в сети. По этим причинам сеть может иметь только сравнительно небольшие размеры.

Смешанная топология – топология, преобладающая в крупных сетях с произвольными связями между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со смешанной топологией.

По административным отношениям между узлами можно выделить локальные сети с централизованным управлением (или с выделенными **серверами**) и сети без централизованного управления – так называемые **одноранжевые** сети.

В локальных сетях с централизованным управлением один из компьютеров является сервером, а остальные

компьютеры – рабочими станциями. **Рабочая станция** – это компьютер, подключенный к сети, через который пользователь получает доступ к сетевым ресурсам. Рабочая станция функционирует как в сетевом, так и локальном режиме и обеспечивает пользователя всем необходимым инструментарием для решения прикладных задач. **Сервер** – это компьютер, выполняющий функции управления сетевыми ресурсами общего доступа: осуществляет хранение данных, управляет базами данных, выполняет удаленную обработку заданий, обеспечивает печать заданий и др.

Аппаратные компоненты компьютерных сетей

Для передачи информации между различными устройствами локальной вычислительной сети используются различные типы **каналов связи**. В локальных вычислительных сетях в качестве передающей среды используются витая пара проводов, коаксиальный и оптоволоконный кабели.

Витая пара состоит из восьми изолированных проводов, свитых по два между собой. Скручивание проводов уменьшает влияние внешних электромагнитных полей на передаваемые сигналы. Невысокая стоимость и небольшая масса этого вида передающей среды делают ее достаточно популярной для локальных вычислительных сетей. Основные недостатки витой пары – плохая помехозащищенность, сравнительно низкая скорость передачи информации, простота несанкционированного подключения, ограничения на количество станций в сети.

В **коаксиальном кабеле** оба проводника тока, образующие электрическую цепь, представляют собой два цилиндра, расположенных вдоль одной оси, разделенных слоем изоляции. Так как внешний проводник одновременно служит электромагнитным экраном, защищающим электрическую цепь тока от влияний извне, коаксиальный кабель обладает высокой помехозащищенностью.

Оптоволоконный кабель состоит из световодов, выполненных из кварцевого стекла толщиной в несколько микрон, помещенных в изоляционное покрытие. Имеет высокую скорость передачи информации, не подвержен действию электромагнитных полей, полностью пожаро- и взрывобезопасен,

практически не имеет излучения. Каждое стеклянное оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон. Имеет следующие недостатки: высокая стоимость, необходимость иметь дополнительное оборудование для преобразования световых сигналов в электрические.

Сетевая плата – периферийное устройство, позволяющее компьютеру взаимодействовать с другими устройствами сети. В настоящее время сетевые платы интегрированы в материнские платы для удобства и удешевления всего компьютера в целом.

Для объединения компьютеров в локальную сеть используются концентраторы и коммутаторы. **Концентратор** имеет несколько портов для подключения компьютеров. После подключения компьютеров к концентратору именно он отвечает за доставку сетевого трафика для этих компьютеров. Когда компьютер отправляет сетевой пакет через концентратор, тот дублирует полученный пакет на все имеющиеся порты. Это означает, что все компьютеры, подключенные к концентратору, могут «видеть» этот пакет, хотя получателем пакета является только один компьютер. Однако на полученный сигнал реагирует только один компьютер, которому адресован пакет. Концентратор, имеющий всего два порта, называют **мостом**.

Коммутатор, в отличие от концентраторов, может распознать, откуда и куда посылается пакет. Когда компьютер отправляет сетевой пакет через коммутатор, пакет будет получен только его «адресатом». Поскольку коммутатор отправляет пакеты только одному получателю, он не занимает линию отправкой таких пакетов всем сетевым устройствам. Это повышает эффективность работы и позволяет справляться с высокими сетевыми нагрузками. Коммутаторы также обеспечивают высокую безопасность, поскольку отправляемый пакет сможет «увидеть» только один сетевой компьютер, являющийся его получателем.

Маршрутизатор позволяет соединить две или несколько сетей между собой. Маршрутизатор имеет как минимум два разъема. К одному из них подключается концентратор или

коммутатор одной сети, а ко второму – другая сеть. Если один из сетевых компьютеров связывается с другой сетью, он посылает соответствующий запрос на маршрутизатор. Маршрутизатор пересылает этот запрос на другую сеть. Если адресат не находится в этой другой сети, следующий маршрутизатор пересылает запрос на следующую сеть и т. д. до тех пор, пока запрос наконец не будет получен адресатом.

Шлюз (англ. gateway) используется для соединения сетей. Чаще всего это выделенный компьютер, на котором установлено специальное программное обеспечение, предназначенное для передачи информации между двумя сетями, использующими разные стандарты. По сути, действие программы сводится к переводу с языка одного стандарта на язык другого. Таким образом обеспечивается совместимость сетей и возможность передачи данных из одной сети в другую.

Модель взаимосвязи открытых систем

Основная задача, решаемая при создании компьютерных сетей, – совместимость оборудования и информационного обеспечения (программ и данных) образующих сеть компонентов. Одним из примеров решения данной задачи является **модель взаимосвязи открытых систем OSI** (англ. *Model of Open System Interconnections*).

Согласно модели OSI архитектуру компьютерных сетей следует рассматривать на разных уровнях. Обмен данными в системах связи происходит путем их перемещения с верхнего уровня на нижний, затем транспортировки и, наконец, обратным воспроизведением на компьютере клиента в результате перемещения с нижнего уровня на верхний. Обработка данных на каждом уровне выполняется в соответствии с сетевыми протоколами. **Сетевой протокол** – совокупность правил, регулирующих порядок функционирования сети.

На **прикладном** уровне (верхний уровень) с помощью специальных приложений пользователь создает документ. На уровне **представления** операционная система фиксирует местоположение созданных данных (в оперативной памяти, в файле на жестком диске и т. п.) и обеспечивает взаимодействие со следующим уровнем.

На *сеансовом* уровне компьютер пользователя взаимодействует с сетью. Протоколы этого уровня проверяют права пользователя на доступ к сети и передают документ на следующий уровень. На *транспортном* уровне документ преобразуется в форму, соответствующую передаче данных в используемой сети. Например, он может разбиваться на небольшие пакеты.

На *сетевом* уровне определяется маршрут движения данных в сети. Так, если на транспортном уровне данные были разбиты на пакеты, то на сетевом уровне каждый пакет должен получить адрес доставки. На уровне *соединения* данные преобразуются в сигналы, циркулирующие на физическом уровне. В компьютере эти функции выполняет сетевая карта или модем. Реальная передача данных происходит на *физическом* уровне (нижний уровень). Восстановление документа происходит на компьютере клиента при переходе с нижнего на верхний уровень модели OSI.

? Вопросы и упражнения для самоконтроля

1. Что такое компьютерная сеть? Назовите известные вам виды сетей.
2. Что такое сервер сети, рабочая станция?
3. Что такое «топология сети»? Опишите виды сетевых топологий.
4. Как называется множество компьютеров, связанных каналами передачи информации и находящихся в пределах одного здания?
5. Опишите виды каналов связи в компьютерных сетях.
6. Каково назначение сетевой платы компьютера?
7. Какую функцию выполняют концентраторы и коммутаторы в компьютерных сетях?
8. В чем заключается различие в принципах действия концентратора и коммутатора?
9. Каковы назначение и принцип действия маршрутизатора?
10. Опишите функции шлюза при построении компьютерных сетей.
11. Каково назначение и структура модели взаимосвязи открытых систем OSI?

3.2. Интернет

Организация сети Интернет

Интернет называют сетью сетей, так как это общедоступное объединение многих компьютерных сетей с установленными правилами обмена информацией (протоколами).

TCP/IP – это единый набор протоколов передачи данных, который должна использовать каждая сеть, если она хочет войти в сообщество интернет-сетей. В названии TCP/IP отражены имена двух главных сетевых протоколов. Каждый подключенный к Интернету компьютер должен быть оснащен программным обеспечением TCP/IP.

TCP (*Transmission Control Protocol*) – протокол контроля передачи данных. Он обеспечивает надежность передачи данных. Интернет построен так, что при пересылке данные делятся на части – пакеты. Пакеты следуют к месту назначения различными маршрутами и прибывают в конечную точку в другом порядке, нежели отправлялись. Они могут теряться и дублироваться. Протокол TCP устраняет все возникающие проблемы и обеспечивает сборку всех пакетов в единое сообщение.

IP (*Internet Protocol*) – межсетевой протокол (протокол маршрутизации, транспортный протокол). Определяет основные правила, которым должны следовать компьютеры для обмена данными: формат пакетов, формат адресов компьютеров сети, маршрут пакета, правила обработки пакетов маршрутизаторами и компьютерами сети.

Провайдеры (англ. *Internet Service Provider*) – фирмы, специализирующиеся на предоставлении доступа к Интернету и подключающие пользователей к сети.

Коммутируемый удаленный доступ (англ. *Dial-up*) – сервис, позволяющий компьютеру посредством модема и телефонной сети общего пользования подключаться к другому компьютеру для доступа в сеть Интернет. Используется телефонная линия (при соединении с Интернетом линия занята). **Модем** (аббревиатура, составленная из слов «модулятор – демодулятор») – устройство, применяющееся в системах связи и выполняющее функцию преобразования сигналов.

С помощью модема персональный компьютер подключается к сети Интернет. При подключении посредством телефонной линии модем преобразует цифровые сигналы в аналоговые (при передаче) и обратно (при приеме).

Технология **ADSL** применяется для высокоскоростного доступа к сети Интернет на абонентских линиях телефонной сети. Технология ADSL использует метод разделения полосы пропускания телефонной линии на несколько частотных полос. Это позволяет одновременно передавать несколько сигналов по одной линии (при соединении с Интернетом телефон остается доступным для использования). Наиболее высокую скорость соединения и обмена данными обеспечивает доступ к Интернету посредством *выделенной линии*.

Каждому компьютеру, подключенному к сети Интернет, присваивается уникальный номер, который называют **IP-адресом**. Это комбинация из четырех десятичных чисел (каждое не более 255), разделенных точками. Например: 195.34.32.11. По IP-адресам компьютеры связываются друг с другом. Компьютер, который подключен к Интернету постоянно, называется *хостом*. IP-адрес хоста всегда постоянный. IP-адрес компьютера с сеансовым способом подключения является динамическим. Такому компьютеру адрес присваивается при входе в сеть из числа тех, что имеются в распоряжении провайдера.

Однако чаще пользователи Интернета применяют более удобную адресацию, которая называется системой имен доменов **DNS** (англ. *Domain Name System, DNS*). **Домен** (англ. *Domain* — область) — это группа хостов, объединенная по определенному признаку и имеющая одно имя. Система доменных имен многоуровневая. Домены первого уровня формируются по территориальному (**ru** — Россия) или функциональному (**edu** — образовательные) признаку. Домены второго уровня группируют хосты по территории (**msk** — Москва) или по принадлежности одной организации.

В доменном адресе имена доменов записываются через точку по возрастанию уровня. Например, если один из компьютеров тольяттинской фирмы «Аист» назван «mail», его доменное имя может быть **mail.aist.tlt.ru**.

Сервисы Интернета

Интернет предоставляет пользователю различные виды услуг. *Электронная почта* является основным видом сетевых услуг. Обмен сообщениями реализуется через систему почтовых серверов. *Telnet* – служба, обеспечивающая доступ к ресурсам удаленных компьютеров сети Интернет. *Gopher* – служба, обеспечивающая доступ к информации через систему связанных текстовых меню.

FTP (англ. *File Transfer Protocol* – протокол передачи файлов) – служба, которую используют для передачи файлов от одного компьютера к другому через Интернет. Интерактивное общение в сети реализуется службой обмена информацией в режиме реального времени и основано на протоколе *IRC* (англ. *Internet Relay Chat*). *Телеконференции* – службы, которые организуют тематический обмен информацией между пользователями сети.

Всемирная паутина (англ. *World Wide Web, WWW*) – служба, которая организует хранение разнообразной информации в виде распределенной по компьютерам сети системы взаимосвязанных документов (*web-страниц*). Связь между документами осуществляется при помощи *гиперссылок*. *Гипертекст* – это способ структурирования документов путем размещения гиперссылок внутри одного документа или между документами. В качестве ссылки могут использоваться как фрагменты текста, так и графические изображения.

Web-страницы – это файлы, создаваемые с использованием языка *HTML* (англ. *Hyper Text Markup Language* – язык гипертекстовой разметки документов). Документ формата HTML может содержать помимо текста, графики и гиперссылок еще и мультимедийные объекты (анимацию, видео, звук). В этом случае говорят, что он строится на основе *технологии гипермедиа*.

Набор web-страниц, который является собственностью какой-либо компании или частного лица, называют *web-сайтом* или просто сайтом. Создавать web-страницы можно как в HTML-редакторах (FrontPage), так и в текстовых редакторах разного уровня (Word, Блокнот). *Тегами* называются команды языка HTML. Чтобы отличить теги от текста

документа, их заключают в угловые скобки. Например, тег **
** означает команду «начало новой строки». Теги могут быть как одиночными, так и парными. Парные теги указывают начало и окончание действия команды. Тег, который указывает на окончание, помечается символом слеш (/).

Рисунки и другие объекты web-страницы хранятся отдельно от HTML-кода. HTML-код содержит команды, которые сообщают браузеру, где хранится объект и в какой области окна он должен быть отображен. Web-страница, как правило, хранится в виде файла формата HTML и папки с объектами.

Для просмотра HTML-документов существуют специальные программы, которые называются *браузеры*. Они интерпретируют теги разметки документа и располагают текст и графику на экране соответствующим образом.

URL (Uniform Resource Locator) – это адрес в системе WWW, при помощи которого однозначно определяется каждый документ. Он используется для записи гиперссылок и для обеспечения доступа к распределенным ресурсам сети Интернет. URL-адрес состоит из трех частей. В первой части указывается протокол, который будет использоваться для передачи данных, во второй части – адрес сервера, в третьей – путь к ресурсу на сервере. Синтаксис URL:

Название протокола: //название_сервера/путь_к_ресурсу

Например, *http://myserver.com/dir1/dir2/dir3/main.html* – URL веб-страницы, сохраненной в файле *main.html* на компьютере-сервере с именем *myserver*, принадлежащем к домену первого уровня *com*, путь к файлу от корневой папки – *dir1/dir2/dir3*, передача страницы выполняется согласно протоколу *http*.

Каждая служба Интернета имеет свой протокол передачи данных. Названия протоколов служб FTP, Telnet и Gopher совпадают с именами служб. Протокол службы WWW называется *Hypertext Transport Protocol* и записывается в URL в виде сокращения *http*.

Современную концепцию развития Интернета принято называть *Веб 2.0* (Web 2.0). Принципиальным отличием Веб 2.0 от традиционной сети является возможность создавать

содержимое Интернета любому пользователю. Сервисы Веб 2.0, или социальные сетевые сервисы – это сетевое программное обеспечение, поддерживающее групповые взаимодействия.

Социальная сеть – это виртуальная сеть, являющаяся средством обеспечения сервисов, связанных с установлением связей между пользователями, а также соответствующими их интересам информационными ресурсами, установленными на сайтах Интернета.

Социальные сервисы представляют собой онлайн-инструменты, с помощью которых пользователи могут не только общаться между собой, но и сами формировать содержание web-страниц. Web-сервис, который позволяет пользователям систематизировать ссылки с описаниями, снабжая их поисковыми критериями, называется **социальными закладками**. С помощью таких сервисов можно хранить закладки (избранное) в Интернете, создавать, делиться и обмениваться закладками и тегами с друзьями и знакомыми, сортировать найденную информацию по темам.

Вики (англ. *Wiki*) – это новая технология коллективного создания сайтов, в рамках которой изменения доступны не только создателю сайта, но и всем желающим. **Википедия** (*Wikipedia*) – социальная система, ориентированная на подготовку энциклопедических статей о любом понятии, которое пользователи сочтут нужным. Основных правил два: статья должна быть нейтральной и объективной, а ее содержание – проверяемым, то есть основываться на уже опубликованных материалах.

Еще один сервис, наращивающий популярность, – служба **«облачных» вычислений**. Отличием от привычного метода работы с программами является то, что пользователь использует ресурсы не своего компьютера, а предоставляемые ему интернет-сервисом. Пользователь имеет полный доступ к собственным данным и возможность работы с ними, но не может управлять операционной системой, программной базой, вычислительными мощностями и прочими ресурсами, с помощью которых эта работа происходит.

Данный подход имеет ряд преимуществ:

- пользователь может задействовать компьютер практически любой конфигурации для выполнения ресурсоемких задач;
- пользователь не привязан к месту работы и может использовать любой компьютер, имеющий доступ в сеть Интернет;
- пользователь застрахован от сбоев в работе и может легко делиться результатами работы с другими либо вести совместную работу.

Дополнительное преимущество заключается в том, что «облачные» сервисы зачастую либо бесплатны, либо имеют довольно низкую стоимость. Также безусловным плюсом является снижение издержек на обслуживание, поддержку, модернизацию и администрирование аппаратного и программного обеспечения.

? Вопросы и упражнения для самоконтроля

1. Что такое IP-адрес?
2. Какой протокол используется для передачи в сети web-страниц?
3. Каково назначение протоколов *FTP*, *IRC*?
4. Какие службы существуют в сети Интернет?
5. Как называется унифицированная форма записи адресов документов в сети Интернет?
6. Какая часть электронного адреса ресурса *http://www.google.com/inf02000/01_02_05/det123.html* описывает сетевой протокол?
7. Что обозначает выделенная курсивом часть электронного адреса ресурса *http://www.google.com/inf02000/det123.html*?
8. Что такое доменное имя? Для чего необходима служба Domain Name System?
9. Что такое гипертекст, гиперссылка? Что такое HTML?
10. Что такое WWW, Web-сайт?
11. Опишите понятия «социальная сеть», «социальный сервис». Назовите известные вам социальные сервисы.

3.3. Организация защиты информации

Угрозы безопасности информации, их виды

Развитие средств, методов и форм автоматизации информационных процессов, массовое применение вычислительной техники делают информацию уязвимой. Информация может быть незаконно изменена, похищена или уничтожена. Защита информации включает решение следующих проблем:

- обеспечение физической целостности информации, т. е. предотвращение уничтожения элементов информации;
- предотвращение изменения (модификации) элементов информации при сохранении ее целостности;
- предотвращение несанкционированного получения информации лицами (процессами), не имеющими на это соответствующих полномочий;
- использование передаваемых данных только в соответствии с оговоренными сторонами условиями.

Функционирование компьютерных технологий выявило наличие слабых мест (*угроз*) в защите информации. Угрозы безопасности делятся на случайные (непреднамеренные) и умышленные (преднамеренные).

Источником *случайных угроз* могут быть [5]:

- отказы и сбои аппаратных средств;
- помехи в каналах и на линиях связи;
- форс-мажорные ситуации (пожар, выход из строя электропитания и т. д.);
- ошибки и просчеты разработчиков и производителей технических средств;
- алгоритмические и программные ошибки;
- неумышленные действия пользователей, приводящие к отказу технологии или разрушению аппаратных, программных, информационных ресурсов (неумышленная порча оборудования, удаление, искажение файлов с важной информацией или программ и т. д.);
- запуск программ, способных при некомпетентном использовании вызывать необратимые изменения (форматирование или реструктуризация носителей информации, удаление данных и т. д.);

- заражение компьютерными вирусами;
- неосторожные действия, приводящие к разглашению конфиденциальной информации или атрибутов разграничения доступа (паролей, ключей шифрования и т. д.).

Умышленные угрозы – активное воздействие человека на объекты и процессы с целью умышленной дезорганизации функционирования информационной технологии, вывода ее из строя, проникновения в систему и несанкционированного доступа к информации.

Умышленные угрозы бывают двух видов: пассивные и активные. **Пассивные угрозы** направлены на несанкционированное использование информационных ресурсов, не оказывая при этом влияния на функционирование информационной технологии. К пассивной угрозе относится, например, попытка получения информации, циркулирующей в каналах связи, посредством их прослушивания. **Активные угрозы** имеют целью нарушение нормального функционирования информационной технологии посредством целенаправленного воздействия на аппаратные, программные и информационные ресурсы. К активным угрозам можно отнести разрушение или радиоэлектронное подавление каналов связи, искажение сведений в базах данных и т. д.

В целом можно выделить следующие **умышленные угрозы** безопасности информации [5]:

- **раскрытие конфиденциальной информации** – бесконтрольный выход информации за пределы круга лиц, для которых она предназначена;
- **несанкционированный доступ к информации** – противоправное преднамеренное овладение конфиденциальной информацией лицом, не имеющим права доступа к охраняемым сведениям;
- **компрометация информации** – факт доступа посторонних лиц к защищаемой информации, а также подозрение на него;
- **отказ от информации** – непризнание получателем или отправителем информации фактов ее получения или отправки;

- *нарушение информационного обслуживания* — задержка в предоставлении информационных ресурсов потребителю;
- *«взлом системы»* — умышленное проникновение в информационную технологию, когда взломщик не имеет санкционированных параметров для входа.

Методы и средства обеспечения безопасности информации

Под *политикой безопасности* понимают совокупность документированных решений, принимаемых руководством организации и направленных на защиту информации. Политика безопасности описывает цели без конкретизации того, как они должны быть достигнуты. *Модель безопасности* реализует принятую в информационной системе политику безопасности, преобразуя абстрактные цели политики в термины информационных систем, точно описывая структуры данных, средства и методы, необходимые для реализации политики безопасности.

Например, если политика безопасности утверждает, что субъекты должны быть авторизованы для доступа к объектам, модель безопасности должна предоставить математические соотношения и формулы, объясняющие, как субъект *X* может получить доступ к объекту *Y* посредством определенных методов. Затем разрабатываются технические требования, согласно которым создаются компьютерные программы для реализации механизмов управления. Эти механизмы представляют сетевому администратору средства настройки разграничения доступа в рамках операционной системы.

К *методам обеспечения безопасности информации* относятся препятствие, управление доступом, маскировка, регламентация, принуждение, побуждение (рис. 3.4).

Препятствие — метод физического преграждения пути злоумышленнику к защищаемой информации (к аппаратуре, носителям информации и т. д.).

Управление доступом включает следующие функции защиты [5]:

- *идентификация специалистов, персонала и ресурсов* — присвоение каждому объекту персонального идентификатора;

- **аутентификация** – опознание объекта или субъекта по предъявленному им идентификатору (например, проверка подлинности пользователя путём сравнения введённого им пароля с паролем в базе данных пользователей);
- **проверка полномочий** – соответствие дня недели, времени суток запрашиваемых ресурсов и процедур установленному регламенту;
- **протоколирование (регистрация)** обращений к защищаемым ресурсам;
- **реагирование (сигнализация, отключение, задержка работ, отказ в запросе)** при попытке несанкционированных действий.

Маскировка – метод защиты информации путем ее криптографического закрытия. Этот метод широко применяется как при обработке, так и при хранении информации. При передаче информации по каналам связи большой протяженности данный метод является единственно надежным.

Регламентация – метод защиты информации, создающий по регламенту такие условия автоматизированной обработки, хранения и передачи защищаемой информации, при которых возможности несанкционированного доступа к ней сводились бы к минимуму.

Принуждение – метод защиты, при котором специалисты и технический персонал вынуждены соблюдать правила обработки, передачи и использования защищаемой информации под угрозой материальной, административной или уголовной ответственности.

Побуждение – метод защиты, побуждающий специалистов и технический персонал не разрушать установленные порядки за счет соблюдения сложившихся моральных и этических норм.

Рассмотренные методы обеспечения безопасности реализуются на практике за счет применения различных средств защиты (рис. 3.4). **Формальные средства защиты** – это средства, выполняющие защитные функции строго по заранее предусмотренной процедуре без непосредственного участия человека. **Неформальные средства защиты** – это средства

защиты, которые определяются целенаправленной деятельностью человека либо регламентируют эту деятельность.



Рис. 3.4. Методы и средства защиты информации

К основным формальным средствам защиты относятся технические и программные средства. *Технические средства* реализуются в виде электрических, электромеханических и электронных устройств. Все технические средства делятся на следующие виды:

- *аппаратные*, представляющие собой устройства, встраиваемые непосредственно в вычислительную технику, или устройства, которые сопрягаются с подобной аппаратурой по стандартному интерфейсу;
- *физические*, представляющие собой автономные устройства и системы, создающие физические препятствия для злоумышленников (замки, решетки, охранная сигнализация и т. д.).

Программные средства – это программное обеспечение, специально предназначенное для выполнения функций защиты информации.

К основным неформальным средствам защиты относятся:

- **организационные средства**, которые представляют собой организационно-технические и организационно-правовые мероприятия, осуществляемые в процессе создания и эксплуатации вычислительной техники, аппаратуры и телекоммуникаций;
- **морально-этические средства**, которые реализуются в виде всевозможных норм, сложившихся традиционно или складывающихся по мере распространения вычислительной техники и средств связи;
- **законодательные средства**, которые определяются законодательными актами страны (регламентируют правила пользования, обработки и передачи информации ограниченного доступа и устанавливают меры ответственности за нарушения этих правил).

? Вопросы и упражнения для самоконтроля

1. Какие проблемы должны быть решены в процессе обеспечения защиты информации?
2. Назовите источники случайных и умышленных угроз безопасности информации.
3. Что такое модель безопасности, политика безопасности?
4. Назовите основные угрозы безопасности информации.
5. Как классифицируются угрозы безопасности?
6. Назовите и охарактеризуйте методы обеспечения безопасности информации.
7. Перечислите и охарактеризуйте функции управления доступом как метода обеспечения безопасности информации.
8. Как классифицируют средства обеспечения безопасности информации?
9. Опишите основные средства обеспечения безопасности информации.

3.4. Антивирусная защита компьютера. Защита информации в локальных и глобальных компьютерных сетях

3.4.1. Антивирусная защита компьютера

Вредоносная программа — это термин для обозначения любого программного обеспечения, специально созданного для того, чтобы причинять ущерб отдельному компьютеру или компьютерной сети. Рассмотрим основные разновидности вредоносных программ [4].

Компьютерный вирус — программа, способная создавать свои копии (необязательно совпадающие с оригиналом) и внедрять их в файлы, системные области компьютера, а также осуществлять иные деструктивные действия. При этом копии сохраняют способность дальнейшего распространения.

Логическая бомба — это программа или фрагмент кода в программе, реализующий некоторую функцию при выполнении определенного условия, например, условием может быть наступление заданной даты. «Взрываясь», логическая бомба реализует нежелательную для пользователя функцию, например, удаляет некоторые данные.

Троянский конь — программа, выполняющая в дополнение к основным еще и дополнительные действия, не описанные в документации. Троянский конь — это дополнительный блок команд, тем или иным образом вставленный в исходную безвредную программу. Троянский конь обычно действует в рамках полномочий одного пользователя, но в интересах другого пользователя (злоумышленника).

Червь (сетевой червь) — тип вредоносных программ, распространяющихся в компьютерной сети, способных к преодолению систем защиты, а также к созданию и дальнейшему распространению своих копий и осуществлению иных вредоносных действий.

Захватчик паролей — это программа, специально разработанная для воровства паролей. Сценарий может быть следующим. Программа выводит на экран сообщение об окончании сеанса работы, а затем — запрос на ввод логина и пароля для входа в систему. Введенные пользователем

данные пересылаются владельцу программы-захватчика. Для предотвращения этой угрозы перед вводом запрашиваемых данных необходимо убедиться, что вы вводите имя и пароль именно системной программе, а не какой-нибудь другой.

Клавиатурный шпион (кейлоггер) – программное или аппаратное средство, основным назначением которого является скрытый мониторинг нажатий клавиш и ведение журнала этих нажатий. Кейлоггер безопасен для системы, однако он может быть очень опасным для пользователя: с помощью кейлоггера можно перехватить пароли и другую конфиденциальную информацию, вводимую пользователем с помощью клавиатуры. В результате злоумышленник узнает коды и номера счетов в электронных платежных системах, логины, пароли к системам электронной почты и т. д. Большинство антивирусных программ распознают известные кейлоггеры, и метод защиты от них не отличается от метода защиты от любого другого вредоносного программного обеспечения.

Условием, способствующим реализации многих видов угроз безопасности информации, является наличие в программном коде «люков». **Люк** – это не описанная в документации на программный продукт возможность работы с этим программным продуктом. В результате пользователь получает доступ к возможностям и данным, которые в обычных условиях для него закрыты. Люки чаще всего являются результатом забывчивости разработчиков. Например, в качестве люка может быть использован временный механизм прямого доступа к частям программы, созданный для облегчения процесса отладки и не удаленный по ее окончании. Защита от люков только одна – не допускать их появления в программе.

Виды компьютерных вирусов, их классификация

Известны десятки тысяч компьютерных вирусов, которые распространяются через Интернет по всему миру. Существует несколько классификаций компьютерных вирусов [4].

По среде обитания различают вирусы *сетевые* (обитают в компьютерных сетях), *файловые* (внедряются в основном в исполняемые файлы с расширениями.COM и.EXE), *загрузочные* (внедряются в загрузочный сектор диска или

в сектор, содержащий программу загрузки системного диска), **файлово-загрузочные** (поражают загрузочные секторы дисков и файлы прикладных программ) и **системные** (проникают в системные модули и драйверы периферийных устройств, таблицы размещения файлов и таблицы разделов).

По способу заражения выделяют вирусы **резидентные** (при заражении компьютера оставляют в оперативной памяти свою резидентную часть, которая затем перехватывает обращение операционной системы к другим объектам заражения, внедряется в них и выполняет свои разрушительные действия вплоть до выключения или перезагрузки компьютера) и **нерезидентные** (не заражают оперативную память и являются активными ограниченное время).

По степени воздействия вирусы бывают **безвредные** (не оказывающие разрушительного влияния на работу компьютера, но способные переполнять оперативную память в результате своего размножения), **неопасные** (не разрушают файлы, но уменьшают свободную дисковую память, выводят на экран графические эффекты, создают звуковые эффекты и т. д.), **опасные** (нередко приводят к различным серьезным нарушениям в работе) и **разрушительные** (приводят к стиранию информации, полному или частичному нарушению работы прикладных программ и пр.).

По особенностям алгоритмов вирусы делят на **паразитические** (изменяющие содержимое файла добавлением в него своего кода, при этом зараженная программа сохраняет полную или частичную работоспособность), **репликаторы** (благодаря своему быстрому воспроизводству приводят к переполнению основной памяти), **невидимки** (перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо себя незараженные объекты), **мутанты** (со временем видоизменяются и самовоспроизводятся, при этом воссоздают копии, которые отличаются от оригинала), **макровирусы** (используют возможности макроязыков, встроенных в офисные программы обработки данных — текстовые редакторы, электронные таблицы и т. д.).

Защита от компьютерных вирусов

Для защиты компьютера от вирусов разрабатывается и постоянно обновляется антивирусное программное обеспечение. Основным средством антивирусной защиты является периодическая проверка компьютера с помощью систематически обновляемого антивирусного программного обеспечения.

Ранние версии антивирусных программ по выполняемым ими функциям подразделялись на детекторы, фаги, ревизоры, фильтры и др. Рассмотрим особенности этих типов программ.

Программы-детекторы позволяют обнаруживать файлы, зараженные одним из нескольких известных вирусов. Эти программы проверяют, имеется ли в файлах на указанном пользователем диске специфическая для данного вируса комбинация байтов. При ее обнаружении в каком-либо файле на экран выводится соответствующее сообщение. Программа-детектор может обнаруживать только те вирусы, которые ей «известны».

Программы-полифаги (сканеры) не только находят зараженные вирусами файлы, но и «лечат» их, т. е. удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние. В начале работы полифаги сканируют оперативную память, обнаруживают вирусы и уничтожают их и только затем переходят к «лечению» файлов.

Программы-ревизоры (CRC-сканеры) используют для поиска вирусов метод обнаружения изменений. Принцип работы CRC-сканеров основан на подсчете CRC-сумм (кодов циклического контроля) для присутствующих на диске файлов и системных секторов.

Программы-фильтры (сторожа) — это небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов.

Программы-вакцины (иммунизаторы) — это резидентные программы, предотвращающие заражение файлов. Вакцины применяют, если отсутствуют программы-доктора, уничтожающие тот или иной вирус. Вакцинация возможна только от известных вирусов. Вакцина модифицирует программу

или диск таким образом, чтобы это не отражалось на их работе, а вирус будет воспринимать их зараженными и поэтому не внедрится.

Современные антивирусные пакеты программ объединяют все средства обнаружения и нейтрализации вирусов:

- **антивирусное ядро** – программные библиотеки поиска и устранения вредоносного кода;
- **вирусная база данных** – описания вредоносных программ и способов их устранения;
- **антивирусный сканер** – программа, выполняющая сканирование файлов на внешних носителях и позволяющая устранить обнаруженные вредоносные программы;
- **антивирусный сторож** – программа, проверяющая файлы при выполнении некоторых типовых операций (записи, запуска и т. д.);
- **специализированный фильтр** – программа, контролирующая сообщения, которыми пользователь обменивается с помощью сети.

Наиболее популярными антивирусными программами являются Антивирусная система Касперского, DrWEB и др.

Создание и распространение вредоносных программ преследуется в России согласно Уголовному кодексу РФ (глава 28, статья 273).

3.4.2. Защита информации в локальных и глобальных компьютерных сетях

Средства защиты информации от несанкционированного доступа

Защита информации от несанкционированного доступа и модификации призвана обеспечить решение одной из наиболее важных задач – защиту хранимой и обрабатываемой информации от всевозможных злоумышленных покушений, которые могут нанести существенный ущерб.

Требования по защите информации от несанкционированного доступа направлены на достижение трех основных свойств защищаемой информации:

- 1) засекреченная информация должна быть доступна только тому, кому она предназначена (**конфиденциальность**);

2) информация, на основе которой принимаются важные решения, должна быть достоверной, точной и должна быть защищена от возможных непреднамеренных и злоумышленных искажений (*целостность*);

3) информация и соответствующие информационные службы должны быть доступны, готовы к обслуживанию всегда, когда в них возникает необходимость (*доступность*).

Наиболее эффективный способ защиты при выходе в сеть Интернет обеспечивает установка системы *FireWall (файервол, межсетевой экран, брандмауэр)*. Это комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами. Основной задачей межсетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Примеры программных файерволов: Agnitum Outpost, Symantec Firewall, ZoneAlarm.

Криптографическая защита информации

Криптография – наука о способах преобразования (шифрования) информации с целью защиты ее от незаконного или нежелательного использования. Применяются различные типы шифрования.

Симметричное шифрование основывается на использовании одного и того же секретного ключа для шифрования и дешифрования. *Асимметричное шифрование* характеризуется тем, что для шифрования используется один ключ, являющийся общедоступным, а для дешифрования – другой, секретный. Знание общедоступного ключа не позволяет определить секретный ключ.

Наряду с шифрованием современная криптография включает асимметричные криптосистемы, системы *электронной цифровой подписи* (ЭЦП), хеш-функции, управление ключами, получение скрытой информации, квантовую криптографию.

Электронная цифровая подпись

Электронная цифровая подпись (ЭЦП) – реквизит электронного документа, позволяющий установить отсутствие

искажения информации в документе с момента формирования ЭЦП и проверить принадлежность подписи владельцу сертификата ключа ЭЦП.

ЭЦП – это небольшой объем информации, содержащий сжатый и зашифрованный образ электронного документа. Цифровая подпись предназначена для аутентификации лица, подписавшего электронный документ. Кроме того, использование цифровой подписи позволяет осуществить:

- **контроль целостности** передаваемого документа (при любом случайном или преднамеренном изменении документа подпись станет недействительной, потому что вычислена она на основании исходного состояния документа и соответствует лишь ему);
- **защиту от изменений (подделки) документа** – гарантия выявления подделки при контроле целостности делает подделывание нецелесообразным в большинстве случаев;
- **невозможность отказа от авторства** (так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец не может отказаться от своей подписи под документом);
- **доказательное подтверждение авторства документа** (так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец пары ключей может доказать свое авторство подписи под документом).

Механизм ЭЦП основывается на алгоритмах асимметричного шифрования и включает две процедуры: формирование подписи отправителем и ее опознание (верификацию) получателем. Первая процедура обеспечивает шифрование блока данных или его дополнение криптографической контрольной суммой, причем в обоих случаях используется секретный ключ отправителя. Вторая процедура основывается на использовании общедоступного ключа, знание которого достаточно для опознавания отправителя.

? Вопросы и упражнения для самоконтроля

1. Охарактеризуйте основные виды вредоносных программ.
2. Каким образом программные «люки» могут быть использованы для реализации угроз безопасности информации?
3. Перечислите основные типы антивирусных программ.
4. Опишите классификацию компьютерных вирусов по среде обитания.
5. Чем резидентные вирусы отличаются от нерезидентных?
6. Перечислите разновидности компьютерных вирусов в зависимости от реализуемого ими алгоритма действий?
7. Какие функции выполняют программы-детекторы и полифаги?
8. Каковы механизмы действия антивирусных программ-ревизоров и фильтров?
9. Сформулируйте цель защиты информации от несанкционированного доступа и модификации.
10. Определите понятия конфиденциальности, целостности и доступности информации.
11. Какова роль системы FireWall в организации безопасной работы в сети Интернет?
12. Чем симметричное шифрование отличается от несимметричного?
13. Каково назначение электронной цифровой подписи?
14. Каков механизм формирования электронной цифровой подписи?

3.5. Практические работы

Практическая работа 3.1. Создание простейшего HTML-документа. Форматирование шрифта и абзаца

Цель работы – научиться:

- создавать простейшие гипертекстовые документы средствами текстового редактора **Блокнот**;

- выполнять форматирование шрифта и абзаца в HTML-документе.

Порядок выполнения работы

☑ **Внимание!** Описание используемых в работе тегов приведено в прил. 5.

1. Создайте файл с гипертекстовым документом:
 - запустите редактор **Блокнот**, введите текст:
Приветствую Вас на моей первой web-страничке!
 - сохраните файл в созданной папке, при сохранении в окне диалога **Сохранить как...** в строке **Тип файла:** выберите вариант **Все файлы(*.*)**, а в строке **Имя файла** задайте имя с расширением **.htm**, например, **1_name.htm** (где **name** – ваше имя);
 - закройте документ, найдите его пиктограмму в окне **Мой компьютер** или в окне программы **Проводник**;
 - откройте файл, проанализируйте, *с помощью какого приложения отображается файл* и как выглядит введенная фраза.
2. Введите в текст документа теги, определяющие структуру html-документа:
 - откройте документ в программе **Блокнот**, введите приведенные ниже теги (между тегами **<TITLE>** **</TITLE>** укажите свою фамилию):
<HTML>
<HEAD> **<TITLE>** **Фамилия** **</TITLE>** **</HEAD>**
<BODY>
Приветствую Вас на моей первой web-страничке!
</BODY>
</HTML>
 - сохраните документ под тем же именем, обновите его отображение в браузере, проанализируйте произошедшие изменения в отображении документа.
3. Отредактируйте документ:
 - добавьте после текста **«Приветствую Вас на моей первой web-страничке!»** текст подписи:
Студент группы NNN Фамилия Имя
 - сохраните документ (но не закрывайте) и обновите его просмотр в браузере;

- используя одиночный тег **
, отредактируйте документ так, чтобы слова **Студент группы NNN располагались во второй строке, а **Фамилия Имя** – в третьей. Просмотрите в браузере новый вариант.

Внимание! После каждого изменения документ нужно сохранять, а просмотр в браузере начинать с обновления загрузки документа с помощью кнопки «Обновить» на панели инструментов.

4. Оформите фрагменты текста с помощью стилей **Заголовков**:

- первую строку документа оформите стилем **Заголовок 1-го уровня** с помощью парного тега **<H1> ...</H1>**. Вторую строку – стилем **Заголовок 6-го уровня** (тег **<H6> ...</H6>**), а третью – стилем **Заголовок 4-го уровня** (тег **<H4> ...</H4>**);
 - просмотрите документ в браузере.
5. Выполните форматирование шрифта:
- после строки **Фамилия Имя** добавьте еще одну любую строку текста;
 - измените оформление текста, используя **разные начертания шрифта** (жирный, курсивный, подчеркнутый и др.);
 - измените оформление текста так, чтобы было использовано **не менее трех цветов шрифта**;
 - просмотрите полученный документ в браузере.
6. Выполните форматирование абзацев:
- создайте новый документ **2_name.htm**, сохраните его в той же рабочей папке;

• введите текст:

```
<HTML>
```

```
<HEAD> <TITLE> Фамилия </TITLE> </HEAD>
```

```
<BODY>
```

```
Омар Хайям <BR>
```

```
Много лет размышлял я над жизнью земной. Непонятно-  
го нет для меня под луной. Мне известно, что мне ничего  
не известно! Вот последняя правда, открытая мной.
```

```
</BODY>
```

```
</HTML>
```

- выровняйте текст по ширине;
- вставьте горизонтальную разделительную линию, которая ограничит текст снизу (тег <HR>);
- скопируйте текст, выровняйте по центру и разбейте его на абзацы:

Много лет размышлял я над жизнью земной.

Непонятого нет для меня под луной.

Мне известно, что мне ничего не известно!

Вот последняя правда, открытая мной.

- сохраните документ, просмотрите его в окне браузера.
7. Выполните оформление списков:
- создайте новый документ **3_name.htm**, сохраните его в той же рабочей папке жесткого диска;
 - введите текст:

```
<HTML>
```

```
<HEAD> <TITLE> Фамилия </TITLE> </HEAD>
```

```
<BODY>
```

Основные параметры форматирования шрифта:

гарнитура,

начертание,

размер,

цвет.

```
</BODY>
```

```
</HTML>
```

- оформите четыре последние строки как **список нумерованный**, для этого используйте следующую конструкцию тегов:

```
<OL>
```

```
<LI> гарнитура, </LI>
```

```
<LI> начертание, </LI>
```

```
<LI> размер, </LI>
```

```
<LI> цвет, </LI>
```

```
</OL>
```

- скопируйте последнюю конструкцию тегов, поменяйте оформление списка нумерованного на **список маркированный**, используя теги , .

8. Сохраните документ, предъявите результат работы преподавателю.

Практическая работа 3.2. Вставка в HTML-документ рисунков. Создание гиперссылок

Цель работы – научиться:

- выполнять вставку рисунков в HTML-документ;
- создавать гиперссылки;
- использовать таблицы для оформления web-страниц.

Порядок выполнения работы

Внимание! Описание используемых в работе тегов приведено в прил. 5.

1. Скопируйте из Интернета или другого источника в рабочую папку три графических файла (например, **1.jpg**, **2.jpg**, **3.jpg**). Убедитесь, что созданные ранее документы **1_name.htm**, **2_name.htm** и **3_name.htm** также находятся в этой папке.

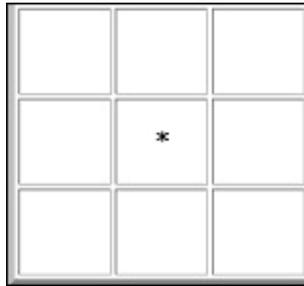
2. Выполните вставку рисунков в документ:

- откройте в **Блокноте** документ **2_name.htm**;
- вставьте рисунок **1.jpg** в начало документа **2_name.htm**, добавив тег `` после тега `<BODY>` (размер рисунка устанавливается равным 50 пикселям по горизонтали и по вертикали);
- сохраните документ с именем **4_name.htm**;
- просмотрите в браузере полученный результат;
- введите в тег рисунка параметр **ALIGN = RIGHT** для выравнивания рисунка по правому краю, просмотрите результат в браузере;
- вставьте рисунок **2.jpg** в конец документа **4_name.htm** (тип выравнивания рисунка – на ваше усмотрение), установите размер рисунка 100 пикселей по горизонтали и по вертикали.

3. Создайте гиперссылки:

- в документе **3_name.htm** закрепите гиперссылки за следующими словами:
 - за словом «**гарнитура**» – гиперссылка на документ **1_name.htm**;
 - за словом «**начертание**» – гиперссылка на документ **2_name.htm**;
- проверьте правильность переходов по гиперссылкам.

4. Закрепите гиперссылку за рисунком:
 - закрепите за рисунком **1.jpg** гиперссылку на документ **3_name.htm**, заключив тег вставки рисунка `` между тегами `<A HREF ...>` и ``;
 - выполните переход по гиперссылке.
5. Создайте новый документ в программе **Блокнот**.
6. Создайте таблицу по приведенному образцу (рис. 3.5), сохраните документ с именем **5_name.htm**. Сверху над таблицей поместите заголовок **Таблица № 1**.



	*	

Рис. 3.5. Таблица 1

При отображении таблицы в браузере должны удовлетворяться следующие условия:

- таблица должна размещаться по центру относительно левого и правого полей страницы;
- в центральной ячейке поместите символ * (звездочка), остальные ячейки должны быть пустыми.

Внимание! Для отображения пустых ячеек в них нужно поместить символьный примитив пробела ** **;

7. В этом же документе создайте копию таблицы № 1, введите заголовок **Таблица № 2**. Модифицируйте таблицу:

- в центральной ячейке поместите рисунок;
- «раскрасьте» все остальные ячейки в различные цвета.

8. Сохраните документ. Предъявите результат работы преподавателю.

Практическая работа 3.3. Организация защиты документов средствами пакета Microsoft Office

Цель работы — научиться организовывать защиту документов в MS Word, защиту электронных таблиц в MS Excel, защиту баз данных в MS Access.

Порядок выполнения работы

1. Выполнить упражнения 1, 2, 3.

Упражнение 1

Внимание! В процессе выполнения работы потребуется ввести пароли на защиту документов. Рекомендуется вводить всегда один и тот же пароль.

В этом упражнении будет создан бланк ведомости выдачи заработной платы, содержащий три раздела: **Раздел 1** — «шапка» ведомости; **Раздел 2** — таблица; **Раздел 3** — дата создания ведомости, фамилии руководителя организации и главного бухгалтера (рис. 3.6). Будет установлена защита документа от изменений так, чтобы изменения можно было вносить только в поля форм и в таблицу с фамилиями и другими данными сотрудников организации. Также будет установлен пароль на открытие файла.

СВОДНАЯ ВЕДОМОСТЬ № []						
НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ						
За [] месяц [] года						
Разрыв раздела (на текущей странице)						
Табельный номер	ФИО	Код подразделения	Должность	Оклад	Процент премии	Начислено
1	Антонов А.А.	1	Начальник отдела	50000	10	
2	Бородин Г.Д.	1	Инженер	20000	20	
3	Воронина И.С.	2	Бухгалтер	30000	10	
4	Громова А.А.	3	Инспектор	20000	15	
5	Ефимов О.Н.	2	Главный бухгалтер	40000	20	
6	Ильина П.А.	3	Инспектор	30000	10	
7	Колосова В.В.	2	Бухгалтер	12000	25	
8	Морозов В.Л.	2	Бухгалтер	15000	30	
9	Титова А.Р.	3	Начальник отдела	40000	5	
10	Фролов И.А.	2	Бухгалтер	20000	10	
Разрыв раздела (на текущей странице)						
Руководитель организации []				Дата создания ведомости []		
Главный бухгалтер []						

Рис. 3.6. Вид документа в режиме просмотра Обычный

1. Введите с клавиатуры текст документа.
2. Чтобы вставить два разрыва раздела (перед таблицей и перед датой создания ведомости), необходимо, установив курсор в место вставки, в окне команды **Вставка – Разрыв** установить переключатель в позицию **Новый раздел – на текущей странице**. Для просмотра вставленных границ разделов можно перейти к просмотру документа в обычном режиме, выполнив команду **Вид – Обычный**. Если границы разделов установлены правильно, можно вернуться к режиму **Разметка страницы**.
3. Для вставки первого поля формы выведите на экран панель инструментов **Формы (Вид – Панели инструментов – Формы)**. Установите курсор в тексте после слов «СВОДНАЯ ВЕДОМОСТЬ №» и выберите инструмент **Текстовое поле** (рис. 3.7). Аналогично вставьте другие поля формы.

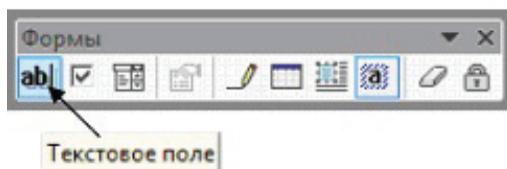


Рис. 3.7. Панель инструментов **Формы**

4. Установите защиту документа, оставив возможность изменять значения полей формы и текст второго раздела (таблица):
 - выполните команду **Сервис – Защитить документ**, откроется область задач **Защита документа** (рис. 3.8);
 - в области задач установите опцию **Разрешить только указанный способ редактирования документа**, затем выберите из списка доступных способов редактирования **Ввод данных в поля форм**;
 - перейдя по ссылке **Выбор разделов...**, в открывшемся окне **Защита раздела** отметьте галочкой только **Раздел 1** и **Раздел 3**, нажмите **ОК**;
 - в окне **Защита документа** пароль на снятие защиты разделов не устанавливайте. Нажмите **ОК**.

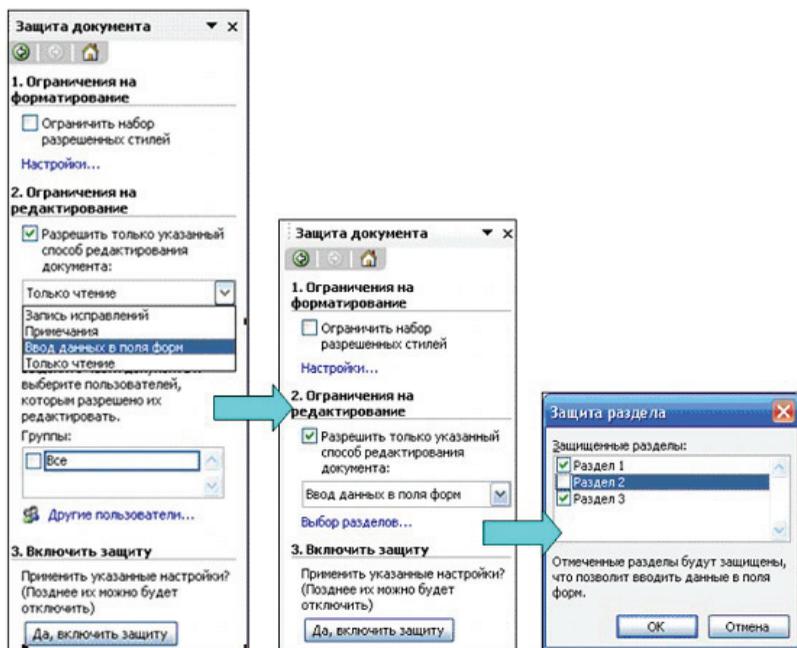


Рис. 3.8. Защита разделов документа

5. Проверьте, что защита установлена, то есть можно вводить текст только в **Раздел 2** и поля ввода из **Раздела 1** и **Раздела 3**.
6. Установите пароль для открытия файла:
 - выполните в меню **Сервис – Параметры**;
 - на вкладке **Безопасность** введите пароль, нажмите **ОК**, затем в открывшемся окне **Подтверждение пароля** введите пароль еще раз.
7. Сохраните документ с именем **Ведомость.doc**.

Упражнение 2

В этом упражнении будет создан документ Microsoft Excel, содержащий три рабочих листа: **Ведомость** – данные о сотрудниках организации, **Подразделения** – данные о подразделениях; **Статистика** – результаты обработки данных с листов **Ведомость** и **Подразделения**.

1. Создайте рабочую книгу Microsoft Excel. Сохраните ее с именем **Организация.xls**.

2. Измените имена листов рабочей книги: **Лист1** – на **Ведомость**, **Лист2** – на **Подразделения**, **Лист3** – на **Статистика**.

3. Откройте созданный в упражнении 1 документ **Ведомость.doc**, введя при открытии пароль.

4. Скопируйте таблицу из документа **Ведомость.doc** на лист **Ведомость** электронной таблицы.

5. В столбец **Начислено** введите формулы для расчета. Размер начисленной суммы определяется по формуле **Начислено = Оклад · (1+Процент премии / 100)**:

- в ячейку **G2** введите формулу **=E2*(1+F2/100)**, используя маркер заполнения, «протяните» формулу вниз на остальные ячейки столбца (рис. 3.9).

	А	В	
1	Табельный номер	ФИО	под
2	1	Антонов А.А.	1
3	2	Бородин Г.Д.	1
4	3	Воронина И.С.	2
5	4	Громова А.А.	3
6	5	Ефимов О.Н.	2
7	6	Ильина П.А.	3
8	7	Колосова В.В.	2
9	8	Морозов В.Л.	2
10	9	Титова А.Р.	3
11	10	Фролов П.А.	2

	Г
	Начислено
10	=E2*(1+F2/100)
20	=E3*(1+F3/100)
10	=E4*(1+F4/100)
15	=E5*(1+F5/100)
20	=E6*(1+F6/100)
10	=E7*(1+F7/100)
25	=E8*(1+F8/100)
30	=E9*(1+F9/100)
5	=E10*(1+F10/100)
10	=E11*(1+F11/100)

Рис. 3.9. Лист **Ведомость** в режиме показа формул (фрагмент)

6. На листе **Подразделения** в диапазоне ячеек **A1:C5** создайте таблицу, содержащую данные о подразделениях организации (рис. 3.10).

7. На листе **Статистика** создайте таблицу обработки данных, содержащихся на листах **Ведомость** и **Подразделения**, таким образом, чтобы при вводе в ячейку **B1** номера подразделения (1, 2 или 3) в ячейках диапазона **B4:B9** появились

следующие данные: наименование подразделения, ФИО руководителя, число сотрудников, размер средней, наибольшей и наименьшей зарплаты в подразделении (рис. 3.11). С этой целью выполните пункты 8–18 данного упражнения.

	A	B	C
1	Подразделения		
2	Код подразделения	Наименование	Руководитель
3	1	Отдел сбыта	Антонов А.А.
4	2	Бухгалтерия	Ефимов О.Н.
5	3	Отдел кадров	Титова А.Р.

Рис. 3.10. Таблица на листе **Подразделения**

	A	B
1	Код подразделения:	2
2		
3		Статистика
4	Наименование подразделения	Бухгалтерия
5	Руководитель	Ефимов О.Н.
6	Число сотрудников	5
7	Средняя зарплата	27 500,00р.
8	Наибольшая зарплата	48 000,00р.
9	Наименьшая зарплата	15 000,00р.

Рис. 3.11. Таблица на листе **Статистика**

8. Введите текст в ячейки столбца A так, как показано на рис. 3.11.

9. Установите проверку вводимых значений в ячейку **B1** (в нее будет заноситься код подразделения):

- выделите ячейку **B1** и выполните в меню **Данные – Проверка...**, в диалоговом окне **Проверка вводимых**

значений на вкладке **Параметры** установите параметры так, как показано на рис. 3.12;

- на вкладке **Сообщение для вывода** в поле **Сообщение** введите текст «**Введите код подразделения 1, 2 или 3**»;
- на вкладке **Сообщение об ошибке** в поле **Сообщение** введите текст «**Повторите ввод данных**»;
- нажмите **ОК**.

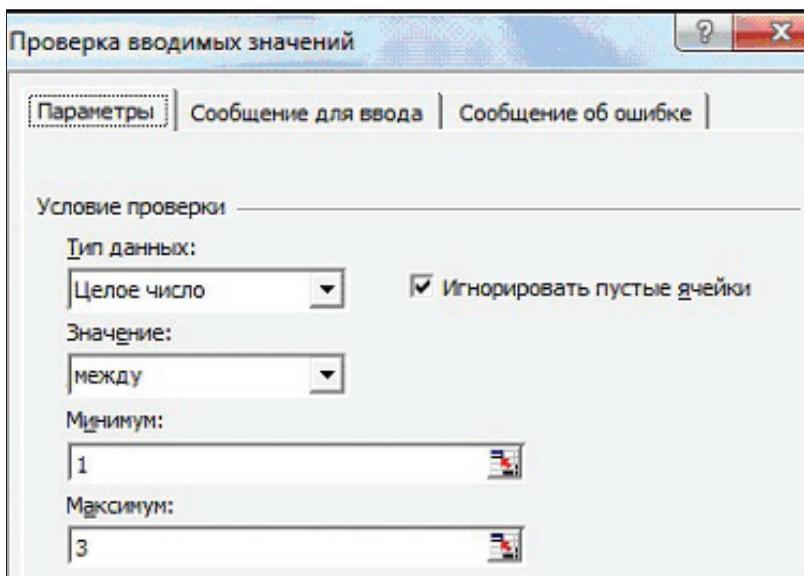


Рис. 3.12. Установка параметров проверки вводимых в ячейку **В1** данных

Теперь при вводе в ячейку **В1** кода подразделения число будет контролироваться на принадлежность множеству значений 1, 2, 3. В случае ввода числа, отличного от 1, 2, 3, будет выводиться сообщение с просьбой повторить ввод данных. Ячейка будет защищена от ввода неверных значений.

10. В ячейке **В4** будет выводиться наименование подразделения, если в эту ячейку ввести формулу **=ВПР(В1;Подразделения!А3:С5;2)**.

Функция **ВПР** просматривает левый столбец диапазона ячеек **А3:С5** (столбец **А**) на листе **Подразделения**.

Если значение в ячейке этого столбца совпало со значением в ячейке **В1** листа **Статистика** (введенный код подразделения), то в ячейку **В4** записывается значение из соответствующей найденному коду подразделения ячейки столбца с номером 2 (столбец **Наименование**).

11. Для вывода в ячейке **В5** ФИО руководителя подразделения введите в нее формулу по аналогии с пунктом 10 (рис. 3.13).

	А	В
1	Код подразделения:	1
2		
3		Статистика
4	Наименование подразделения	=ВПР(В1;Подразделения!А3:С5;2)
5	Руководитель	=ВПР(В1;Подразделения!А3:С5;3)
6	Число сотрудников	=СЧЁТЕСЛИ(Ведомость!С2:С11;В1)
7	Средняя зарплата	=СУММ(D1:D10)/B6
8	Наибольшая зарплата	=МАКС(D1:D10)
9	Наименьшая зарплата	=МИН(E1:E10)

Рис. 3.13. Таблица на листе **Статистика** в режиме показа формул

12. В ячейке **В6** будет выводиться число сотрудников в интересующем нас подразделении, если в эту ячейку ввести формулу **=СЧЁТЕСЛИ(Ведомость!С2:С11;В1)**. Функция **СЧЁТЕСЛИ** вычисляет количество ячеек диапазона **С2:С11** на листе **Ведомость**, значения в которых равны введенному в ячейку **В1** номеру подразделения.

13. Для определения средней и наибольшей зарплаты сформируйте вспомогательный массив ячеек **D1:D10**, записав в него значения из столбца **Начислено** листа **Ведомость** только для сотрудников рассматриваемого подразделения (для остальных сотрудников запишем нули). Для этого введите в ячейку **D1** формулу **=ЕСЛИ(Ведомость!С2=Статистика!\$B\$1; Ведомость!G2;0)**. Выполните автозаполнение остальных ячеек столбца **D**.

14. В ячейку **В7** введите формулу **=СУММ(D1:D10)/B6** (сумма заработных плат сотрудников подразделения делится на число сотрудников).

15. В ячейку **B8** введите формулу для вычисления максимального значения в ячейках диапазона **D1:D10**.

16. Для определения наименьшей зарплаты сформируйте вспомогательный массив ячеек **E1:E10**, записав в него значения из столбца **Начислено** листа **Ведомость** только для сотрудников рассматриваемого подразделения (для остальных сотрудников запишем заведомо невозможное значение зарплаты 100000000). Для этого введите в ячейку **E1** формулу **=ЕСЛИ(Ведомость!C2=Статистика!\$B\$1; Ведомость!G2;100000000)**. Выполните автозаполнение остальных ячеек столбца **E**.

17. В ячейку **B9** введите формулу для вычисления минимального значения в ячейках диапазона **E1:E10** (рис. 3.13).

18. Скройте столбцы **D** и **E** на листе **Статистика**, выделив эти столбцы и выполнив в меню **Формат – Столбец – Скрыть**.

Внимание! В дальнейшем, если потребуются отобразить скрытые столбцы, нужно будет выделить две любые смежные ячейки столбцов **C** и **F** и выполнить **Формат – Столбец – Отобразить**.

19. Сделайте доступным внесение данных в ячейку **B1** после того, как будет установлена защита от изменений листа **Статистика**. Для этого установите курсор в ячейку **B1**, в окне команды **Формат – Ячейки** выберите вкладку **Защита**, снимите флажок **Защищаемая ячейка** и нажмите **ОК**.

20. Скройте расчетные формулы на листе **Статистика**, выделив ячейки **B4:B9** и установив флажок **Скрыть формулы** в окне команды **Формат – Ячейки – вкладка Защита**.

21. Установите защиту листа **Статистика** от изменений (**Сервис – Защита – Защитить лист**).

22. Проверьте возможность ввода данных в ячейку **B1** и отсутствие такой возможности для остальных ячеек.

23. Присвойте диапазону ячеек **A1:F11** листа **Ведомость** имя **Сотрудники**. Для этого выделите его и выполните команду **Вставка – Имя – Присвоить**. В строке **Имя** появившегося окна **Присвоение имени** введите имя **Сотрудники** и нажмите **ОК**.

24. Присвойте диапазону ячеек **A2:C5** на листе **Подразделения** имя **Подразделения**.

25. Сохраните файл и завершите работу с MS Excel.

Упражнение 3

1. Запустите программу Microsoft Access и создайте новую базу данных с именем **Организация.mdb**.

2. Создайте таблицу базы данных **Сотрудники**, выполнив импорт данных из именованного диапазона **Сотрудники** с листа **Ведомость** файла электронной таблицы **Организация.xls**:

- в окне базы данных активируйте вкладку **Таблицы** и выполните команды **Файл – Внешние данные – Импорт**;
- в окне **Импорт** выберите тип файла **Microsoft Excel (*.xls)**, выберите файл **Организация.xls** и нажмите кнопку **Импорт**;
- в первом окне мастера **Импорт электронной таблицы** установите переключатель на пункт **именованные диапазоны** и выберите диапазон **Сотрудники**, нажмите кнопку **Далее**;
- в следующем окне установите флажок параметра **Первая строка содержит заголовки столбцов** и нажмите **Далее**;
- в следующем окне в группе **Данные необходимо сохранить** установите переключатель на пункт **в новой таблице**, нажмите **Далее**;
- в следующем окне измените описание поля **Табельный номер**, выбрав в поле **индекс** значение **Да (Совпадения не допускаются)**, нажмите **Далее**;
- в следующем окне мастера определите ключ **Табельный номер** и нажмите **Далее**;
- в последнем окне мастера введите имя таблицы **Сотрудники** и нажмите кнопку **Готово**;
- в появившемся окне об успешном импорте данных нажмите **ОК**.

3. Аналогично создайте таблицу **Подразделения**, импортировав данные о подразделениях из именованного диапазона **Подразделения** файла **Организация.xls**. Ключевое поле таблицы – **Код подразделения**.

4. Просмотрите полученные таблицы **Сотрудники** и **Подразделения** (рис. 3.14).

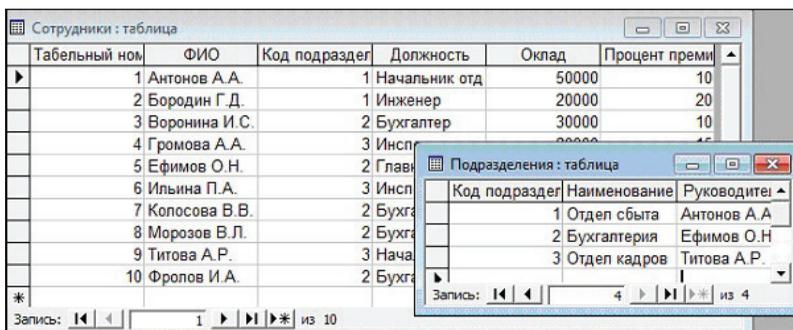


Рис. 3.14. Таблицы **Сотрудники** и **Подразделения** в окне базы данных

5. Установите связь между таблицами **Подразделения** и **Сотрудники** по полю **Код подразделения** (связь **один-ко-многим**).
6. Закройте файл базы данных.
7. Зашифруйте созданную базу данных:
 - не открывая базы данных, выполните в окне программы MS Access команду **Сервис – Защита – Шифровать/Расшифровать...**;
 - в появившемся окне выберите базу данных **Организация.mdb**;
 - в следующем окне в строку **Имя файла** введите имя новой (зашифрованной) базы данных **Организация_шифр.mdb** и нажмите кнопку **Сохранить**.
8. Защитите открытие зашифрованной базы паролем:
 - в окне программы MS Access выполните **Файл – Открыть**, выберите файл **Организация_шифр.mdb**;
 - щелкните мышью на указателе выпадающего списка (стрелочка вниз) кнопки **Открыть** и выберите из списка пункт **Монопольно**;
 - выполните команду **Сервис – Защита – Задать пароль базы данных...**;
 - в окне ввода пароля дважды введите пароль и нажмите **ОК**.
9. Закройте базу данных **Организация_шифр** и вновь откройте ее, воспользовавшись паролем.

10. Выполните статистическую обработку данных из таблиц **Сотрудники** и **Подразделения** по аналогии с обработкой данных, выполненной в упражнении 2 с помощью электронной таблицы. Сначала создайте запрос с вычисляемым полем **Начислено** (в таблице **Сотрудники** базы данных это поле отсутствует):

- в запрос следует включить поле **Код подразделения** из таблицы **Подразделения** и поля **ФИО**, **Оклад**, **Процент премии** из таблицы **Сотрудники**;
- для создания нового поля **Начислено** введите формулу **Начислено: [Оклад]*(1+[Процент премии]/100)** в заголовке поля (рис. 3.15);
- сохраните запрос с именем **Запрос1**.

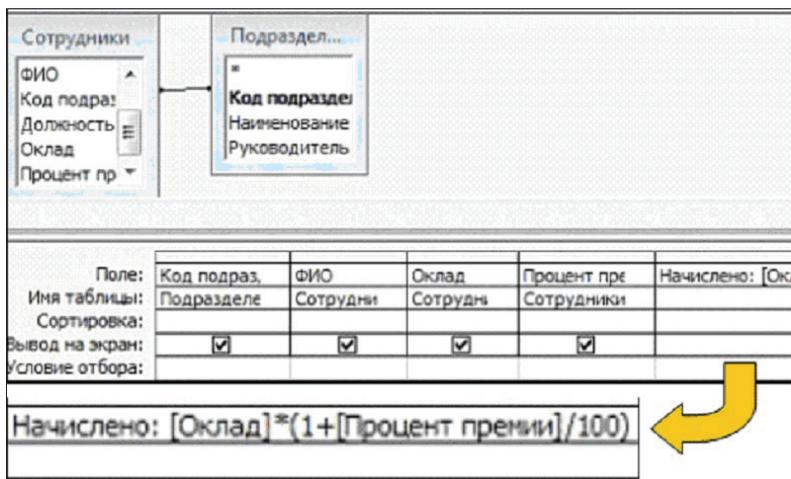


Рис. 3.15. Запрос с вычисляемым полем **Начислено**

11. Создайте итоговый запрос для определения среднего, наибольшего и наименьшего значения по полю **Начислено** в каждом подразделении организации:

- при создании запроса на этапе добавления таблиц выберите **Запрос1**;
- включите в запрос поле **Код подразделения** и поле **Начислено** трижды;
- активируйте инструмент **Групповые операции** ;

- по полю **Код подразделения** установите группировку, по полю **Начислено** – функции для вычисления среднего, наибольшего и наименьшего значений (рис. 3.16);
- сохраните и выполните запрос (рис. 3.17).

Код подразделения	Начислено	Начислено	Начислено
Запрос1	Запрос1	Запрос1	Запрос1
Группировка	Avg	Max	Min
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 3.16. Итоговый запрос

	Код подраздел	Avg-Начислено	Max-Начислено	Min-Начислено
▶	1	39500	55000	24000
	2	27500	48000	15000
	3	32666,666667	42000	23000

Рис. 3.17. Результат выполнения итогового запроса

12. Сравните результаты выполнения запроса с результатами расчетов на листе **Статистика** электронной таблицы.

13. Предъявите результаты работы преподавателю.

Практическая работа 3.4. Криптографические преобразования средствами Microsoft Excel

Цель работы – научиться:

- применять стандартные функции MS Excel для работы с текстовой информацией;
- выполнять простейшее криптографическое шифрование текстовой информации в среде электронных таблиц.

Порядок выполнения работы

1. Создайте новую книгу MS Excel, откройте **Лист2**, в ячейках **A1:A40** наберите 33 заглавные буквы русского алфавита, а также специальные символы: «.», «,», пробел, «:», кавычки, «!», «;».

2. Выделите диапазон ячеек **A1:A40** и присвойте ему имя **ABC** (**Вставка – Имя – Присвоить...**).

3. На **Листе1** в ячейке **B1** наберите текст, который необходимо зашифровать. При наборе текста используйте только русские буквы и специальные символы, которые входят в алфавит.

4. В ячейке **B3** запишите формулу **=ПРОПИСН(B1)**. Функция **ПРОПИСН** переводит символы в строке ячейки **B1** в заглавные буквы.

5. В ячейке **D3** запишите формулу **=ДЛСТР(B3)**. Функция **ДЛСТР** позволяет определить длину строки, это необходимо для кодировки исходной строки.

6. В ячейку **D4** запишите значение **k**.

При шифровании исходного текста каждая буква будет заменена другой буквой того же алфавита по следующему правилу: заменяющая буква определяется путем смещения по алфавиту к концу от исходной буквы на **k** букв. То есть **k** должно быть больше 0, но меньше длины строки **N** (**N** рассчитано в ячейке **D3**).

7. В столбце **A**, начиная с ячейки **A6**, введите числа от **1** до **N**.

8. В ячейку **B6** запишите формулу **=ПСТР(B\$3;A6;1)**, которая разделяет кодируемый текст на отдельные символы. Скопируйте эту формулу в ячейки столбца **B** от **B7** до ячейки, которая соответствует числу **N** в столбце **A**.

9. В ячейку **C6** запишите формулу **=ПОИСКПОЗ(B6;ABC;0)**. Скопировать содержимое ячейки **C6** в ячейки, соответствующие заполненным ячейкам столбца **A**.

Функция **ПОИСКПОЗ** производит поиск индекса (номера позиции) символа в массиве **ABC**, который был определен на **Листе2**.

10. Получив номер символа в алфавите **ABC**, нужно произвести сдвиг нумерации алфавита для кодируемой последовательности символов. В ячейку **D6** запишите формулу **=ЕСЛИ (ПОИСКПОЗ(B6; ABC; 0) + \$D\$4 > 40; ПОИСКПОЗ(B6; ABC; 0) + \$D\$4 – 40; ПОИСКПОЗ(B6; ABC; 0) + \$D\$4)**. Содержимое **D6** скопируйте в соответствующие ячейки столбца **D**.

Эта формула производит сдвиг номеров символов алфавита на величину k и определяет номер заменяющего символа из алфавита **ABC**.

11. Выберите символы из алфавита **ABC** в соответствии с новыми номерами. В ячейку **E6** запишите формулу **=ИНДЕКС(ABC;D6)**. Скопируйте содержимое ячейки **E6** в соответствующие ячейки столбца **E**.

12. Для получения строки закодированного текста необходимо в ячейку **F6** записать формулу **=E6**, в ячейку **F7** – формулу **=F6&E7**. Далее скопировать содержимое ячейки **F7** в соответствующие ячейки столбца **F**. В последней заполненной ячейке столбца **F** будет получен зашифрованный текст.

13. Для проверки шифрования произведем дешифрование полученного текста и сравним его с исходным. На **Листе3** выполните дешифрование аналогично пунктам 3–12 практической работы. При этом необходимо учесть следующие особенности:

- в ячейку **B1** нужно ввести зашифрованный текст (можно использовать копирование из соответствующей ячейки на листе 1: **Правка – Копировать**, затем **Правка – Специальная вставка – значения**);
- в пункте 10 в ячейку **D6** следует записать формулу **=ЕСЛИ(ПОИСКПОЗ(B6; ABC; 0) – \$D\$4 < 0; ПОИСКПОЗ(B6;ABC;0)–\$D\$4 + 40; ПОИСКПОЗ(B6; ABC; 0) – \$D\$4)**.

14. Получение исходного текста в последней заполненной ячейке столбца **F Листа3** свидетельствует о корректном выполнении практической работы.

15. Сохраните документ.

3.6. Индивидуальное домашнее задание 2. Создание web-сайта

Цель работы – научиться создавать простейшие тематические web-сайты.

Задание на выполнение

1. Выбрать тему из табл. 3.1 в соответствии с номером варианта.

2. Выполнить поиск в сети Интернет материалов по заданной теме.
3. Спроектировать структуру web-сайта.
4. Выбрать стиль оформления сайта.
5. Создать главную страницу сайта, содержащую ссылки на другие страницы, средствами языка HTML в редакторе **Блокнот**.
6. Создать остальные страницы сайта средствами текстового процессора Microsoft Word (**Файл – Сохранить как web-страницу...**).

Требования к содержанию и рекомендации по оформлению сайта

Первая страница должна содержать тему сайта, рисунок по теме и содержание сайта с гиперссылками на остальные страницы. Страницы, отражающие основное содержание представленной темы, должны включать подзаголовки, текст, рисунки по теме и гиперссылки: **Назад** – на предыдущую страницу сайта; **Далее** – на следующую страницу сайта (кроме последней страницы); **Оглавление** – на главную страницу сайта.

Страница **Источники информации** должна содержать список библиографических ссылок и (или) список интернет-ресурсов с указанием адресов ресурсов и их кратким описанием (название сайта, кому принадлежит ресурс, авторы публикации). На странице **Об авторе** должна быть информация об авторе сайта.

Пример структуры сайта, содержащего 6 страниц (Главная страница, 3 содержательные страницы, Источники информации, Об авторе), приведен на рис. 3.18.

Важно подобрать правильное сочетание цветов для фона и шрифта: например, фон – светлый, а шрифт – темный или наоборот. Не следует использовать в качестве фона фотографии и рисунки, поскольку это резко ухудшает восприятие текста. При создании сайта следует использовать не более двух шрифтов. Много шрифтов на одной странице отвлекает от содержания.

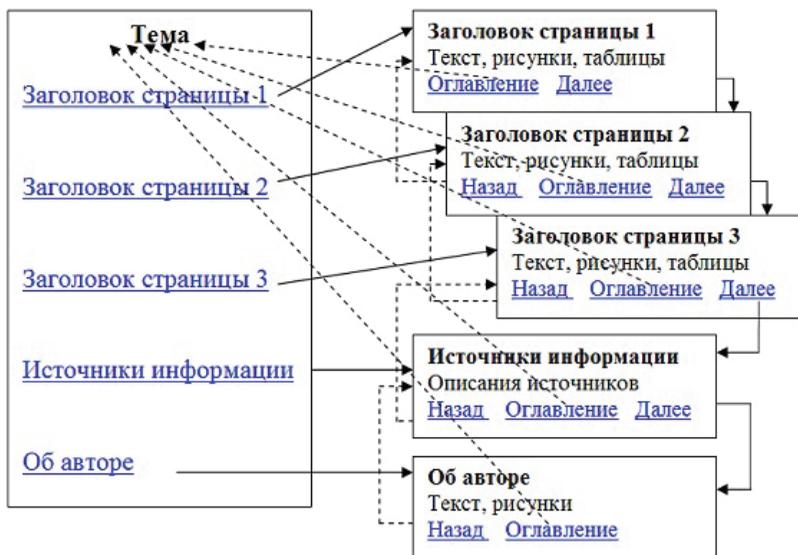


Рис. 3.18. Пример структуры сайта

Таблица 3.1

Варианты заданий для ИДЗ 2

№ варианта	Тема
1	Информационная безопасность и ее составляющие
2	Угрозы безопасности информации и их классификация
3	Основные виды защищаемой информации
4	Проблемы информационной безопасности в мировом сообществе
5	Законодательные и иные правовые акты Российской Федерации, регулирующие правовые отношения в сфере информационной безопасности и защиты государственной тайны
6	Административно-правовая ответственность в информационной сфере
7	Уголовная ответственность в информационной сфере
8	Политика безопасности и методы защиты информации

№ варианта	Тема
9	Антивирусная защита
10	Специфика обработки конфиденциальной информации в компьютерных системах
11	Классификация вирусов
12	Криптографические методы защиты информации
13	Электронная цифровая подпись
14	Требования к системе информационной безопасности
15	Информационные ресурсы ограниченного доступа
16	Обеспечение защищенности информационных ресурсов, составляющих государственную тайну
17	Симметричные и асимметричные алгоритмы шифрования
18	Классификация антивирусных программ

3.7. Тестовые задания

1. Обобщенная геометрическая характеристика компьютерной сети называется...
 - a) сетевой картой;
 - b) сетевой топологией;
 - c) логической архитектурой сети;
 - d) физической архитектурой сети.
2. Предотвратить проникновение вредоносных программ на подключенный к сети компьютер помогает...
 - a) резервное копирование;
 - b) антивирусный монитор;
 - c) электронная подпись;
 - d) наличие электронного ключа.
3. FTP-сервер – это компьютер, на котором...
 - a) хранится архив почтовых сообщений;
 - b) существует система разграничения прав доступа к файлам;

- c) содержатся файлы, предназначенные для открытого доступа;
 - d) содержатся файлы, предназначенные только для администратора сети.
4. Географическими доменами являются:
- a) ua;
 - b) net;
 - c) ru;
 - d) kz;
 - e) gov;
 - f) org.
5. Компьютер, предоставляющий ресурсы клиентам сети, – это...
- a) модем;
 - b) сервер;
 - c) рабочая станция;
 - d) шлюз.
6. Для передачи вируса посредством электронной почты обычно используется...
- a) область адреса получателя;
 - b) вложение письма;
 - c) тема письма;
 - d) служебные заголовки письма.
7. Почтовыми программами являются...
- a) Skype;
 - b) The Bat;
 - c) MS Outlook;
 - d) Mozilla Firefox;
 - e) ICQ.
8. Антивирусные программы, имитирующие заражение файлов компьютера вирусами, – это...
- a) программы-вакцины;
 - b) программы-доктора;
 - c) программы-черви;
 - d) программы-брандмауэры.
9. При создании цифровой подписи задается(-ются)...
- a) два открытых ключа;
 - b) два секретных ключа;

- c) два ключа: секретный и открытый;
- d) один секретный ключ.

10. Протоколы POP3 и SMTP используются для организации сервиса...

- a) IRC;
- b) FTP;
- c) Telnet;
- d) электронная почта.

11. Объединение компьютеров в сеть посредством подключения каждого компьютера сети к концентратору с помощью отдельного кабеля образует топологию...

- a) «звезда»;
- b) «общая шина»;
- c) «кольцо»;
- d) «дерево».

12. Программно-аппаратное устройство, соединяющее сети, работающие по разным протоколам, — это...

- a) мост;
- b) шлюз;
- c) концентратор;
- d) брандмауэр.

Библиографический список

Рекомендуемая литература

1. Васильков, А.В. Безопасность и управление доступом в информационных системах : учеб. пособие / А.В. Васильков, И.А. Васильков. – М. : ФОРУМ, 2010. – 367 с.
2. Информатика : учеб. / Б.В. Соболев [и др.]. – Ростов н/Д : Феникс, 2010. – 446 с.
3. Информатика : учеб. / под ред. Н.В. Макаровой. – 3-е изд., перераб. – М. : Финансы и статистика, 2003. – 767 с.
4. Информатика для юристов и экономистов / под ред. С.В. Симоновича. – СПб. : Питер, 2008. – 687 с.
5. Коноплева, И.А. Информационные технологии : учеб. пособие для вузов / И.А. Коноплева, О.А. Хохлова, А.В. Денисова. – М. : Проспект, 2007. – 295 с.
6. Кузин, А.В. Базы данных : учеб. пособие для вузов / А.В. Кузин, С.В. Левонисова. – М. : Академия, 2008. – 315 с.
7. Мадера, А.Г. Математические модели в управлении: компьютерное моделирование в MS Excel : лаб. практикум / А.Г. Мадера. – М. : РГГУ, 2007. – 118 с.
8. Могилев, А.В. Информатика : учеб. пособие для вузов / А.В. Могилев, Н.И. Пак, Е.К. Хеннер ; под ред. Е.К. Хеннера. – М. : Академия, 2004. – 842 с.
9. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. пособие для вузов / В.Г. Олифер, Н.А. Олифер. – СПб. : Питер, 2008. – 957 с.
10. Степанов, А.Н. Архитектура вычислительных систем и компьютерных сетей : учеб. пособие для вузов / А.Н. Степанов. – СПб. : Питер, 2007. – 508 с.

Интернет-ресурсы

11. Алексеев, Е.Г. Информатика : учеб. [Электронный ресурс] / Е.Г. Алексеев, С.Д. Богатырев. – Режим доступа: <http://inf.e-alekseev.ru/>, свободный.
12. Кириллов, В.В. Основы проектирования реляционных баз данных : учеб. пособие [Электронный ресурс] / В.В. Кириллов. – Режим доступа: <http://citforum.ru/database/dbguide/index.shtml>, свободный.

13. Путеводитель по обучению работе с Excel 2003 на сайте корпорации Microsoft [Электронный ресурс]. – Режим доступа: <http://office.microsoft.com/ru-ru/excel-help/NA001116128.aspx?CTT=1>, свободный.
14. Энциклопедия персонального компьютера и Интернета Кирилла и Мефодия [Электронный ресурс]. – Режим доступа: http://mega.km.ru/pc_2001/, свободный.

Глоссарий

ADSL — технология высокоскоростного доступа к сети Интернет на абонентских линиях телефонной сети. Технология ADSL использует метод разделения полосы пропускания телефонной линии на несколько частотных полос.

IP-адрес — числовой адрес, задающий точное расположение компьютера в сети Интернет.

IRC (англ. *Internet Relay Chat*) — сетевой протокол, на котором основана служба обмена информацией в режиме реального времени.

FTP (англ. *File Transfer Protocol*) — служба сети Интернет, которую используют для передачи файлов от одного компьютера к другому.

HTML (англ. *Hyper Text Markup Language*) — язык гипертекстовой разметки документов.

HTTP (англ. *Hypertext Transport Protocol*) — транспортный протокол службы WWW в сети Интернет.

TCP/IP — единый набор протоколов передачи данных в сети Интернет. В названии TCP/IP отражены имена двух главных сетевых протоколов. **TCP** (*Transmission Control Protocol*) — протокол контроля передачи данных. Он обеспечивает надежность передачи данных. **IP** (*Internet Protocol*) — протокол маршрутизации. Определяет основные правила, которыми должны следовать компьютеры для обмена данными.

URL (англ. *Uniform Resource Locator*) — унифицированный указатель ресурса, адрес в системе WWW (всемирная паутина), при помощи которого однозначно определяется каждый документ.

Асимметричное шифрование — способ шифрования, при котором для шифрования используется один ключ, являющийся общедоступным, а для дешифрования — другой, являющийся секретным. При этом знание общедоступного ключа не позволяет определить секретный ключ.

Атрибут — характеристика, отличительное свойство некоторой сущности (реального объекта, явления, процесса, события).

Аутентификация – процедура входа в систему с предоставлением идентификационных данных, доказывающих аутентичность (уникальность) и легитимность пользователя.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, независимая от прикладных программ. Является информационной моделью предметной области.

Веб-обозреватель, браузер – программное обеспечение для запроса веб-страниц, их обработки, вывода и перехода от одной страницы к другой.

Векторная графика – разновидность компьютерной графики, в которой изображение формируется из совокупности отдельных геометрических примитивов, задаваемых с помощью математического описания.

Виртуальная реальность – высокотехнологичная форма компьютерного моделирования, которая позволяет пользователю погрузиться в модельный мир и непосредственно действовать в нем. Зрительные, слуховые, осязательные и моторные ощущения пользователя при этом заменяются их имитацией, генерируемой компьютером.

Всемирная паутина (англ. *World Wide Web*) – распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключённых к Интернету. Всемирную паутину образуют миллионы **web-серверов**. Гипертекстовые документы, размещаемые во всемирной паутине, называются **web-страницами**. Несколько web-страниц, объединённых общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же web-сервере, называются **web-сайтом**.

Геоинформационные системы (ГИС) – новая информационная технология, интегрирующая методы и технологии баз данных, геодезии, картографии, космонавтики, навигации, компьютерной графики.

Гиперссылка – часть гипертекстового документа, ссылающаяся на другой элемент в самом документе, на другой

объект, расположенный на локальном диске или в компьютерной сети, либо на элементы этого объекта. Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется графически.

Глобальная компьютерная сеть (англ. *Wide Area Network, WAN*) — телекоммуникационная структура, объединяющая локальные информационные сети, имеющие общий протокол связи, методы подключения и протоколы обмена данными.

Захватчик паролей — программа, специально разработанная для воровства паролей.

Идентификация пользователя — присвоение уникального кода пользователю информационной системы.

Иерархическая модель данных — логическая модель данных в виде древовидной структуры, представляющая собой совокупность элементов данных, расположенных в порядке их подчинения и образующих перевернутое дерево. Один или несколько элементов более низкого уровня соединяется при помощи связи с одним элементом более высокого уровня. Обобщением иерархической модели является **сетевая модель данных**. Каждый элемент данных в этой модели может быть связан с несколькими другими элементами.

Имитационное моделирование — разновидность моделирования, реализуемого с помощью компьютерных программ, позволяющих посредством процессов-аналогов провести исследование структуры и функций объекта моделирования.

Интернет (англ. *Internet*) — открытая мировая коммуникационная инфраструктура, состоящая из взаимосвязанных компьютерных сетей, обеспечивающая доступ к удаленной информации и обмен информацией между компьютерами.

Интернет-провайдер (англ. *Internet Service Provider, ISP*) — организация, предоставляющая услуги доступа к Интернету и иные связанные с Интернетом услуги.

Информационная модель — описание моделируемого объекта на одном из языков кодирования информации.

Информационный объект — описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных атрибутов.

Клавиатурный шпион (кейлоггер) – программное или аппаратное средство, основным назначением которого является скрытый мониторинг нажатий клавиш и ведение журнала этих нажатий.

Компьютерная модель – модель, реализованная средствами программной среды.

Компьютерная сеть – система связи компьютеров и компьютерного оборудования (серверы, маршрутизаторы и другое оборудование). Для передачи информации могут быть использованы различные физические явления (различные виды электрических сигналов, световых сигналов или электромагнитного излучения).

Компьютерный вирус – программа, имеющая возможность создавать свои дубликаты (не обязательно совпадающие с оригиналом) и внедрять их в вычислительные сети и/или файлы, системные области компьютера и прочие объекты с целью искажения и уничтожения данных и программ. При этом дубликаты сохраняют способность к дальнейшему распространению.

Криптографические методы защиты информации – специальные способы и методы шифрования информации, в результате чего данная информация становится недоступной для посторонних лиц. Данные технологии используются для конфиденциальной и безопасной передачи информации.

Криптография – наука о способах преобразования (шифрования) информации с целью защиты ее от незаконного или нежелательного использования.

Линия тренда – графическое представление общей закономерности изменения ряда данных.

Логическая бомба – программа или фрагмент кода в программе, реализующий некоторую функцию при выполнении определенного условия, например, условием может быть наступление заданной даты.

Локальная вычислительная сеть (англ. *Local Area Network, LAN*) – компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий. Компьютеры могут соединяться между собой с использованием различных средств связи: витая пара, оптические кабели, беспроводные технологии.

Математическая модель – совокупность записанных на языке математики соотношений (формул, неравенств, уравнений, логических соотношений), определяющих характеристики состояния объекта в зависимости от его свойств, параметров, внешних воздействий.

Метод Монте-Карло – метод моделирования на основе статистических испытаний, проводимых на компьютере с помощью программ-генераторов псевдослучайных чисел.

Моделирование – процесс исследования реального объекта с помощью модели. Исходный объект называется при этом **прототипом** или **оригиналом**.

Модель – материальный или воображаемый объект, который в процессе познания замещает реальный объект, сохраняя при этом его существенные свойства.

Модель данных – совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними.

Модем – устройство, применяющееся в системах связи и выполняющее функцию преобразования сигналов.

Нормализация таблиц – процесс приведения таблиц реляционной базы данных к выбранной нормальной форме.

Первичный ключ реляционной таблицы – поле или группа полей, которые позволяют однозначно определить каждую строку в таблице. Если первичный ключ состоит из одного поля, то он называется **простым ключом (ключевым полем)**. Если первичный ключ состоит из нескольких полей, то такая таблица имеет **составной ключ**.

Подбор параметра в Microsoft Excel – инструмент, позволяющий целенаправленно перебрать множество значений одиночного параметра с одновременным контролем результирующего значения.

Поиск решения в Microsoft Excel – набор управляемых пользователем алгоритмов решения оптимизационных задач.

Постреляционная модель данных – расширенная реляционная модель, снимающая ограничение неделимости данных, допускающая поля, значения которых состоят из подзначений.

Предметная область — некоторая часть реального мира, информация о которой представлена в базе данных.

Принуждение — метод защиты информации, при котором пользователи вынуждены соблюдать правила обработки, передачи и использования информации под угрозой материальной, административной или уголовной ответственности.

Протокол — набор правил, которым следуют компьютеры и программы при обмене информацией. Существуют различные протоколы, которые управляют всеми аспектами связи и передачи данных — от аппаратного до прикладного уровня.

Рабочая станция — компьютер, подключенный к сети, через который пользователь получает доступ к сетевым ресурсам. Рабочая станция функционирует как в сетевом, так и в локальном режиме и обеспечивает пользователя всем необходимым инструментарием для решения прикладных задач.

Растровая графика — разновидность компьютерной графики, использующая представление изображения с помощью набора пикселей, имеющих различные характеристики. **Пиксель** — наименьший неделимый элемент растрового графического изображения.

Регламентация — метод защиты информации, создающий по регламенту такие условия автоматизированной обработки, хранения и передачи защищаемой информации, при которых возможности несанкционированного доступа к ней сводились бы к минимуму.

Реляционная модель данных — логическая модель данных, использующая организацию данных в виде двумерных реляционных таблиц (отношений).

Сервер — компьютер, выполняющий функции управления сетевыми ресурсами общего доступа: осуществляет хранение данных, управляет базами данных, выполняет удаленную обработку заданий, обеспечивает печать заданий и др.

Сетевая модель OSI (*базовая эталонная модель взаимодействия открытых систем*, англ. *Open Systems Interconnection Basic Reference Model*) — абстрактная сетевая модель для коммуникаций и разработки сетевых протоколов.

Сетевая топология — способ описания конфигурации сети, схема расположения и соединения сетевых устройств. Существует множество способов соединения сетевых устройств. Выделяют три базовые топологии (шина, кольцо, звезда) и дополнительные — ячеистая, решётка, дерево, полносвязная. Дополнительные топологии являются комбинациями базовых.

Сетевой червь — тип вредоносных программ, распространяющихся в компьютерной сети, способных к преодолению систем защиты, а также к созданию и дальнейшему распространению своих копий и осуществлению иных вредоносных действий.

Симметричное шифрование — это способ шифрования, использующий один и тот же секретный ключ для шифрования и дешифрования.

Система управления базами данных (СУБД) — комплекс программных средств, предназначенных для создания структуры новой базы данных, наполнения ее содержимым, редактирования и обработки содержимого.

Социальная сеть — виртуальная сеть, являющаяся средством обеспечения сервисов, связанных с установлением связей между пользователями сети Интернет.

Сущность — это объект, о котором хранится информация в базе данных.

Таблица реляционной базы данных (реляционная таблица) — заданная структура полей, состоящая из конечного набора однотипных записей. **Поле** — элементарная единица логической организации данных, которая соответствует одному атрибуту информационного объекта (*столбец* реляционной таблицы). **Запись** — совокупность логически связанных полей (обобщенная строка реляционной таблицы). **Экземпляр записи** — отдельная реализация записи, содержащая конкретные значения ее полей (конкретная строка реляционной таблицы).

Теги — команды языка HTML.

Телеконференции — службы Интернета, которые организуют тематический обмен информацией между пользователями компьютерной сети.

Трёхмерная графика (3D-графика) — разновидность компьютерной графики, использующая методы создания объёмных моделей объектов, которые максимально соответствуют реальным.

Троянский конь — программа, выполняющая в дополнение к основным еще и дополнительные действия, не описанные в документации. Троянский конь — это дополнительный блок команд, тем или иным образом вставленный в исходную безвредную программу.

Файервол (межсетевой экран, брандмауэр) — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

Фрактальная графика — разновидность компьютерной графики, использующая принцип построения изображений на основе геометрических фракталов, обладающих свойством самоподобия, когда небольшая часть фрактала содержит информацию обо всем фрактале.

Шлюз — выделенный компьютер, на котором установлено специальное программное обеспечение, предназначенное для передачи информации между двумя сетями, использующими разные стандарты

Электронная почта (англ. *electronic mail, e-mail*) — технология и предоставляемые ею услуги по пересылке и получению электронных сообщений по компьютерной сети.

Электронная цифровая подпись — реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа, позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе.

Язык SQL (Structured Query Language — язык структурированных запросов) — информационно-логический язык для описания, изменения и извлечения данных, хранимых в реляционных базах данных. SQL является механизмом связи между прикладным программным обеспечением и базой данных.

Приложение 2

Ответы к тестовым заданиям

Раздел «Модели решения функциональных и вычислительных задач»

Номер вопроса	1	2	3	4	5	6	7	8	9	10	11	12
Номер(а) правильных ответов	c	a	b	d	b	a	b	c	d	a	c	b

Раздел «Базы данных»

Номер вопроса	1	2	3	4	5	6	7	8	9	10	11	12
Номер(а) правильных ответов	c	a, c, f	d	b	d	b	e	a	c	d	c	a

Раздел «Компьютерные сети. Защита информации»

Номер вопроса	1	2	3	4	5	6	7	8	9	10	11	12
Номер(а) правильных ответов	a	b	c	a, c, d	b	b	b, c	a	c	d	a	b

Приложение 3

Вопросы для подготовки к экзамену (зачету)

1. Понятие модели и моделирования. Классификация моделей по различным признакам.
2. Этапы математического моделирования. Типы математических моделей.
3. Математическое моделирование стохастических процессов. Моделирование на основе метода Монте-Карло.
4. Компьютерная графика: разновидности, принципы построения графических изображений, сравнительный анализ возможностей, достоинств и недостатков.
5. Применение в моделировании средств компьютерной графики.
6. Использование в моделировании инструмента «Подбор параметра» табличного процессора Microsoft Excel: возможности, ограничения, способ применения, пример.
7. Использование в моделировании инструмента «Поиск решения» табличного процессора Microsoft Excel: возможности, ограничения, способ применения, пример.
8. Понятие базы данных и системы управления базами данных. Модели представления данных (иерархическая, сетевая).
9. Основные понятия реляционной модели данных: реляционная таблица (отношение), поле (атрибут), запись, первичный ключ, внешний ключ.
10. Нормализация реляционных таблиц: понятие нормализации, определение 1, 2, 3 нормальных форм.
11. Основы работы с СУБД Microsoft Access: основные объекты базы данных.
12. Работа с таблицами в среде СУБД Microsoft Access: создание и сохранение таблицы, ввод и редактирование данных. Типы связей между таблицами.
13. Работа с запросами в среде СУБД Microsoft Access: типы запросов, процедура создания запроса на выборку в режиме конструктора.
14. Понятие, классификация и виды компьютерных сетей.
15. Аппаратные компоненты компьютерных сетей.
16. Организация сети Интернет. Сервисы Интернета.

17. Угрозы безопасности информации, их виды. Методы и средства обеспечения безопасности информации.

18. Понятие и виды вредоносных программ. Виды компьютерных вирусов, их классификация. Защита от компьютерных вирусов.

19. Средства защиты информации от несанкционированного доступа.

20. Криптографическая защита информации. Электронная цифровая подпись.

Приложение 4

Формулировка задания для выполнения ИДЗ по теме «Базы данных» (Задание 1)

Вариант 1. База данных «Платный прием в поликлинике»	
Описание предметной области	Минимальный набор полей базовых таблиц
<p>Платный прием пациентов проводится врачами разных специальностей (хирург, терапевт, кардиолог и т. д.). При оформлении приема должна быть сформирована квитанция об оплате приема, в которой указывается информация о пациенте, враче, стоимости приема, дате приема.</p> <p>Пациент платит за прием некоторую сумму, которая устанавливается персонально для каждого врача. За каждый прием врачу отчисляется 40% от стоимости приема</p>	<ol style="list-style-type: none"> 1. ФИО врача 2. Специальность врача 3. Стоимость приема 4. ФИО пациента 5. Дата рождения пациента 6. Адрес пациента 7. Дата приема
	<p style="text-align: center;">Предлагаемый набор базовых таблиц</p>
	<ol style="list-style-type: none"> 1. ВРАЧИ 2. ПАЦИЕНТЫ 3. ПРИЕМ ПАЦИЕНТОВ
Вариант 2. База данных «Прокат автомобилей»	
Описание предметной области	Минимальный набор полей базовых таблиц
<p>Фирма выдает напрокат автомобили. При этом фиксируются данные о клиенте, об автомобиле, дата начала проката и количество дней проката, стоимость одного дня проката. Стоимость одного дня проката может отличаться для разных автомобилей и изменяться периодически. Стоимость проката автомобиля определяется как</p> <p>Стоимость одного дня проката · Количество дней проката</p>	<ol style="list-style-type: none"> 1. ФИО клиента 2. Серия, номер паспорта клиента 3. Модель автомобиля 4. Год выпуска автомобиля 5. Госномер автомобиля 6. Стоимость одного дня проката 7. Дата начала проката 8. Количество дней проката

	Предлагаемый набор базовых таблиц
	1. КЛИЕНТЫ 2. АВТОМОБИЛИ 3. ПРОКАТ
Вариант 3. База данных «Учет оптовых продаж магазина»	
Описание предметной области	Минимальный набор полей базовых таблиц
<p>Оптовый магазин закупает товар по цене закупки и продает товар по цене продажи, которая может изменяться. Разница между ценой продажи и ценой закупки составляет доход магазина от реализации каждой единицы товара.</p> <p>В магазине работают несколько продавцов. Каждый продавец получает комиссионное вознаграждение за проданный товар. Процент комиссионных назначается индивидуально каждому продавцу</p>	1. Наименование товара 2. Единица измерения товара 3. Цена закупки 4. Цена продажи 5. Дата продажи 6. Количество проданных единиц товара 7. ФИО продавца 8. Процент комиссионных продавца
	Предлагаемый набор базовых таблиц
	1. ТОВАРЫ 2. ПРОДАВЦЫ 3. ПРОДАЖИ
Вариант 4. База данных «Учет нарушений правил дорожного движения»	
Описание предметной области	Минимальный набор полей базовых таблиц
<p>При нарушении правил дорожного движения (ПДД) фиксируется информация об автомобиле, водителе, виде нарушения, дате нарушения, размере штрафа.</p> <p>Размер штрафа определяется видом нарушения</p>	1. Модель автомобиля 2. Год выпуска 3. Госномер 4. ФИО владельца 5. Серия и номер паспорта владельца 6. Вид нарушения ПДД 7. Размер штрафа 8. Дата нарушения ПДД 9. ФИО водителя

	Предлагаемый набор базовых таблиц
	1. АВТОМОБИЛИ 2. ВИДЫ НАРУШЕНИЙ 3. ФАКТЫ НАРУШЕНИЙ
Вариант 5. База данных «Туристическое агентство»	
Описание предметной области	Минимальный набор полей базовых таблиц
<p>Фирма предоставляет клиентам услуги по организации зарубежных поездок. При оформлении поездки устанавливается фиксированная стоимость 1 дня пребывания в той или иной стране, включающая стоимость проживания, питания, экскурсионного обслуживания и других услуг. Эта стоимость является характеристикой каждого конкретного маршрута.</p> <p>Стоимость поездки может быть вычислена как Стоимость 1 дня пребывания · Количество дней + Стоимость транспортных услуг + Стоимость оформления визы</p>	1. ФИО клиента 2. Серия и номер паспорта клиента 3. Страна назначения 4. Стоимость 1 дня пребывания 5. Стоимость транспортных услуг 6. Стоимость оформления визы 7. Дата начала поездки 8. Количество дней поездки
	Предлагаемый набор базовых таблиц
	1. КЛИЕНТЫ 2. МАРШРУТЫ 3. ПОЕЗДКИ
Вариант 6. База данных «Учет подписки на периодические печатные издания»	
Описание предметной области	Минимальный набор полей базовых таблиц
<p>Требуется создать базу данных для хранения информации о подписке на периодические печатные издания. При оформлении подписки на то или иное печатное издание следует указать данные о подписчике, об издании, дату начала подписки и количество месяцев, на которые оформляется подписка (срок подписки).</p>	1. ФИО подписчика 2. Улица 3. Номер дома 4. Номер квартиры 5. Индекс издания по каталогу 6. Вид издания (газета или журнал) 7. Название издания

<p>Стоимость подписки может быть вычислена как Стоимость 1 месяца подписки · Срок подписки</p>	<p>8. Стоимость 1 месяца подписки 9. Начало подписки (номер месяца) 10. Срок подписки</p>
	<p>Предлагаемый набор базовых таблиц</p>
	<p>1. ИЗДАНИЯ 2. ПОЛУЧАТЕЛИ 3. ДОСТАВКА</p>
<p>Вариант 7. База данных «Учет сделок с недвижимостью»</p>	
<p>Описание предметной области</p>	<p>Минимальный набор полей базовых таблиц</p>
<p>Фирма занимается оформлением сделок с объектами жилой недвижимости. При оформлении сделки фиксируется информация о продаваемой квартире, риэлторе, оформляющем сделку купли-продажи, дате оформления сделки. Риэлтор, оформивший сделку, получает комиссионное вознаграждение, которое вычисляется как Цена квартиры · Процент вознаграждения. Процент вознаграждения является индивидуальным и фиксированным для каждого риэлтора</p>	<p>1. Название улицы 2. Номер дома 3. Номер квартиры 4. Площадь квартиры 5. Количество комнат 6. Дата сделки 7. Цена квартиры 8. ФИО риэлтора 9. Процент вознаграждения</p>
	<p>Предлагаемый набор базовых таблиц</p>
	<p>1. КВАРТИРЫ 2. РИЭЛТОРЫ 3. СДЕЛКИ</p>
<p>Вариант 8. База данных «Учет договоров страхования»</p>	
<p>Описание предметной области</p>	<p>Минимальный набор полей базовых таблиц</p>
<p>Договор страхования заключается между страховой компанией и клиентом на 1 год. При заключении договора указывается вид страхования, страховая сумма, дата начала действия договора.</p>	<p>1. ФИО клиента 2. Процент скидки 3. Вид страхования (имущество, автомобиль и т. д.) 4. Тариф (зависит от вида страхования)</p>

<p>Каждый клиент выплачивает при заключении договора страховую премию. Размер страховой премии зависит от суммы страхования, тарифа и индивидуальной скидки клиента: Страховая премия = Сумма страхования · (Тариф – Процент скидки).</p> <p>Тариф принимает значения от 5 до 10 процентов, индивидуальная скидка – от 1 до 5 процентов.</p> <p>Страховой агент получает вознаграждение за договор, равное произведению страховой суммы на процент вознаграждения</p>	<p>5. Сумма страхования 6. Дата заключения договора 7. ФИО агента 8. Процент вознаграждения</p>
	<p>Предлагаемый набор базовых таблиц</p> <p>1. КЛИЕНТЫ 2. СТРАХОВЫЕ АГЕНТЫ 3. ДОГОВОРЫ</p>
<p>Вариант 9. База данных «Учет результатов сдачи вступительных экзаменов»</p>	
<p>Описание предметной области</p>	<p>Минимальный набор полей базовых таблиц</p>
<p>База данных должна содержать информацию об абитуриентах, экзаменаторах и результатах сдачи вступительных экзаменов.</p> <p>О каждом факте сдачи экзамена указываются дата сдачи экзамена, название экзамена, кто сдавал экзамен, кто принимал экзамен, оценка за экзамен.</p> <p>Экзаменатор получает за прием экзамена установленную оплату, которая назначается индивидуально</p>	<p>1. ФИО абитуриента 2. Год рождения абитуриента 3. Серия/номер паспорта абитуриента 4. ФИО экзаменатора 5. Размер оплаты за прием экзамена 6. Дата сдачи экзамена 7. Название экзамена (история, физика и т. д.) 8. Оценка за экзамен</p>
	<p>Предлагаемый набор базовых таблиц</p> <p>1. АБИТУРИЕНТЫ 2. ЭКЗАМЕНАТОРЫ 3. ЭКЗАМЕНЫ</p>

Приложение 5

Основные теги для создания web-страниц

Назначение	Вид тега	Комментарии
<i>Общая структура документа HTML</i>		
Тип документа	<HTML> теги </HTML>	Начало и конец документа
Имя документа	<HEAD> текст </HEAD>	Не отображается браузером
Заголовок	<TITLE> текст </TITLE>	Содержимое строки заголовка страницы в окне браузера
Тело документа	<BODY> теги страницы </BODY>	Границы WEB-страницы
<i>Форматирование абзаца</i>		
Границы абзаца	<P> текст абзаца </P>	
Перевод строки внутри абзаца	 	Одиночный тег
Выравнивание абзаца	<P ALIGN = LEFT>текст </P> <P ALIGN = CENTER>текст </P> <P ALIGN = RIGHT> текст</P> <P ALIGN = JUSTIFY> текст </P>	LEFT – по левому краю CENTER – по центру RIGHT – по правому краю JUSTIFY – по ширине
Заголовки разных уровней	<Hn> текст </Hn> <i>Пример:</i> <H2> текст </H2> – заголовков 2-го уровня	n – номер уровня заголовка (от 1 до 6) Заголовок – это заранее определенный стиль оформления текста

<i>Форматирование шрифта</i>	
Жирный	 текст
Жирный шрифт	
Курсив	<I> текст </I>
Курсивный шрифт	
Подчеркнутый	<U> текст </U>
Подчеркнутый шрифт	
Перечеркнутый	<S> текст </S>
Перечеркнутый шрифт	
Верхний индекс	^{текст}
Верхний индекс	
Нижний индекс	_{текст}
Нижний индекс	
Размер шрифта	 текст Пример: текст — увеличение текущего размера на 2 условные единицы
Размер шрифта (от 1 до 7 условных единиц) или относительное изменение размера шрифта (на-пример, +2)	
Цвет шрифта	 текст Пример: текст — текст будет выведен шрифтом зеленого цвета
Цвет задается ключевым словом (RED — красный, BLUE — голубой, GREEN — зеленый, YELLOW — желтый, NAVY — темно-синий и т. д.)	

Создание списков	
Нумерованный список	 элементы списка
Маркированный список	 элементы списка
Элемент списка	 текст
Вставка рисунков	
Вставка рисунка	<p></p> <p><i>Пример</i></p> <p></p>
Вставка гиперссылки	

Пример	
Теги	Результат
	1. Москва
 Москва 	2. Киров
 Киров 	
	
	• Москва
 Москва 	• Киров
 Киров 	
	

Ссылки на другую страницу	<code></code> текст <code></code>	<code></code> Страница 2 <code></code>				
<i>Оформление таблиц</i>						
Определить таблицу	<code><TABLE ALIGN = LEFT RIGHT CENTER</code> <code>WIDTH = n% ></code> теги таблицы <code></TABLE></code>	n% – ширина таблицы в процентах по отношению к ширине страницы				
Строка таблицы	<code><TR></code> теги ячеек <code></TR></code>					
Ячейка таблицы	<code><TD></code> содержимое ячейки <code></TD></code>					
Выравнивание текста, установка ширины и цвета заливки ячейки	<code><TD ALIGN=LEFT RIGHT CENTER</code> <code>WIDTH = ширина</code> <code>BACKGROUND-COLOR = цвет</code> <code>></code> содержимое ячейки <code></TD></code>	ширина – ширина ячейки (в пикселях или в процентах от ширины строки); цвет – цвет заливки ячейки (RED, BLUE, GREEN и т. д.)				
Заголовков столбца или строки таблицы	<code><TH></code> текст <code></TH></code>	Текст в ячейке выравнивается по центру, устанавливается жирный шрифт				
<i>Пример</i>	<code><TABLE ALIGN =CENTER WIDTH=50% ></code> <code><TR >Товар</TR></code> <code><TH>Цена</TH></code> <code></TR></code> <code><TR >Телефон</TR></code> <code><TD>2000 </TD></code> <code></TR></code> <code></TABLE></code>	<i>Результат</i> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Товар</td> <td>Цена</td> </tr> <tr> <td>Телефон</td> <td>2000</td> </tr> </table>	Товар	Цена	Телефон	2000
Товар	Цена					
Телефон	2000					

Учебное издание

Содержание

Предисловие	3
1. МОДЕЛИ РЕШЕНИЯ ФУНКЦИОНАЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ	8
1.1. Понятия «модель» и «моделирование». Классификация моделей	9
1.2. Компьютерное математическое моделирование	18
1.3. Использование в моделировании средств компьютерной графики	28
1.3.1. Разновидности компьютерной графики	28
1.3.2. Применение компьютерной графики в моделировании	32
1.4. Моделирование средствами табличного процессора Microsoft Excel	36
1.5. Практические работы	44
<i>Практическая работа 1.1.</i> Применение в моделировании деловой графики и инструмента «Подбор параметра» MS Excel	44
<i>Практическая работа 1.2.</i> Использование функций работы с массивами Microsoft Excel в моделировании (на примере упрощенной модели межотраслевого баланса)	53
<i>Практическая работа 1.3.</i> Моделирование на основе метода Монте-Карло	60
<i>Практическая работа 1.4.</i> Решение задачи об оптимальном распределении ресурсов при выпуске продукции	71
1.6. Тестовые задания	84
2. БАЗЫ ДАННЫХ	86
2.1. Базы данных и модели данных. Основы проектирования реляционных баз данных	87
2.2. Основы работы с СУБД Microsoft Access. Основные объекты базы данных. Работа с таблицами	102
2.3. Основы работы с СУБД Microsoft Access. Работа с запросами, формами и отчетами	106
2.4. Практические работы	113
<i>Практическая работа 2.1.</i> Работа с таблицами и формами базы данных Microsoft Access	113
<i>Практическая работа 2.2.</i> Работа с запросами на выборку в базе данных Microsoft Access	122

2.5. Индивидуальное домашнее задание 1. Проектирование базы данных для заданной предметной области	138
2.6. Тестовые задания	143
3. ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ. ЗАЩИТА ИНФОРМАЦИИ В СЕТЯХ	147
3.1. Компьютерные сети	148
3.2. Интернет	155
3.3. Организация защиты информации	161
3.4. Антивирусная защита компьютера. Защита информации в локальных и глобальных компьютерных сетях	167
3.4.1. Антивирусная защита компьютера	167
3.4.2. Защита информации в локальных и глобальных компьютерных сетях	171
3.5. Практические работы	174
<i>Практическая работа 3.1. Создание простейшего HTML-документа. Форматирование шрифта и абзаца</i>	<i>174</i>
<i>Практическая работа 3.2. Вставка в HTML-документ рисунков. Создание гиперссылок</i>	<i>178</i>
<i>Практическая работа 3.3. Организация защиты документов средствами пакета Microsoft Office</i>	<i>180</i>
<i>Практическая работа 3.4. Криптографические преобразования средствами Microsoft Excel</i>	<i>191</i>
3.6. Индивидуальное домашнее задание 2. Создание web-сайта	193
3.7. Тестовые задания	196
Библиографический список	199
Приложения	201

Учебное издание

*Глазова Вера Федоровна
Богданова Анна Владимировна
Тонких Артем Петрович*

ИНФОРМАТИКА

Учебно-методическое пособие

В двух частях

Часть 2

Редактор *Г.В. Данилова*
Технический редактор *З.М. Малявина*
Компьютерная верстка: *И.И. Шишкина*
Дизайн обложки: *Г.В. Карасева*

Подписано в печать 28.08.2013. Формат 60×84/16.
Печать оперативная. Усл. п. л. 12,96.
Тираж 100 экз. Заказ № 1-67-12.

Издательство Тольяттинского государственного университета
445667, г. Тольятти, ул. Белорусская, 14

