

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет
архитектуры и строительства»
(ПГУАС)

Т.А. Глебова, М.А. Чиркина, И.Г. Гвоздева

ИНФОРМАТИКА

Рекомендовано Редсоветом университета
в качестве учебного пособия для студентов,
обучающихся по направлению подготовки 38.03.03
«Управление персоналом»

Пенза 2016

УДК 681.32
ББК 32.973-018.1
Г53

Рецензент – генеральный директор ООО «Креомастер»
С.Н. Даянов

Глебова Т.А.
Г53 Информатика: учеб. пособие по направлению подготовки
38.03.03 «Управление персоналом» / Т.А. Глебова, М.А. Чиркина,
И.Г. Гвоздева. – Пенза: ПГУАС, 2016. – 124 с.

Изложены основные методы информатики и их реализация в пакетах прикладных программ. Пособие направлено на формирование компетенций, предусмотренных рабочей программой по дисциплине «Информатика».

Подготовлено на кафедре «Информационно-вычислительные системы» и предназначено для использования студентами, обучающимися по направлению подготовки 38.03.03 «Управление персоналом», при изучении дисциплины «Информатика».

© Пензенский государственный университет
архитектуры и строительства, 2016
© Глебова Т.А., Чиркина М.А.,
Гвоздева И.Г., 2016

ВВЕДЕНИЕ

В сегодняшних условиях стремительного развития компьютерной и телекоммуникационной техники требования к интеллектуальному, нравственному развитию человека должны существенно возрасти – человеку необходимо «опережать» компьютерный прогресс, выступая «заказчиком» новых компьютерных услуг.

От человека, живущего в мире информатизации, требуется повышение его интеллектуального и гуманистического потенциала. Информатизация – это системно-деятельностный процесс овладения информацией как ресурсом управления и развития (А.Д. Урсул).

Процесс информатизации должен включать в себя три диалектически взаимосвязанных процесса:

- медиатизацию – процесс совершенствования средств сбора, хранения и распространения информации;
- компьютеризацию – процесс совершенствования средств поиска и обработки информации, а также
- интеллектуализацию – процесс развития знаний и способностей людей к восприятию и порождению информации, что закономерно обуславливает повышение интеллектуального потенциала общества, включая возможность использование средств искусственного интеллекта.

1. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

Информатика – это наука о методах создания, хранения, обработки и передачи информации с помощью технических средств. Термин информатика (от французского *informatique*) образована путем слияния начало слова *information* (информация) и конца слова *automatique* (автоматика), что означает «автоматизированная переработка информации».

Главная функция информатики заключается в разработке методов и средства преобразования информации и их использовании в организации технологического процесса переработки информации.

Задачи информатики состоят в следующем:

- исследование информационных процессов;
- разработка новой технологий переработки информации;
- решение научных и инженерных проблем создания, внедрения и пользования компьютерной техники и информационной технологии.

Понятие информации.

Термин «информация» происходит от латинского слова «*informatio*», что означает сведения, разъяснения, изложение. Несмотря на широкое распространение этого термина, понятие информации является одним из самых дискуссионных в науке. В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию *информация*, но пока это понятие во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности. Информация в исторической эволюции: сведения, передаваемые людьми устным, письменным или другим способом. Более узкое определение дается в технике, где это понятие включает в себя сведения, являющиеся объектом хранения, передачи и преобразования. Наиболее общее определение имеет место в философии, где под информацией понимается отражение реального мира.

Клод Шеннон, американский учёный, заложивший основы теории информации – науки, изучающей процессы, связанные с передачей, приёмом, преобразованием и хранением информации, – рассматривает информацию как снятую неопределенность наших знаний о чем-то.

Другие популярные определения:

- *Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний* (Н.В. Макарова);
- *Информация – это обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспособления к нему наших чувств* (Норберт Винер)
- *Информация – это отрицание энтропии* (Леон Бриллюэн);
- *Информация – это мера сложности структур* (Моль);

- *Информация – это отраженное разнообразие (Урсул);*
- *Информация – это содержание процесса отражения (Тузов);*
- *Информация – это вероятность выбора (Яглом).*

Информация содержится в человеческой речи, текстах книг, журналов и газет, сообщениях радио и телевидения, показаниях приборов и т. д. Человек воспринимает информацию с помощью органов чувств, хранит и перерабатывает ее. Передаваемая информация обычно касается каких-то предметов и связана с событиями, происходящими в окружающем нас мире. В рамках науки информация является первичным и неопределяемым понятием. Оно предполагает наличие материального носителя информации, источника информации, передатчика информации, приемника и канала связи между источником и приемником. Понятие информации используется во всех сферах. Конкретное толкование элементов, связанных с понятием информации, зависит от метода конкретной науки, цели исследования или просто от наших представлений.

С понятием информации связаны такие понятия, как сигнал, сообщение и данные.

Сигнал представляет собой любой процесс, несущий информацию.

Сообщение – это информация, представленная в определенной форме и предназначенная для передачи.

Данные – это информация, представленная в формализованном виде и предназначенная для обработки ее техническими средствами, например, ЭВМ.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодированных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объем сообщения.

Различают две формы представления информации – непрерывную и дискретную. Поскольку носителями информации являются сигналы, то в качестве последних могут использоваться физические процессы различной природы. Например, процесс протекания электрического тока в цепи.

По месту возникновения информацию можно разделить на входную, то есть поступающую на вход какой-то системы, и выходную, покидающую данную систему. По отношению к той же системе информация может быть внутренней, возникающей внутри ее, и внешней, существующей за пределами системы.

В процессе обработки информация может менять структуру и форму. Признаком структуры являются элементы информации и их взаимосвязь. Формы представления информации могут быть различны. Основными из них являются: символьная (основана на использовании различных симво-

лов), текстовая (текст – это символы, расположенные в определенном порядке), графическая (различные виды изображений), звуковая.

В повседневной практике такие понятия, как информация и данные, часто рассматриваются как синонимы. На самом деле между ними имеются различия. Данными называется информация, представленная в удобном для обработки виде. Данные могут быть представлены в виде текста, графики, аудиовизуального ряда. Представление данных называется языком информатики, представляющим собой совокупность символов, соглашений и правил, используемых для общения, отображения, передачи информации в электронном виде.

Качество информации определяется ее свойствами, отвечающими потребностям пользователя.

Свойства информации:

- объективность,
- полнота,
- достоверность,
- адекватность,
- доступность,
- актуальность.

Понятие *объективности* информации является относительным, это понятно, если учесть, что методы являются субъективными. Более объективной принято считать ту информацию, в которую методы вносят меньший субъективные элемент.

Полнота информации во многом характеризует её качество и определяет достаточность данных для принятия решений или для создания новых данных на основе имеющихся. Чем полнее данные, тем шире диапазон методов, которые можно использовать, тем проще подобрать метод, вносящий минимум погрешностей в ход информационного процесса.

Данные возникают в момент регистрации сигналов, но не все сигналы являются «полезными» – всегда присутствует какой-то уровень посторонних сигналов, в результате чего полезные данные сопровождаются определённым уровнем «информационного шума». Если полезный сигнал зарегистрирован более чётко, чем посторонние сигналы, достоверность информации может быть более высокой. При увеличении уровня шумов достоверность информации снижается. В этом случае при передаче того же количества информации требуется использовать либо больше данных, либо более сложные методы.

Адекватность информации – степень соответствия реальному объективному состоянию дела. Неадекватная информация может образовываться при создании новой информации на основе неполных или недостоверных данных. Однако и полные, и достоверные данные могут приводить к соз-

данию неадекватной информации в случае применения к ним неадекватных методов.

Доступность информации – мера возможности получить ту или иную информацию. На степень доступности информации влияют одновременно как доступность данных, так и доступность адекватных методов для их интерпретации. Отсутствие доступа к данным или отсутствие адекватных методов обработки приводят к одинаковому результату: информация оказывается недоступной.

Актуальность информации – степень соответствия информации текущему моменту времени. Нередко с актуальностью, как и с полнотой, связывают коммерческую ценность информации. Поскольку информационные процессы растянуты во времени, то достоверная и адекватная, но устаревшая информация может приводить к ошибочным решениям. Необходимость поиска (или разработки) адекватного метода для работы с данными может приводить к такой задержке получения информации, что она становится неактуальной и ненужной. На этом, в частности, основаны многие современные системы шифрования данных с открытым ключом. Лица, не владеющие ключом (методом) для чтения данных, могут заняться поиском ключа, поскольку алгоритм его работы доступен, но продолжительность этого поиска столь велика, что за время работы информация теряет актуальность и, естественно связанную с ней практическую ценность.

Люди обмениваются информацией в форме сообщений. Сообщение – это форма представления информации в виде речи, текстов, жестов, взглядов, изображений, цифровых данных, графиков, таблиц и т.п. *Одно и то же информационное сообщение* (статья в газете, объявление, письмо, телеграмма, справка, рассказ, чертёж, радиопередача и т.п.) *может содержать разное количество информации для разных людей – в зависимости от их предшествующих знаний, от уровня понимания этого сообщения и интереса к нему.*

2. СИСТЕМЫ СЧИСЛЕНИЯ И ОСНОВЫ ЛОГИКИ

2.1. Представление информации.

Количество и единицы измерения информации

Количеством информации называют числовую характеристику сигнала, характеризующую *степень неопределенности* (неполноту знаний), которая исчезает после получения сообщения в виде данного сигнала. Эта мера неопределенности в теории информации называют *энтропией*. Если в результате получения сообщения достигается полная ясность в каком-то вопросе, говорят, что была получена полная или исчерпывающая информация, и необходимости в получении дополнительной информации нет.

В определенных, весьма широких условиях можно пренебречь качественными особенностями информации, выразить её количество числом, а также сравнить количество информации, содержащейся в различных группах данных.

За единицу измерения информации принимается такое количество информации, которое содержит сообщение, уменьшающее неопределенность (неполноту знаний) в два раза.

Количество информации, которое можно получить при ответе на вопрос типа «да – нет», называется *битом* (двоичная единица). *Бит – минимальная единица количества информации*, ибо получить информацию, меньшую, чем 1 бит невозможно.

В *вычислительной технике* битом называют наименьшую «порцию» памяти компьютера, необходимую для хранения одного из двух знаков «0» и «1», используемых для внутримашинного представления данных и команд.

Рассмотрим на примере, что означает 1 бит информации.

Если положить в мешок два шарика разного цвета, то, загадывая какой шар будет вынут, будем иметь два варианта. Причем, ни один из этих вариантов не имеет преимуществ перед другим. Таким образом, неопределенность знаний о цвете вынутого шара перед тем, как шар будет вынут, равна двум. После того как шар вынули, произошло одно из двух возможных событий. Неопределённость знаний уменьшилась в два раза: было два варианта, остался один. Значит, узнав цвет вынутого шара, получили 1 бит информации.

Бит – слишком мелкая единица измерения. На практике чаще применяется более крупная единица – **байт**, равная **восемь битам**. Именно восемь битов требуется для того, чтобы закодировать любой из 256 символов алфавита клавиатуры компьютера ($256=2^8$).

Широко используются также ещё **более крупные производные единицы информации**:

- 1 Килобайт (Кбайт) = 1024 байт = 2^{10} байт,
- 1 Мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт,
- 1 Гигабайт (Гбайт) = 1024 Мбайт = 2^{30} байт.
- 1 Терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт,
- 1 Петабайт (Пбайт) = 1024 Тбайт = 2^{50} байт.

В настоящее время получили распространение подходы к определению понятия «количество информации», основанные на том, что информацию, содержащуюся в сообщении, можно нестрого трактовать в смысле её новизны или, иначе, уменьшения неопределённости наших знаний об объекте. Эти подходы используют математические понятия *вероятности* и *логарифма*.

Подходы к определению количества информации. Формулы Хартли и Шеннона.

Американский инженер Р. Хартли в 1928 г. процесс получения информации рассматривал как выбор одного сообщения из конечного наперёд заданного множества из N равновероятных сообщений, а количество информации I , содержащееся в выбранном сообщении, определял как двоичный логарифм N .

Формула Хартли: $I = \log_2 N$.

Определим теперь, являются ли равновероятными сообщения «*первой выйдет из дверей здания женщина*» и «*первым выйдет из дверей здания мужчина*». Однозначно ответить на этот вопрос нельзя. Все зависит от того, о каком именно здании идет речь. Если это, например, станция метро, то вероятность выйти из дверей первым одинакова для мужчины и женщины, а если это военная казарма, то для мужчины эта вероятность значительно выше, чем для женщины.

Формулу для вычисления количества информации, учитывающую неодинаковую вероятность событий, предложил К. Шеннон в 1948 г. Количественная зависимость между вероятностью события p и количеством информации I в сообщении о нем выражается формулой:

$$I = -\log_2 p.$$

Связь между вероятностью события и количеством информации в сообщении об этом событии можно выразить следующим образом – чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.

Количество информации для событий с различными вероятностями p_1, p_2, \dots, p_n определяется по формуле (эту формулу также называют формулой Шеннона):

$$I = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N)$$

$$\text{или } I = - \sum_{i=1}^n p_i \cdot \log p_i,$$

где p_i – вероятность того, что именно i -е сообщение выделено в наборе из N сообщений.

Если вероятности p_1, \dots, p_N равны, то каждая из них равна $1/N$, и формула Шеннона превращается в формулу Хартли.

2.2. Системы счисления, используемые в компьютере. Представление чисел в памяти ЭВМ

Объем данных V_d в сообщении измеряется количеством символов (разрядов) в этом сообщении. В различных системах счисления один разряд имеет различный вес и соответственно меняется единица измерения данных.

Алфавит – упорядоченный набор символов, используемый для кодирования сообщений на некотором языке.

Мощность алфавита – количество символов алфавита. Алфавитный подход – объективный подход к измерению информации. Он удобен при использовании технических средств работы с информацией.

Чтобы определить объем информации в сообщении при алфавитном подходе, нужно последовательно решить задачи:

1. Определить количество информации I в одном символе по формуле $2^I = N$, где N – мощность алфавита
2. Определить количество символов в сообщении n
3. Вычислить объем информации по формуле: $V = i n$.

Семантическая мера информации. Используется для измерения смыслового содержания информации. Наибольшее распространение здесь получила тезаурусная мера, связывающая семантические свойства информации со способностью пользователя принимать поступившее сообщение.

Тезаурус – это совокупность сведений, которыми располагает пользователь или система.

Относительной мерой семантической информации может служить коэффициент содержательности C , который определяется как отношение количества семантической информации к ее объему:

$$C = \frac{I_c}{V_d}.$$

Прагматическая мера информации. Определяет ее полезность, ценность для достижения пользователем поставленной цели. Эта мера также величина относительная, обусловленная особенностями использования этой информации в той или иной системе.

Информационным объемом сообщения называется количество двоичных символов, которое используется для кодирования этого сообщения.

Пусть M – количество символов (мощность) алфавита, в котором записано сообщение, N – количество символов в записи сообщения. Тогда информационный объем сообщения

$$I = N \log_2 M.$$

Если $\log_2 M$ не является целым числом, то его нужно округлить в большую сторону.

Системы кодирования.

Система кодирования в вычислительной технике называется двоичным кодированием и основана на представлении данных последовательностью всего двух знаков: 0 и 1. Эти знаки называются двоичными цифрами.

Одним битом могут быть выражены два понятия: 0 или 1. Если количество битов увеличить до двух, то уже можно выразить четыре различных понятия: 00 01 10 11.

Количество значений, в данной системе может быть выражено

$$N = 2^m,$$

где N – количество независимых кодируемых значений; m – разрядность двоичного кодирования, принятая в данной системе.

Кодирование информации.

Кодировка представляет собой таблицу символов, где каждой букве алфавита (а также цифрам и специальным знакам) присвоен свой уникальный номер – код символа.

Для того чтобы закодировать один символ, число, звук и т.д. естественно не обойтись одним битом. Поэтому биты объединяются в байты.

В одном байте можно закодировать $2^8 = 256$ комбинаций состояний битов от 00000000 до 11111111.

Кодирование целых и действительных чисел.

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). Шестнадцать бит позволяют закодировать целые числа от 0 до 65 535, а 24 бита – уже более 16,5 миллионов разных значений.

Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразуется в нормализованную форму:

$$3,1415926 = 0,31415926 \cdot 10^1$$

$$300\,000 = 0,3 \cdot 10^6$$

$$123\,456\,789 = 0,123456789 \cdot 10^{10}$$

Первая часть числа называется мантиссой, а вторая – характеристикой. Большую часть из 80 бит отводят для хранения мантиссы (вместе со зна-

ком) и некоторое фиксированное количество разрядов отводят для хранения характеристики (тоже со знаком).

Кодирование текстовых данных.

Если каждому символу алфавита сопоставить определенное целое число (например, порядковый номер), то с помощью двоичного кода можно кодировать и текстовую информацию. Восемью двоичных разрядов достаточно для кодирования 256 различных символов.

Для английского языка, Институт стандартизации США (*ANSI – American National Standard Institute*) ввел в действие систему кодирования ASCII (*American Standard Code for Information Interchange* – стандартный код информационного обмена США). В системе ASCII закреплены две таблицы кодирования – базовая и расширенная. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255.

Аналогичные системы кодирования текстовых данных были разработаны и в других странах. Отсутствие единого стандарта в этой области привело к множественности одновременно действующих кодировок. Только в России можно указать три действующих стандарта кодировки и еще два устаревших.

Так, например, кодировка символов русского языка, известная как кодировка *OS Windows-1251*, была введена «извне» – компанией *Microsoft*. Эта кодировка используется на большинстве локальных компьютеров, работающих на платформе *OS Windows*.

Другая распространенная кодировка носит название КОИ-8 (код обмена информацией, восьмизначный). Сегодня кодировка КОИ-8 имеет широкое распространение в компьютерных сетях на территории России и в российском секторе Интернета.

Международный стандарт, в котором предусмотрена кодировка символов русского алфавита, носит название кодировки *ISO (International Standard Organization* – Международный институт стандартизации). На практике данная кодировка используется редко.

В связи с избытком систем кодирования текстовых данных, действующих в России, возникает задача межсистемного преобразования данных – это одна из распространенных задач информатики.

Универсальная система кодирования текстовых данных.

Эта система, основана на 16-разрядном кодировании символов, получила название – *UNICODE*. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65 536 различных символов – этого поля достаточно для размещения в одной таблице символов большинства языков планеты.

Кодирование графических данных.

Графическая информация представляется в растровом или векторном форматах. В векторных форматах записаны команды для принтера или плоттера, такой способ кодирования графических документов (по существу в виде команд, хранимых в текстовом файле) компактен, точен, позволяет легко изменять масштаб изображения, но применяется только для представления формализованной информации – графиков, схем, чертежей, карт.

Растровый способ позволяет закодировать любое изображение – картину, фотографию и т.д. Для каждой точки запоминается цвет. Файл в таком формате получается очень большим, поэтому все форматы хранения графических образов предусматривают сжатие информации, которое может осуществляться без потерь (*bmp, gif, psx, tiff*) или с потерями (*jpeg*). Формат *jpeg* уменьшает размер данных в десятки раз с потерями, которые не заметны для человеческого глаза.

Применяют несколько систем кодирования: *HSB, RGB* и *СМУК*. Первая цветовая модель проста и интуитивно понятна, т.е. удобна для человека, вторая наиболее удобна для компьютера, а последняя модель *СМУК* – для типографий.

Модель *HSB* характеризуется тремя компонентами:

- оттенок цвета (*Hue*), Значение цвета выбирается как вектор, выходящий из центра окружности. Направление вектора задается в угловых градусах и определяет цветовой оттенок.
- насыщенность цвета (*Saturation*). Насыщенность цвета определяется длиной вектора,
- яркость цвета (*Brightness*). Яркость цвета задается на отдельной оси, нулевая точка которой имеет черный цвет. Точка в центре соответствует белому (нейтральному) цвету, а точки по периметру – чистым цветам

Принцип метода *RGB* заключается в следующем: известно, что любой цвет можно представить в виде комбинации трех цветов: красного (*Red, R*), зеленого (*Green, G*), синего (*Blue, B*). Другие цвета и их оттенки получаются за счет наличия или отсутствия этих составляющих. По первым буквам основных цветов система и получила свое название – *RGB*. Данная цветовая модель является аддитивной, то есть любой цвет можно получить сочетание основных цветов в различных пропорциях.

- Метод *СМУК*. Эта цветовая модель используется при подготовке публикаций к печати. Каждому из основных цветов ставится в соответствие дополнительный цвет (дополняющий основной до белого). Получают дополнительный цвет за счет суммирования пары остальных основных цветов.

В основном такой метод принят в полиграфии. Но там еще используют черный цвет (*Black*, так как буква *B* уже занята синим цветом, то обозначают буквой *K*). Это связано с тем, что наложение друг на друга дополнительных цветов не дает чистого черного цвета.

Различают несколько режимов представления цветной графики:

а) полноцветный (*True Color*) – для кодирования яркости каждой из составляющих используют по 256 значений;

б) *High Color* – это кодирование при помощи 16-разрядных двоичных чисел;

в) индексный – к графическим данным прилагается палитра (справочная таблица), без которой воспроизведение будет неадекватным. Сам код точки растра в данном случае означает не сам по себе цвет, а только его номер (индекс) в палитре.

Чтобы рассчитать объем видеопамати, надо количество экранных точек (пикселей) умножить на глубину цвета.

Кодирование звуковой информации

Можно выделить два основных направления кодирования звуковой информации.

Метод *FM (Frequency Modulation)* основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а следовательно, может быть описан числовыми параметрами, то есть кодом.

В настоящее время наиболее популярны форматы сжатия без потерь – *WAV, RA*, сжатия с потерями – *MP3, VQF*. При сжатии *MPEG*’ом в 11-12 раз для записи одной секунда звучания требуется 16 Кбайт, т.е. в 1 Гб памяти влезает 18 часов музыки. Такое сжатие считается оптимальным, при большем сжатии потери становятся заметными.

Метод таблично-волнового (*Wave-Table*) синтеза заключается в том, что где-то в заранее подготовленных таблицах хранятся образцы звуков для множества различных музыкальных инструментов. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звука

Качество двоичного кодирования звука определяется частотой дискретизации и глубиной кодирования.

Для определения информационного объема звукового файла нужно глубину кодирования умножить на частоту дискретизации и на длительность звучания (в секундах), а в случае стереоаудиофайла еще и на два.

Видео информация – кодируется аналогично графической и звуковой.

Системы счисления.

Совокупность приемов наименования и записи чисел называется *счислением*. Под *системой счисления* понимается способ представления любого числа с помощью ограниченного алфавита символов, называемых цифрами. Счисление представляет собой частный случай кодирования, где слово, записанное с использованием определенного алфавита и по определенным правилам, называется *кодом*. Применительно к счислению это *код числа*.

Системы счисления делятся на непозиционные и позиционные.

Непозиционная система счисления – система, в которой значение символа не зависит от его положения в числе. Непозиционные системы счисления возникли раньше позиционных систем. Они использовались в древности римлянами, египтянами, славянами и другими народами. Например, запись *MDCLXVIII* означает, что записано число 1668 (*M* – тысяча, *D* – пятьсот, *C* – сто, *L* – пятьдесят, *X* – десять, *V* – пять, *I* – единица и т.д.).

Основной недостаток непозиционных систем – большое число различных знаков и сложность выполнения арифметических операций.

Позиционная система счисления – система, в которой значение символа зависит от его места в ряду цифр, изображающих число.

Основание (базис) позиционной системы счисления – количество знаков или символов, используемых в разрядах для изображения числа в данной системе счисления (q).

Так для записи числа в десятичной системе счисления используются 10 цифр: **0,1,2,3,4,5,6,7,8,9**, в двоичной системе – две цифры: **0** и **1** и т.д. Каждой позиции символа в записи числа, которое в общем случае можно записать в виде десятичной дроби, соответствует свой весовой множитель. Пусть основание системы счисления равно a , тогда весовой множитель первого символа, стоящего слева от точки, разделяющей целую и дробную часть равен a^0 . Весовой множитель второго символа слева от точки равен a^1 и т.д. Соответственно весовые множители символов, стоящих справа от точки будут равны a^{-1} , a^{-2} и т.д. Например, число 23,45 в десятичной системе счисления ($a=10$) можно записать как

$$23,45_{10} = 2 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 5 \cdot 10^{-2} = 20 + 3 + 0,4 + 0,05.$$

Та же самая запись числа в восьмеричной системе будет выглядеть так:

$$23,45_8 = 2 \cdot 8^1 + 3 \cdot 8^0 + 4 \cdot 8^{-1} + 5 \cdot 8^{-2}.$$

Наибольшее распространение получила *двоичная система счисления*. В этой системе для представления любого числа используются два символа – цифры *0* и *1*. Основание системы счисления $q = 2$.

Основание	Система счисления	Знаки
2	<i>Двоичная</i>	0, 1
3	Троичная	0, 1, 2
4	Четверичная	0, 1, 2, 3
5	Пятеричная	0, 1, 2, 3, 4
8	<i>Восьмеричная</i>	0, 1, 2, 3, 4, 5, 6, 7
10	Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
12	Двенадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
16	<i>Шестнадцатеричная</i>	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Правило 1.

Для перевода целого числа A из десятичной системы счисления в систему с основанием q необходимо A разделить с остатком (нацело) на число q , записанное в десятичной системе. Затем неполное частное, полученное от деления, нужно снова разделить с остатком на q и т.д., пока последнее полученное неполное частное не станет равным нулю. Представлением числа A в новой системе счисления будет последовательность остатков деления, изображенных q -й цифрой и записанных в порядке, обратном порядку их получения.

Правило 2.

Для перевода дроби из десятичной системы счисления в систему с основанием q необходимо последовательно выполнять умножение исходной дроби и получаемых дробных произведений на основание системы счисления q до тех пор, пока не получится нулевая дробная часть или не будет достигнута требуемая точность вычислений. Представлением дроби в новой системе счисления будет последовательность полученных целых частей произведения, записанных в порядке их получения.

Например, перевести число $37,45_{10}$ из десятичной системы счисления в двоичную систему счисления.

$ \begin{array}{r} 37 \overline{) 2} \\ \underline{-36} \\ 1 \overline{) 18} \\ \underline{-18} \\ 0 \overline{) 9} \\ \underline{-8} \\ 1 \overline{) 4} \\ \underline{-4} \\ 0 \overline{) 2} \\ \underline{-2} \\ 0 \end{array} $	$ \begin{array}{r} 0, \overline{) 45} \\ \underline{\times 2} \\ 0 \overline{) 90} \\ \underline{\times 2} \\ 1 \overline{) 80} \\ \underline{\times 2} \\ 1 \overline{) 60} \\ \underline{\times 2} \\ 1 \overline{) 20} \\ \vdots \dots \end{array} $
<p>остатки 0 1 0 1 0 1 — последнее частное</p>	
<p>целая часть $X_{(2)} = 100101$</p>	<p>дробная часть $X_{(2)} = 0,0111\dots$</p>

Перевод чисел из десятичной системы счисления в восьмеричную и шестнадцатеричную производится (по аналогии с двоичной системой счисления) с помощью делений и умножений на 8 или 16.

Перевод чисел из одной произвольной системы в другую.

В этом случае сначала следует выполнить перевод числа в десятичную систему, а затем из десятичной в требуемую.

Особым способом выполняется перевод чисел для систем с кратными основаниями.

Пусть p и q – основания двух систем счисления. Будем называть эти системы системами счисления с кратными основаниями, если $p = qn$ или $q = pn$, где n – натуральное число. Так, например, системы счисления с основаниями 2 и 8 являются системами счисления с кратными основаниями.

Пусть $p = qn$ и требуется перевести число из системы счисления с основанием q в систему счисления с основанием p . Разобьем целую и дробную части записи числа на группы по n последовательно записанных цифр влево и вправо от запятой. Если количество цифр в записи целой части числа не кратно n , то надо дописать слева соответствующее количество нулей. Если количество цифр в записи дробной части числа не кратно n , то нули дописываются справа. Каждая такая группа цифр числа в старой системе счисления будет соответствовать одной цифре числа в новой системе счисления.

Пример:

Переведем $1100001,111_2$ в четверичную систему счисления.

Дописав нули и выделив пары цифр, получим $01100001,1110_2$.

Теперь выполним перевод отдельно каждой пары цифр, пользуясь пунктом Перевод чисел из одной произвольной системы в другую.

$$01_2 = 1_{10} = 1_4$$

$$10_2 = 2_{10} = 2_4$$

$$00_2 = 0_{10} = 0_4$$

$$01_2 = 1_{10} = 1_4$$

$$11_2 = 3_{10} = 3_4$$

$$10_2 = 2_{10} = 2_4$$

Итак, $1100001,111_2 = 01100001,1110_2 = 1201,324$.

Пусть теперь требуется выполнить перевод из системы с большим основанием q , в систему с меньшим основанием p , т.е. $q = pn$. В этом случае одной цифре числа в старой системе счисления соответствует n цифр числа в новой системе счисления.

Пример: Выполним проверку предыдущего перевода числа.

$$1201,324 = 1100001,1110_2 = 1100001,111_2$$

В шестнадцатеричной системе есть цифры с числовыми значениями 10, 11, 12, 13, 14, 15. Для их обозначения используют первые шесть букв латинского алфавита A, B, C, D, E, F .

Приведем таблицу чисел от 0 до 16, записанных в системах счисления с основаниями 10, 2, 8 и 16.

Число в десятичной системе счисления	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
В восьмеричной	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
В двоичной	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
В шестнадцатеричной	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

Пример: Переведем число $1101010010101010100,11_2$ в шестнадцатеричную систему счисления.

Воспользуемся кратностью оснований систем счисления ($16=2^4$). Сгруппируем цифры по четыре, дописав, слева и справа нужное количество нулей

$$0001101010010101010100,1100_2$$

и, сверяясь с таблицей, получим: $1A9554,C_{16}$.

Арифметические операции в позиционных системах счисления

Правила выполнения арифметических действий для всех позиционных систем счисления одинаковы и совпадают с правилами для десятичной системы счисления.

Сложение

Если результат сложения двух цифр в системе счисления с основанием q больше q (т. е. полученное число двузначное), то старшая цифра результата равна 1. Таким образом, при сложении в следующий разряд может переходить только единица, а результат сложения в любом разряде будет меньше, чем q . Результат сложения двух положительных чисел имеет столько же значащих цифр, что и максимальное из двух слагаемых, либо на одну цифру больше, но этой цифрой может быть только единица.

Примеры

Сложить числа:

$$100110011_2 + 1101001_2 = 110011100_2.$$

Вычитание

Если необходимо вычесть из цифры a цифру b и $a < b$, то в столбце b таблицы сложения ищем значение числа a . Самая левая цифра в строке, в которой найдено значение числа a , и будет результатом вычитания. Если же $a < b$, то нужно заимствовать единицу из левого разряда, поэтому в столбце ищем число $1a$, и левая цифра в соответствующей строке будет результатом вычитания.

Примеры

Выполнить вычитание чисел:

$$1100000,001_2 - 101101,1_2 = 110010,101_2.$$

Умножение

Умножение выполняется столбиком с использованием соответствующих таблиц умножения и сложения.

Отметим, что во всех позиционных системах счисления с любым основанием q умножение на числа вида q^m , где m – целое число, сводится просто к перенесению запятой умножаемого на m разрядов вправо или влево (в зависимости от знака m), так же, как и в десятичной системе счисления.

Примеры

Выполнить умножение чисел:

$$10011_2 \cdot 100101_2 = 101011111_2.$$

Деление

Как для умножения, так и для деления нужны обе таблицы – умножения и сложения в соответствующей системе счисления. Само деление выполняется уголком с последующим вычитанием сомножителей.

Выполнить деление:

$$110101112 : 1010112 = 1012$$

2.3. Алгебра логики. Основные логические операции.

Основные законы преобразования алгебры логики.

Логические основы ЭВМ

Алгебра логики – это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Алгебра логики возникла в середине 19 в. в трудах английского математика Джорджа Буля. Ее создание представляло собой попытку решать традиционные логические задачи алгебраическими методами. **Логическое высказывание** – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно или ложно. Так, например, предложение «6 – четное число» следует считать высказыванием, так как оно истинно. Предложение «Рим – столица Франции» тоже высказывание, так как оно ложно. Не всякое предложение является логическим высказыванием. **Высказывательная форма** – это повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную или становится высказыванием, когда все переменные замещаются своими значениями. Алгебра логики рассматривает любое высказывание только с одной точки зрения – является оно ли истинным или ложным. Употребляемые в обычной речи слова и словосочетания «не», «и», «или», «если..., то», «тогда и только тогда» и др. позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками**. Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**. Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначения:

1. Операция, выражаемая словом «не», называется отрицанием и обозначается: **НЕ А, \bar{A} , $\neg A$, NOT A.**

Определение отрицания может быть записано с помощью таблицы истинности:

A	$\neg A$
1	0
0	1

2. Операция, выражаемая связкой «и», называется конъюнкцией (от лат. – соединение) или логическим умножением и обозначается: **A и B , $A \wedge B$, $A \& B$, $A \cdot B$**

Определение конъюнкции может быть записано с помощью таблицы истинности:

A	B	$A \& B$
1	1	1
1	0	0
0	1	0
0	0	0

3. Операция, выражаемая связкой «или», называется дизъюнкцией или логическим сложением и обозначается: **A или B , $A \vee B$, $A | B$, $A + B$**

Определение дизъюнкции может быть записано с помощью таблицы истинности:

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

4. Операция, выражаемая связками «если ..., то», «из ... следует», «...влечет ...», называется импликацией и обозначается знаком \rightarrow .

Определение импликации может быть записано с помощью таблицы истинности:

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

5. Операция, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «... равносильно ...», называется эквиваленцией или двойной импликацией и обозначается: $A \leftrightarrow B$, $A \equiv B$, $A \sim B$.

Определение эквиваленции может быть записано с помощью таблицы истинности:

A	B	$A \leftrightarrow B$
1	1	1
1	0	0
0	1	0
0	0	1

Логическая формула

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, т.е. заменить формулой. Дадим определение **логической формуле**:

1. Всякая логическая переменная и символы «истинна» («1») и ложь («0») – формулы.

2. Если A и B – формулы, то A , $(A*B)$, $(A+B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$ – формулы.

3. Никаких других формул в алгебре логики нет. В п. 1 определены **элементарные формулы**, в п. 2 даны **правила образования из любых данных формул новых формул**.

Для каждого составного высказывания (логического выражения) можно построить *таблицу истинности*, которая определяет истинность или ложность логического выражения при всех возможных комбинациях исходных значений простых высказываний (логических переменных).

При построении таблиц истинности целесообразно руководствоваться определенной последовательностью действий:

1) записать выражение и определить порядок выполнения операций

2) определить количество строк в таблице истинности. Оно равно количеству возможных комбинаций значений логических переменных, входящих в логическое выражение (определяется по формуле $Q = 2^n$, где n – количество входных переменных)

3) определить количество столбцов в таблице истинности (= количество логических переменных + количество логических операций)

Операции в логическом выражении выполняются слева направо с учетом скобок в следующем порядке:

1. инверсия;

2. конъюнкция;

3. дизъюнкция;

4. импликация и эквивалентность.

Для изменения порядка выполнения логических операций используются круглые скобки.

Пример. Составить таблицу истинности для выражения

$$F = (A \vee B) \& (\neg A \vee \neg B)$$

A	B	$A \vee B$	$\neg A$	$\neg B$	$\neg A \vee \neg B$	F
1	1	1	0	0	0	0
1	0	1	0	1	1	1
0	1	1	1	0	1	1
0	0	0	1	1	1	0

Логические законы и правила преобразования логических выражений

Если логическое выражение содержит большое число операций, то составлять для него таблицу истинности достаточно сложно.

В таких случаях формулы удобно привести к нормальной форме.

Формула имеет нормальную форму, если в ней отсутствуют знаки эквивалентности, импликации, двойного отрицания, при этом знаки отрицания находятся только при логических переменных.

Для приведения формулы к нормальной форме используют законы логики и правила логических преобразований.

Закон непротиворечия $A \& A = 0.$

Закон исключения третьего $A \vee A = 1.$

Закон двойного отрицания $A = A.$

Законы де Моргана $A \vee B = \neg(\neg A \& \neg B).$

(законы общей инверсии) $A \& B = \neg(\neg A \vee \neg B).$

Правило коммутативности:

$$A \& B = B \& A,$$

$$A \vee B = B \vee A.$$

Правило ассоциативности:

$$(A \& B) \& C = A \& (B \& C),$$

$$(A \vee B) \vee C = A \vee (B \vee C).$$

Правило дистрибутивности:

$$(A \& B) \vee (A \& C) = A \& (B \vee C),$$

$$(A \vee B) \& (A \vee C) = A \vee (B \& C).$$

Правила равносильности:

$$A \vee A = A,$$

$$A \& A = A.$$

Правила исключения констант:

$$A \vee 1 = 1,$$

$$A \vee 0 = A,$$

$$A \& 1 = A,$$

$$A \& 0 = 0.$$

Пример. Упростить логическое выражение

$$(A \& B) \vee (A \& B).$$

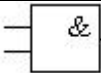
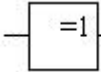
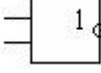


Решение.

$$(A \& B) \vee (A \& B) = A \& (B \vee B) = A \& 1 = A.$$

Логический элемент компьютера

Логический элемент компьютера – часть электронной логической схемы, которая реализует элементарную логическую функцию. Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ и др. (называемые также **вентильями**), а также **триггер**. С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентиляей бывает от двух до восьми входов и один или два выхода. Чтобы представить два логических состояния «1» и «0» в вентиляях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например, +5 вольт и 0 вольт. Высокий уровень обычно соответствует значению «истина» («1»), а низкий- значению «ложь» (0). Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем. Работу логических элементов описывают с помощью таблиц истинности. **Таблица истинности** – это табличное представление логической схемы, в котором перечислены все возможные сочетания значений истинности входных сигналов вместе со значением истинности выходного сигнала для каждого из этих сочетаний.

Логические элементы

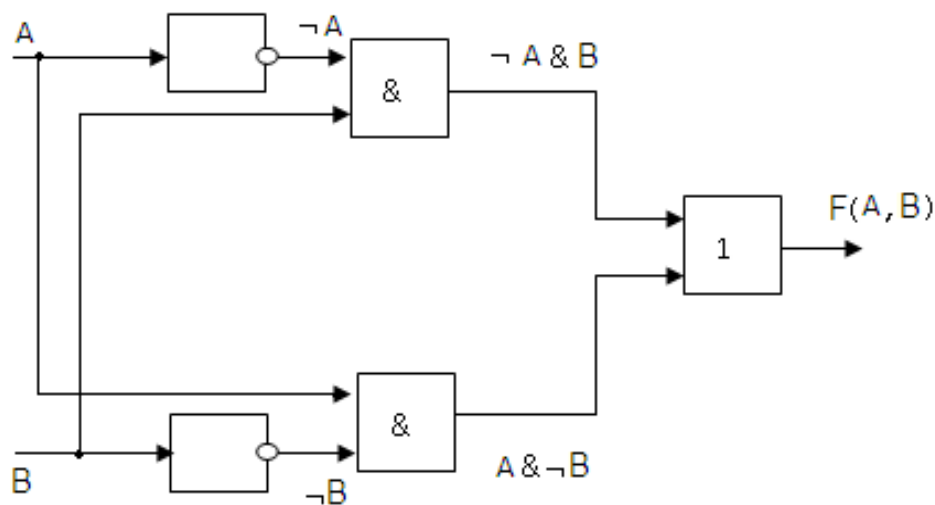
Функция	Операционные символы	Обозначения, названия
$F_1 = x_1 \cdot x_2$	$x_1 \wedge x_2$	 И – лог. умножитель
$F_6 = \bar{x}_1 x_2 + x_1 \bar{x}_2$	$x_1 \oplus x_2$	 Исключающее ИЛИ
$F_7 = x_1 + x_2$	$x_1 \vee x_2$	 ИЛИ – лог. сумматор
$F_8 = \overline{x_1 + x_2}$	$x_1 \downarrow x_2$	 ИЛИ – НЕ, функция Пирса
$F_{10} = \bar{x}_2$	\bar{x}_2	 НЕ – инвертор x_2
$F_{11} = x_1 + \bar{x}_2$	$x_2 \rightarrow x_1$	 Импликатор
$F_{14} = \overline{x_1 \cdot x_2}$	$x_1 x_2$	 И – НЕ, функция Шеффера

Алгоритм построения логических схем.

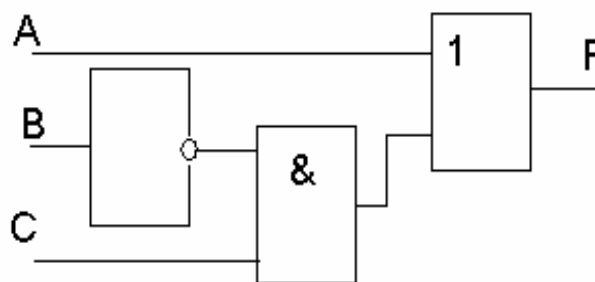
- Определить число логических переменных.
- Определить количество логических операций и их порядок.
- Изобразить для каждой логической операции соответствующий ей логический элемент.
- Соединить логические элементы в порядке выполнения логических операций.

Пример 1. По заданной логической функции построить логическую схему.

$$F = (\neg A \ \& \ B) \vee (A \ \& \ \neg B)$$



Пример 2. Записать формулу для данной логической схемы и построить таблицу истинности:



$$F = (\neg B \wedge C) \vee A$$

A	B	C	$\neg B$	$\neg B \wedge C$	$(\neg B \wedge C) \vee A$
0	0	0	1	0	0
1	0	0	1	0	1
0	1	0	0	0	0
0	0	1	1	1	1
1	1	0	0	0	1
1	0	1	1	1	1
0	1	1	0	0	0
1	1	1	0	0	1

3. УСТРОЙСТВО КОМПЬЮТЕРА

3.1. Основные устройства компьютера

В вычислительной системе принято рассматривать отдельно две основные и взаимосвязанные части: аппаратное и программное обеспечение.

К *аппаратному обеспечению* относятся устройства и приборы, входящие в состав вычислительной системы. Это компьютеры, системы передачи данных и т.п.

К программному обеспечению относятся программы, обеспечивающие функционирование вычислительной системы. Программа представляет упорядоченную последовательность команд (инструкций) для компьютера.

Программное и аппаратное обеспечение компьютера работают в непосредственной связи и непрерывном взаимодействии.

Архитектура, состав и назначение основных элементов персонального компьютера.

По определению, архитектура – это описание сложной системы, состоящей из множества элементов, как единого целого.

Архитектура современного персонального компьютера является обобщением принципов построения ЭВМ, предложенным группой ученых во главе с Джоном фон Нейманом. В основу построения большинства компьютеров положены следующие принципы, сформулированные Джоном фон Нейманом:

- принцип использования двоичной системы представления данных;
- принцип программного управления;
- принцип однородности памяти;
- принцип адресности.

В классической архитектуре фон – неймановской ЭВМ можно выделить 5 основных блоков, показанных на рис. 1.

С помощью устройств ввода (УВв) данные и программы, представленные в двоичной форме, попадают в оперативно-запоминающее устройство (ОЗУ). Для реализации команд, образующих программу, используется арифметико-логическое устройство (АЛУ), выполняющее арифметические операции, операции сравнения, алгебры логики и др. Взаимодействие ОЗУ и АЛУ осуществляет устройство управления (УУ). С его помощью программа из ОЗУ передается в АЛУ, отыскиваются нужные данные, выполняются вычисления, происходит запись в память и организуется выдача результата посредством устройства вывода (УВыв).

Реальная структура современного компьютера значительно сложнее, что обусловлено стремлением к повышению его производительности и функциональных возможностей.



Рис. 1. Классическая архитектура фон-неймановской ЭВМ

Так, в структуре персональной ЭВМ появилась кэш-память, введены каналы прямого доступа к оперативной памяти, используемые для обмена данными с устройствами ввода/вывода, минуя микропроцессор.

Периферийные устройства подключаются к аппаратуре компьютера через специальные контроллеры (К) или адаптеры (А) – устройства управления, освобождая процессор от непосредственного управления данным оборудованием.

В архитектуре персонального компьютера появился сопроцессор – устройство, функционирующее параллельно с главным процессором и выполняющее специфические операции: например, математический сопроцессор предназначен для сложных математических вычислений.

Системный блок является центральной частью ПК. В корпусе системного блока размещены внутренние устройства ПК. В состав системного блока входят следующие устройства:

- системная (материнская) плата с микропроцессором;
- оперативная память;
- накопитель на жестком магнитном диске;
- контроллеры или адаптеры для подключения и управления внешними устройствами ПК (монитором, звуковыми колонками и др.);
- порты для подключения внешних устройств (принтер, мышь и др.);
- внешние запоминающие устройства (ВЗУ).

Системная плата является интегрирующим (объединяющим) узлом ПК. Системная плата во многом определяет конфигурацию ПК, поскольку от ее параметров зависит тип используемого микропроцессора, максимальный

объем оперативной памяти, количество и способы подключения внешних устройств ПК и другие характеристики.

Микропроцессор (или процессор) – это главная микросхема компьютера. Он запускает программный код, находящийся в памяти, и управляет всеми устройствами компьютера либо напрямую, либо через соответствующие контроллеры.

Основой любого микропроцессора является ядро, которое состоит из миллионов транзисторов, расположенных на кристалле кремния.

Обмен данными и командами между внутренними устройствами ПК происходит по проводникам многожильного кабеля – системной шине. Основной задачей системной шины является передача данных между процессором и остальными электронными узлами компьютера. Различают три вида шин:

- шина данных;
- шина адреса;
- шина команд.

Шина данных. По этой шине происходит передача данных из оперативной памяти в *РОН* процессора и наоборот. Шина адреса. По этой шине передаются адреса ячеек оперативной памяти, где находятся команды, которые необходимо выполнить процессору. Кроме этого, по этой шине передаются данные, с которыми оперируют команды. В современных процессорах адресная шина 32-разрядная. Шина команд. По этой шине из оперативной памяти поступают команды, выполняемые процессором. Команды представлены в виде байтов. Простые команды занимают один байт, а более сложные – два, три и больше байтов. Большинство современных процессоров имеют 32-разрядную командную шину, хотя существуют 64-разрядные процессоры с 64-разрядной командной шиной.

Рассмотрим основные шинные интерфейсы системных плат, но более подробно остановимся на шине *USB*.

USB (Universal Serial Bus). Универсальная последовательная шина *USB* является обязательным элементом современного ПК, она пришла на смену устаревшим параллельным и последовательным портам. Шина *USB* представляет собой последовательный интерфейс передачи данных для средне- и низкоскоростных периферийных устройств. Она позволяет подключить до 256 разных устройств с последовательным интерфейсом. Шина *USB* поддерживает автоопределение новых устройств, а также так называемое «горячее» подключение, то есть подключение к работающему компьютеру без его перезагрузки. Скорость передачи данных по *USB* составляет 1,5 Мбит/с.

Все виды запоминающих устройств, расположенные на системной плате, образуют внутреннюю память ПК, к которой относятся:

- оперативная память;

- сверхоперативная память (кэш-память);
- постоянная память.

Оперативная память *RAM (Random Access Memory)* используется для хранения исполняемых в данный момент программ и необходимых для этого данных. Особенностью оперативной памяти является ее энергозависимость, т. е. данные хранятся в ней только при включенном компьютере.

Кэш память (*Cache memory*) используется для повышения быстродействия ПК. Принцип «кэширования» заключается в использовании быстродействующей памяти для хранения наиболее часто используемых данных или команд, при этом сокращается количество обращений к более медленной оперативной памяти.

Постоянная память *ROM* предназначена для хранения неизменяемой информации и размещается в микросхеме постоянного запоминающего устройства (ПЗУ). Микросхема ПЗУ способна продолжительное время сохранять информацию даже при отключенном компьютере, поэтому постоянную память называют энергонезависимой памятью.

Комплект программ, находящийся в ПЗУ, составляет базовую систему ввода/вывода *BIOS (Basic Input/Output System)*. *BIOS* содержит программы управления клавиатурой, видеокартой, дисками, портами и другими устройствами. Основное назначение этих программ состоит в том, чтобы проверить состав и работоспособность системы и обеспечить взаимодействие основных узлов ПК до загрузки какой-либо операционной системы. Кроме этого, в *BIOS* входит программа тестирования, которая выполняется при включении компьютера.

Память персонального компьютера подразделяется на внутреннюю и внешнюю.

Внутренняя память предназначена для временного хранения программ и обрабатываемых в текущий момент данных (оперативная память, кэш-память), а также для долговременного хранения информации о конфигурации ПК (ПЗУ). Внешняя память используется для длительного хранения большого объема данных и программ.

Внешняя память – это память, реализованная в виде внешних относительно материнской платы устройств с разными принципами хранения информации и типами носителей, предназначенных для долговременного хранения данных. В современных компьютерах используются три вида носителей: электрические, магнитные, оптические.

Жесткий диск (винчестер) состоит из одной или нескольких круглых пластин. Для хранения информации используются обе поверхности пластины. Дорожки одинакового радиуса составляют цилиндр.

Наименьшей физической единицей хранения данных является сектор. Размер сектора равен 512 байт. Группы секторов условно объединяются в кластеры. Кластер является наименьшей единицей адресации к данным.

Размер кластера, в отличие от размера сектора, не фиксирован и зависит от емкости диска.

Кластер – это минимальная единица размещения информации на диске. Каждому файлу выделяется участок памяти, кратный определенному количеству кластеров.

Перспективными носителями данных являются электронные схемы, использующие *Flash*- (флэш-) память. *Flash*-память относится к статической энергонезависимой памяти.

Устройства ввода и вывода (ввода/вывода) являются обязательными элементами любой ЭВМ, поскольку именно они обеспечивают взаимодействие пользователя с вычислительной системой.

С одной стороны, пользователь вводит команды или данные в компьютер через устройства ввода для их обработки, с другой стороны, вычислительная система выдает пользователю результаты своей работы посредством устройств вывода.

К основным устройствам ввода относятся клавиатура, мышь, сканер, джойстик, трекбол, графический планшет.

Сканер – это устройство оптического ввода, предназначенное для ввода в ПК черно-белых или цветных изображений, а также для считывания текста с бумажного носителя для последующей обработки. Сканированные оригиналы оцифровываются и после ввода в ПК обрабатываются с помощью специального программного обеспечения (например, для распознавания текста – программа *Fine Reader*), а затем сохраняются в виде текстового или графического файла. Сканеры подразделяются на планшетные и ручные.

В планшетных сканерах оригинал помещается на стекло, под которым перемещается оптико-электронное считывающее устройство. В барабанных сканерах оригинал протягивается барабаном мимо неподвижного считывающего устройства. Барабанные сканеры не позволяют сканировать книги, переплетенные брошюры и т. п. Ручной сканер плавно перемещается вручную по поверхности оригинала.

Основными характеристиками сканеров являются разрешающая способность и глубина цвета.

Разрешающая способность. Сканер воспринимает любой объект как набор отдельных точек (пикселей). Плотность пикселей (количество на единицу площади) называется разрешающей способностью сканера и измеряется в dpi (количество точек на дюйм). Качество сканирования зависит от оптической разрешающей способности (определяется количеством светочувствительных элементов матрицы на дюйм) и механической разрешающей способности (определяется дискретностью движения светочувствительного элемента или системы зеркал относительно листа). Существуют черно-белые и цветные сканеры. Современные цветные сканеры под-

держивают глубину цвета до 32 разрядов (т.е. цвет одного пикселя кодируется 32-разрядным двоичным словом) при разрешающей способности 1200 dpi.

К основным устройствам вывода относятся монитор и принтер.

Монитор (дисплей) – так же, как и клавиатура, является обязательным элементом ПК, предназначенным для вывода на экран текстовой и графической информации. Соответственно монитор может работать в двух режимах: текстовом и графическом. В текстовом режиме экран разбивается на знакоместа, на каждое из которых выводится один из 256 символов таблицы стандарта ASCII. В графическом режиме на экран выводятся изображения по пикселям. По принципу формирования изображения наиболее распространенными являются мониторы на электронно – лучевой трубке – *CRT (Cathode-Ray Tube)* и жидкокристаллические мониторы на тонкопленочных транзисторах – *TFT-LCD (Thin-Film Transistor Liquid-Crystal Display)*.

В основе *LCD* мониторов лежит технология формирования изображения с помощью жидких кристаллов. Жидкие кристаллы представляют собой молекулы, которые могут перетекать как жидкость. Эти молекулы пропускают свет, но под действием электрического заряда изменяют ориентацию.

Выводом изображения на экран монитора управляет специальное устройство – видеоадаптер (видеокарта).

Принтер – устройство вывода текста и графики на печать. По способу нанесения красителя на бумагу различают следующие виды принтеров: матричные, струйные, лазерные.

В матричном принтере изображение выводится на бумагу с помощью специальной движущейся головки, в которой содержится несколько (9, 24 или 48) иголок, наносящих удары по листу бумаги через красящую ленту.

В струйных принтерах красящее вещество (чернила) из специальной емкости выдувается на бумагу с помощью системы капиллярных распылителей, число которых в среднем от 16 до 400, а в некоторых моделях цветных струйных принтеров и более 400.

Лазерные принтеры являются самыми перспективными принтерами, обеспечивающими высокое качество (черно-белой) печати при низкой стоимости. Элементами лазерного принтера являются источник света (лазер), светочувствительный барабан (фотобарабан), красящий порошок (тонер) и блок термического закрепления тонера. В лазерных принтерах реализован метод электрофотографической печати.

Плоттер (графопостроитель) – устройство для вывода чертежей на бумагу. Устройство плоттеров аналогично устройству струйных принтеров, только они значительно больше по размерам. По конструкции плоттеры делятся на планшетные и рулонные. В планшетных плоттерах материал, на

который идет вывод, фиксируется, а печатающий узел перемещается в двух направлениях – по осям X и Y . В рулонных плоттерах бумага перемещается в вертикальном направлении, а печатающий узел – в горизонтальном направлении. Более высокое качество вывода графической информации обеспечивают планшетные плоттеры.

Устройства связи предназначены для приема и передачи (ввода и вывода) информации между двумя и более ПК. В зависимости от целого ряда параметров (типа линии связи, вида подключения, удаленности ПК и др.) используются различные устройства связи.

Модем (модулятор-демодулятор) – устройство, преобразующее информацию к виду, в котором ее можно передавать по телефонным линиям связи. Модемы выполняют цифроаналоговое преобразование цифровых сигналов ПК для их передачи по телефонной линии связи или аналогово-цифровое преобразование аналоговых сигналов из линии связи в цифровые сигналы для обработки в ПК. Модемы передают данные по обычным телефонным каналам со скоростью до 56000 бит в секунду. Также модемы осуществляют сжатие данных перед отправлением, поэтому их реальная скорость может превышать максимальную скорость модема.

Сетевой адаптер (сетевая плата) – электронное устройство, выполненное в виде платы расширения (адаптер может быть интегрирован в системную плату) с разъемом для подключения к линии связи. Сетевой адаптер используется для подключения ПК к локальной компьютерной сети.

3.2. Программное обеспечение компьютера. Операционная система

Программа – упорядоченная последовательность команд (инструкций) компьютера для решения задачи.

Программное обеспечение – совокупность программ обработки данных и необходимых для их эксплуатации документов.

Программы предназначены для машинной реализации задач. Термины задача и приложение имеют очень широкое употребление в контексте информатики и программного обеспечения.

Задача – проблема, подлежащая решению.

Приложение – программная реализация на компьютере решения задачи.

Программирование – теоретическая и практическая деятельность, связанная с созданием программ.

Классификация программных продуктов.

Для поддержки информационной технологии выделим три класса программных продуктов:

1. Системное программное обеспечение, выполняющее различные функции, например:

- управление ресурсами компьютера;
- создание копий используемой информации;
- проверка работоспособности устройств компьютера;
- выдача справочной информации о компьютере и др.

2. Прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;

3. Инструментарий технологии программирования, облегчающий процесс создания новых программ для компьютера.

Системное программное обеспечение (*System Software*) – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ, обеспечивает работу других программ, управляет аппаратными ресурсами вычислительной системы и т.д.

Системное программное обеспечение направлено:

– на создание операционной среды функционирования других программ;

– на обеспечение надежной и эффективной работы самого компьютера и вычислительной сети;

– на проведение диагностики и профилактики аппаратуры компьютера и вычислительных сетей;

– на выполнение вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и баз данных и т.д.).

Данный класс программных продуктов тесно связан с типом компьютера и является его неотъемлемой частью. Системные программные продукты в основном ориентированы на квалифицированных пользователей

Системное программное обеспечение – делится на базовое программное обеспечение, которое, как правило, поставляется вместе с компьютером, и сервисное программное обеспечение, которое может быть приобретено дополнительно.

Базовое программное обеспечение – минимальный набор программных средств, обеспечивающих работу компьютера и включает в себя: операционные системы, интерфейсные системы текстовые и графические, сетевые операционные системы.

Операционная система представляет собой комплекс программ и предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами компьютера.

ОС можно разбить на три основные группы: однозадачные (однопользовательские), многозадачные (многопользовательские) и сетевые.

Однозадачные ОС предназначены для работы одного пользователя в каждый конкретный момент с одной конкретной задачей. Типичным представителем таких ОС является *MS-DOS*, разработанная фирмой *Microsoft*.

Многозадачные ОС обеспечивают коллективное использование ЭВМ в мультипрограммном режиме, когда в памяти ЭВМ находится несколько задач-программ, а процессор распределяет ресурсы между ними. К многозадачным ОС относятся: *UNIX*, *OS/2*, *Windows*, *Linux*.

Сетевые ОС связаны с появлением локальных и глобальных сетей и служат для обеспечения доступа пользователя ко всем ресурсам вычислительной сети. В число сетевых ОС входят: *Net Ware*, *MS Windows NT*, *Unix* и др.

Интерфейсные системы являются естественным продолжением операционной системы и модифицируют как пользовательский, так и программный интерфейсы, а также реализуют дополнительные возможности по управлению ресурсами ЭВМ.

Сетевое ПО предназначено для управления общими ресурсами в распределенных вычислительных системах: сетевыми накопителями на магнитных дисках, принтерами, сканерами, передаваемыми сообщениями.

Сервисное программное обеспечение – программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя, включает: архиваторы, антивирусы, программы обслуживания дисков, программы обслуживания сени, программы диагностики работы компьютера.

Драйвер – это программа, обеспечивающая управление периферийным устройством (мышью, монитором, принтером и др.). Часть этих программ встроена в ОС, а другая создается разработчиками оборудования на языках низкого уровня (*Assembler*, *C*) и поставляется вместе с устройствами.

Операционная система представляет собой комплекс программ и предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами компьютера.

Основные функции операционных систем (ОС) заключаются в управлении ресурсами (физическими и логическими) и процессами вычислительных систем. Физическими ресурсами являются: оперативная память, процессор, монитор, печатающее устройство, магнитные и оптические диски. К логическим ресурсам можно отнести программы, файлы, события и т.д. Под процессом понимается некоторая последовательность действий, предписанная соответствующей программой и используемыми ею данными.

Все современные дисковые операционные системы обеспечивают создание файловой системы, предназначенной для хранения данных на дисках и обеспечения доступа к ним. Для того, чтобы данные могли быть записаны на диск, его поверхность необходимо структурировать, т.е. разделить на дорожки и сектора.

FAT-таблица размещения файлов. Для организации доступа к файлу операционная система должна иметь сведения о номерах кластеров, где размещается каждый файл. В этом ей помогает *FAT*-таблица, которая предназначена для размещения и поиска файлов на диске, содержит информацию о свободном пространстве на диске, о неисправных секторах, а также код формата диска.

Принцип организации файловой системы – табличный. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных таблицах размещения файлов (*FAT*-таблицах), она существует в двух экземплярах, идентичность которых регулярно контролируется средствами операционной системы.

В файлах может содержаться текстовая информация, графика, числовые данные, программные модули.

Правила именования файлов:

1. В имени файла может содержаться максимум 256 символов
2. Могут содержать пробелы, но нельзя в имени файла использовать следующие знаки: . , / ? < > ; *
3. ОС не различает на каком регистре набраны символы например: ИВАНОВ.txt и Иванов.txt системой будут восприниматься как один файл.
4. При поиске и просмотре списков файлов можно использовать символы * и ?. Например: Задав ZX*.* вы обратитесь ко всем файлам начинающимся с букв ZX, Nr???.txt, вы можете найти все файлы состоящие из 5 символов имени которых начинаются с Nr.

Каталог представляет собой справочник, содержащий сведения о местоположении, размере, дате и времени обновления файлов. Для каждого файла и каталога на диске имеется один элемент в определенном каталоге. Один элемент корневого каталога выделяется для метки диска. Для каждого каталога есть один элемент в его родительском каталоге.

Прикладные программы

Сюда относятся программные комплексы, обеспечивающие выполнение различных прикладных задач, т.е. выполнение фактических задач пользователей.

К типовому прикладному ПО относят следующие программы:

- текстовые редакторы;
- табличные процессоры;
- системы иллюстративной и деловой графики (графические процессоры);
- системы управления базами данных;
- экспертные системы;
- программы математических расчетов, моделирования и анализа экспериментальных данных.

Среди пакетов программ следует выделить *MS Office*. Большое развитие получили ППП для различных предметных областей: бухгалтерского учета, финансового менеджмента, правовых систем и т. д.

К бухгалтерским ППП следует отнести «1С. Бухгалтерия», «Квестор», «Бест», «Монолит-Инфо» и др.

ППП финансового менеджмента в основном ориентированы на финансовый анализ предприятий (ЭДИП – «Центр-инвест Софт», Альт-Финансы – «Альт», Финансовый анализ – «Инфософт») и оценку эффективности капиталовложений и инвестиций (Project Expert – PRO-Invest Consulting, Альт-Инвест – «Альт», FOCCAL – «ЦентринвестСофт»).

ППП правовых справочных систем насчитывают более десятка правовых систем, среди которых наиболее известными следует считать ППП «КонсультантПлюс», «Гарант» и др.

Инструментарий технологии программирования – совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.

В рамках этих направлений сформировались следующие группы программных продуктов:

– Средства для создания приложений, включающие:

- локальные средства, обеспечивающие выполнение отдельных работ по созданию программ;

- интегрированные среды разработчиков программ, обеспечивающие выполнение комплекса взаимосвязанных работ по созданию программ;

– CASE-технология, представляющая методы анализа, проектирования и создания программных систем и предназначенная для автоматизации процессов разработки и реализации информационных систем.

Процесс создания загрузочного модуля. Программа, подготовленная на языке высокого уровня, проходит несколько этапов:

1 этап. В текстовом редакторе пишется исходный код программы на алгоритмическом языке и сохраняется в файле.

2 этап. Трансляция, происходит преобразование исходного кода программы в объектный код, т.е. происходит проверка синтаксиса написания операторов, и если ошибок в написании нет, осуществляется перевод на язык машинных кодов.

Трансляторы предназначены для проверки правильности написания операторов и преобразования программ, написанных на языках программирования, в программы на машинном языке. Трансляторы делятся на два класса: компиляторы и интерпретаторы. Компиляторы транслируют всю программу, но без ее выполнения. Интерпретаторы, в отличие от компиляторов, выполняют пооператорный перевод на машинный язык и выполнение всей программы.

3 этап. Компоновка, когда происходит обработка объектного кода редактором связей, специальной программой осуществляющей построение загрузочного модуля, пригодного к выполнению .

Компоновщик, или редактор связей – системная обрабатывающая программа, редактирующая и объединяющая объектные модули в единые загрузочные, готовые к выполнению программные модули. Загрузочный модуль может быть помещен ОС в основную память и выполнен.

Отладчик позволяет управлять процессом исполнения программы, является инструментом для поиска и исправления ошибок в программе.

4. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

4.1. Технология обработки текстовой информации

При описании текстовых процессоров следует выделить две группы программных продуктов этого типа. Первая группа ориентирована на создание документов разной степени сложности, с мощными средствами форматирования и включения графики. Вторая группа текстовых процессоров ориентирована для работы с чисто текстовыми файлами.

Текстовый процессор – прикладное программное обеспечение, используемое для создания текстовых документов.

Типовая структура интерфейса включает ряд элементов:

Строка меню – содержит группы команд, объединенных по функциональному признаку. Она расположена в верхней части экрана под строкой заголовка. Выбор режима из строки меню открывает соответствующее подменю, а выбор определенной опции в нем обеспечивает доступ к меню более низкого уровня. Такая система вложенных (ниспадающих) меню составляет основу интерфейса текстового процессора.

Строка состояния – содержит имя редактируемого документа и определяет текущее положение курсора в этом документе. В строке выводится справочная информация.

Строка подсказки – содержит информацию о возможных действиях пользователя в текущий момент.

Координатная линейка – определяет границы документа и позиции табуляции. По умолчанию координатная линейка градуирована по умолчанию.

Линейка прокрутки – служит для перемещения текста документа в рабочем поле окна. Линейка обеспечивает вертикальное и горизонтальное перемещение.

Курсор – активная точка экрана.

Операции производимые с документом. Минимальный набор типовых операций:

- Создание нового документа.
- Загрузка предварительно созданного документа в оперативную память.
- Сохранение документа.
- Удаление документа.
- Распечатка документа.
- Ввод текста.
- Редактирование текста.

Работа с фрагментами текста.

Фрагментом называется непрерывная часть текста. Выделение фрагмента делает его объектом последующей команды. Выделить объект – значит «подсветить» его при помощи мыши или клавиатуры. Выделенный фрагмент может быть строчным, блочным или линейным. *Строчный* – состоит из последовательности целых строк. *Блочный* объединяет обычно части строк, образующие прямоугольник. *Линейный* – может включать в себя последовательность необязательно целых строк.

Копирование и перемещение фрагментов текста – наиболее часто используемые операции в текстовых процессорах. Эти операции выполняются с применением буфера обмена, так и без него (перетаскивание мышью).

Технология:

- выделение части текста;
- перенос выделенного фрагмента в буфер обмена;
- перемещение курсора в нужное место документа;
- копирование (перенос) из буфера обмена.

Поскольку все приложения ОС Windows работают через общий системный буфер промежуточного хранения, операции с фрагментами текста могут затрагивать не только внешние документы, но и другие приложения.

Удаление текста. В современных текстовых процессорах можно удалять не только символ, слово, строку, но и фрагмент текста. Удаление символа или выделенного фрагмента производится клавишами BackSpace, если символ находится справа от курсора и DELETE, если символ находится справа от курсора. Также для удаления фрагмента текста можно использовать перемещение его в буфер обмена.

Форматирование текста. Операции форматирования включают в себя разбивку текста на строки и страницы, выбор расположения абзаца, отступов и отбивок между абзацами, обтекание отдельных абзацев, а также видов и начертаний шрифтов. Суть форматирования заключается в способности текстового редактора изменять оформление документа на странице.

Форматирование текста осуществляется средствами меню Формат или панели форматирования. Основные приемы форматирования включают:

- Выбор и изменение гарнитуры шрифтов;
- Управление размером;
- Управление начертанием и цветом;
- Управление методом выравнивания;
- Создание маркированных и нумерованных списков (в том числе многоуровневых);
- Управление параметрами абзаца.

Приемы и средства автоматизации разработки документов.

К наиболее общим средствам автоматизации разработки документов относятся *стили оформления абзацев, шаблоны документа и темы оформления*.

Абзац – элементарный элемент оформления любого документа.

Стиль оформления – это именованная совокупность настроек параметров шрифта, абзаца, языка и некоторых элементов оформления абзацев (линий и рамок).

Работа со стилями состоит в создании, настройке и использовании стилей. Некоторые стандартные стили присутствуют в настройке программы по умолчанию, сразу после ее установки. Их используют путем выбора нужного стиля из раскрывающегося списка на панели Форматирования.

Совокупность удачных стилевых настроек сохраняется с готовым документом, но желательно иметь средство, позволяющее сохранять их и вне документа. Тогда их можно использовать для подготовки новых документов. Такое средство есть – это шаблоны. По своей сути, шаблоны – это тоже документы, а точнее говоря, заготовки будущих документов. От обычных документов они отличаются тем, что в них приняты специальные меры, исключающие их повреждения. При сохранении нужен новый документ, а шаблон, использованный в качестве его основы, остается в неизменном виде и пригоден для дальнейшего использования.

Последняя версия текстового процессора Microsoft Word имеет специальное средство автоматического оформления, предназначенное в первую очередь для электронных документов. Это средство называется Тема.

Тема представляет собой совокупность следующих элементов оформления:

- фоновый узор;
- стили оформления основного текста и заголовков;
- стиль оформления маркированных списков;
- стиль графических объектов оформления (линий).

Доступ к выбору темы выполняется командой Формат → Темы.

Кроме создания *простых текстовых документов* (т.е. документов не содержащих встроенных объектов) средствами текстового процессора Microsoft Word, можно создавать и *комплексные текстовые документы* (содержащие объекты встроенные в текст):

- формулы;
- таблицы;
- диаграммы;
- графические объекты (рисунки и изображения);
- надписи.

Формулы

В программе Microsoft Word имеется простое средство для ввода простых формул – Microsoft Equation 3.0. Он позволяет создавать формульные объекты и вставлять их в текстовый документ. При необходимости вставленный объект можно редактировать непосредственно в поле самого документа.

Таблицы

Данные, представленные в табличной форме, отличаются наглядностью. Таблицы всегда были неотъемлемым атрибутом печатной научно-технической документации. Текстовый редактор Microsoft Word обладает удивительно гибкими и мощными средствами создания таблиц как для печатных. Так и для электронных документов. Три основные средства создания таблиц – это:

- кнопка *Добавить таблицу* на панели инструментов *Стандартная*;
- диалоговое окно «Вставки таблицы»;
- средство рисования таблиц «Таблицы и границы».

Редактирование таблиц включает в себя:

- добавление заданного количества строк и столбцов;
- удаление выделенных ячеек, строк и столбцов;
- слияние выделенных ячеек;
- разбиение ячеек.

Диаграммы

Диаграммы является удобным средством визуального представления данных наряду с таблицами и очень широко используются в научно-технической документации. Для создания диаграмм текстовый процессор Microsoft Word имеет подключаемое средство Microsoft Graph. Существует два метода для вставки диаграмм в документ. Более общий метод основан на том, что сначала вставляется в документ некая произвольная диаграмма, с которой связана некая базовая таблица данных. Далее производится настройка диаграммы, которая состоит в настройке внешнего вида и в редактировании базовой таблицы. Второй, частный, метод основан на том, что диаграмма создается на базе конкретной таблице, имеющейся в документе. В этом случае настройка диаграммы состоит только в настройке внешнего вида.

Настройка диаграммы состоит в выборе *элементов оформления диаграммы* и *элементов представления данных*. Элементы представления данных – это точки на графиках, столбцы гистограмм, секторы круговых диаграмм – в общем, что служит для непосредственного отображения данных. *Элементы оформления* – это название диаграммы, название ее осей, легенда (специальное поле, в котором приведены условные обозначения для групп элементов данных), подписи к элементам данных и линии координатной сетки. Настройку осуществляют подключением или отключением

ем тех или иных элементов. Элементы диаграммы бывают *связанными* или *присоединенными*. Присоединенные элементы можно редактировать отдельно, связанные элементы редактировать отдельно нельзя, так как они связаны с элементами базовой таблицы.

Графические объекты.

В документах Microsoft Word можно использовать 2 типа графических объектов: *рисунки и изображения*.

Рисунки – объекты векторной природы (линии, геометрические фигуры). Простейшие средства для их создания есть в самом текстовом процессоре.

Изображения – растровые объекты. Текстовый процессор не имеет средств для их создания, поэтому они вставляются в текст как внешние объекты.

Для создания художественных графических надписей, например заголовков, текстовый процессор Microsoft Word имеет специальное программное средство WordArt.

4.2. Технология обработки числовой информации

Для представления данных в удобном виде используют таблицы. Компьютер позволяет представлять их в электронной форме, а это дает возможность не только отображать, но и обрабатывать данные. Класс программ, используемых для этой цели, называется *электронными таблицами*.

Электронная таблица – компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: текст, даты, числа, формулы.

Результат вычисления формулы в клетке изображением этой клетки. Числовые данные и даты могут рассматриваться как частный случай формул. Для управления электронной таблицей используется специальный комплекс программ – табличный процессор.

Главное достоинство электронной таблицы – это возможность мгновенного пересчета всех данных, связанных формульными зависимостями при изменении значения любого операнда.

Для хранения и автоматизации расчета данных, представленных в табличной форме, использует табличный процессор *Excel*. Документы, созданные в среде *Excel*, называют рабочими книгами. Рабочие книги записываются как файлы с расширением *.xls*.

Рабочая область электронной таблицы состоит из строк и столбцов, имеющих свои имена. Имена строк – это их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом установленным для данной программы. Имена столбцов – это буквы латинского алфавита сначала от *A* до *Z*, затем от *AA* до *AZ*, *BA* до *BZ* и т.д.

Максимальное количество строк и столбцов определяется особенностями используемой программы и объемом памяти компьютера. Современ-

менные программы дают возможность создавать электронные таблицы, содержащие более 1 млн. ячеек, хотя для практических целей в большинстве случаев этого не требуется.

Пересечение строки и столбца образует ячейку таблицы, образует ячейку таблицы, имеющую свой уникальный адрес. Для указания адресов ячеек в формулах используются ссылки (A2 или C2).

Ячейка – область, определяемая пересечением столбца и строки электронной таблицы. Адрес ячейки определяется названием (номером) столбца и номером строки.

Ссылка – способ (формат) указания адреса ячейки.

В электронной таблице существует понятие блока (диапазона) ячеек, также имеющего свой уникальный адрес. В качестве блока рассматривается строка или часть строки, столбец или часть столбца, а также прямоугольник, состоящий из нескольких строк или столбцов или их частей. Адрес блока ячеек задается указанием ссылок первой и последней его ячеек, между которого ставится разделительный символ – двоеточие.

При работе с электронной таблицей на экран выводятся рабочее поле таблицы и панель управления. Панель управления обычно включает: Главное меню, вспомогательную область управления, строку ввода и строку подсказки. Расположение этих областей на экране может быть произвольным и зависит от особенностей конкретного табличного процессора.

Строка главного меню содержит имена меню основных режимов программы. Выбрав один из них, пользователь получает доступ к ниспадающего меню, содержащему перечень входящих в него команд.

Вспомогательная область управления включает:

- строку состояния;
- панели инструментов;
- вертикальную горизонтальную линейки прокрутки.

Строка ввода отображает вводимые в ячейку данные. В ней пользователь может рассматривать или редактировать содержание текущей ячейки. Текущей (активной) называется ячейка электронной таблицы, в которой в данный момент находится курсор. Часть электронной таблицы, которую мы видим на экране монитора, называется текущим (активным) экраном.

Данные, хранимые в ячейках электронной таблицы.

Символьные (текстовые) – данные имеют описательный характер. Они могут включать в себя алфавитные, числовые, и специальные символы.

Числовые – не могут содержать алфавитных и специальных символов, поскольку с ними производятся математические операции. Единственным исключением являются десятичная точка (запятая) и знак числа, стоящий перед ним.

Формулы – видимое на экране содержимое ячейки, возможно, – результат вычислений, произведенных по имеющейся, но не видимой в ней фор-

муле. Формула может включать ряд арифметических, логических, и прочих действий, производимых с данными из других ячеек.

Функция – представляет собой программу с уникальным именем, для которой пользователь должен задать конкретные значения аргументов функции, стоящих в скобках после ее имени. Функцию, (так же, как и число) можно рассматривать частным случаем формулы. Различают статистические, логические, финансовые, математические и др. функции.

Дата. Особым типом входных данных являются даты. Этот тип данных обеспечивает выполнение таких функций, как добавление к дате числа (пересчет дат вперед и назад) или вычисление разности двух дат (длительности периода). Даты имеют внутренний и внешний формат. Внутренний формат отображает количество дней от начала 1900 года или порядковый номер дня по Юлианскому календарю. Внешний формат используется ввода и отображения дат. Внешний формат бывает полным, кратким, средним. Чаще используется краткий формат даты.

Тип входных данных, содержащихся в каждой ячейке, определяется первым символом, который должен трактоваться не как часть данных, а как команда переключения режима:

- Если в ячейке содержатся числа, то первый их символ является либо цифрой, либо десятичной точкой, либо знаком числа.
- Если в ячейке содержится формула или функция, то первый ее символ должен быть знак “=”.
- Ячейка, содержащая символьные данные начинается с пробела или любого другого символа кроме знаков -,=,+.

Форматирование числовых данных

Можно использовать различные форматы представления числовых данных в рамках одной и той же электронной таблицы. По умолчанию числа располагаются в клетке, выравниваясь по правому краю. Наиболее распространенные форматы представления числовых данных:

- основной формат используется по умолчанию, обеспечивая запись числовых данных в ячейках в том же виде, в котором они вводятся;
- формат с фиксированным количеством десятичных знаков обеспечивает представление чисел в ячейках с заданной точностью;
- процентный формат обеспечивает представление введенных данных в форме процентов со знаком %;
- денежный формат обеспечивает представление введенных данных в денежном выражении;
- научный формат обеспечивает представление очень маленьких или очень больших чисел в виде двух компонентов:
 - мантиссы;
 - порядка числа.

Форматирование символьных данных по умолчанию они выравниваются по левому краю числа. Форматирование символьных данных можно производить при помощи панели форматирования.

Формулы Вычисления в таблицах производятся с помощью формул. Результат вычисления помещается в ячейку, в которой находится формула. При вычислениях с помощью формул соблюдается порядок действий, принятый в математике. Формулы состоят из операторов и операндов, расположенных в определенном порядке. В качестве операндов используются данные, а также ссылки отдельных ячеек или блоков ячеек. Операторы в формулах обозначают действия, производимые с операндами. Операторы различают арифметические и логические:

Арифметические:

- + сложение;
- вычитание;
- * умножение;
- / деление;
- ^ возведение в степень;

Логические:

- NOT* – логическое отрицание;
- OR* – логическое “ИЛИ”;
- AND* – логическое “И”.

Логические формулы определяют, выражение истинно или ложно. Истинным выражением присваивается численная величина 1, а ложным – 0.

По умолчанию электронная таблица вычисляет формулы при их вводе, пересчитывает их повторно при каждом изменении входящих в них исходных данных. Формулы могут включать функции.

Функции

Под функцией понимают зависимость одной переменной (y) от одной переменной (x) или нескольких переменных ($x_1, x_2 \dots x_n$). Причем каждому набору значений переменной x будет соответствовать единственное значение определенного типа зависимой переменной y . Функции вводят в таблицу в составе формул. В электронных таблицах могут быть представлены следующие виды функций:

- математические;
- статистические;
- текстовые;
- финансовые;
- функции даты и время.

Буфер промежуточного хранения

Важной особенностью многих электронных таблиц является буфер промежуточного хранения. Буфер используется при выполнении команд

копирования и перемещения для временного хранения копируемых или перемещаемых данных, после которого они направляются по новому адресу. При удалении данных они также помещаются в буфер. Содержимое буфера сохраняется до тех пор. Пока в него не будет записана новая порция данных.

Абсолютная и относительная адресация.

При копировании или перемещении формулы в другое место таблицы необходимо организовать управление форматированием адресов исходных данных. Поэтому в электронной таблице при написании формул наряду с введенным ранее понятием ссылки используются понятия абсолютной и относительной ссылки.

Абсолютная ссылка – это не изменяющийся при копировании или перемещении формулы адрес ячейки, содержащей исходное данное (операнд).

Для указания абсолютной адресации вводится символ \$. Различают два типа абсолютной ссылки: полная и частичная.

Полная абсолютная ссылка указывается, если при копировании или перемещении адрес клетки, содержащей исходное данное не меняется. Для этого символ \$ ставится перед наименованием столбца и номером строки. (*\$B\$5*)

Частичная абсолютная ссылка – указывается, при копировании или перемещении не меняется номер строки или наименование столбца. При этом символ \$ в первом случае ставится перед номером строки, а во втором наименование столбца. (*B\$5* – строка или *\$B5* – столбец)

Относительная ссылка – это изменяющийся при копировании или перемещении формулы адрес ячейки, содержащей исходное данное (операнд). Изменение адреса происходит по правилу относительной ориентации клетки с исходной формулой и клеток с операндами. Форма написания относительной ссылки совпадает с обычной записью.

Итоговые вычисления

Итоговые вычисления предполагают получение числовых характеристик, описывающих определенный набор данных в целом. Проведение итоговых вычислений в программе Excel выполняется при помощи встроенных функций.

Суммирование

Для итоговых вычислений применяют ограниченный набор функций, наиболее типичной из которых является функция суммирования. Это единственная функция, для применения которой есть отдельная кнопка на панели инструментов (кнопка Автосумма).

Функции, предназначенные для выполнения итоговых вычислений, часто применяют при использовании таблиц *Excel* в качестве баз данных.

Построение диаграмм

В программе *Excel* термин диаграмма используется для обозначения всех видов графического представления данных. Диаграмма представляет собой вставной объект, внедренный на один из листов рабочей книги. Она может располагаться на том же листе, где находятся данные или на любом другом рабочем листе. Диаграмма сохраняет связь с данными, на основе которых она построена, и при обновлении этих данных немедленно изменяет свой вид.

Консолидация данных

Инструмент Консолидация позволяет объединить однотипные данные таблиц, находящиеся в разных местах и даже в разных листах и книгах. Итоговая таблица будет содержать только одну строку с ключевым полем, а данные в ней будут результатом применения одной из перечисленных итоговых сумм. Ключевым считается самое левое поле (столбец) таблицы или ее выделенной части. Консолидация может быть применена и к одной таблице.

Сводная таблица

Инструмент Сводная таблица из меню Данные является очень удобным и мощным средством анализа и преобразования данных. С его помощью можно провести любое разбиение таблицы с целью получения значений итоговых функций для произвольной группы данных. Сводная таблица во многом объединяет возможности таких инструментов, как Итоги и Консолидация.

Сводная таблица предоставляет возможность создавать некоторое количество страниц, равное числу параметров выбранного поля исходной таблицы. На каждой странице формируются заголовки строк, заголовки столбцов и данные таблицы. Последние затем подвергаются обработке с помощью выбранной итоговой функции. Для построения сводных таблиц в *Excel* используется Мастер сводных таблиц.

4.3. Технология хранения, поиска и сортировки информации

Исторически сложившееся развитие вычислительных систем обусловило необходимость хранения в электронном виде все большего количества информации. Одновременно с совершенствованием и дальнейшим развитием вычислительных систем росли объемы информации, подлежащей обработке и хранению. Сложности, возникшие при решении на практике задач структурированного хранения и эффективной обработки возрастающих объемов информации, стимулировали исследования в соответствующих областях. Задачи хранения и обработки данных были формализованы. Была создана теоретическая база для решения задач такого класса, результатом реализации на практике которой стали системы, предназначенные

для организации обработки, хранения и предоставления доступа к информации. Позже такие системы стали называть системами баз данных.

Одновременно с развитием систем баз данных, происходило интенсивное развитие средств вычислительной техники, используемой для работы с большими объемами информации. Вычислительная мощность и, особенно, объемы запоминающих устройств первых вычислительных систем были недостаточны для хранения и обработки информации в объемах, необходимых на практике.

По мере развития систем баз данных, менялись принципы организации данных в них: первоначально данные представлялись на основе иерархической, а в последствии сетевой модели. В конце 1970-х – начале 1980-х годов начали появляться первые реляционные продукты. В настоящее время системы баз данных на основе реляционной модели занимают лидирующее положение.

Понятие БД и СУБД

Система баз данных (банка данных) – это компьютеризированная система основная задача которой – хранение информации и предоставление доступа к ней по требованию.

Система баз данных включает в себя:

1. данные, непосредственно сохраняемые в базе данных;
2. аппаратное обеспечение;
3. программное обеспечение;
4. пользователей:
 - 4.1. прикладные программисты;
 - 4.2. конечные пользователи;
 - 4.3. администраторы баз данных.

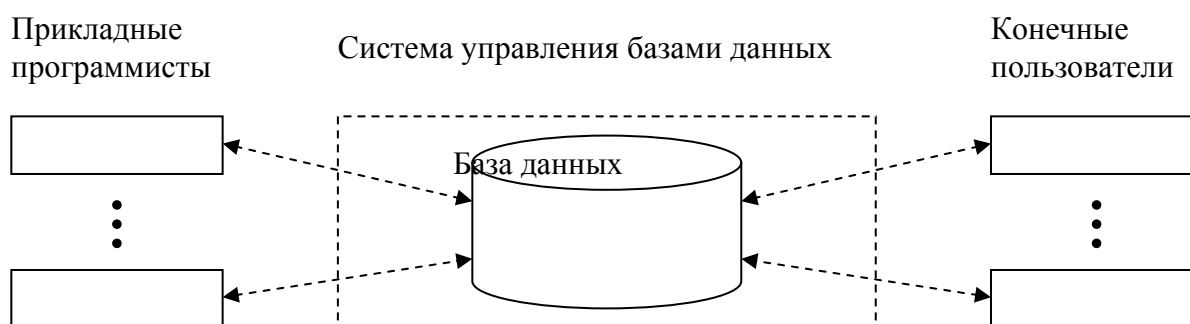


Рис. 2. Система баз данных

Данные.

Данные в базе данных являются интегрированными и, как правило, общими. Понятие интегрированных данных подразумевает возможность представления базы данных как объединение нескольких отдельных фай-

лов данных, полностью или частично не перекрывающихся. Понятие общие подразумевает возможность использования отдельных областей данных, в базе данных несколькими различными пользователями.

Аппаратное обеспечение.

К аппаратному обеспечению системы относятся накопители для хранения информации, вместе с устройствами ввода-вывода, контролерами устройств и т.д.; вычислительная техника, используемая для поддержки работы ПО системы.

Программное обеспечение.

Программное обеспечение является промежуточным слоем между собственно физической базой данных и пользователями системы и называется диспетчером базы данных или системой управления базами данных, СУБД (DBMS). Все запросы пользователей обрабатываются СУБД.

Таким образом, СУБД – это специализированное программное обеспечение, предоставляющее пользователю базы данных возможность работать с ней, не вникая в детали хранения информации на уровне программного обеспечения.

Пользователи.

Прикладные программисты – отвечают за написание прикладных программ, использующих базу данных.

Конечные пользователи – работают с базой данных непосредственно, через рабочую станцию или терминал. Конечный пользователь может получить доступ к базе данных, используя соответствующее прикладное ПО.

Администраторы базы данных – технические специалисты, осуществляющие создание БД, технический контроль работы СУБД и др. операции. Администраторы базы данных отвечают за реализацию решений администратора данных. Администратор данных решает, какие данные необходимо хранить в БД, обеспечивает поддержание порядка при обслуживании и использовании хранимых в БД данных.

Использование баз данных для хранения информации позволяет организовать централизованное управление данными, что обеспечивает следующие преимущества:

- 1) возможность сокращения избыточности;
- 2) возможность устранения (до некоторой степени) противоречивости;
- 3) возможность общего доступа к данным;
- 4) возможность соблюдения стандартов;
- 5) возможность введения ограничений для обеспечения безопасности;
- 6) возможность обеспечения целостности данных;
- 7) возможность сбалансировать противоречивые требования;

8) возможность обеспечения независимости данных. Поскольку программное обеспечение отделяется от данных, хранимых БД, изменения, вносимые в структуру БД, в большинстве случаев не приводят к необходимости внесения радикальных изменений в программное обеспечение.

База данных в широком смысле слова – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Под предметной областью подразумевается часть реального мира, подлежащего изучению и использованию для решения различных практических задач.

БД представляют собой разновидность информационной модели, для которой определены такие понятия, как информационный объект, реквизит, отношения и связи.

Информационным объектом называется описание реального объекта, процесса или явления в виде совокупности его характеристик, называемых реквизитами. Информационный объект однозначно идентифицируется именем и заданием ключевого реквизита (ключа). Остальные реквизиты в информационных объектах являются описательными.

Пример Информационный объект СТУДЕНТ имеет реквизиты: номер (номер зачетной книжки – ключевой реквизит), фамилия, имя, отчество, дата рождения, код места обучения.

Информационный объект ПРЕПОДАВАТЕЛЬ: код (ключевой реквизит), кафедра, фамилия, имя, отчество, ученая степень, ученое звание, должность.

Отношения, существующие между реальными объектами, определяются в информационных моделях как связи. Существует три вида связей: один к одному (1:1), один ко многим (1:n) и многие ко многим (n:m).

Связь один к одному определяет соответствие одному информационному объекту не более одного информационного объекта и наоборот.

Связь один ко многим предполагает, что одному экземпляру информационного объекта X может соответствовать любое количество информационных объектов Y_1, Y_2, \dots, Y_n , но каждый объект $Y_i, i = 1, 2, \dots, n$, связан не более чем с одним объектом X .

Связь многие ко многим предполагает соответствие каждому информационному объекту $X_j, j = 1, 2, \dots, m$, любого количества экземпляров объекта $Y_i, i = 1, 2, \dots, n$, и наоборот.

Пример.

Информационные объекты СТУДЕНТ и ПРЕПОДАВАТЕЛЬ имеют связь многие ко многим. Каждый студент обучается у множества преподавателей, а каждый преподаватель учит множество студентов.

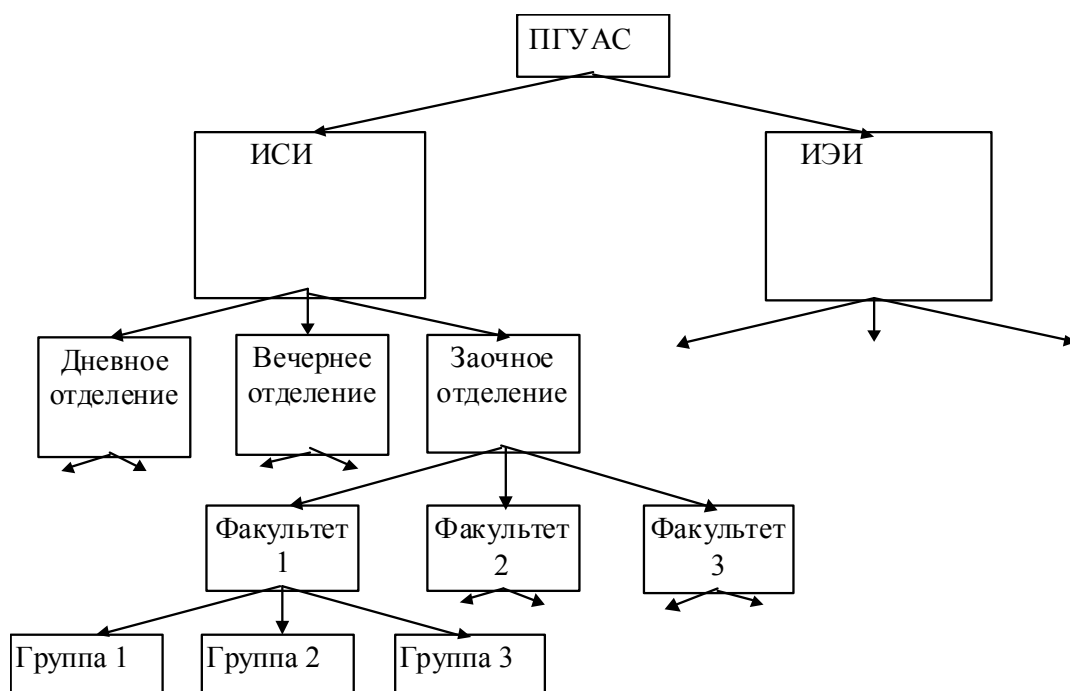
Теперь определим базу данных как информационную модель, состоящую из совокупности связанных информационных объектов в исследуемой предметной области. Ядром любой БД является модель данных, кото-

рая представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

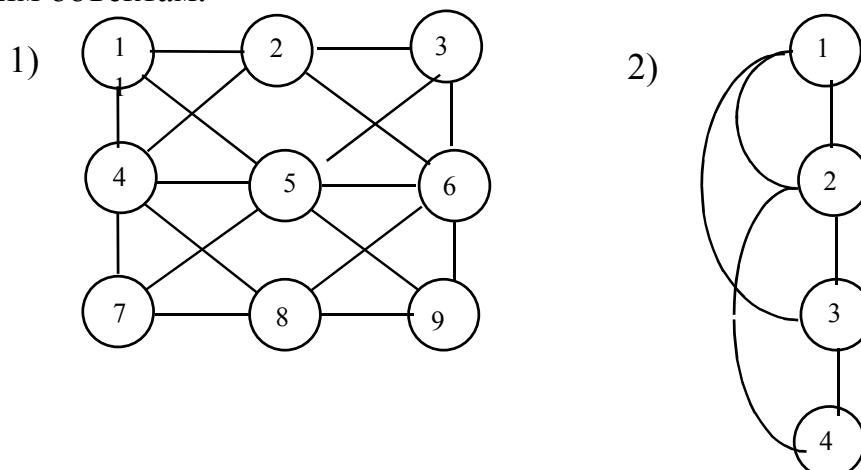
Модель данных – это совокупность структур данных и операций их обработки.

Различают иерархическую, сетевую и реляционную модели данных.

В иерархической модели данные представлены в виде древовидной (иерархической) структуры. Непосредственный доступ возможен лишь к объекту самого высокого уровня. К другим объектам доступ осуществляется по связям от объекта на вершине модели.



В сетевых моделях непосредственный доступ обеспечивается к любому объекту независимо от уровня его расположения в модели, а также по связям к другим объектам.



Обе эти модели не получили широкого распространения из-за сложности реализации графов в виде машинных структур данных, кроме того, в них сложно осуществить операции поиска информации.

Наибольшее распространение получила третья модель данных – реляционная модель.

Понятие «реляционный» (англ. relation – отношение) связано с работами Е. Кодда – известного специалиста в области БД.

Реляционная модель данных ориентирована на организацию данных в виде двумерной таблицы (отношения), обладающей следующими свойствами:

- каждый элемент таблицы – один элемент данных;
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, текстовый и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Таблицу называют отношением, строки – записями, столбцы – полями, а строку заголовков – схемой отношения.

Для однозначного определения записей служит первичный ключ, который может быть простым или составным.

Простой ключ (ключевое поле) – это поле, каждое значение которого однозначно определяет запись.

Пример.

База данных «Поставщик – изделия» содержит три типа информации о строительной компании:

Информацию только об изделиях, используемых при строительстве зданий. Эта информация находится в отношении ИЗДЕЛИЯ.

Информацию только о поставщиках, поставляющих изделия на строительные площадки. Эта информация содержится в отношении ПОСТАВЩИК.

Информация о номерах и количестве изделий, поступающих от каждого поставщика. Эта информация содержится в отношении П_И. Отношение П_И является связующим для независимых отношений ПОСТАВЩИК и ИЗДЕЛИЯ.

ИЗДЕЛИЯ

Дном	Дназв	Дмасса	Дцена
101	панель	1500	50
102	плита	1200	70
103	блок	400	30

ПОСТАВЩИК

Пном	Пимя	Город
П1	ЖБИ-1	Пенза
П2	ДСК-2	Кузнецк

П_И

Пном	Дном	шт.
П1	101	12
П2	102	46
П3	103	24

Если все экземпляры участвуют в связи, то связи называются обязательными, если не участвуют, то необязательными. Другими словами, если некоторым значениям первой сущности нет соответствующих значений другой сущности, то первая сущность является необязательной. Эта характеристика связи описывает класс принадлежностей входящих в нее сущностей.

Это могут быть:

1. Обязательный класс принадлежностей.
2. Необязательный класс принадлежностей.

Вид зависимости и класс принадлежности определяют кардинальность сущности.

В настоящее время используют 7 правил генерации отношений из диаграмм ER-типа.

ПРАВИЛО 1. Если степень бинарной связи равна 1:1 и класс принадлежности обеих сущностей является обязательным, то требуется только одно связное отношение. Первичным ключом этого отношения может быть ключ любой из двух сущностей.

ПРАВИЛО 2. Если степень бинарной связи равна 1:1 и класс принадлежности одной сущности является обязательным, а другой необязательным, то необходимо построение двух отношений. Под необязательную сущность необходимо выделить одно отношение, при этом ключ сущности должен служить первичным ключом для соответствующего отношения. Кроме этого, ключ сущности, для которого класс принадлежности является необязательным, добавляется в качестве атрибута в отношение, выделенное для сущности с обязательным классом принадлежности.

ПРАВИЛО 3. Если степень бинарной связи равна 1:1 и класс принадлежности ни одной сущности не является обязательным, то необходимо использовать три отношения: по одному для каждой сущности, ключи которых служат в качестве первичных в соответствующих отношениях, и одного для связи, в котором будет иметься по одному ключу от каждой сущности.

ПРАВИЛО 4. Если степень бинарной связи равна 1: N и класс принадлежности обеих сущностей является обязательным, то достаточно использовать два отношения, одно на n -связную сущность, при условии, что ключ каждой сущности служит в качестве первичного ключа для соответствующего отношения и одно связное дополнительно.

Ключ n -связной сущности должен быть представлен как атрибут в отношении, отводимое для n -связной сущности.

ПРАВИЛО 5. Если степень бинарной связи равна 1: N и класс принадлежности n -связной сущности является необязательным, то достаточно использовать три отношения, по одному на каждую сущность, при условии, что ключ каждой сущности служит в качестве первичного ключа для соот-

ветствующего отношения. Дополнительно ключ одной сущности должен быть представлен как атрибут в отношении, отводимое для n -связной сущности.

ПРАВИЛО 6. Если степень бинарной связи равна $1:N$ и классы принадлежности сущностей являются необязательными, то необходимо формирование трех сущностей: по одному для каждой сущности, причем ключ сущности служит первичным ключом соответствующего отношения, и одного отношения для связи. Связное отношение должно иметь ключи каждой сущности.

ПРАВИЛО 7. Если степень бинарной связи равна $m:n$, то для хранения данных необходимы три отношения: по одному для каждой сущности, причем ключ каждой сущности используется в качестве первичного ключа соответствующего отношения, одного отношения для связи. Последнее отношение должно иметь в числе своих атрибутов ключ каждой сущности. Принцип построения отношений соответствует предыдущему примеру и отличается перечислением в связном отношении всех связей между ключевыми элементами объектных отношений.

Языки и формы запросов

Операции реляционной алгебры (РА) заложены не только в обработке информации командами СУБД, отмечено их эффективное применение в языках запросов, значительно повышающих эффективность поиска незапланированных запросов.

Тенденция развития СУБД, прослеживаемая особенно в последнее время для применений под управлением WINDOWS, показывает, что представленные для коммерческой продажи они имеют ряд общих признаков: СУБД под Windows сравнимы по интегрированности, т.е. содержат: редакторы, компиляторы, отладчики, генераторы отчетов, графическое обеспечение, средства проектирования. Также имеются средства поддержки работы в локальной сети.

Но самым интересным дополнением в СУБД является включение языков запросов QBE (Query By Example) и SQL (Structure Query Language). Появление SQL-Server позволяет проектировать и осуществлять доступы к распределенным базам данных, т.е. обеспечивать оперативность и достаточную надежность работы СУБД в локальной сети, без значительных дополнительных усилий разработчиков сети.

Языки запросов также важны и с точки зрения приложений СУБД, например, в АРМ. Объяснение этому достаточно естественно. Разработанные АРМ имеют, как правило, четко определенное меню, т.е. отработанные формы запросов, оформленные в систему выбора (да/нет).

Поэтому необходимо применять языки запросов для реализации диалога в командах, что требует квалификации программиста. Языки запросов снимают и эту проблему, но порождаются две других.

Во-первых, для формулировки запросов необходимо знать язык запросов. Если в QBE этому обучиться достаточно просто, благодаря развитому экранному представлению примера запроса в среде WINDIWS и QBE, то в SQL, представляющий своеобразный язык, требуется определенная подготовка, может быть даже сравнимая с изучением языка одной из СУБД.

Во-вторых, и QBE, и SQL требует дополнительных ресурсов от ЭВМ, что не всегда реально в медленно обновляющемся парке ЭВМ.

И, наконец, еще одно замечание о включении языков запросов QBE и SQL в уже действующие АРМы или СУБД. А, следует ли включать такие мощные средства из-за 10-20 нестандартных запросов, поступающих, например, в течение полгода обслуживания СУБД или АРМов.

Может сложиться ситуация, что разработчик сформулирует вывод: «Язык запросов необходимо включить в АРМ, но ни оперативной, ни дисковой памяти не хватит». В этом случае, надеясь, что разработчик знает систему команд СУБД, он может сам написать транслятор для языка запросов, основанного на элементах реляционной алгебры, представленных в предлагаемых методических указаниях.

К базе данных БД могут быть предъявлены различные вопросы на поиск информации, удовлетворяющие тем или иным условиям, или ограничениям. В результате поиска из набора файлов, хранящих объектные или связные отношения, будет получено отношение, которое может быть сохранено в виде файла, либо выведено пользователю на экран дисплея или принтер.

Формулировки вопросов могут быть представлены разнообразными формами запросов, простыми и сложными, к одному файлу или к нескольким и т.д. Однако все запросы могут быть структурированы, а следовательно, формализованы, что и представляет возможность их реализации в языковой (программной среде). Рассмотрим наиболее простые типы запросов.

Обычный запрос. $A(E) = ?$ Каково значение атрибута А объекта Е (напомним, в понятие атрибут входит имя атрибута и значения поля (столбца) в реляционном отношении. Объектом называем строку (запись) в отношении. Аналогичное можно распространить на плоский файл.

Пример. Определить заработок у доцента института. Здесь значение атрибута А – заработок, объект – доцент. Доцент в свою очередь является значением атрибута должность (доцент, профессор, старший преподаватель, преподаватель, ассистент).

Инвертированный запрос. $A(?) = V$. Какой объект Е имеет значение атрибута А равное V. Пример. Какие должности преподавательского состава имеют заработок (=) равный V.

Атрибутный запрос. $?(E)=V$ (\leq , $=$, \neq , \geq). Какой атрибут объекта E имеет значение V.

Пример. Найти имя атрибута имеющего значение 10000р у должности доцента.

Запрос на поиск информации для заданного объекта $? (E) = ?(\leq, =, \neq, \geq)$. Выбрать всю информацию атрибута и их значение для объекта E.

Пример. Выдать все атрибуты и их значения известных для профессии «доцент».

Запрос на поиск значений заданного атрибута для всех объектов. $A(?) = ?$ Для заданного атрибута A найти все значения для всех объектов. Выдать величину зарплат для всех должностей.

Запрос на поиск всех объектов и всех атрибутов имеющих значение $?(?) = V$ (\leq , $=$, \neq , \geq). Найти все объекты и все атрибуты, для которых задано значение V.

Пример. Найти все атрибуты и все объекты, для которых значение неизвестного атрибута равно 100000р.

Сложный запрос на одном отношении: Запрос основывается на перечисленных типах, усложнением структуры запроса, логическими операндами «или» = «or», «и» = «AND» и вычислительными операциями.

Пример. Перечислить кафедры, на которых работают и доценты, и профессора. Перечислить кафедры, на которых имеется должность с окладом условно 10000 р. и 15000р. Перечислить фамилии работающих на кафедре «Физика» в должности доцента.

Сложный запрос на 2-х и больше отношениях. Запрос требует обработки 2-х и более отношений. Может иметь логические операнды и вычислительные операции и требовать дополнительных операций над отношениями: проекция, выбор, объединение, соединение и т.д., что более подробно рассмотрено ниже.

Из анализа перечисленных типов запросов можно сделать следующий вывод. Для формализации запросов необходимо иметь:

- имена отношений (имена файлов);
- имена атрибутов (имена полей);
- значения атрибутов (значения полей);
- правила формирования условий (\leq , $=$, \neq , \geq);
- правила логических операций (AND, OR);
- правила реляционной алгебры для отношений.

Последнее правило является базовым для формирования сложных структур запросов, особенно для запросов на обработку нескольких отношений (файлов). Следует отметить, что все реляционные СУБД, как правило, в одном файле содержат одно отношение. SQL в одном файле содержит несколько отношений, точнее все, которые обрабатываются в запросе. Может показаться, что перечисленные типы запросов достаточно просто

формализуются. Это не всегда так. Если рассуждать командами СУБД типа `filtr`, `select`, `sort on`, `list`, `display`, то наиболее простые запросы формализуются легко. Но запрос может быть сложным, композиционным, содержащим в себе многоуровневое вложение элементарных запросов с логическими и арифметическими операциями, в этом случае необходимо правило декомпозиции запроса и его формализации. А для этого необходимо иметь представление об операциях в реляционной алгебре. Это полезно не только для разработки языка запроса в конкретной СУБД, QBE или SQL, но также и при концептуальном моделировании при проектировании АРМ.

В современной технологии БД предполагается, что создание БД, ее поддержка и обеспечение доступа пользователей к ней осуществляется с помощью специального программного продукта, именуемого системой управления базами данных.

Система управления базами данных (СУБД) – это комплекс программных и языковых средств, необходимых для создания БД, поддержания их в рабочем состоянии и организации поиска в них необходимой информации.

Свойства полей базы данных

Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Microsoft Access.

- **Имя поля** – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).
- **Тип поля** – определяет тип данных, которые могут содержаться в данном поле.
- **Размер поля** – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.
- **Формат поля** – определяет способ форматирования данных в ячейках, принадлежащих полю.
- **Маска ввода** – определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных).
- **Подпись** – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство **Имя поля**).
- **Значение по умолчанию** – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).
- **Условие на значение** – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

- Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство Условие на значение).

- Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы;

- Пустые строки – свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым).

- Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Типы данных

С основными типами данных мы уже знакомы. Так, например, при изучении электронных таблиц Microsoft Excel мы видели, что они работают с тремя типами данных: текстами, числами и формулами. Таблицы баз данных, как правило, допускают работу с гораздо большим количеством разных типов данных. Так, например, базы данных Microsoft Access работают со следующими типами данных

- Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

- Поле Мемо – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он хранится в другом месте базы данных, а в поле хранится указатель на него, но для пользователя такое разделение заметно не всегда.

- Числовой – тип данных для хранения действительных чисел.

- Дата/время – тип данных для хранения календарных дат и текущего времени.

- Денежный – тип данных для хранения денежных сумм. Теоретически, для их записи можно было бы пользоваться и полями числового типа, но для денежных сумм есть некоторые особенности (например, связанные с правилами округления), которые делают более удобным использование специального типа данных, а не настройку числового типа.

- Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.

- Логический – тип для хранения логических данных (могут принимать только два значения, например Да или Нет).

- Поле объекта OLE – специальный тип данных, предназначенный для хранения объектов *OLE*, например мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае полей MEMO, они хранятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной).

- Гиперссылка – специальное поле для хранения адресов *URL Web*-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

- Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод в данных поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

Проектирование баз данных

Объекты базы данных

Кроме таблиц база данных может содержать и другие типы объектов. Привести полную классификацию возможных объектов баз данных затруднительно, поскольку каждая система управления базами данных может реализовать свои типы объектов. В **Microsoft Access** СУБД позволяет создавать и использовать объекты семи различных типов.

Таблицы. Это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства).

Запросы. Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную *результующую таблицу*.

Формы. Если запросы – это специальные средства для отбора и анализа данных, то формы – это средства для ввода данных. Смысл их тот же – предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочие) для автоматизации ввода.

С помощью форм данные можно не только вводить, но и отображать. Запросы тоже отображают данные, но делают это в виде результирующей таблицы, не имеющей почти никаких средств оформления. При выводе

данных с помощью форм можно применять специальные средства оформления.

Отчеты. По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер). В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т. п.)

Страницы. Это специальные объекты баз данных. Более корректно их называть *страницами доступа к данным*. Физически это особый объект, выполненный в коде *HTML*, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является базой данных, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и базой данных, размещенной на сервере.

Макросы и модули. Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с системой управления базами данных, так и для создания новых функций путем программирования. В СУБД **Microsoft Access** *макросы* состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. *Модули* создаются средствами внешнего языка программирования, в данном случае языка **Visual Basic for Applications**. Это одно из средств, с помощью которых разработчик базы может заложить в нее нестандартные функциональные возможности, удовлетворить специфические требования заказчика, повысить быстродействие системы управления, а также уровень ее защищенности.

Проектирование базы данных

Мы лишь рассмотрели основные понятия баз данных и еще не знакомы ни с одной системой управления базами данных, но, тем не менее, поднимаем вопрос о проектировании базы. Это не случайно. Методически правильно начинать работу с карандашом и листом бумаги в руках, не используя компьютер. На данном этапе он просто не нужен. Неоптимальные решения и прямые ошибки, заложенные на этапе проектирования, впоследствии очень трудно устраняются, поэтому этот этап является основополагающим.

Разработка технического задания. Техническое задание на проектирование базы данных должен предоставить заказчик. Однако для этого он должен владеть соответствующей терминологией и знать, хотя бы в общих

чертах, технические возможности основных систем управления базами данных. К сожалению, на практике такое положение встречается не всегда. Поэтому обычно используют следующие подходы:

- демонстрируют заказчику работу аналогичной базы данных, после чего согласовывают спецификацию отличий;
- если аналога нет, выясняют круг задач и потребностей заказчика, после чего помогают ему подготовить техническое задание.

При подготовке технического задания составляют:

- список исходных данных, с которыми работает заказчик;
- список выходных данных, которые необходимы заказчику для управления структурой своего предприятия;
- список выходных данных, которые не являются необходимыми для заказчика, но которые он должен предоставлять в другие организации (в вышестоящие структуры, в органы статистического учета, прочие административные и контролирующие организации).

При этом очень важно не ограничиваться взаимодействием с головным подразделением заказчика, а провести обсуждение со всеми службами и подразделениями, которые могут оказаться поставщиками данных в базу или их потребителями. Так, например, при подготовке базы данных для учета абитуриентов и студентов в высшем учебном заведении, необходимо не только изучить документооборот ректората и всех деканатов, но и понять, что хотели бы получить от базы данных службы. Следует изучить работу подразделений, распределяющих учебную нагрузку преподавателей, отвечающих за распределение аудиторного фонда, за проживание студентов в общежитии и других. В расчет должны приниматься и такие службы, как библиотека, отдел кадров и прочие. В любой момент может выясниться, например, что администрация библиотеки должна периодически предоставлять кому-то отчеты, характеризующие читательскую активность студентов в зависимости от пола, возраста и социального положения. К возможным пожеланиям заказчика следует готовиться на этапе проектирования, до создания базы.

Разработка структуры базы данных. Выяснив основную часть данных, которые заказчик предоставляет, можно приступить к созданию структуры базы, то есть структуры ее основных таблиц.

1. Работа начинается с составления генерального списка полей – он может насчитывать десятки и даже сотни позиций.

2. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

3. Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Цель – обеспечить, чтобы ввод данных в одну таблицу производился,

по возможности, в рамках одного подразделения, а еще лучше – на одном рабочем месте.

Наметив столько таблиц, сколько подразделений охватывает база данных, приступают к дальнейшему делению таблиц. Критерием необходимости деления является факт множественного повтора данных в соседних записях.

4. В каждой из таблиц намечают *ключевое поле*. В качестве такового выбирают поле, данные в котором повторяться не могут. Например, для таблицы данных о студентах таким полем может служить индивидуальный шифр студента. Для таблицы, в которой содержатся расписания занятий, такого поля можно и не найти, но его можно создать искусственным комбинированием полей «Время занятия» и «Номер аудитории». Эта комбинация неповторима, так как в одной аудитории в одно и то же время не принято проводить два различных занятия.

Если в таблице вообще нет никаких полей, которые можно было бы использовать как ключевые, всегда можно ввести дополнительное поле типа Счетчик – оно не может содержать повторяющихся данных по определению.

5. С помощью карандаша и бумаги расчерчивают связи между таблицами. Такой чертеж называется *схемой данных*.

Существует несколько типов возможных связей между таблицами. Наиболее распространенными являются связи «один ко многим» и «один к одному». Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым, то есть на стороне «один» должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения. Значения на стороне «многие» могут повторяться.

Рассмотрим таблицу Клиенты. Здесь поле Код клиента является ключевым. Это понятно, поскольку у каждого клиента должен быть свой уникальный код, идентифицирующий его однозначно. Если мы рассмотрим таблицу Заказы, то увидим, что в ней код клиента не может быть уникальным, поскольку каждый клиент мог сделать сколь угодно много заказов. На схеме данных эти поля соединены *линией связи*. С одной стороны эта линия маркирована знаком «1», с другой стороны – значком «бесконечность». Это графический метод изображения связи «один ко многим».

Ключевым полем в таблице заказов является Код заказа – он однозначно идентифицирует, кто, когда, что заказал и на какую сумму. Здесь же можно узнать, какой сотрудник принял заказ к исполнению. Поскольку один сотрудник может принять множество заказов, поле Код сотрудника в таблице заказов не является ни уникальным, ни ключевым, зато в таблице Сотрудники это поле уникально.

Про подобные таблицы говорят, что они связаны *реляционными отношениями*. Соответственно, системы управления, способные работать со связанными таблицами, называют *системами управления реляционными базами данных*, а схему данных в технической литературе могут называть *схемой реляционных отношений*.

6. Разработкой схемы данных заканчивается «бумажный» этап работы над техническим предложением. Эту схему можно согласовать с заказчиком, после чего приступить к непосредственному созданию базы данных.

Следует помнить, что по ходу разработки проекта заказчику непременно будут приходить в голову новые идеи. На всех этапах проектирования он стремится охватить единой системой все новые и новые подразделения и службы предприятия. Возможность гибкого исполнения его пожеланий во многом определяется квалификацией разработчика базы данных. Если схема данных составлена правильно, подключать к базе новые таблицы нетрудно. Если структура базы нерациональна, разработчик может испытать серьезные трудности и войти в противоречия с заказчиком.

Противоречия исполнителя с заказчиком всегда свидетельствуют о недостаточной квалификации исполнителя. Именно поэтому этап предварительного проектирования базы данных следует считать основным. От его успеха зависит, насколько база данных станет удобной и будут ли с ней работать пользователи. Если отмечается, что пользователи базы «саботируют» ее эксплуатацию и предпочитают работать традиционными методами, это говорит не о низкой квалификации пользователей, а о недостаточной квалификации разработчика базы.

5. КОМПЬЮТЕРНЫЕ КОММУНИКАЦИИ

5.1. Компьютерные коммуникации. Интернет

При физическом соединении двух или более компьютеров образуется компьютерная сеть. В общем случае для создания компьютерных сетей необходимо специальное аппаратное обеспечение (сетевое оборудование) и специальное программное обеспечение (сетевые программные средства). Простейшее соединение двух компьютеров для обмена данными называется прямым соединением. Для создания прямого соединения компьютеров работающих в операционной системе Windows не требуется не специального аппаратного не программного обеспечения. В этом случае аппаратными средствами являются стандартные устройства (порты ввода, вывода; последовательные или параллельные). А в качестве программного обеспечения используется стандартное средство имеющееся в составе операционной системы (пуск – программы – стандартные – связь – прямое кабельное соединение).

Основной задачей решаемой при создании компьютерных сетей является обеспечение совместимости оборудования по электрическим и механическим характеристикам и обеспечение совместимости информационного обеспечения (программ и данных) по системе кодирования и формата данных. Решение этой задачи относится к области стандартизации и основана на так называемой модели OSI (модель взаимодействия открытых систем). Она создана на основе технических предложений международного института стандарта.

Согласно модели OSI архитектуру компьютерных сетей следует рассматривать на разных уровнях (общее число уровней до 7). Самый верхний уровень – прикладной. На этом уровне пользователь взаимодействует с вычислительной системой. Самый нижний уровень – физический. Он обеспечивает обмен сигналами между устройствами. Обмен данными в системах связи происходит путем их перемещения с верхнего уровня на нижний, затем транспортировки и далее обратно воспроизведение на компьютере пользователя, в результате перемещения с нижнего уровня на верхний. Для обеспечения необходимой совместимости на каждом из 7 уровней архитектуры компьютерной сети действуют специальные стандарты, называемые протоколами. Они определяют характер аппаратного взаимодействия компонентов в сети (аппаратные протоколы) и характер взаимодействия программ и данных (программные протоколы).

Физически функции поддержки протоколов исполняют аппаратные устройства (интерфейсы), а также программные средства (программы поддержки протоколов). Программы, выполняющие поддержку протоколов также называют протоколами. Например, если два компьютера соединены

между собой прямым соединением то на низшем уровне протокол их взаимодействия определяет конкретное устройство физического порта (параллельного или последовательного) и механические компоненты (разъемы, кабели и так далее). На более высоком уровне взаимодействия между компьютерами определяют программные средства управляющих передач данных через порты. Для стандартных портов они находятся в базовой системе ввода-вывода (BIOS). На самом высоком уровне протокол взаимодействия обеспечивают приложения операционной системы. Например, для Windows 98 это стандартная программа – прямое кабельное соединение. В соответствии с используемыми протоколами кабельные сети принято разделять на локальные (LAN) и глобальные (WAN). Компьютеры локальной сети преимущественно используют единый комплект протоколов для всех участков. По территориальному признаку локальные сети отличаются компактностью. Они могут объединять компьютеры одного помещения, этажа и так далее, или группы компактно расположенных сооружений. Глобальные сети имеют как правило увеличенные географические размеры – земной шар. Они могут объединять как отдельные компьютеры, так и отдельные локальные сети, в том числе использующие различные протоколы. Назначение всех видов компьютерных сетей определяется двумя функциями:

- 1) обеспечение совместного использования аппаратных и программных ресурсов,
- 2) обеспечение совместного доступа к ресурсам данных.

Например, все участники локальной сети могут совместно использовать одно общее устройство печати (сетевой принтер) или например, ресурсы жестких дисков одного выделенного компьютера (файлового сервера). Это же относится и к программному и к информационному обеспечению. Если в сети имеется специальный компьютер выделенный для совместного использования участниками сети. Он называется файловым сервером. Компьютерные сети в которых нет выделенного сервера, а все локальные компьютеры могут общаться друг с другом на равных правах (обычно это небольшие сети и называются они одноранговыми). Группа сотрудников работающих над одним проектом в рамках локальной сети называется рабочей группой. В рамках одной локальной сети могут работать несколько рабочих групп. У участников рабочих групп могут быть разные права для доступа к общим ресурсам сети. Совокупность приемов разделения и ограничения прав участников в компьютерной сети называется политикой сети. Управление сетевыми политиками (их может быть несколько в одной сети) называется администрированием сети. Лицо, управляющее организацией работы участников локальной компьютерной сети называется системным администратором. Создание локальных сетей характерно для отдельных предприятий или отдельных подразделений пред-

приятый. Если предприятие занимает обширную территорию, то отдельные локальные сети могут объединяться в глобальные сети. В этом случае локальные сети связывают между собой с помощью любых традиционных каналов связи (кабельных, спутниковых и так далее). Для этой цели могут быть использованы так же телефонные каналы, хотя они в наименьшей степени удовлетворяют требованиям цифровой связи. Для связи между собой нескольких локальных сетей, работающих по разным протоколам служат специальные средства называемые шлюзами. Шлюзы могут быть как аппаратными так и программными. Например, это может быть специальный компьютер (шлюзовой сервер), а может быть и специальная компьютерная программа. В последнем случае компьютер может выполнить не только функцию шлюза, но и какие-то иные функции, типичные для рабочих станций. При подключении локальной сети предприятия к глобальной сети важную роль играет понятие сетевой безопасности, в частности должен быть ограничен доступ в локальную сеть для посторонних лиц извне, а также ограничен выход за пределы локальной сети для сотрудников предприятия не имеющих соответствующих прав. Для обеспечения сетевой безопасности между локальной и глобальной сетью устанавливают так называемые брандмауэры. В качестве брандмауэра может быть использован специальный компьютер или компьютерная программа препятствующая несанкционированным перемещениям данных между сетями.

Сетевые службы. Основные понятия.

Понятия виртуального соединения.

Рассмотрим простой пример взаимодействия двух корреспондентов с помощью обычной почты. Если они регулярно отправляют друг другу письма и соответственно получают их, то они могут полагать, что между ними существует соединение на пользовательском (прикладном) уровне. Однако это не совсем так. Такое соединение можно назвать виртуальным. Такое соединение было бы физическим если бы каждый из корреспондентов лично относил другому письмо и вручал в собственные руки. В реальной жизни он бросает его в почтовый ящик и ждет ответа. Сбором писем из общественных почтовых ящиков и доставкой корреспонденции в личные почтовые ящики занимаются местные почтовые службы. Это другой уровень модели связи, лежащий ниже. Для того чтобы наше письмо достигло адресата в другом городе должна существовать связь между нашей местной почтовой службой и его местной почтовой службой. Это еще один пример виртуальной связи. Поскольку никакой физической связью эти службы не обладают вступившую почтовую корреспонденцию они только сортируют и передают на уровень федеральной почтовой службы. Федеральная почтовая служба в своей работе опирается на службы очередного уровня. Например, на почтовую багажную службу железнодорожного ведомства и только рассмотрев работы этой службы мы найдем наконец при-

знаки физического соединения. Например, железнодорожный путь связывающий два города. Это очень простой пример поскольку в реальности даже доставка обычного письма может затронуть гораздо большее количество служб. Но нам важно обратить внимание на то, что в нашем примере образовалось несколько виртуальных соединений между аналогичными службами находящимися в пунктах отправки и приема. Не вступая в прямой контакт эти службы взаимодействуют между собой.

Модели взаимодействия открытых систем

Выше мы упомянули о том что согласно рекомендателям международного института стандартизации системы компьютерной связи рекомендуется рассматривать на 7 разных уровнях. Каждый новый уровень все больше и больше увеличивает функциональность системы связи, то есть чем выше уровень в модели связи, тем больше различных функциональных служб его используют.

Системы компьютерной связи рассмотрим как модели OSI. Происходит обмен данными между пользователями на разных континентах:

- на прикладном уровне с помощью специального приложения пользователь создает документ (сообщение, рисунок),

- на уровне представления операционная система его компьютера фиксирует, где находятся созданные данные (в оперативной памяти, в файле, на жестком диске и так далее) и обеспечивает взаимодействие со следующим уровнем,

- на сеансовом уровне компьютер пользователя взаимодействует с локальной или глобальной сетью. Протоколы этого уровня проверяют права пользователя на выход в эфир и передают документ протоколам транспортного уровня,

- на транспортном уровне документ преобразуется в ту форму, в которой положено передавать данные в используемой сети. Например, он может нарезаться на небольшие пакеты стандартного размера.

Сетевой уровень. Определяет маршрут движения данных в сети. Например, если на транспортном уровне данные были нарезаны на пакеты, то на сетевом уровне каждый пакет должен получить адрес, по которому он должен быть доставлен независимо от прочих пакетов.

Уровень соединения. Необходим для того, чтобы промодулировать сигналы циркулирующие на физическом уровне, в соответствии с данными полученными с сетевого уровня. Например, в компьютере эти функции выполняет сетевая карта или модем.

Реальная передача данных происходит на физическом уровне. Здесь нет ни документов ни пакетов ни даже байтов, а есть только биты – то есть элементарные единицы представления данных. Восстановление документа из них произойдет постепенно при переходе с нижнего на верхний уровень на компьютере клиента.

Средства физического уровня лежат за пределами компьютера. В локальных сетях это оборудование самой сети. При удаленной связи с использованием телефонных модемов это линия телефонной связи, коммутационное оборудование телефонных станций и т.п. На компьютере получателя информации происходит обратный процесс преобразования данных от битовых сигналов до документа.

Особенности виртуальных соединений

Разные уровни протоколов сервера и клиента не взаимодействуют друг с другом напрямую, но они взаимодействуют через физический уровень. Постепенно переходя с верхнего уровня на нижний данные непрерывно преобразуются, то есть обрастают другими данными, которые анализируются протоколами соответствующих уровней на сопредельной стороне. Это создает эффект виртуального взаимодействия уровней между собой. Не смотря на виртуальность это все-таки соединение, через которое проходят данные. Одновременно с теми запросами на поставку данных, которые клиент направляет серверу, передается масса служебной информации, которая может быть как желательной так и не желательной. Например, обязательно передаются данные о текущем адресе клиента, о дате и времени запроса, о версии его операционной системы, о его правах доступа, запрашиваемых данных и прочее. Передается немало косвенной информации, например, по какому адресу он посылал предыдущий запрос. Иногда даже передаются идентификационные коды процессоров компьютера.

На использовании виртуальных соединений основаны такие позитивные свойства электронных систем связи как возможность работать по одному физическому каналу сразу с несколькими серверами.

Но на них же основаны такие негативные средства как троянские программы. Троянская программа – разновидность компьютерного вируса, создающая во время сеанса связи виртуальные соединения для передачи данных о компьютере на котором установлена. Среди этих данных может быть парольная информация о содержании жесткого диска и т.п. В отличие от обычных компьютерных вирусов троянские программы как правило не производят разрушительных действий на компьютере и поэтому лучше маскируются.

Сетевые службы

На виртуальных соединениях основаны все службы современного Интернета. Например, пересылка сообщения от сервера к клиенту может проходить через десятки различных компьютеров. Это совсем не означает что на каждом компьютере сообщение должно пройти через все уровни. Ему достаточно подняться до сетевого уровня (определяющего адресацию) при приеме и вновь опуститься до физического уровня при передаче. В данном случае служба передачи сообщений основывается на виртуальном соединении сетевого уровня и соответствующих ему протоколов.

Интернет. Основные понятия.

Дословно, в переводе на русский язык означает “может”, то есть в узком смысле слова Интернет – это объединение сетей. Однако в последние годы у этого слова появился и более широкий смысл – “всемирная компьютерная сеть”.

Интернет можно рассматривать в физическом смысле как несколько миллионов компьютеров, связанных друг с другом всевозможными линиями связи. Но в последнее время Интернет можно рассматривать как некое информационное пространство. Интернет это не совокупность прямых соединений между компьютерами. Например, если два компьютера находятся на разных континентах, то это совсем не значит, что между ними находится прямое или виртуальное соединение, хотя они обмениваются данными в интернете. Данные, которые они посылают друг другу разбиваются на пакеты и даже в одном сеансе связи разные пакеты одного сообщения могут пройти разными маршрутами. Какими бы маршрутами не двигались пакеты данных они все равно достигнут пункта назначения и будут собраны вместе в цельный документ. При этом данные отправленные позже могут приходить раньше. Это не мешает правильно собрать документ поскольку каждый пакет имеет свою маркировку.

Интернет представляет собой как бы пространство внутри которого осуществляется непрерывная циркуляция данных. В этом смысле его можно сравнить с теле- и радиоэфиром. Разница в том что в эфире никакая информация храниться не может, а в Интернете она перемещается между компьютерами составляющими в узлы сети и какое-то время хранится на жестких дисках.

Теоретические основы Интернета

Ранние эксперименты по передаче и приему информации с помощью компьютеров начались еще в 50-х годах и имели лабораторный характер. В конце 60-х годов на средства агентства перспективных разработок министерство обороны США (DARPA) была создана первая сеть национального масштаба. По имени агентства она получила название ARPANET. Эта сеть связала несколько крупных научно-исследовательских и образовательных центров. Её основной задачей стала координация групп коллективов работающих над едиными научно-техническими проектами. Основным назначением стало обмен электронной почтой и файлами с научной и проектно-конструкторской документацией. Сеть ARPANET заработала в 1969 году. Немногочисленные узлы, входившие в неё в то время были связаны выделенными линиями. Прием и передача информации обеспечивались программами, работающими на узловых компьютерах. Сеть постепенно расширялась за счет подключения новых узлов. К началу 80-х годов на базе наиболее крупных узлов были созданы свои региональные сети, которые воссоздавали общую архитектуру сети ARPANET, но на более низком

уровне (в региональном или локальном масштабе). Всякий раз, когда мы говорим о вычислительной технике надо иметь в виду принцип единства аппаратного и программного обеспечения. Пока глобальное расширение ARPANET происходило за счет механического подключения все новых и новых аппаратных средств (узлов и сетей) до Интернета было далеко. Рождением Интернета принято считать 1983 год. Считается, что в этом году произошли революционные изменения в программном обеспечении компьютерной связи. Днем рождения Интернета в современном понимании этого слова стала дата стандартизации протокола связи TCP/IP, лежащего в основе всемирной сети по нынешний день. Уточним, что в современном понимании TCP/IP это не один сетевой протокол, а два протокола лежащих на разных уровнях (так называемый стек протокол). Протокол TCP это протокол транспортного уровня. Он управляет тем, как происходит передача информации. Протокол IP-адресный, он принадлежит сетевому уровню и определяет куда происходит передача информации.

Протокол TCP

Согласно протоколу TCP отправляемые данные нарезаются на небольшие пакеты. После чего каждый пакет маркируется таким образом, чтобы в нем были данные необходимые для правильной сборки документа на компьютере получателя. Для правильного понимания сути протокола TCP можно представить игру в шахматы по переписке. Когда двое участников разыгрывают одновременно десятки партий, каждый ход записывается на отдельной открытке с указанием номера партии и номера хода. В этом случае между двумя партнерами через один и тот же почтовый канал работает как бы десятки соединений (по одному на партию). Два компьютера, связанные между собой одним физическим соединением могут точно так же поддерживать одновременно несколько TCP соединений. Так, например, два промежуточных сетевых сервера могут одновременно по одной линии связи передавать друг другу в обе стороны множество TCP-пакетов от многочисленных клиентов.

5.2. Приемы защиты информации

Сложность и объем информационных потоков в современном мире возрастает с каждым днем, информация оказывает непосредственное влияние на деятельность государственных органов, предприятий, организаций, жизнь частных лиц. От того как обеспечена защита циркулирующей в любом срезе общества информации, зависит его жизнедеятельность и безопасность людей. Поэтому вопросам защиты информации во все времена придавалось и придается огромное значение.

Информационная безопасность – это защищенность информационной среды посредством различных средств и методов.

Под информационной средой понимается совокупность информационных ресурсов и наличие соответствующей инфраструктуры их создания и использования. Целью информационной безопасности является предотвращение влияния неблагоприятных событий (угроз) или обеспечение минимального ущерба от них информационной среде.

Информационная безопасность определяется способностью государства, общества, личности

- обеспечивать защищенность информационных ресурсов для поддержания своей жизнедеятельности и развития;
- противостоять негативным информационным воздействиям на сознание и психику людей, а также на компьютерные сети и другие технические источники информации;
- вырабатывать личностные и групповые навыки безопасного поведения;
- поддерживать постоянную готовность к мерам информационного противоборства.

Информационная безопасность должна обеспечивать конфиденциальность, точность, полноту и доступность информации. Эффективная информационная безопасность должна основываться на теории защиты информации, в которой рассматриваются следующие направления:

- сбор, систематизация и анализ сведений о проблеме защиты информации;
- формирование на основе собранных сведений научно обоснованных прогнозов о возможности возникновения угроз;
- научно обоснованная постановка задачи защиты информации в современных условиях;
- разработка мероприятий по организации защиты информации;
- разработка методологии и инструментальной базы защиты информации.

В Российской Федерации был выработан ряд подходов к защите информации, закрепленный государственными документами и стандартами. Одним из таких документов является «Доктрина информационной безопасности Российской Федерации». Согласно этому документу, государственная политика обеспечения информационной безопасности основывается на следующих основных принципах:

- соблюдении Конституции РФ, законодательства РФ, общепризнанных принципов и норм международного права при осуществлении деятельности по обеспечению информационной безопасности РФ;
- информированности общества о деятельности федеральных органов государственной власти и общественных объединений;

- правовом равенстве всех участников процесса информационного взаимодействия вне зависимости от их политического, социального и экономического статуса;
- приоритетном развитии отечественных информационных и телекоммуникационных технологий, производстве технических и программных средств в целях соблюдения жизненно важных интересов РФ.

Виды угроз в информационной сфере

Угрозы информационной безопасности РФ иллюстрирует рис.5.1.

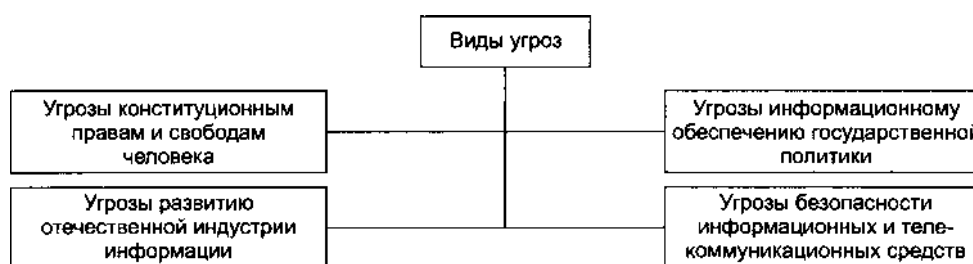


Рис. 5.1. Виды угроз информационной безопасности

Угрозами конституционным правам и свободам человека и гражданина в области духовной жизни и информационной деятельности могут являться:

- принятие федеральными органами государственной власти правовых актов, ущемляющих конституционные права и свободы граждан в области духовной жизни и информационной деятельности;
- нерациональное, чрезмерное ограничение доступа к общественно необходимой информации;
- противоправное применение специальных средств воздействия на индивидуальное, групповое и общественное сознание;
- неисполнение органами государственной власти, организациями и гражданами требований федерального законодательства, регулирующего отношения в информационной сфере;
- вытеснение российских информационных агентств, средств массовой информации с внутреннего информационного рынка;
- манипулирование информацией (дезинформация, сокрытие или искажение информации).

Угрозами информационному обеспечению государственной политики РФ могут являться:

- монополизация информационного рынка России отечественными и зарубежными компаниями;
- блокирование деятельности государственных средств массовой информации по информированию российской и зарубежной аудитории;

- дефицит квалифицированных кадров;
- отсутствие системы формирования и реализации государственной информационной политики.

Угрозами развитию отечественной индустрии информации, включая индустрию средств информатизации, телекоммуникации и связи, могут являться:

- противодействие доступу РФ к новейшим информационным технологиям;
- создание условий для технологической зависимости РФ в области современных информационных технологий;
- закупка органами государственной власти импортных средств информатизации, телекоммуникации и связи при наличии отечественных аналогов;
- вытеснение с отечественного рынка российских производителей средств информатизации, телекоммуникации и связи;
- отток за рубеж специалистов и правообладателей интеллектуальной собственности.

Угрозами безопасности информационных и телекоммуникационных средств и систем могут являться:

- нарушения технологии обработки информации;
- внедрение в аппаратные и программные изделия компонентов, реализующих не предусмотренные документацией функции;
- разработка и распространение программ, нарушающих нормальное функционирование информационных и информационно-телекоммуникационных систем;
- уничтожение или повреждение систем обработки информации;
- воздействие на системы защиты автоматизированных систем обработки и передачи информации;
- утечка информации по техническим каналам;
- внедрение электронных устройств для перехвата информации в технические средства обработки, хранения и передачи информации по каналам связи, а также в служебные помещения органов государственной власти, предприятий, учреждений и организаций независимо от формы собственности;
- уничтожение, повреждение, разрушение или хищение машинных и других носителей информации;
- перехват информации в сетях передачи данных и на линиях связи, дешифрование этой информации и навязывание ложной информации;
- использование сертифицированных отечественных и зарубежных информационных технологий при создании и развитии российской информационной инфраструктуры;
- несанкционированный доступ к информации в базах данных;
- нарушение законных ограничений на распространение информации.

Внутренние и внешние источники угроз

Источники угроз информационной безопасности РФ подразделяются на внешние и внутренние.

К *внешним источникам угроз* относятся:

- деятельность иностранных разведок в информационной сфере;
- ущемление интересов России в мировом информационном пространстве;
- обострение международной конкуренции за обладание информационными технологиями и ресурсами;
- деятельность международных террористических организаций;
- отставание РФ в сфере создания конкурентоспособных информационных технологий;
- разработка концепций информационных войн.

К *внутренним источникам угроз* относятся:

- критическое состояние отечественных отраслей промышленности;
- неблагоприятная криминогенная обстановка, рост количества преступлений в информационной сфере;
- плохая координация деятельности органов государственной власти в области обеспечения информационной безопасности РФ;
- отсутствие нормативной правовой базы в информационной сфере, а также слабая правоприменительная практика;
- неразвитость институтов гражданского общества;
- недостаточное финансирование мероприятий по обеспечению информационной безопасности РФ;
- недостаток квалифицированных кадров в области обеспечения информационной безопасности;
- отставание России от ведущих стран мира по уровню информатизации органов государственной власти.

Организационно-правовое обеспечение информационной безопасности

Организационно-правовое обеспечение информационной безопасности – это совокупность решений, законов, нормативов, регламентирующих организацию работ, создание и функционирование систем защиты информации.

Основные функции организационно-правовой базы следующие:

- разработка основных принципов отнесения конфиденциальных сведений к защищаемой информации;
- определение системы органов и должностных лиц, ответственных за обеспечение и регулирование информационной безопасности в стране;
- создание комплекса нормативно-правовых документов, регламентирующих обеспечение информационной безопасности;
- определение мер ответственности за нарушения правил защиты;
- определение порядка разрешения спорных и конфликтных ситуаций по вопросам защиты информации.

Созданием законодательной базы в области информационной безопасности каждое государство стремится защитить свои информационные ресурсы. Информационные ресурсы государства в самом первом приближении могут быть разделены на три большие группы:

- открытую информацию, на распространение и использование которой не имеется ограничений;
- запатентованную информацию, которая охраняется законодательством или международными соглашениями как объект интеллектуальной собственности;
- информацию, «защищаемую» ее владельцем с помощью механизмов защиты государственной, коммерческой или другой охраняемой тайны.

Одним из важнейших в системе защиты государственных секретов в любой стране является понятие государственной тайны. Определение этого понятия дано в Законе РФ «О государственной тайне»:

Государственная тайна – это защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности РФ.

В зависимости от вида, содержания и размеров ущерба можно выделить группы некоторых видов ущерба при утечке сведений, составляющих государственную тайну.

Политический ущерб может наступить при утечке сведений политического и внешнеполитического характера, о разведывательной деятельности спецслужб государства и др. В результате политического ущерба могут произойти серьезные изменения в международной обстановке, утрата страной политических приоритетов в каких-то областях, ухудшение отношений с какой-либо страной и т.д.

Экономический ущерб может наступить при утечке сведений любого содержания: политического, экономического, военного, научно-технического и т.д. Экономический ущерб может быть выражен в денежном исчислении в виде прямых и косвенных потерь. Например, прямые потери могут наступить в результате утечки секретной информации о системах вооружения и, как следствие, потребовать крупных затрат на их замену. Косвенные потери чаще всего выражаются в виде размера упущенной выгоды, например, из-за срыва переговоров с иностранными фирмами, утраты приоритета в научном исследовании, и т.п.

Моральный ущерб наступает от утечки информации, вызвавшей противоправную государству пропагандистскую кампанию, подрывающую репутацию страны.

Засекречивать информацию имеют право органы власти, управления и должностные лица, наделенные соответствующими полномочиями. Осуществляется эта деятельность посредством совокупности организационно-правовых мер, регламентированных законами и другими нормативными актами на распространение и использование информации в интересах ее собственника.

Государственная система правового обеспечения защиты информации в Российской Федерации

Государственная система обеспечения информационной безопасности создается для решения следующих проблем, требующих законодательной поддержки:

- защита персональных данных;
- борьба с компьютерной преступностью, в первую очередь в финансовой сфере;
- защита коммерческой тайны и обеспечение благоприятных условий для предпринимательской деятельности;
- защита государственных секретов;
- создание системы взаимных финансовых расчетов в электронной форме с элементами цифровой подписи;
- обеспечение безопасности автоматизированных систем управления (АСУ) потенциально опасными производствами;
- страхование информации и информационных систем;
- сертификация и лицензирование в области безопасности, контроль безопасности информационных систем;
- организация взаимодействия в сфере защиты данных со странами-членами СНГ и другими государствами.

Общая организация и координация работ в стране по защите информации, обрабатываемой техническими средствами, осуществляется Федеральной службой по техническому и экспортному контролю (ФСТЭК России). ФСТЭК России является федеральным органом исполнительной власти, осуществляющим реализацию государственной политики, организацию межведомственной координации и взаимодействия, специальные и контрольные функции в области государственной безопасности по следующим вопросам в области информационной безопасности:

- обеспечение безопасности информации в системах информационной и телекоммуникационной инфраструктуры;
- противодействие иностранным техническим разведкам на территории РФ;
- обеспечение защиты (некриптографическими методами) информации, отнесенной к государственной тайне, предотвращение ее утечки по техническим каналам и несанкционированного доступа к ней;
- защита информации при разработке, производстве, эксплуатации и утилизации неинформационных комплексов, систем и устройств.

Основными задачами в области обеспечения информационной безопасности для ФСТЭК России являются:

- реализация государственной политики в области обеспечения безопасности информации;
- организация деятельности государственной системы противодействия техническим разведкам;

- самостоятельное нормативно-правовое регулирование вопросов, связанных с безопасностью информации;
- обеспечение в пределах своей компетенции безопасности информации в ключевых системах информационной инфраструктуры государства;
- прогнозирование развития сил, средств и возможностей технических разведок, выявление угроз безопасности информации;
- осуществление центральным аппаратом ФСТЭК России организационно-технического обеспечения деятельности Межведомственной комиссии по защите государственной тайны.

Нормативные правовые акты и методические документы, изданные по вопросам деятельности ФСТЭК России, обязательны для исполнения аппаратами федеральных органов государственной власти и органов государственной власти субъектов РФ, федеральными органами исполнительной власти, органами исполнительной власти субъектов РФ, органами местного самоуправления и организациями.

Разработка правового обеспечения защиты информации идет по трем направлениям:

- защита прав личности на частную жизнь;
- защита государственных интересов;
- защита предпринимательской и финансовой деятельности.

Структура нормативной базы по вопросам информационной безопасности включает:

- Конституцию РФ;
- федеральные законы и законы РФ;
- кодексы РФ (уголовный, гражданский, об административных правонарушениях);
- постановления правительства РФ;
- ведомственные нормативные акты, ГОСТы, руководящие документы.

Угрозы в информационных системах

Угрозы информационной безопасности информационных систем можно подразделить на несколько групп в зависимости от того, на какой параметр информации воздействует угроза (рис. 5.2).

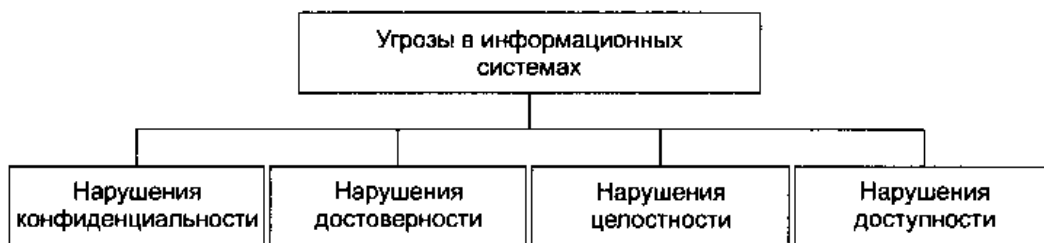


Рис. 5.2. Угрозы в информационных системах

Нарушения конфиденциальности

Нарушение конфиденциальности является одной из наиболее тяжелых с точки зрения ущерба угроз информационной безопасности. Если в случае

нарушений достоверности, целостности и доступности информации факт угрозы обнаруживается немедленно или в течение сравнительно короткого промежутка времени, то в случае нарушения конфиденциальности деструктивное воздействие может продолжаться длительное время, нанося тяжелый материальный, моральный и политический ущерб.

Нарушение конфиденциальности наступает в результате следующих действий:

- разглашение – умышленные или неосторожные действия должностных лиц и граждан, приведшие к ознакомлению посторонних лиц с конфиденциальными сведениями;
- утечка – бесконтрольный выход информации за пределы круга лиц или организаций, которым она была доверена;
- несанкционированный доступ – противоправное преднамеренное овладение конфиденциальной информацией лицом, не имеющим на это права.

Пример. Сбор конфиденциальных (в том числе военных) сведений разведками иностранных государств или промышленный шпионаж, позволяющий похитить информацию, дающую конкурентные преимущества.

Нарушения достоверности

Нарушение достоверности – это вторая по тяжести последствий после нарушения конфиденциальности угроза. Нарушение достоверности информации происходит в результате фальсификации, подделки или мошенничества. Ущерб от этой угрозы может быть менее масштабным, чем от нарушения конфиденциальности, но при этом последствия могут быть катастрофическими.

Пример. Подмена кодов или координат цели в системе наведения стратегических ядерных ракет, изменение режимов синхронизации в глобальной энергетической системе, вызывающее разрушение генераторов и массовое отключение электроэнергии.

Нарушения целостности

Нарушение целостности информации обычно происходит в каналах передачи или в хранилищах информации и заключается в искажениях, внесении ошибок и потере части передаваемой или хранимой информации. Обычно нарушение целостности информации выявляется гораздо раньше, чем нарушение достоверности, поскольку целостность информации легче поддается автоматизированному (аппаратному) контролю. Последствия нарушения целостности информации обычно носят тактический характер, а тяжесть последствий зависит от ситуации.

Пример. Нарушение целостности информации в распределенной информационной системе управления предприятием может привести к финансовым потерям одной фирмы, но такое же нарушение в распределенной

информационной системе противовоздушной обороны в момент ракетного нападения может быть фатальным.

Нарушения доступности

Нарушение доступности происходит вследствие блокирования, невозможности искажения или уничтожения информации. Последствия от нарушения доступности, так же как и в случае нарушения целостности, могут иметь разную тяжесть в зависимости от ситуации.

Пример. Стирание информации на магнитных носителях в банковской системе может иметь тяжелые финансовые и экономические последствия, в то время как физическое уничтожение библиотеки уникальных документов может быть культурной катастрофой.

Классификация угроз информации

Классификацию угроз иллюстрирует рис. 5.3.

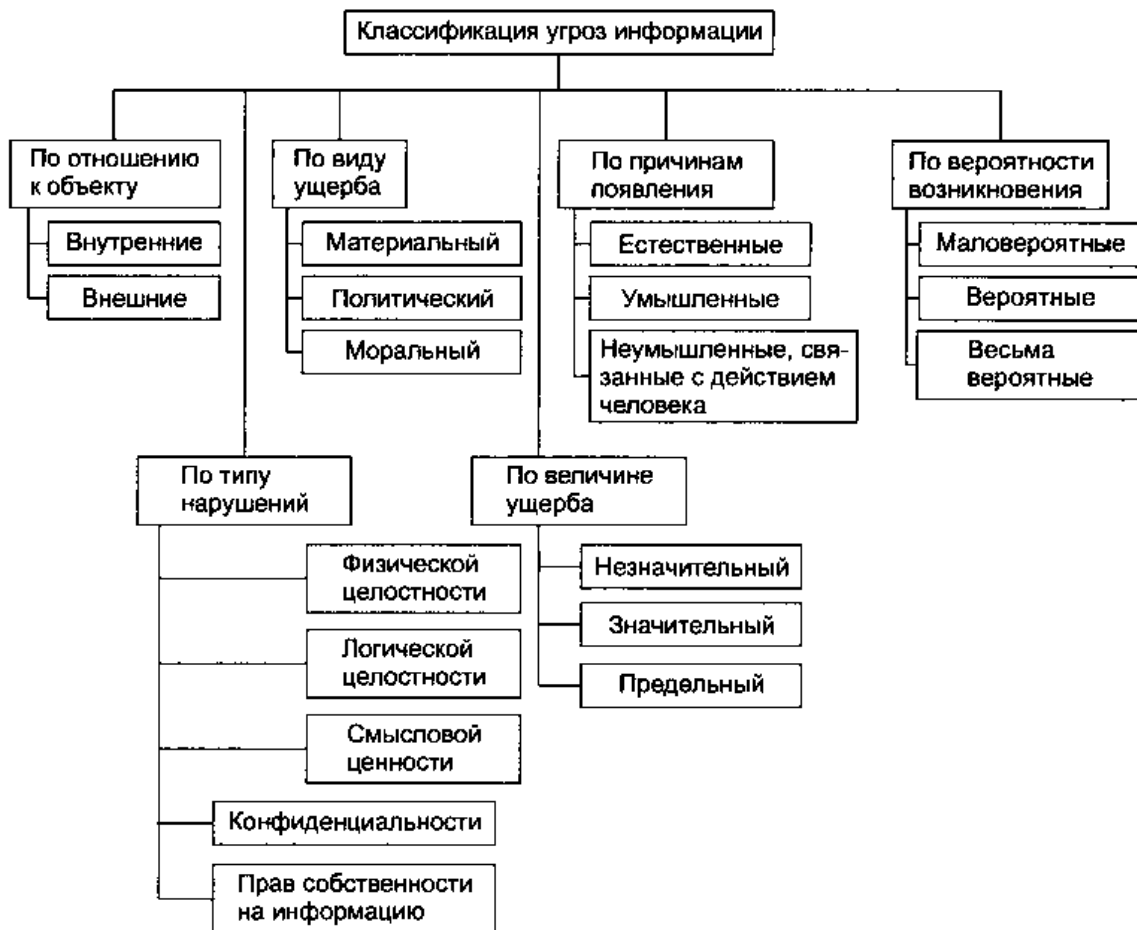


Рис. 5.3. Классификация угроз

Классификация угроз по отношению к объекту

Разделение угроз на внутренние и внешние зависит от того, где проводится граница между объектом, содержащим информацию, и внешним миром. Так, для персонального компьютера внешней угрозой могут быть дей-

ствия пользователя локальной сети. Эта же угроза с точки зрения организации, которой принадлежит данная локальная сеть, будет внутренней.

Наиболее опасной и вероятной внутренней угрозой для любой информационной системы или системы охраны информации являются низкая квалификация, халатность или враждебные действия персонала (пользователей).

Классификация угроз по виду ущерба

Разделение ущерба на материальный и моральный в достаточной мере условно: тот и другой, в конце концов, ведут к материальным потерям, а размеры материального ущерба, косвенно вызванного моральным ущербом, могут быть больше, чем прямые материальные потери.

Пример. Рассылка в локальной сети организации писем, содержащих провокационные сведения, приводит к созданию атмосферы разобщенности, недоверия, дезорганизованности в коллективе. Такая атмосфера может послужить почвой как для намеренных, так и для ненамеренных действий, приводящих к значительному материальному ущербу (разглашение конфиденциальных данных, техническая авария крупных размеров).

Классификация угроз по причинам появления

По причинам появления угрозы можно разделить на связанные с явлениями природы (стихийные бедствия или эксцессы) и действиями человека. В свою очередь, человеческие действия, составляющие угрозу, могут совершаться на основе умысла (шпионаж, диверсии, кража или порча конфиденциальной информации) или без умысла (халатность, некомпетентность, результат плохого самочувствия).

Классификация угроз по вероятности возникновения

Вероятность возникновения того или иного вида угроз не является постоянной величиной и зависит от комплекса причин. В разных ситуациях возникновение одной и той же угрозы может иметь разную вероятность.

Пример. Угроза физического разрушения сетевой инфраструктуры или информационных хранилищ маловероятна в мирное время, однако во время вооруженных конфликтов она может стать весьма вероятной и даже основной.

Классификация угроз по типу нарушений

Под нарушениями в первую очередь понимаются нарушения целостности информации: физической (уничтожение, разрушение элементов), логической (разрушение логических связей), смысловой (навязывание ложной информации). Нарушение конфиденциальности выражается в разрушении защиты и несанкционированном овладении конфиденциальной информацией. Все чаще в последнее время в нашей стране обращают внимание на нарушение прав собственности на информацию (несанкционированное копирование, использование).

Защита информации в информационных системах

Требования к защите информации

Конкретные требования к защите, обусловленные спецификой автоматизированной обработки информации, определяются совокупностью следующих факторов:

- характером обрабатываемой информации;
- объемом обрабатываемой информации;
- продолжительностью пребывания информации в автоматизированной системе обработки информации;
- структурой автоматизированной системы обработки данных;
- видом защищаемой информации;
- технологией обработки информации;
- организацией информационно-вычислительного процесса в автоматизированной системе обработки данных;
- этапом жизненного цикла автоматизированной системы обработки данных. Информация должна защищаться во всех структурных элементах информационной системы.

На рис. 5.4 представлена структурная схема автоматизированной системы обработки информации с мэйнфреймовой структурой (одним центральным вычислителем и пользовательскими местами в виде терминалов). Именно такая система обеспечивает наиболее эффективную защиту информации. Если терминалами пользователя являются персональные компьютеры, необходимо сделать конструктивно невозможными бесконтрольное снятие и ввод информации через устройства ввода-вывода (например, физически заблокировать или удалить все разъемы подключения USB-устройств, порты ввода-вывода, устройства чтения гибких и лазерных дисков).

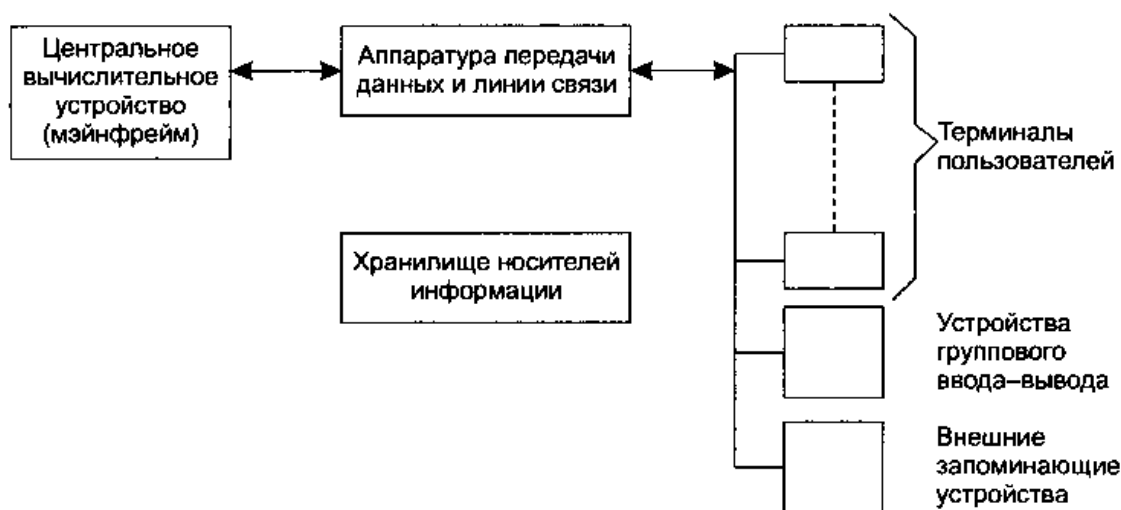


Рис. 5.4. Структурная схема информационной системы

Защита информации в терминалах пользователей: защищаемая информация может находиться только во время сеанса; должна быть исключена возможность просмотра отображаемой информации со стороны; информация, имеющая ограничительный гриф, должна выдаваться (отображаться) совместно с этим грифом.

Защита информации в устройствах группового ввода-вывода: в устройствах группового ввода-вывода информация может находиться только во время решения задач либо с нормированной длительностью хранения;

- устройства отображения и фиксации информации должны исключать возможность просмотра отображаемой информации со стороны;
- информация, имеющая ограничительный гриф, должна выдаваться (отображаться) совместно с этим грифом.

Защита информации в аппаратуре и линиях связи:

- в аппаратуре и линиях связи защищаемая информация должна находиться только в течение сеанса;
- линии связи, по которым защищаемая информация передается в явном виде, должны находиться под непрерывным контролем на все время сеанса передачи;
- перед началом каждого сеанса передачи должна осуществляться проверка адреса выдачи данных;
- при передаче большого объема защищаемой информации проверка адреса передачи должна производиться периодически (через заданные промежутки времени или после передачи заданного числа знаков сообщения);
- при наличии в составе аппаратуры связи процессоров и запоминающих устройств должна вестись регистрация данных обо всех сеансах передачи защищаемой информации.

Защита информации в центральном вычислителе:

- защищаемая информация в оперативном запоминающем устройстве может находиться только во время сеансов решения соответствующих задач, во внешних запоминающих устройствах – минимальное время, определяемое технологией решения соответствующей прикладной задачи в автоматизированной системе обработки данных;
- устройства отображения и фиксации информации должны исключать возможность просмотра отображаемой информации со стороны;
- информация, имеющая ограничительный гриф, должна выдаваться (отображаться) совместно с этим грифом;
- при обработке защищаемой информации должно осуществляться установление подлинности всех участвующих в обработке устройств и пользователей с ведением протоколов их работы;
- всякое обращение к защищаемой информации должно проверяться на санкционированность;

- при обмене защищаемой информацией, осуществляемом с использованием линий связи, должна осуществляться проверка адресов корреспондентов. Защита информации во внешних запоминающих устройствах.

- сменные носители информации должны находиться на устройствах управления в течение минимального времени, определяемого технологией автоматизированной обработки информации;

- устройства управления внешних запоминающих устройств, на которых установлены носители с защищаемой информацией, должны иметь замки, предупреждающие несанкционированное изъятие или замену носителя.

Защита информации в хранилище носителей:

- все носители, содержащие защищаемую информацию, должны иметь четкую и однозначную маркировку, которая, однако, не должна раскрывать содержания записанной на них информации;

- носители, содержащие защищаемую информацию, должны храниться таким образом, чтобы исключались возможности несанкционированного доступа к ним;

- при выдаче и приемке носителей должна осуществляться проверка личности получающего (сдающего) и его санкции на получение (сдачу) этих носителей. Защита информации для всех устройств: для всех устройств должна быть предусмотрена возможность аварийного уничтожения информации.

Способы и средства защиты информации

Содержание способов обеспечения безопасности представлено на рис. 5.5.

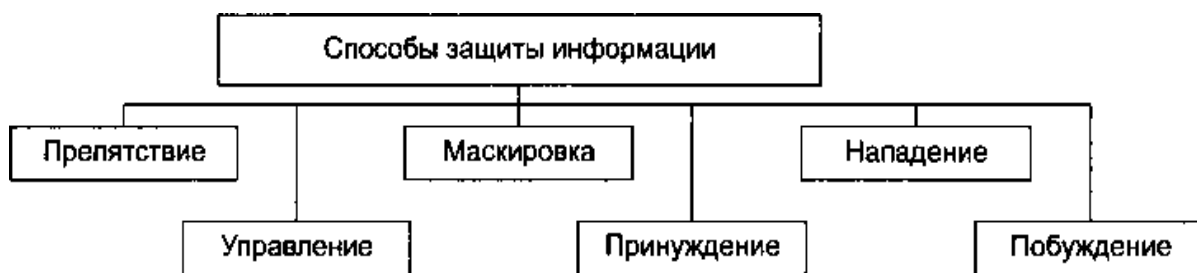


Рис. 5.5. Методы защиты информации

Препятствие заключается в создании на пути возникновения или распространения угрозы барьера, не позволяющего угрозе принять опасные размеры.

Пример. Блокировки, не позволяющие техническому устройству или программе выйти за опасные границы; создание физических препятствий на пути злоумышленников, экранирование помещений и технических средств и т. п.

Управление (или регламентация) заключается в определении алгоритмов функционирования систем обработки информации, а также процедур и

правил работы предприятий и учреждений, препятствующих возникновению угрозы.

Пример. Управление доступом на секретный объект, включающее в себя процедуры и правила, которые должен выполнять персонал объекта, а также алгоритмы работы механизмов и устройств слежения, фиксации и ограничения доступа.

Маскировка – преобразование информации или скрываемого объекта, вследствие которого снижается степень распознавания скрываемой информации или объекта и затрудняется доступ к ним.

Пример. Шифрование информации, дезинформация о месте нахождения конфиденциальной информации, создание легенд, намеренное внесение помех.

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

Побуждение есть способ защиты информации, при котором пользователи и персонал объекта внутренне (благодаря материальным, моральным, этическим, психологическим и другим мотивам) побуждаются к соблюдению всех правил обработки информации.

Нападение – способ защиты, применяемый в активной фазе информационной войны. Цель – заставить противника сосредоточить усилия на защите, ослабив усилия на создание угроз.

Рассмотренные способы обеспечения защиты информации реализуются с применением различных средств (рис. 5.6).

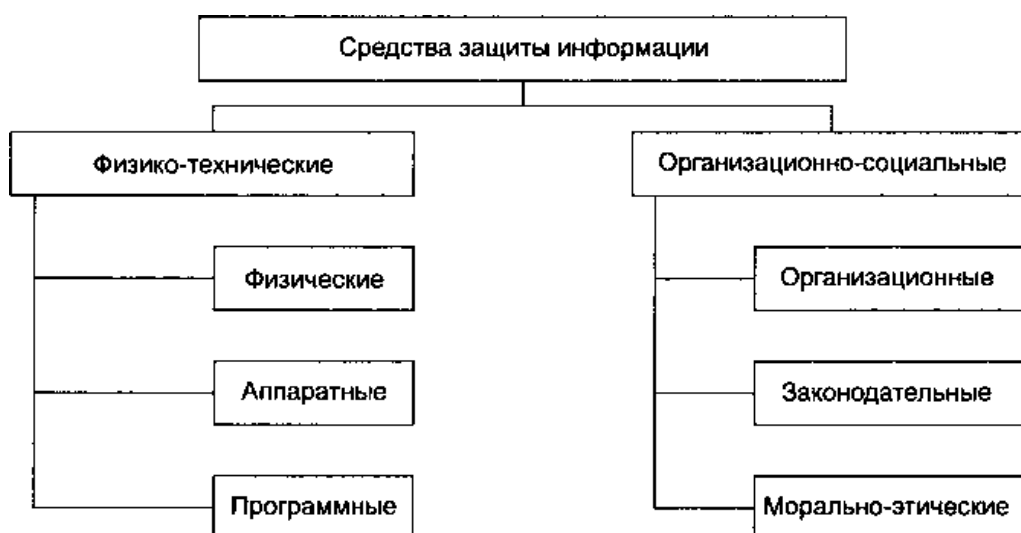


Рис. 5.6. Средства защиты информации

Физико-технические средства делятся на физические, аппаратные и программные.

- Физические средства – механические, электрические, электромеханические и т.п. устройства и системы, которые функционируют автономно, создавая различного рода препятствия на пути угроз.

- Аппаратные средства – различные электронные и электронно-механические и т.п. устройства, встраиваемые в схемы аппаратуры системы обработки данных или сопрягаемые с ней специально для решения задач защиты информации. Например, для защиты от утечки по техническим каналам используются генераторы шума.

Физические и аппаратные средства объединяются в класс технических средств защиты информации.

- Программные средства – специальные пакеты программ или отдельные программы, включаемые в состав программного обеспечения автоматизированных систем с целью решения задач защиты информации. Это могут быть различные программы по криптографическому преобразованию данных, контролю доступа, защиты от вирусов и др.

Организационно-социальные средства делятся на организационные, законодательные и морально-этические.

- Организационные средства – специально предусматриваемые в технологии функционирования объекта организационно-технические мероприятия для решения задач защиты информации, осуществляемые в виде целенаправленной деятельности людей.

- Законодательные средства – нормативно-правовые акты, с помощью которых регламентируются права и обязанности всех лиц и подразделений, связанных с защитой информации, а также устанавливается ответственность за нарушение правил обработки информации, следствием чего может быть нарушение защищенности информации.

- Морально-этические средства защиты информации предполагают, прежде всего, воспитание сотрудника, допущенного к секретам, то есть проведение специальной работы, направленной на формирование у него системы определенных качеств, взглядов и убеждений (патриотизма, понимания важности и полезности защиты информации и для него лично), обучение сотрудника, осведомленного в сведениях, составляющих охраняемую тайну, правилам и методам защиты информации, привитие ему навыков работы с носителями секретной и конфиденциальной информации.

Системы защиты информации

С целью всесторонней защиты информации рекомендуется создать единую, целостную систему, являющуюся функционально самостоятельной подсистемой любого объекта обработки информации. Такое решение позволяет объединить все ресурсы, средства и методы, а также полноценно координировать мероприятия по защите информации.

Организационно система защиты информации (СЗИ) должна состоять из трех механизмов:

- обеспечения защиты информации;
- управления средствами защиты;
- общей организации работы системы.

Ядро системы защиты информации

Ядро системы защиты предназначено для объединения всех подсистем системы защиты информации в единую целостную систему и организации ее функционирования.

Ядро может включать в себя организационные и технические составляющие. Организационная составляющая представляет собой совокупность специально выделенных для защиты информации сотрудников, выполняющих свои функции в соответствии с разработанными правилами, а также нормативную базу, регламентирующую выполнение этих функций.

Техническая составляющая обеспечивает техническую поддержку организационной составляющей и представляет собой совокупности технических средств отображения состояний элементов системы защиты информации, контроля доступа к ним, управления их включением и т.д. Чаще всего эти средства объединены в соответствующий пульт управления системы защиты информации.

Ядро системы защиты информации должно обеспечивать выполнение двух функций.

- Включение компонентов системы защиты информации в работу при поступлении запросов на обработку защищаемой информации и блокирование бесконтрольного доступа к ней. Для этого требуется:

- оборудование объекта средствами охранной сигнализации;
- хранение носителей защищаемой информации в отдельных хранилищах (документация, шифры, магнитные носители и т. д.);
- включение блокирующих устройств, регулирующих доступ к элементам системы защиты информации при предъявлении соответствующих полномочий и средств сигнализации.

- Организация и обеспечение проверок правильности функционирования системы защиты информации. При этом реализуется проверка:

- аппаратных средств – по тестовым программам и организационно;
- физических средств – организационно (плановые проверки средств охранной сигнализации, сигнализации о повышении давления в кабелях и т.д.);
- программных средств – по специальным контрольным суммам (на целостность) и по другим идентифицирующим признакам.

Ресурсы системы защиты информации

Ресурсы информационно-вычислительной системы, необходимые для создания и поддержания функционирования системы защиты информации,

как и любой другой автоматизированной системы, объединяются в техническое, математическое, программное, информационное и лингвистическое обеспечение.

Техническое обеспечение – совокупность технических средств, необходимых для технической поддержки решения всех тех задач защиты информации, которые возникают в процессе функционирования системы защиты информации.

Математическое обеспечение – совокупность математических методов, моделей и алгоритмов, необходимых для оценки уровня защищенности информации и решения других задач защиты.

Программное обеспечение – совокупность программ, реализующих программные средства защиты, а также программ, необходимых для решения задач управления механизмами защиты. К ним должны быть отнесены также сервисные и вспомогательные программы системы защиты информации.

Информационное обеспечение – совокупность систем классификации и кодирования данных о защите информации, массивы данных системы защиты информации, в также входные и выходные документы системы защиты информации.

Лингвистическое обеспечение – совокупность языковых средств, необходимых для обеспечения взаимодействия компонентов системы защиты информации между собой, с компонентами объекта обработки информации, с внешней средой.

Организационное построение

Организационное построение системы защиты информации в самом общем случае может быть представлено совокупностью следующих рубежей защиты:

- территории, занимаемой системой защиты информации;
- зданий, расположенных на территории;
- помещений внутри здания, в которых расположены ресурсы системы защиты информации и защищаемая информация;
- ресурсов, используемых для обработки и хранения информации и самой защищаемой информации;
- линий связи, проходящих в пределах одного и того же здания;
- линий (каналов) связи, проходящих между различными зданиями, расположенными на одной и той же охраняемой территории;
- линий (каналов) связи, соединяющих системы защиты информации с другими объектами вне охраняемой территории.

Компьютерные вирусы

Среди огромного разнообразия видов компьютерных программ существует одна их разновидность, заслуживающая особого упоминания. Глав-

ное отличие этих программ от всех остальных состоит в том, что они вредны, т.е. предназначены для нанесения ущерба пользователям ЭВМ. Это – компьютерные вирусы.

Компьютерным вирусом называется программа, обычно малая по размеру (от 200 до 5000 байт), которая самостоятельно запускается, многократно копирует свой код, присоединяя его к кодам других программ («размножается») и мешает корректной работе компьютера и/или разрушает хранимую на магнитных дисках информацию (программы и данные).

Существуют вирусы и менее «злокачественные», вызывающие, например, переустановку даты в компьютере, музыкальные (проигрывающие какую-либо мелодию), приводящие к появлению на экране дисплея какого-либо изображения или к искажениям в отображении дисплеем информации, «осыпанию букв» и т.д.

Создание компьютерных вирусов можно квалифицировать с юридической точки зрения как преступление.

Главную опасность представляют не сами по себе компьютерные вирусы, а пользователи компьютеров и компьютерных программ, не подготовленные к встрече с вирусами, ведущие себя неквалифицированно при встрече с симптомами заражения компьютера, легко впадающие в панику, что парализует нормальную работу.

Разновидности компьютерных вирусов

Рассмотрим подробнее основные особенности компьютерных вирусов, характеристики антивирусных программ и меры защиты программ и данных от компьютерных вирусов в наиболее распространенной операционной системе MS DOS.

По приближенным оценкам к 1997 г. существовало около 7000 различных вирусов. Подсчет их осложняется тем, что многие вирусы мало отличаются друг от друга, являются вариантами одного и того же вируса и, наоборот, один и тот же вирус может менять свой облик, кодировать сам себя. На самом деле основных принципиальных идей, лежащих в основе вирусов, не очень много (несколько десятков).

Среди всего разнообразия вирусов следует выделить следующие группы:

- **загрузочные (boot) вирусы** заражают программу начальной загрузки компьютера, хранящуюся в загрузочном секторе дискеты или винчестера, и запускающиеся при загрузке компьютера;

- **файловые вирусы** в простейшем случае заражают пополняемые файлы, но могут распространяться и через файлы документов (системы Word for Windows) и даже вообще не модифицировать файлы, а лишь иметь к ним какое-то отношение;

- **загрузочно-файловые вирусы** имеют признаки как загрузочных, так и файловых вирусов;

- **драйверные вирусы** заражают драйверы устройств компьютера или запускают себя путем включения в файл конфигурации дополнительной строки.

Из вирусов, функционирующих не на персональных компьютерах под операционной системой MS DOS, следует упомянуть **сетевые вирусы**, распространяющиеся в сетях, объединяющих многие десятки и сотни тысяч компьютеров.

Рассмотрим принципы функционирования загрузочных вирусов. На каждой дискете или винчестере имеются служебные сектора, используемые операционной системой для собственных нужд, в том числе сектор начальной загрузки. В нем помимо информации о дискете (число дорожек, число секторов и пр.) хранится небольшая программа начальной загрузки.

Простейшие загрузочные вирусы, резидентно находясь в памяти зараженного компьютера, обнаруживают в дисководе незараженную дискету и производят следующие действия:

- выделяют некоторую область дискеты и делают ее недоступной операционной системе (помечая, например, как сбойную – bad);
- замещают программу начальной загрузки в загрузочном секторе дискеты, копируя корректную программу загрузки, а также свой код, в выделенную область дискеты;
- организуют передачу управления так, чтобы вначале выполнялся бы код вируса и лишь затем – программа начальной загрузки.

Магнитные диски компьютеров винчестерского типа обычно бывают разбиты на несколько логических разделов. Программы начальной загрузки при этом имеются в MBR (Master Boot Record – главная загрузочная запись) и в загрузочном разделе винчестера, заражение которых может происходить аналогично заражению загрузочного сектора дискеты. Однако, программа начальной загрузки в MBR использует при переходе к программе загрузки загрузочного раздела винчестера, так называемую таблицу разбиения (Partition table), содержащую информацию о положении загрузочного раздела на диске. Вирус может исказить информацию Partition table и таким образом передать управление своему коду, записанному на диск, формально не меняя загрузочной программы.

Теперь рассмотрим принципы функционирования **файловых вирусов**. Файловый вирус не обязательно является резидентным, он может, например, внедриться в код исполняемого файла. При запуске зараженного файла вирус получает управление, выполняет некоторые действия и возвращает управление коду, в который он был внедрен. Действия, которые выполняет вирус, включают поиск подходящего для заражения файла, внедрение в него так, чтобы получить управление при запуске файла, производство некоторого эффекта, например, звукового или графического. Если файловый вирус резидентный, то он устанавливается в памяти и получает воз-

возможность заражать файлы и проявляться независимо от первоначального зараженного файла.

Заражая файл, вирус всегда изменит его код, но далеко не всегда производит другие изменения. В частности, может не изменяться начало файла и его длина (что раньше считалось признаком заражения). Например, вирусы могут искажать информацию о файлах, хранящуюся в служебной области магнитных дисков -таблице размещения файлов (FAT – file allocation table), – и делать таким образом невозможной любую работу с файлами. Так ведут себя вирусы семейства «Dir».

Загрузочно-файловые вирусы используют принципы как загрузочных, так и файловых вирусов, и являются наиболее опасными.

Антивирусные средства

К настоящему времени накоплен значительный опыт борьбы с компьютерными вирусами, разработаны антивирусные программы, известны меры защиты программ и данных. Происходит постоянное совершенствование, развитие антивирусных средств, которые в короткий срок с момента обнаружения вируса от недели до месяца оказываются способными справиться с вновь появляющимися вирусами.

Создание антивирусных программ начинается с обнаружения вируса по аномалиям в работе компьютера. После этого вирус тщательно изучается, выделяется его сигнатура – последовательность байтов, которая полностью характеризует программу вируса (наиболее важные и характерные участки кода), выясняется механизм работы вируса, способы заражения. Полученная информация позволяет разработать способы обнаружения вируса в памяти компьютера и на магнитных дисках, а также алгоритмы обезвреживания вируса (если возможно, удаления вирусного кода из файлов).

Известные ныне антивирусные программы можно разделить на несколько типов, перечисленных ниже.

- **Детекторы.** Их назначение – лишь обнаружить вирус. Детекторы вирусов могут сравнивать загрузочные сектора дискет с известными загрузочными секторами, формируемыми операционными системами различных версий, и таким образом обнаруживать загрузочные вирусы или выполнять сканирование файлов на магнитных дисках с целью обнаружения сигнатур известных вирусов. Такие программы в чистом виде в настоящее время редки.

- **Фаги.** Фаг – это программа, которая способна не только обнаружить, но и уничтожить вирус, т.е. удалить его код из зараженных программ и восстановить их работоспособность (если возможно). Известнейшим в России фагом является Aidstest. К январю 1997 года эта программа была способна обнаружить и обезвредить около 1600 вирусов. Еженедельно по-

являются новые версии этой программы, рассчитанные на обезвреживание десятков новых вирусов.

Очень мощным и эффективным антивирусным средством является фаг Doctor Web. Детектор этого фага не просто сканирует файлы в поисках одной из известных вирусных сигнатур. Doctor Web реализует эвристический метод поиска вирусов, может находить и обезвреживать, так называемые, полиморфные вирусы (не имеющие определенной сигнатуры), проверять файлы, находящиеся в архивах. Для нахождения вирусов Doctor Web использует программную эмуляцию процессора, т.е. он моделирует выполнение остальных файлов с помощью программной модели микропроцессора 1-8086 и тем самым создает среду для проявления вирусов и их размножения. Таким образом, программа Doctor Web может бороться не только с полиморфными вирусами, но и с вирусами, которые только еще могут появиться в перспективе. Специалисты рекомендуют использовать Aidstest и Doctor Web в комплексе.

- **Ревизоры.** Программа-ревизор контролирует возможные пути распространения программ-вирусов и заражения компьютеров. Программы-ревизоры относятся к самым надежным средствам защиты от вирусов и должны входить в арсенал каждого пользователя. Ревизоры являются единственным средством, позволяющим следить за целостностью и изменениями файлов и системных областей магнитных дисков. Наиболее известна в России программа-ревизор ADinf.

- **Сторожа.** Сторож – это резидентная программа, постоянно находящаяся в памяти компьютера, контролирующая операции компьютера, связанные с изменением информации на магнитных дисках, и предупреждающая пользователя о них. В состав операционной системы MS DOS, начиная с версии 6.0, входит сторож VSAFE. Однако, из-за того, что обычные программы выполняют операции, похожие на те, что делают вирусы, пользователи обычно не используют сторожа, так как постоянные предупреждения мешают работе.

- **Вакцины.** Так называются антивирусные программы, ведущие себя подобно вирусам, но не наносящие вреда. Вакцины предохраняют файлы от изменения и способны не только обнаружить факт заражения, но и в некоторых случаях «вылечить» пораженные вирусами файлы. В настоящее время антивирусные программы-вакцины широко не применяют, так как в прошлые годы некоторыми некорректно работающими вакцинами был нанесен ущерб многим пользователям.

Помимо программных средств защиты от вирусов существуют и специальные дополнительные устройства, обеспечивающие надежную защиту определенных разделов винчестера. Примером такого рода устройств является плата Sheriff. Несмотря на кажущееся обилие программных антивирусных средств, даже все вместе они не обеспечивают полной защиты про-

грамм и данных, не дают 100 %-ной гарантии от воздействия вирусных программ. Только комплексные профилактические меры защиты обеспечивают надежную защиту от возможной потери информации. В комплекс таких мер входит:

- регулярное архивирование информации (создание резервных копий важных файлов и системных областей винчестера);
- избегание использования случайно полученных программ (старайтесь пользоваться только законными путями получения программ);
- входной контроль нового программного обеспечения, поступивших дискет;
- сегментация жесткого диска, т.е. разбиение его на логические разделы с разграничением доступа к ним;
- систематическое использование программ-ревизоров для контроля целостности информации;
- при поиске вирусов (который должен происходить регулярно!) старайтесь использовать заведомо чистую операционную систему, загруженную с дискеты. Защищайте дискеты от записи, если есть хоть малая вероятность заражения.

При неаккуратной работе с антивирусными программами можно не только переносить с ними вирусы, но и вместо лечения файлов безнадежно их испортить. Полезно иметь хотя бы общее представление о том, что могут и чего не могут компьютерные вирусы, об их жизненном цикле, о важнейших методах защиты.

6. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

6.1. Понятие алгоритма. Свойства алгоритма

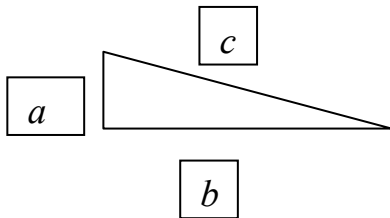
Алгоритм – это точное описание последовательности действий, которые необходимо совершить, чтобы перевести имеющиеся данные в нужный результат.

Существует несколько способов описания алгоритмов:

- словесная форма;
- графическое описание (блок-схема);
- на алгоритмическом языке.

Словесное описание:

пусть требуется определить площадь треугольника, заданного тремя сторонами: a , b , c .



Начало;

ввести данные в ячейки: a , b , c ;

присвоить $p = \frac{(a + b + c)}{2}$;

присвоить $S = \sqrt{(p(p - a)(p - b)(p - c))}$;

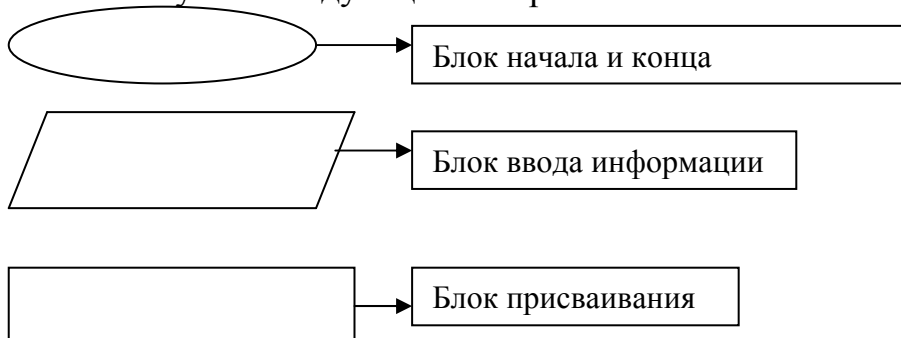
печатать S ;

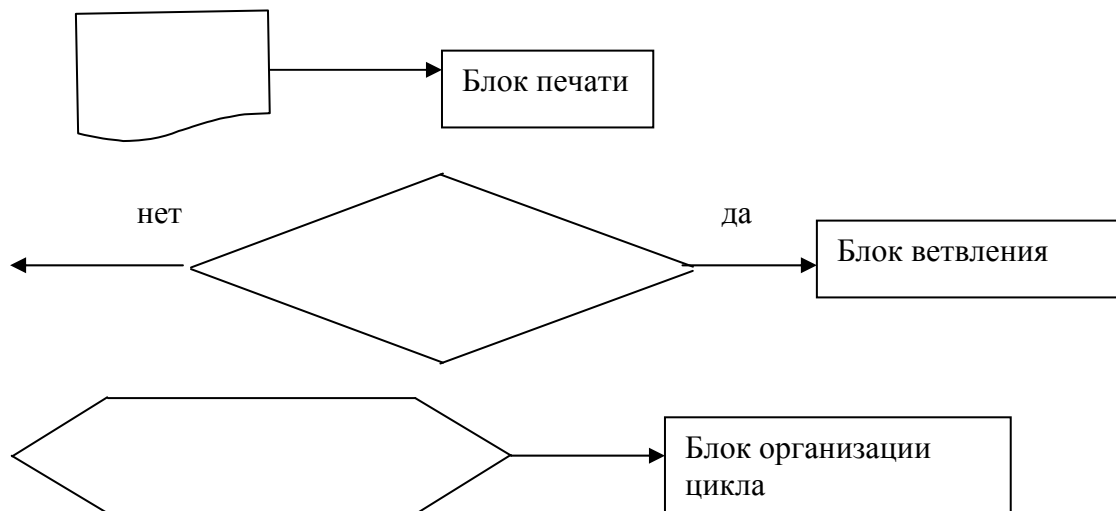
конец.

Операция присваивания – запись информации в ячейку. Ячейка может быть названа либо одним символом, либо набором символов. Записать информацию в ячейку можно либо вводом с клавиатуры, либо операцией вычисления, либо путем копирования из других ячеек.

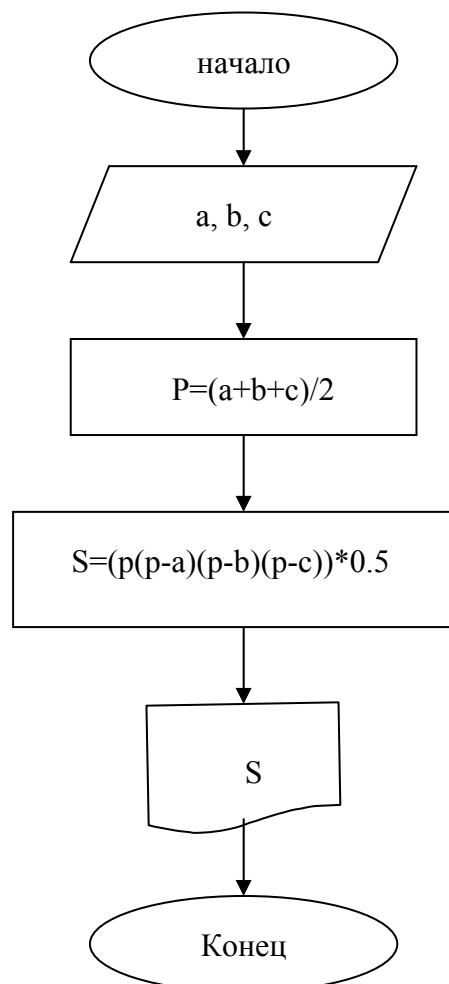
Графическое описание

Используется следующее начертание блоков:





Пример:



Свойства алгоритмов и этапы решения задач

Различают следующие свойства алгоритмов:

Массовость – это свойство говорит о том что, алгоритм должен быть применен к решению целого ряда задач этого рода.

Результативность – это свойство означает, что в результате выполнения алгоритма должны получить результат в виде числа, сообщения, графика, изображения, звука и так далее.

Дискретность – алгоритм состоит из определенного набора действий.

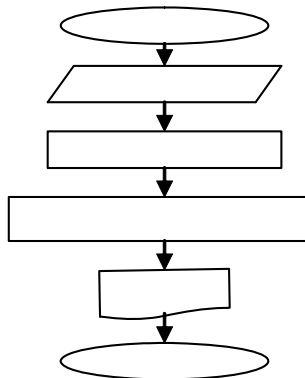
Конечность – это говорит о том, что после выполнения алгоритма и получения результата система должна выйти на останов.

Этапы решения задач на ПК:

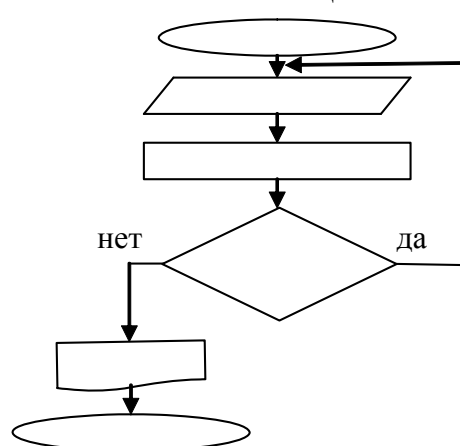
1. Постановка задачи (определение данных, которые имеются и определение результатов),
2. Выбор математической модели (подбор математических методов и способов, с помощью которых данные превращаются в результат),
3. Описание алгоритма (словесное, блок-схема и так далее),
4. Создание программного кода (подбирается алгоритмический язык, который наилучшим образом отображает описанный алгоритм),
5. Тестирование программного кода (отладка программы, выявление всех ошибок и их исправление),
6. Получение результатов.

Виды алгоритмов:

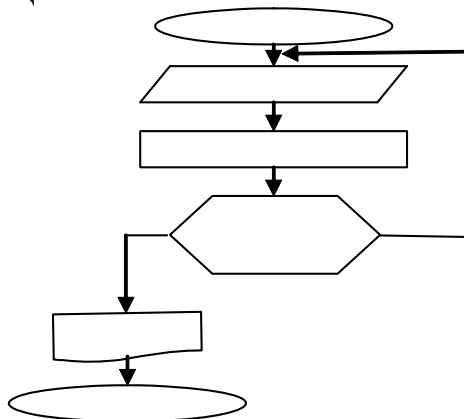
Линейные



Разветвляющиеся

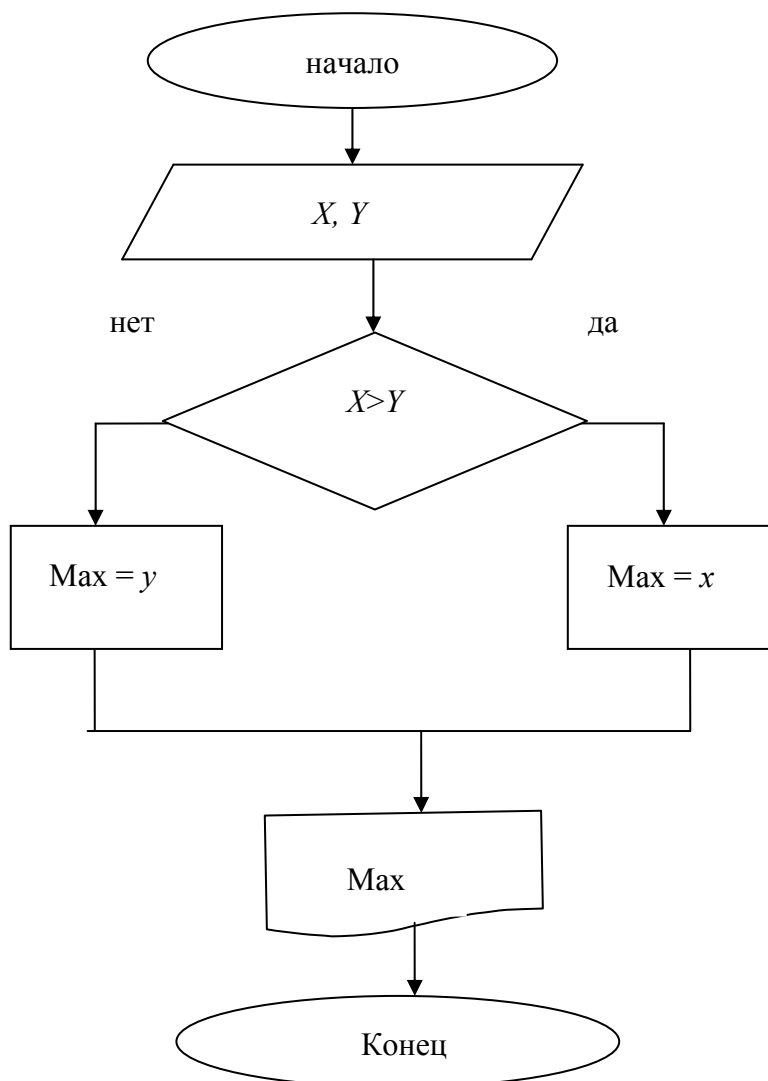


Циклические



Пример разветвляющейся структуры:

Даны два действительных различных числа x и y , выбрать среди них максимальное.

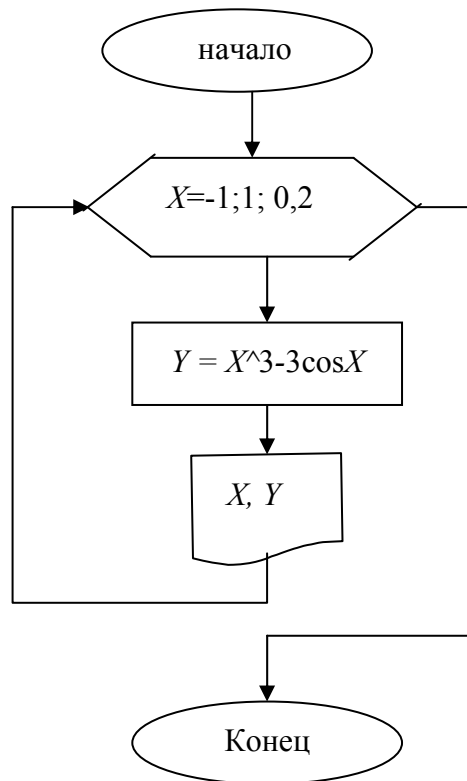


Пример циклической структуры:

Найти все значения функции $y=x^3-3\cos x$, если x изменяется в интервале от -1 до 1 , шаг изменения x равен $0,2$.

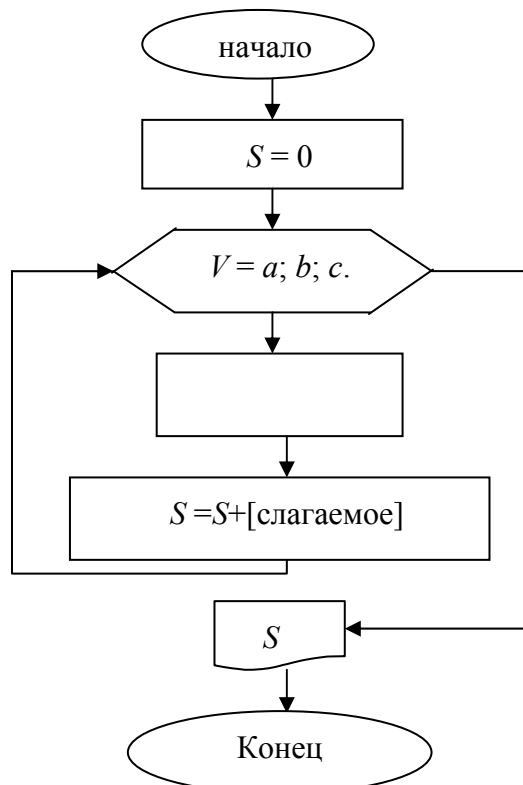
Переменная, которая отвечает за количество повторений, называется параметром цикла. Тело цикла – это повторяющаяся часть алгоритма. Условие задачи можно записать в следующем виде: $y=x^3-3\cos x$, $-1 \leq x \leq 1$, $\Delta x=0,2$, где $a=-1$, $b=1$. Тогда количество повторений тела цикла можно рассчитать по формуле

$$n = \frac{b - a}{\Delta x} + 1.$$



При описании алгоритмов часто используют стандартный набор определенных структур. Например, на практике часто приходится находить такие значение, как сумма, произведение, максимальное и минимальное значение. Пусть V – параметр цикла, и V изменяется в пределах: $a \leq V \leq b$, $\Delta V = c$.

Пример нахождения суммы.



Массивы информации

Массивы информации бывают одномерными, двумерными, n -мерными.

В начале пишется имя (идентификатор массива), а в скобках записываются индексы. Для одномерного массива – 1 индекс ($A(k)$), а для двумерного – 2, разделенные запятой ($A(n,m)$). Одномерный массив – вектор, двумерный массив – матрица.

6.2. Технология программирования на языке высокого уровня

Как и любой другой язык программирования имеет некоторую лексическую структуру. Алфавит языка программирования – это набор символов, разрешенных к использованию и воспринимаемые компилятором, с помощью которых могут быть образованы выражения и операторы данного языка. Для образования чисел используются символы: арабские цифра от 0 до 9, точка и буква E . Для образования имени переменных можно использовать буквы строчные и прописные латинского алфавита, арабские цифры от 0 до 9, точка, знак подчеркивания и следующие символы @, %, &, \$, !, #. Для образования строки можно использовать все символы алфавита. Разделителями служат: пробел, двоеточие, точка с запятой, кроме того для образования конструкций можно использовать символы + - * / \ : ; ‘ “ . < > =

VBA различает 4 класса слов: имена, ключевые слова, числа, строки. Слова разделяются друг от друга разделителями или специальными символами.

Имя (идентификатор) – это набор символов не превышающий 255 и начинающийся с буквы. Имя должно быть отлично от ключевых слов. В конце имени может стоять символ типа. В написание имени строчные и прописные буквы не различаются.

Строка – это последовательность символов заключенная в кавычки.

Числа – это лексема, служащая для записи числовых значений. Различают числа целые и вещественные. Целые записываются обычно. Вещественные (действительные) в двух видах – с плавающей запятой и фиксированной запятой.

Ключевое слово – это специальная лексема, смысл которой зафиксирован правилами языка и по которым транслятор определяет основные языковые конструкции.

Типы данных

Данные в *VBA* характеризуются типом которые определяют формат представления данных в памяти компьютер, область возможных значений

и область допустимых операций. Типы данных делятся на простые (встроенные) и структурированные.

Byte – представляет числа целые без знака, представленные от 0 до 255.

Integer – целые числа со знаком от -32768 до 32767 .

Long – длинное целое от -2 млрд до 2 млрд.

Currency – десятичные числа с фиксированным числом знаков после запятой.

Single – число с плавающей запятой обычной точности от $-3.4E38$ до $3.4E38$.

Double – числа с плавающей запятой двойной точности от $-1.7E308$ до $1.7E308$.

Decimal – любые числа, включающие 28 знаков.

Boolean – логический тип (*True*, *False*) Принимает одно из значений – либо истина, либо ложь.

Date – используется для хранения дат.

String – для хранения строковых значений.

Object – содержит ссылку на объект.

Variant – используется для хранения значений любых типов.

Операции

Операция – это действие над данными. Любая операция имеет результат. Операнд – это элемент данных участвующих в операции.

$A+B$; A и B операнды

Каждая операция имеет ограничение на типы данных которые можно использовать в этой операции. Для каждой операции определен тип результата. Если в операции участвуют операнды различных типов, то происходит их автоматическое преобразование в эквивалентное значение других типов. Не всякое преобразование возможно. Например, целое число можно представить как вещественное. $3 \rightarrow 3.0$, $3.5 \rightarrow$ нельзя.

Операции делятся на арифметические, логические, операции отношения и строковые.

Арифметические операции

Это операции где операнды числовые величины.

Пример:

$(a+v; a-v; a*v; a/v; a^v; a \setminus v; a \bmod v)$.

Логические операции:

$Not(a)$ – логическое нет;

$a \text{ AND } b$ – логическое и;

$a \text{ OR } b$ – логическое или;

$a \text{ XOR } b$ – исключаящее или;

$a \text{ EQV } b$ – эквивалентность;

$a \text{ IMP } b$ – импликация.

Операции отношения – операции которые возвращают булево значение в результат сравнения двух операндов. Операндами могут быть как переменные так и сложные выражения. Знаки: = <> > >= < <= *IS like*

IS – операция проверки вхождения. Имеет значение истина если элемент 1 входит в элемент 2.

Like – сравнивает строковую переменную с любым набором символов.

Выражение – это последовательность операций. Если выражение состоит больше чем из одной операции, то они выполняются в порядке приоритетности. Для того чтобы изменить приоритетность используют круглые скобки. Можно использовать две операции + и &:

+ – для объединения только строковых величин;

& – для любых величин которые преобразуются к строковым.

Объявление переменных

Переменные – это элемент данных в программе которым присвоено имя. Значение переменной может задаваться и изменяться в программе, но в соответствии с типом данной переменной. Переменная объявляется следующим образом: задается её имя и тип.

Объявление начинается с ключевого слова *Dim* <список переменных> *AS* <тип>.

Такое объявление называется явным.

Можно использовать и неявные объявления, например:

% – Integer

\$ – String

@ – Currency

– Double

& – Long

! – Single

Константы

В отличие от переменной константа – это элемент данных не меняющий своего значения. Синтаксис объявления констант: *Const* имя константы [*AS*<тип данных>]=выражение.

Выражение – это любое значение или формула, возвращающее значение используемое в качестве константы (*const P_i = 3.14*).

Рекомендуется давать константе осмысленные имена и описывать их в самом начале модуля (*Const P_i = π*).

Процедуры

Под процедурами понимают последовательность объявлений и инструкций, объединенных для выполнения:

SUB<имя>()

<инструкции>

End Sub

Операторы

Любая программа представляет собой последовательность операторов. В одной строке может быть несколько операторов. В этом случае они должны определяться друг от друга двоеточием. Операторы присваивания служат для изменения значений переменной. В *VBA* существует 4 типа операторов присваивания.

LET – прямое присваивание.

LSET – левое присваивание.

RSET – правое присваивание.

SET – объективное присваивание.

LSET – значение выражения присваивается переменной стоящей слева от знака =.

При записи арифметических выражений в операторе присваивания используются следующие правила:

1. выражение записывается в одну строку,
2. знак умножения ставится обязательно,
3. можно использовать только круглые скобки,
4. аргумент функции обязательно записывается в круглых скобках.

Можно использовать следующие встроенные функции: одним из классов часто употребляемых функций является функция ввода-вывода, обеспечивающая диалог пользователя с компьютером. *MsgBox* – функция вывода.

MsgBox <сообщение>

Для ввода информации используется функция *Input Box*.

Input Box <сообщение>[заголовок][значение по умолчанию][позиция x][позиция y]

Значение по умолчанию – это значение, которое автоматически появится в поле ввода.

Asc – строковое выражение возвращает код одного символа.

Chr – код символа.

Str – число переводит в строку.

Round () – округляет дробные числа до нужного количества знаков.

Is Numeric (выражение) – показывает является ли результат числовым значением.

Val – строка. Переводит строковое выражение в числовое.

Математические функции:

abs(x) – |x|;

atn(x) – arctg x;

cos(x) – cos x;

sin(x) – sin x;

tan(x) – tg x;

$\log(x) - \ln x$;
 $\exp(x) - e^x$;
 $\text{sqr}(x) - \sqrt{x}$;
 $\text{rnd}(x)$ – случайное число;
 $\text{sgn}(x) = 0, \begin{cases} 1, \text{ если } x > 0 \\ \text{если } x = 0 \\ -1, \text{ если } x < 0 \end{cases}$

$$\log_a b = \frac{\ln a}{\ln b};$$

$$\arcsin x = \text{arctg} \frac{x}{\sqrt{1-x^2}};$$

$$\arccos x = \text{arctg} \frac{\sqrt{1-x^2}}{x};$$

$$\text{arctg} x = \text{arctg} \frac{1}{x}.$$

Условные операторы

В VBA имеются различные управляющие конструкции, позволяющие изменять порядок выполнения операторов программы.

IF ... THEN ELSE

Имеются следующие два вида синтаксиса:

1) IF<условие> THEN<блок операторов>[ELSE<блок операторов>]

2) IF<условие1> THEN

<блок операторов>

[ELSE IF<условиен>THEN<блок операторовn>]

[ELSE блок операторов]

End IF

В качестве условия можно использовать логическое выражение.

Sub Pr 1()

Dim x As Integer, DAS Integer, SAS String

Input Box (“введите x=”, “ввод”)

x=Val(S)

If x<=10 Then

D=250

Else

D=50

End If

Msg Box D

End Sub

В случае с простым разветвлением удобно использовать функцию ИФ (выражение, значение если да, если нет)

ИФ (x<=10, 250, 50)

Select Case.

В принципе любое разветвление можно организовать пи помощи оператора ИФ, но иногда для разветвления удобнее использовать другой оператор Select Case.

Select Case выражение

[Case<список значений 1>]

[блок операторов]

[Case<список значений 2>]

[блок операторов 2]

.....

Case Else

[блок операторов]

End Select

Проверяемое выражение вычисляется в начале работы оператора Select Case. Элементы списка значений могут иметь одну из следующих форм:

значение. В этом случае проверяется совпадение значения выражения со значениями в списке значение 1 То значение 2. Проверяется находится ли значение выражения в указанном диапазоне.

IS – логическая операция значения. Проверяется входит ли подмножество в множество.

Помимо условных операторов в VBA существует оператор безусловного перехода, который также меняет порядок выполнения операторов в программе.

Go To – имя метки.

Применять оператор безусловного перехода не желательно.

Операторы цикла

Циклы позволяют многократно выполнять отдельный оператор или группы операторов. Операторы цикла VBA делятся на 3 группы:

- 1) циклы со счетчиком For Next;
- 2) циклы с условием Do Loop;
- 3) циклы по структуре данных For Each Next.

Оператор For Next

Данный оператор позволяет выполнять группы операторов, которые называются телом цикла заранее определенное количество раз.

При этом параметр цикла последовательно принимает ряд значений определенный пользователем в заголовке цикла.


```

Структура:
For i=a to b step n
Тело цикла(группа операторов)
Next i

```

i – параметр цикла
a – начальное значение параметра цикла
b – конечное значение параметра цикла
n – шаг параметра цикла

В заголовок оператора в первом шаге цикла параметру цикла присваивается начальное значение. Затем выполняется группа операторов, так называемое тело цикла, начиная с оператора который следует за оператором For. После выполнения последнего оператора тела цикла оператор Next *i* передает управление оператору For, который увеличивает значение параметра цикла на величину шага параметра цикла. Затем выполняется тело цикла и так далее. В операторе For происходят в каждом шаге сравнения значения параметра цикла *i* с конечным значением параметра цикла *b*. При $i \leq b$ циклический процесс продолжается. Как только значение *i* станет больше *b*, управление передается оператору следующему за оператором Next.

Прервать данный процесс можно с помощью оператора Exit For, предназначенный для выхода из цикла не дожидаясь его завершения и передачи управления оператору, следующему за оператором Next.

$$S = \sum_{i=1}^{10} 2i/i!$$

```

Sub Pr 2()
Dim i, P As Integer
Dim S, a As Single
P=1
S=0
For i=1 to 10
P=P*i
a=a^i/p
S=S+a
Next i
MsgBox "сумма=" & S
End Sub

```

Оператор цикла Do...Loop

Если заранее неизвестно сколько раз должно выполняться тело цикла, но будет известно условие, при котором цикл будет продолжаться или завершит свою работу, в этом случае используют оператор цикла с условием.

Существует два вида операторов циклов с условием:

1. с предусловием,
2. с постусловием.

Поэтому оператор Do имеет две конструкции:

– с предусловием:

```
Do {While/Until<условие>}
```

```
<блок операторов 1>
```

```
[Exit Do]
```

```
<блок операторов 2>
```

```
Loop
```

– с постусловием:

```
Do
```

```
<блок операторов 1>
```

```
[Exit Do]
```

```
<блок операторов 2>
```

```
Loop [While/Until<условие>]
```

Различие операторов в том, что в первом случае условие проверяется до выполнения блока операторов, а во втором – после.

Блок операторов выполняется пока условие истинно. Если используется функция Until, блок операторов выполняется пока условие ложно.

$$S = \sum_{i=1}^{\infty} \frac{1}{i^2 + 1}, a > E$$

```
Sub Pr 3()
```

```
Dim i As Integer
```

```
Dim S, E, a As Single, V As String
```

```
V=Input Box (“введите E=”, “ввод”, “0.001”)
```

```
E=Val (V)
```

```
S=0
```

```
a=1
```

```
i=1
```

```
Do while a>E
```

```
a=1/(i^2 + 1)
```

```
S=S+a
```

```
i=i+1
Loop
Msg Box “сумма=” & S
End Sub
```

Дано n, найти сумму цифр его образующих.

```
Sub Pr 4()
Dim n As Long
Dim S, K, C As Integer
Dim V As String
V=Input Box (“введите n =”, “ВВОД”)
n=Val (V)
S=0
K=0
While n <> 0
C= n mod 10
S=S + C
K=K + 1
n= n\10
Wend
Msg Box “сумма=” & S & “кол-во=” & K
End Sub
```

Оператор For Each ... Next

Оператор цикла по структуре. Он используется при обработке массивов или семейств однородных объектов.

```
For Each элемент In группа
<блок операторов 1>
[Exit For]
<блок операторов 2>
Next [элемент]
```

Элемент – это переменная, пробегающая все значения входящие в группу. Переменная должна быть типа “вариант”.

Массивы

Массивы незаменимы при работе с большими объемами данных. Массивы – это пронумерованная группа значений одного и того же типа. Действия с массивами похожи на действия с обычными переменными. Прежде чем искать массив его необходимо объявить. Указав сколько значений будет в этом массиве, какого типа значения будут храниться в этом массиве.

```
Dim A(10) As Integer – одномерный массив
```

Dim S (2, 3) As String – двумерный массив

Dim B (2 to 8) As Single – одномерный массив у которого индекс меняется от 2 до 8

Dim C (1 to 6, 2 to 4) As Data – двумерный массив у которого индексы меняются от 1 до 6 и от 2 до 4

Пример:

Найти количество элементов больше 7 в массиве A, состоящего из 10 элементов.

```
Sub Pr 1()
```

```
Dim k, i As Integer
```

```
Dim A(10) As Single
```

```
Worksheets (1). Select
```

```
Randomize
```

```
For i= 1 to 10
```

```
a(i)=Int (100*rnd())
```

```
Cells(1,i)=a(i)
```

```
If a(i)>7 Then k=k+1
```

```
Next i
```

```
Msg Box “количество=” & k
```

```
End Sub
```

Пример 2:

```
Sub Pr 2()
```

```
Dim Range 1 As Range
```

```
Set Range 1 Worksheets (1). Range (“A1”)
```

```
Range 1. Value=500
```

```
Disp Mes 1 _ Range 1.Value
```

```
End Sub
```

```
Sub Disp Mes 1 (Value 1)
```

```
Msg Box “значение=” & Value 1
```

```
End Sub
```

Подпрограммы и функции

В VBA допускается вызов одной программы из другой, что позволяет разделить программный код на логически обособленные фрагменты. Эта возможность имеет два плюса:

- если нужно несколько раз выполнить одну и ту же последовательность действий, удобнее задать её один раз, оформив в виде подпрограммы а затем по мере необходимости вызывать из других программ, при этом облегчается её редактирование, так как изменение вносится только один раз;

- код VBA разделенный на подпрограммы обретает четкую логическую структуру. Его легче разрабатывать, отлаживать и обслуживать.

Фрагментируя код, обязательно руководствуются правилами логики. Одна подпрограмма должна отвечать за выполнение одной операции.

Вызывая подпрограмму в неё можно передавать значения. Эта возможность еще более расширяет область применения подпрограмм и функций, что демонстрирует пример 2.

В вызывающей подпрограмме параметр, который передается в вызываемую подпрограмму, называется фактическим параметром. (Range 1. Value). А в вызываемой подпрограмме этот параметр (Value 1) называется формальным.

Передача параметров по ссылке и по значению

Передача аргументов из одной программы в другую осуществляется двумя способами:

- 1) по ссылке (by Reference);
- 2) по значению (by Value).

В первом способе передается сама переменная, а во втором только значение. Выбор способа передачи определяется ключевыми словами: by Ref (по ссылке) и by Val (по значению), которые указываются в круглых скобках в заголовке вызываемой подпрограммы. При отсутствии этих ключевых слов передача осуществляется по умолчанию, по ссылке. При передаче аргументов подпрограммы по значению сама переменная не передается и поэтому изменить её значение в подпрограмме нельзя.

Пример:

Sub Pr ()	}	вызывающая
Dim n As Integer		
n=5		
Pr 2 (By Val n)		
Msg Box "n=" & n		
End Sub	}	вызываемая подпрограмма
Sub Pr 2 (n)		
n=3		
End Sub		

Функции

Подпрограммы в VBA делятся на 2 группы: сами подпрограммы и функции, они имеют три основных отличия.

Функция начинается ключевым словом Function и заканчивается End Function.

```
Function  
{блок операторов  
End Function
```

Функцию можно вызвать из любой формулы ячейки Excel

Функция обязательно возвращает в вызываемую подпрограмму значение.

Пример:

$$y = \arcsin x + \arcsin ax / \arcsin (a^2 + x^2)$$
$$\arcsin x = \arctg x / \sqrt{1-x^2}$$

```
Function Asn (z) или Asn (By Val z As Single) As Single
```

```
Asn = Atn (z/SQR(1-z^2))
```

```
End Function
```

```
Dim a, x, z As Single
```

```
Sub Pr 3()
```

```
Dim S As String, y As Single
```

```
S=Input Box (“введите a=”, “ввод”)
```

```
a=Val (S)
```

```
S=Input Box (“введите x=”, “ввод”)
```

```
x=Val (S)
```

```
y=(Asn(x) + Asn(a*x))/Asn(a^2 + x^2)
```

```
Msg Box “y=” & y
```

```
End Sub
```

При использовании функции мы можем в скобках указать тип передаваемых параметров и тип результата. По умолчанию используется тип вариант.

Существуют следующие правила передачи аргументов в подпрограммы и функции: поскольку функция возвращает в вызывающую подпрограмму значение, её можно применять в выражениях, то есть там где используются обычные переменные. Помещая функцию в правой части оператора присваивания, аргументы должны заключаться в круглые скобки.

Для вызова подпрограмм и функций можно использовать ключевое слово: Call <имя подпрограммы/функции (аргументы)>

Если подпрограмма или функция вызываются сами по себе, то аргументы в скобки не заключаются.

Области видимости переменных

В VBA различают 3 уровня видимости переменных:

1) уровень процедуры;

2) уровень модуля;

3) уровень проекта.

Переменная уровня процедуры доступна на уровне процедуры. И значение этой переменной в другой процедуре не определено (если она описана внутри процедуры, то она является видимой).

Чтобы переменные были доступны в процедурах одного модуля, то их объявляют перед первой подпрограммой или процедурой данного модуля.

Ключевое слово объявляет переменные видимыми на уровне модуля. Если объявить переменные при помощи ключевого слова `Public`, то в этом случае они будут видимыми на уровне проекта.

Обработка символьной информации

Очень часто при работе с данными приходится проводить обработку символьной информации. Для этого используются стандартные функции или пользовательские, те которые созданы пользователем.

Из стандартных используются такие:

`Asc` (символ) – выдает код одного символа.

`Chr` (код) – по коду выдается символ.

`Val` (строка) – преобразует строковое значение в числовое если это возможно.

`Str` (число) – преобразует число в строку.

`Len` (строка) – выдает длину строки.

`Space` (число) – формирует строку содержащую указанное число пробелов.

`Left` (строка, n) – выделяет из строки слева n-символов.

`Right` (строка, n) – выделяет из строки справа n-символов.

`Mid` (строка, n, k) – выделяет из строки начиная с n-й позиции k символов.

`Replace` (строка, строка Поиск строка _ Замена) – заменяет в строке параметр строка поиск на строка замена.

`Ltrim` (строка) – удаляет левые пробелы в строке.

`Rtrim` (строка) – удаляет концевые пробелы в строке.

`Trim` (строка) – удаляет все пробелы.

`Lcase` (строка) – преобразует прописные в строчные.

`Ucase` (строка) – преобразует строчные в прописные.

Пример:

Ввести строку S, подсчитать кол-во букв a, входящих в эту строку.

```
Sub Pr 4 ()
```

```
Dim S, C As String
```

```
Dim n, k, i As Integer
```

```
S=Input Box (“vвод строку”, “vвод”)
```

```
n=Len (S)
```

```
For i=1 to n
```

```
  C=Mid (S, i, 1)
```

```
  If C= “a” Then k=k+1
```

```
Next i
```

```
Msg Box "кол-во=" & k
End Sub
```

Файлы данных

Различают файлы прямого и последовательного доступа. Последовательные файлы состоят из ряда строк текста. Каждая строка текста должна заканчиваться либо нажатием клавиши Ctrl/z либо нажатием клавиши Enter.

VBA содержит группу операторов предназначенную для записи или чтения целой строки одновременно. Поэтому такие файлы идеально подходят для хранения текстов. Создание последовательных файлов состоит из трёх этапов:

- 1) открытие файла;
- 2) запись в этот файл информации;
- 3) его закрытие.

Открытие осуществляется командой

```
Open filename For Output As # код,
```

где filename – это имя файла создаваемое на диске;
код – любое числовое значение от 1 до 255.

Код связывается с именем файла.

Запись данных в файле данных осуществляется командами:

```
Print # kod, C1, C2
```

```
Write # kod, Z1, Z2
```

По первой команде значение переменных записывается в файл данных без разделителей (в одну строку), вторая команда записывает файл данных разделяя их запятыми. Закрывается файл командой Close # kod.

Если повторно открыть существующий файл для записи, то все данные в нем исчезают и создается новый пустой файл с этим же именем. Поэтому открывать повторно файл для записи нельзя.

№ п/п	Наименование	Цена	Кол-во	Стоимость
1	Масло	55	100	
2	Сыр	78	70	
3	Кефир	4.5	120	
4	Сметана	20	97	
5	Ряженка	5	83	
6	Варенец	4	64	

```
Sub Pr1()
```

```
OPEN "basa" For OutPut As#1
```



```

Dim i As Integer
Dim t,S As String
Dim C,K,ST As Single
For i=1 to 4
S= InputBox(“введите название=”, “ввод”)
t= InputBox(“введите цену=”, “ввод”)
C=Val(t)
t= InputBox(“введите количество=”, “ввод”)
K=Val(t)
ST= C*K
WRITE # 1,i,S,C,K,ST
Next i
Close # 1
End Sub

```

Открывает файл данных для дополнения команда :OPEN filename For APPEND AS # kod. ПО этой команде указатель записи устанавливается после последней строки, имеющейся в формуле данных и происходит ввод новых данных.

```

Sub Pr2 ()
OPEN “baza” For APPEND As # 2
Dim i As Integer
Dim t, S As String
Dim C, K, ST As Single
For i= 1 to 2
S= InputBox(“введите название=”, “ввод”)
T= InputBox(“введите цену=”, “ввод”)
K=Val(t)
ST = C*K
WRITE # 2, i+4S, C, K, ST
Next i
Close # 2
End Sub

```

Открытие файла для чтения осуществляется командой:
Open filename For Input As # kod
Чтение данных из файла данных осуществляется двумя командами:
Line Input # kod, S (читает всю строку целиком без разделителей)
Input # kod, C1, C2 (помещает значение из файла данных в переменные C1, C2 и так далее)

```
Sub Pr3()  
OPEN "baza" For Input As # 1  
Dim n, i As Integer  
Dim t, S As String  
Dim C, K, ST As Single  
For i=1 to 6  
Input # 1, n, S, C, K, ST  
Msg Box n&S&C&K&ST  
Next i  
Close # 1  
End Sub
```

Используют функции для определения конца файла:

EOF (kod) – возвращает значение истина, когда достигнут конец файла

While Not EOF (1) пока не достигнут конец файла

LOF (kod) – выдает кол-во строк в файле данных (в байтах)

LOC (kod) – выдает текущее значение (число символов прочитанных на данный момент).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Абрамов, С.А. Задачи по программированию [Текст] / С.А. Абрамов [и др.]. – М.: Наука, 1988.
2. Аиламазян, А.К. Информатика и теория развития [Текст] / А.К. Аиламазян, Е.В. Стась. – М.: Наука, 1992.
3. Алексеев, В.Е. Вычислительная техника и программирование. Практикум по программированию [Текст] / В.Е. Алексеев [и др.]. – М.: Высшая школа, 1991.
4. Безрукое, Н.Н. Компьютерные вирусы [Текст] / Н.Н. Безрукое. – М.: Наука, 1991.
5. Брой, М. Информатика. Основополагающее введение [Текст]: в 3 ч. / М. Брой. – М.: Диалог-МИФИ, 1996.
6. Брябрин, В.М. Программное обеспечение персональных ЭВМ [Текст] / В.М. Брябрин. – М.: Наука, 1990.
7. Буч, Г. Объектно-ориентированное программирование с примерами применения [Текст] / Г. Буч. – Киев: Диалектика, 1992.
8. Вирт, Н. Алгоритмы и структуры данных [Текст] / Н. Вирт. – М.: Мир, 1989.
9. Гриценко, В.И. Информационная технология: вопросы развития и применения [Текст] / В.И. Гриценко, Б.Н. Панышин. – Киев: Наукова Думка, 1988.
10. Дайтибегов, Д.М. Основы алгоритмизации и алгоритмические языки [Текст] / Д.М. Дайтибегов, Е.А. Черноусов. – М.: Финансы и статистика, 1992.
11. Илюшечкин, В.М. Системное программное обеспечение [Текст] / В.М. Илюшечкин, А.Е. Костин. – М.: Высшая школа, 1991.
12. Как работать с Microsoft Office для Windows 10 [Текст]: техническое описание Microsoft Corporation. – 2015.
13. Каратыгин, С. Базы данных: простейшие средства обработки информации [Текст] / С. Каратыгин, А. Тихонов, В. Долголаптев. – М.: ФИА, 1995. – Т. 1, 2.
14. Каныгин, Ю.М. Основы теоретической информатики [Текст] / Ю.М. Каныгин, Г.И. Калитич. – Киев: Наукова Думка, 1990.
15. Кирмайер, М. Мультимедиа [Текст] / М. Кирмайер. – СПб.: ВHV-Санкт-Петербург, 1994.
16. Компьютерная вирусология [Текст]. – Киев: Укр. сов. энцикл., 1991.
17. Матросов, В.Л. Теория алгоритмов [Текст] / В.Л. Матросов. – М.: Прометей, 1989.
18. Першиков, В.И. Толковый словарь по информатике [Текст] / В.И. Першиков, В.М. Савинков. – М.: Финансы и статистика, 1991.
19. Свириденко, С.С. Современные информационные технологии [Текст] / С.С. Свириденко. – М.: Радио и связь, 1989.
20. Смирнов, Н.Н. Программные средства персональных ЭВМ [Текст] / Н.Н. Смирнов. – Л.: Машиностроение, 1990.
21. Успенский, В.А. Теория алгоритмов: основные открытия и приложения [Текст] / В.А. Успенский, А.Л. Семенов. – М.: Наука, 1987.
22. Хартли, Р. Передача информации [Текст] / Р. Хартли // Теория информации и ее приложения: сб. – М.: ИЛ, 1959.

ГЛОССАРИЙ

Операционная система – набор специальных программ для управления работой компьютера (всеми его ресурсами: аппаратными и программными).

Программа – упорядоченная последовательность команд для решения задачи.

Задача в терминах ОС – программа, загруженная с внешнего носителя в оперативную память.

Файл – именованный набор данных, находящихся на внешнем носителе.

Компьютер – вычислительная машина – физическая система (устройство), предназначенная для механизации и автоматизации процесса алгоритмической обработки информации.

Информация –

1) (с философской точки зрения) отражение реального мира с помощью сведений.

2) (с механической точки зрения) сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состояниях, которые уменьшают о них степень неопределенности, неполноту знаний.

Сведения (сообщения) – это форма представления информации в виде речи, текста, видео, чисел, формул, таблиц...

Данные – зарегистрированные сигналы.

Данные несут в себе информацию о событиях, происшедших в материальном мире, поскольку они являются регистрацией сигналов, возникших в результате этих событий.

Данные можно сохранить, затем применить к ним методы обработки и получить необходимую информацию.

Информация – используемые данные, которые уменьшают степень неопределенности об объекте и времени.

Информатика – область человеческой деятельности, связанная с процессами образования информации с помощью компьютеров и их взаимодействием со средой компьютеров.

Информационные технологии –

1) процесс, использующий совокупность научных методов и технических приемов производства информационных продуктов и услуг с применением всего многообразия технологий и связи.

2) процесс, использующий новые средства и методы создания, обработки, передачи, отображения и хранения информации.

Телекоммуникация – дистанционная передача данных на базе компьютерных сетей и современных технических средств.

Мультимедиа – объединение нескольких средств представления информации в одной системе.

Информационная система – взаимосвязанная совокупность средств, методов, и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Структура информатики

Информатика

Технические средства	Алгоритмические средства	Программные средства

Технические средства – составляют аппаратную конфигурацию вычислительных систем, включая средства сбора, обработки.

Алгоритмические средства – разработка методов и технологий обработки информации, проектирования информационных систем.

Программные средства – совокупность программ обработки данных и необходимых для их эксплуатации документов.

Ярлык – ссылка на объект (файл, папка), для быстрого доступа к нему.

Информатику можно рассматривать и так:

1) отрасль народного хозяйства:

- производство компьютерной техники;
- производство программных продуктов;
- разработка современной технологии переработки информации.

2) фундаментальная наука:

- методология создания информационного обеспечения;
- теория информационных систем и технологий.

Цель фундаментальных исследований в информатике – получение обобщенных знаний, выявление общих закономерностей в их построениях.

3) прикладная информатика:

- изучение закономерностей в информационных процессах;
- создание информационных моделей;
- разработка информационных систем и технологий.

Информатизация общества.

В истории развития цивилизаций произошло несколько информационных революций:

- 1) изобретение письменности;
- 2) изобретение книгопечатания (сер. 16 века);
- 3) изобретение электричества (конец 19 века);
- 4) изобретение микропроцессорной техники (70-е-80-е г.г. 20 века);
- 5) информационная индустрия (производство технических средств, технологии для знаний).

Состав вычислительной системы:

- 1) аппаратная;
- 2) программная.

Назначение вычислительной системы – принимать, преобразовывать, хранить и выдавать информацию, следуя командам человека. Для работы системы нужны следующие блоки:

- 1) для хранения информации – запоминающие устройства;
- 2) для обработки – преобразующие устройства;
- 3) для ввода/вывода – устройства ввода/вывода.

Запоминающие устройства.

- ПЗУ (постоянные)
- ОЗУ (оперативные)
- ВЗУ (внешние)

ПЗУ хранятся в специальных микросхемах; программные данные записываются в микросхемы в процессе заготовки.

ПЗУ содержит служебную информацию.

ОЗУ является хранилищем обрабатываемой информации, (хранятся программы, данные).

ВЗУ – внешние устройства для долговременного хранения.

Процессор – основная микросхема компьютера, которая выполняет программу.

Микропроцессор состоит:

1) устройство управления (УУ) – формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы); формирует адреса ячеек памяти используемых выполняемой операцией; управляет работой компьютера.

2) арифметико-логическое устройство (АЛУ) – предназначено для выполнения арифметических и логических операций с числовыми и символическими данными.

3) микропроцессорная память (МП) – служит для кратковременного хранения информации, используемых в вычислениях.

4) интерфейсная система.

Генератор тактовых импульсов – генерирует последовательность электрических импульсов; частота этих импульсов определяет тактовую частоту компьютера.

Такт – промежуток между соседними импульсами.

Устройства ввода/вывода.

Внешние периферийные устройства.

- 1) ВЗУ;
- 2) диалоговые средства пользователя:
 - видеомонитор;
 - устройство речевого ввода/вывода;

3) устройства ввода информации:

- клавиатура;
- графические планшеты (дигитайзеры);
- сканеры;
- барабанные;
- штрих-сканеры;
- цифровые фотокамеры;
- манипуляторы;
- мышь;
- джойстик;
- световое перо;

4) устройства вывода информации:

- принтеры;
- матричные;
- лазерные;
- струйные светодиодные;
- графопостроители (плоттеры);
- устройства связи и телекоммуникации;
- модем;
- линии связи;
- телефонные;
- проводные;
- оптоволоконные;
- кабельные;
- радиочастотные;
- спутниковые.

Системная шина.

Это основная интерфейсная система компьютера, обеспечивающая сопряжение связь всех его устройств между собой. Все блоки через соответствующие разъемы подключаются к шине однообразно: через контроллеры и адаптеры.

Базовая конфигурация компьютера:

- 1) системный блок;
- 2) монитор;
- 3) клавиатура;
- 4) мышь.

Системный блок представляет собой узел, в котором установлены наиболее важные компоненты.

Устройства внутри системного блока – **внутренние**, остальные **внешние**.

Состоит из:

- > системный модуль;

> системная шина.

«Материнская плата» – основная плата ПК.

На ней расположены:

> процессор;

> микропроцессорный комплект (чипсет – набор микросхем);

> шины – набор проводников, по которым происходит обмен сигналами;

> оперативная память;

> ПЗУ;

> разъемы (слоты).

Современный компьютер – это прибор. Его принцип действия – электронный. Назначение – автоматизация операции с данными.

Гибкость автоматизации основана на том, что операция с данными производится по заранее подготовленной программе.

Основой системы является двоичная 01. Физические основы могут быть разные -механические, световые, электронные.

В восьми разрядах уместится 256 целых чисел для того, чтобы дать уникальное восьми битовое обозначение каждой заглавной или строчной букве 2-х алфавитов, всем цифрам, знакам.

8 разрядов → 1 байт.

Существует таблица кодирования ASCII.

Каждому символу соответствует определенный код, номер в таблице.

Принцип Неймана.

1) основными блоками компьютера является блок управления, арифметико-логические устройства, память и устройства ввода/вывода.

2) информация кодируется в двоичной форме и разделяется на единицы, называемые словами. Алгоритм представляется в форме последовательности управляющих форм.

3) эти управляющие слова называются командами. Их совокупность представляющих алгоритм называются программами.

4) программы и данные хранятся в одной и той же памяти. Разнотипные слова различаются по способу представления.

5) устройства управления и арифметико-логическое устройство объединены вместе и называют центральным процессором.

Они определяют действия, подлежащие выполнению путем считывания команд из ОЗУ. Обработкой информации, предписанной алгоритмом сводится к последовательному выполнению программ в порядке однозначно определяемых программ.

Программная конфигурация вычислительной системы.

Уровни ПО.

Прикладное ПО.

Служебное ПО.

Системное ПО.

Базовое ПО.

Базовый уровень – базовое ПО.

Оно отвечает за взаимодействие с базовыми аппаратными средствами. Находятся в ПЗУ.

Системный уровень – переходный.

Программы этого уровня организуют взаимодействие всех прочих систем с базовыми:

- драйверы;
- пользовательский интерфейс.

Совокупность программного обеспечения системного уровня образует ядро ОС. ОС опирается на базовое ПО. На системном уровне:

- взаимодействие с устройствами;
- взаимодействие с пользователем (интерфейс).

Файл – логически связанная совокупность данных или программ, для размещения которой во внешней памяти выделяется именованная область.

Имя файла состоит из трех частей Имя. тип.

Логические диски

Создается и управляется специальной программой.

Имя логического диска (D, E)

Служебный уровень

Основное назначение служебных программ состоит в автоматизации работ.

Утилит.

Классификация служебных средств:

- 1) диспетчер файлов;
- 2) средства сжатия;
- 3) средства диагностики;
- 4) средства коммуникации;
- 5) средства обеспечения компьютерной безопасности:
 - пассивная защита (резервные копии);
 - активная защита (антивирусные программы);
 - защита от несанкционированного доступа.

Прикладное программное обеспечение.

Представляет собой комплекс прикладных программ, с помощью которых на данном рабочем месте выполняются задания.

Классификация прикладных программ.

1. Текстовые редакторы.

Назначение текстовых процессоров – создание комплексных документов с фрагментами из других программ.

2. Графические редакторы:

- растровые;
- векторные;

- трехмерные.
3. Электронные таблицы.
 4. Системы управления базами данных СУБД.

Базы данных – огромные массивы данных, организованных в табличной структуре. **Основные функции СУБД:**

- 1) структура БД;
- 2) заполнение БД;
- 3) поиск и фильтрация данных по БД;
- 4) анализ и подведение итогов данных;
- 5) системы автоматизированного проектирования;
- 6) настольно-издательские системы – автоматизация полиграфии;
- 7) системы делопроизводства – автоматизация рабочего места руководителя;
- 8) бухгалтерские системы;
- 9) юридические;
- 10) браузеры.

Вопросы компьютерной безопасности. Понятие безопасность:

- Надежность работы компьютера.
- Сохранность ценных данных.
- Защита информации от внесения в нее изменений неуполномоченными лицами.
- Сохранение тайны переписки в электронной связи.

Компьютерный вирус – программный код, встроенный в другую программу или документ или в определенную область носителя данных и предназначенный для выполнения несанкционированных действий.

Основными типами компьютерных вирусов являются:

- программные;
- загрузочные;
- макровирусы;
- троянские.

Программные вирусы – это блоки программного кода, целенаправленно внедренные внутрь других прикладных программ.

Вирусный программный код может воспроизводить себя в теле других программ.

Этот процесс называется **размножением**.

Вирусная атака – это процесс в результате которого программное обеспечение уничтожается.

Программный вирус может вывести из строя и программное обеспечение. Тогда надо менять микросхему.

Загрузочные.

От программных вирусов отличаются методом распространения.

Поражают не файлы, а магнитные носители.

ФАТ-таблица находится в нулевом секторе.

Макровирусы – разновидность вирусов, которые поражают некоторые документы прикладных программ, имеющих средства выполнения макрокоманд.

Методы защиты

Три рубежа:

- 1) предотвращение поступления вирусов;
- 2) предотвращение вирусной атаки;
- 3) предотвращение разрушительных последствий.

Методы реализации защиты:

- программные;
- аппаратные;
- организационные.

Средства антивирусной защиты:

- резервное копирование;
- антивирусные программы.

Программные средства антивирусной защиты:

- 1) создание образа на внешних носителях;
- 2) регулярное сканирование;
- 3) контроль за изменением размеров и других атрибутов файлов;
- 4) контроль за обращениями к жесткому диску.

ЛОГИЧЕСКИЕ ФУНКЦИИ В EXCEL

Логические выражения:

- арифметические выражения;
- операции сравнения;
- логические операции;
- логические величины (константы).

О Г Л А В Л Е Н И Е

ВВЕДЕНИЕ.....	3
1. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ.....	4
2. СИСТЕМЫ СЧИСЛЕНИЯ И ОСНОВЫ ЛОГИКИ.....	8
2.1. Представление информации. Количество и единицы измерения информации....	8
2.2. Системы счисления, используемые в компьютере. Представление чисел в памяти ЭВМ.....	10
2.3. Алгебра логики. Основные логические операции. Основные законы преобразования алгебры логики. Логические основы ЭВМ.....	20
3. УСТРОЙСТВО КОМПЬЮТЕРА.....	26
3.1. Основные устройства компьютера.....	26
3.2. Программное обеспечение компьютера. Операционная система.....	32
4. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.....	38
4.1. Технология обработки текстовой информации.....	38
4.2. Технология обработки числовой информации.....	42
4.3. Технология хранения, поиска и сортировки информации.....	47
Понятие БД и СУБД.....	48
5. КОМПЬЮТЕРНЫЕ КОММУНИКАЦИИ.....	65
5.1. Компьютерные коммуникации. Интернет.....	65
5.2. Приемы защиты информации.....	71
6. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ.....	94
6.1. Понятие алгоритма. Свойства алгоритма.....	94
6.2. Технология программирования на языке высокого уровня.....	99
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	115
ГЛОССАРИЙ.....	116

Учебное издание

Глебова Татьяна Александровна
Чиркина Марина Александровна
Гвоздева Ирина Геннадьевна

ИНФОРМАТИКА

Учебное пособие

по направлению подготовки 38.03.03 «Управление персоналом»

В авторской редакции
Верстка Н.А. Сазонова

Подписано в печать 8.08.16. Формат 60×84/16.
Бумага офисная «Снегурочка». Печать на ризографе.
Усл.печ.л. 7,2. Уч.-изд.л. 7,75. Тираж 80 экз.
Заказ № 503.

Издательство ПГУАС.
440028, г. Пенза, ул. Германа Титова, 28.