

Тема 1 Введение. Информатика как наука. Информация

1.1 Что такое информатика

Информатика – это фундаментальная естественная наука, изучающая структуру и общие свойства информации, а также вопросы, связанные с процессами сбора, хранения, поиска, передачи, переработки, преобразования и использования информации в различных сферах человеческой деятельности с помощью средств вычислительной и организационной техники.

Информатика связана с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации (вычислительная техника, программное обеспечение), затрагивает организационные, коммерческие, административные и социальнополитические аспекты компьютеризации во всех областях жизни людей, изучает общие закономерности, свойственные информационным процессам, свойства информации, методы и средства ее обработки (сбор, хранение, преобразование, перемещение, выдача).

Под *информацией* (от лат. *informatio* – разъяснение, изложение) первоначально понимались сведения, передаваемые людьми устным, письменным или другим способом с помощью условных сигналов, технических средств и т.п. С середины XX века информация является общенаучным понятием, включающим в себя: обмен сведениями между людьми, человеком и автоматом, автоматом и автоматом; обмен сигналами в животном и растительном мире; передачу признаков от клетки к клетке, от организма к организму.

Информацию можно определить также как совокупность сведений, уменьшающих степень неопределенности знания о конкретных событиях, процессах, явлениях.

В зависимости от сферы использования информация может быть экономической, технической, генетической.

Под экономической информацией понимается информация, характеризующая производственные отношения в обществе.

К ней относятся сведения экономического характера о процессах производства, материальных ресурсах, процессах управления производством, финансовых процессах, которые циркулируют в экономической системе.

Экономической информатикой называется наука, изучающая методы автоматизированной обработки экономической информации с помощью средств вычислительной и организационной техники.

Термин «информатика» (фр. *informatique*) происходит от французских слов *information* (информация) и *automatique* (автоматизация) – «информационная автоматизация». Синонимы названия информатика: «Computer science» – «компьютерная наука», «Information science» – «информационная наука».

Предметом экономической информатики являются: технология и этапы разработки систем автоматизированной обработки экономической информации и обоснование целесообразности такой обработки, функциональный анализ предметной области, алгоритмическое представление задачи и программная ее реализация.

С современной точки зрения информатику можно рассматривать и как науку, и как прикладную дисциплину, и как отрасль народного хозяйства.

Информатика включает в себя

- научную составляющую – комплекс научных дисциплин, связанных с методами, средствами и процессами описания, получения, передачи и обработки информации в различных областях человеческой деятельности;
- прикладную составляющую – программный и технический инструментарий для создания и эксплуатации информационных систем (ИС), информационных технологий (ИТ); □
бизнес-сферу.

1.2 Информатика как наука

Информатика как *наука* нацелена на получение знаний об информации и ее свойствах, теоретическое обоснование основ построения ИС и ИТ, использует разнообразные методы научно-исследовательской работы, понятийный аппарат. Теоретическая информатика использует достижения фундаментальных наук: теории информации, теории алгоритмов, математической логики, теории вероятности и статистика и т.д.

Можно выделить ряд научных направлений, которые связаны с информатикой:

- Теоретическая информатика.
- Кибернетика.
- Программирование.
- Искусственный интеллект.
- Информационные системы.
- Вычислительная техника.
- Информатика в обществе.
- Информатика в природе.

Теоретическая информатика распадается на ряд самостоятельных дисциплин: математическая логика для анализа информационных процессов (теория алгоритмов, теория параллельных вычислений, теория автоматов, теория сетей Петри), вычислительная математика, теория кодирования информации, системный анализ, имитационное моделирование процессов, протекающих в реальных объектах, теория массового обслуживания, теория принятия решений, математическое программирование, исследование операций, теория коллективного поведения.

Кибернетика изучает общие законы и принципы управления в объектах различной природы, создала ряд теорий (автоматического управления, технической диагностики, распознавания образов и др.).

Теоретическое программирование обеспечивает развитие идей построения программ для компьютеров, процедур программирования, создание языков программирования, трансляторов, операционных систем, сетевых протоколов связи и др.

В начале 70-х гг. 20 века появились **искусственный интеллект** и **когнитивная психология**, исследования которых направлены на раскрытие закономерностей и механизмов принятия решений, распознавания образов, построение интеллектуальных систем, робототехники и др.

Информационные системы обеспечивают реализацию информационных процессов системы управления. Большое значение имеет анализ и прогнозирование информационных потоков, развитие теоретических основ организации обработки, хранения и использования информации.

Вычислительная техника развивается в направлении совершенствования элементной базы вычислительных машин (микроэлектроника), создания многомашинных и многопроцессорных комплексов (суперЭВМ) новой архитектуры построения.

Информатика в обществе связана с созданием информационной среды, обеспечивающей удовлетворение информационных потребностей общества, разработкой и реализацией концепции «информационное общество».

Информатика в природе связана с изучением информационных процессов, протекающих в биологических системах, и использованием накопленных знаний при организации и управлении природными системами и создании технических систем. Это:

- биокибернетика – анализ информационно управляющих процессов, протекающих в живых организмах, диагностика заболеваний и поиск путей их лечения, оценка биологической активности химических соединений, исследования моделей внутриклеточных процессов, лежащих в основе всего живого;

- бионика – наука о применении в технических устройствах и системах принципов организации, свойств, функций и структур живой природы;
- биогеоценология – наука, исследующая строение и функционирование биогеоценозов – сообществ живых и неживых объектов, создание информационных моделей поддержания и сохранения равновесия природных систем и поиска таких воздействий на них, которые стабилизируют разрушающее воздействие человеческой цивилизации на биомассу Земли.

1.3 Информационное общество

Концепция информационного общества сложилась в конце 20 века, она тесно связана с понятием постиндустриального общества, новой фазой в развитии всей нашей цивилизации. Отличительные черты информационного общества:

- главный продукт производства – информация и знания;
- изменение структуры общества: возрастание доли людей, занятых информационными технологиями, коммуникациями и производством информационных продуктов и услуг;
- сплошная информатизация общества на основе современных ИКТ (телефония, радио, телевидение, Интернет, электронные СМИ);
- глобализация информационного пространства;
- развитие электронной демократии, информационной экономики, электронного государства, электронного правительства, цифровых рынков и сетевой экономики;
- в управлении социальными и экономическими отношениями возрастание роли личности.

В РФ подготовлен проект программы «Информационное общество», которая включает следующие направления:

- создание электронного правительства,
- повышение качества жизни граждан (предоставление полной информации в удобном виде в приемлемое время),
- преодоление цифрового неравенства (развитие телекоммуникационной инфраструктуры),
- обеспечение безопасности в информационном обществе, □ сохранение культурного наследия (создание цифрового контента для музейных фондов, архивов),
- развитие рынка ИКТ (распространение системы технопарков в том числе).

Развитие электронного документооборота, ЭЦП, безопасность передачи данных, Электронная Россия, предоставление государственных и муниципальных услуг в цифровом виде, создание поисковика, способного искать тексты, изображения, аудио и видео и др. – вот основное содержание будущих проектов. Предполагается создание национальной информационно-коммуникационной платформы для распространения цифрового контента (аналог сервиса Apple iTunes). В «Информационное общество» вошли проекты по развитию электронных госуслуг, национальной программной и облачной платформ, платежной системы, создание единого реестра автодорог, туристического атласа страны, базы данных документов об образовании, федерального электронного кадастра недвижимости и других сервисов, доступных через ИНТЕРНЕТ. В результате через развитие ИКТ будет обеспечен технологический прорыв страны. К 2020 г. Россия должна будет войти в первую десятку международного рейтинга по индексу развития ИКТ, в двадцатку рейтингов по развитию электронного правительства и сетевого общества. Непосредственные расходы на реализацию госпрограммы в период с 2011 по 2020 гг. из федерального бюджета составят 88,03 млрд руб. Основные показатели этой программы к 2015 г.:

- предоставление основных федеральных услуг в электронном виде (около 200);
- увеличение удельного веса ИКТ в ВВП России (до 6%, сейчас около 4%),
- рост экспорта товаров, связанных с ИКТ до 6 млрд долл. в год, □ увеличение доли отечественных товаров и услуг в объеме внутреннего рынка ИКТ до 15% (сейчас менее 5%);
- проникновение широкополосного доступа в ИНТЕРНЕТ;
- рост количества компьютеров на домохозяйство;
- доля населения, способного принимать цифровой телевизионный сигнал;
- рост объема инвестиций;
- доля товаров и услуг, размещенных в электронном виде. Главная проблема – безопасность личной информации и персональных данных.

ОАО «Ростелеком» на период до 2020 г. обеспечит:

1. Развитие единого портала государственных и муниципальных услуг (функций).
2. Разработку механизмов, позволяющих использовать мобильные устройства для доступа к сервисам электронного правительства.
3. Развитие сервисов взаимодействия граждан с органами государственной власти при помощи электронной почты, созданной на

базе единого портала государственных и муниципальных услуг (функций).

4. Развитие центров телефонного обслуживания.

5. Формирование единого пространства доверия электронной цифровой подписи.

6. Развитие межведомственного электронного взаимодействия на основе системы взаимодействия.

7. Формирование и развитие инфраструктуры универсальной электронной карты (в части, относящейся к компетенции федеральных органов исполнительной власти).

8. Создание единой системы справочников и классификаторов, используемых в государственных и муниципальных информационных системах.

9. Создание единой системы учета записей актов гражданского состояния (электронный загс).

10. Развитие государственной автоматизированной системы «Управление».

11. Развитие мероприятия «Электронный регион».

12. Создание технологической инфраструктуры для осуществления электронных платежей за государственные услуги (на основе единого портала государственных и муниципальных услуг (функций)).

13. Создание системы контроля реализации поручений Правительственной комиссии по внедрению информационных технологий в деятельность государственных органов и органов местного самоуправления.

14. Реализация мероприятий по координации расходования бюджетных средств органов государственной власти на использование информационных технологий.

15. Создание национальной платформы для распределенной обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

16. Обеспечение развития информационных систем в сфере государственных закупок и торгов.

17. Развитие средств поиска информации по различным видам контента.

Прикладная информатика нацелена на практическое применение теоретических основ, методов и средств для построения ИС, реализацию ИТ сбора, передачи, хранения, обработки, распространения и использования информации для различных областей деятельности (экономика, медицина, юриспруденция, научная сфера и т.п.), решение прикладных задач системы управления, автоматизацию и информатизацию производственной и управленческой деятельности.

Основные разделы прикладной информатики:

- моделирование информационных процессов различных предметных областей;
- программные средства для реализации ИТ сбора, регистрации, передачи, хранения и обработки информации;
- архитектура информационно-вычислительных систем, компьютерных сетей и телекоммуникаций;
- методы управления информационными ресурсами;
- методология проектирования информационных систем (ИС); методы и средства алгоритмизации и программирования, системы и языки программирования;
- инструментарий для создания программных средств, CASE (Computer-Aided Software Engineering)-технологии.

1.5 Информатика как бизнес-сфера

Информатика как *бизнес-сфера* охватывает 4 сегмента рынка:

- программное обеспечение,
- аппаратное обеспечение,
- ИТ-услуги,
- телекоммуникации.

Это наиболее динамичные секторы экономики в мире и в России в том числе, это обусловлено возрастанием роли ИТ и ИС в различных сферах деятельности. Мировые расходы на ИТ в 2009 году составили \$1,5 трлн, с учетом сферы телекоммуникаций – почти \$3 трлн. Больше всего выросли расходы на аппаратное обеспечение: средства хранения данных, персональные компьютеры, серверы. Несколько снились темпы роста затрат на программное обеспечение и услуги, что связано с развитием облачных вычислений и технологий ИНТЕРНЕТ.

В бизнес-сферу информатики включены фирмы – производители средств вычислительной техники, коммуникаций и связи, программного обеспечения,

разработчики ИС и ИТ, различного рода консалтинговые и аналитические компании, фирмы, осуществляющие подготовку пользователей современных ИС и ИТ. Среди гигантов компании: Microsoft, Oracle, IBM, SAP.

1.6 Единицы измерения информации

На синтаксическом уровне изучения информации дается оценка объема данных и количества разнообразных кодовых комбинаций, зависящих от выбранной системы кодирования. Так, количество разнообразных кодов (N) для информационного сообщения длиной n разрядов, представленного с помощью системы кодирования с основанием m , равно:

$$N = m^n$$

Наиболее известны формулы **Р. Хартли** (1928), **Х. Найквиста** (1924), **Клода Шеннона** (1948), используемые для оценки количества информации. К. Шеннон использовал понятие *энтропии* – меры неопределенности состояния системы. Информация снимает неопределенность, уменьшает энтропию системы, поэтому количество информации в сообщении можно измерить через уменьшение энтропии системы.

Максимальное количество информации о системе, имеющей N разнообразных состояний¹ с помощью информационного сообщения длиной n разрядов и системы кодирования с основанием m , равно:

$$I = \log N = n \cdot \log m$$

Для оценки количества информации используются основания логарифма 2 и 10, соответственно, *бит*, *дит*. (см. табл. 1).

Табл. 1

Название	Символ	Степень
байт	B	10^0
килобайт	kB	10^3
мегабайт	MB	10^6
гигабайт	GB	10^9

терабайт	ТВ	10^{12}
петабайт	ПВ	10^{15}
эксабайт	ЕВ	10^{18}
зеттабайт	ЗВ	10^{21}
йоттабайт	УВ	10^{24}

Например, информационное сообщение о том, что из 16 возможных цветов выбран один определенный цвет, содержит:

$$I = \log_2 16 = 4 \text{ бита}$$

$$I = \log_{10} 16 \cong 1,204 \text{ дита}$$

Информационное сообщение о том, что мальчик Петров и девочка Иванова сидят вместе за одной партой, при условии, что в классе 12 мальчиков и 18 девочек, содержит:

$$I = \log_2 (12 + 18) = \log_2 216 = 7,75 \text{ бита}$$

$$I = \log_{10} (12 + 18) = \log_{10} 30 = 2,33 \text{ (дита)}$$

Характерным носителем информации является сообщение – все то, что подлежит передаче. Это может быть электрический сигнал или сигнал другого рода энергии, передаваемый по выбранной физической среде.

Рассмотрим виды информации:

1. Научная информация. Это логическая информация, адекватно отображающая объективные закономерности природы общества мышления. Научную информацию делят по областям получения или использования (техническая биологическая политическая и т.д.); по назначению: массовая и специальная; по видам носителя: на бумаге – документальная, на магнитной ленте, в памяти ЭВМ.

2. Техническая информация. Она используется и возникает при решении новых задач (конструирование, технологические процессы и т.д.).

3. Научно-техническая информация – объединение первых двух.

4. Технологическая информация: она циркулирует в сфере материально-технического производства.

5. Планово-экономическая информация содержит интегральные сведения о ходе производств, об экономических показателях.

Верхним уровнем информации являются **знания**. Знания возникают как итог теоретической и практической деятельности. Информация в виде знаний

¹ Когда несколько объектов рассматриваются как один, количество возможных состояний перемножается, а количество информации – складывается.

отличается высокой степенью структурированности. По мере развития общества информация как совокупность научно-технических знаний превращается в базу информационного обслуживания общества во всех видах его деятельности.

Наряду с энергией, полезными ископаемыми и т.д., информация является ресурсом общества. По мере продвижения технологического прогресса информационный ресурс становится наиболее важным национальным ресурсом. Эффективность промышленной эксплуатации информационных ресурсов определяет экономическую мощь страны.

Технологическую базу формирования и эксплуатации информационных ресурсов создает индустрия ЭВМ. Однако перекачивание трудовых ресурсов из сфер материального производства в информационную сферу ведет к эре «информационного кризиса». В настоящее время:

- Удвоение объема информации, накопленных научных знаний – 2-3 года.
- Материальные затраты на хранение, передачу и переработку информации превышают расходы на энергетику.
- Уровень радиоизлучений на отдельных участках Земли приближается к уровню радиоизлучения Солнца.

1.07 Связь понятий «информация», «данные», «знание» В информатике **данные** (*data*) – это представление фактов и идей в формализованном виде на материальном носителе, пригодном для передачи и обработки в некотором информационном процессе. Они выступают как источник информации, извлекаемой из данных путем обработки. При этом результаты обработки фиксируются в виде новых данных.

Термин **знание** имеет многообразную трактовку: форма существования и систематизации результатов познавательной деятельности человека; субъективный образ реальности, в форме понятий и представлений, информация, позволяющая решать поставленную задачу, достичь поставленной цели. В информатике знание является обязательным атрибутом интеллектуальных систем, представляется в виде базы знаний и правил вывода для принятия решений.

Знания могут быть представлены в форме данных (в виде текста на некотором формальном языке, в виде сети, задающей связи разного рода между элементами знаний), но знания более высокая степень организации данных, они обладают способностью управлять информационными процессами (вычислениями). Знания делятся на отдельные фрагменты – *фреймы*, которые могут быть связаны друг с другом родовидовыми отношениями, могут быть и узлами семантических сетей. Знания должны иметь прагматическую ценность.

Процесс формирования знания схематично выглядит так: данные – информация – знание – понимание – мудрость (рис. 5).

Данные – это неупорядоченные символы, рассматриваемые безотносительно к какому-либо контексту. Данные рассматриваются как сырье, факты.

Информация – выделенная, упорядоченная часть базы данных, обработанная для использования. Информация рассматривается как результат обработки данных в соответствии с алгоритмом обработки.

Знание – это формулировка существующих тенденций или существенных связей между явлениями, представленными в информации, которая может служить руководством для принятия конкретных решений и действий. Знание является результатом восприятия и понимания информации в контексте опыта при систематическом изучении.

Понимание – осознание закономерности, содержащейся во множестве разрозненных знаний.

Мудрость – оцененное понимание закономерности с точки зрения прошлого и будущего. Мудрость обеспечивает эффективное использование знаний для получения новой информации и расширения области познаний.

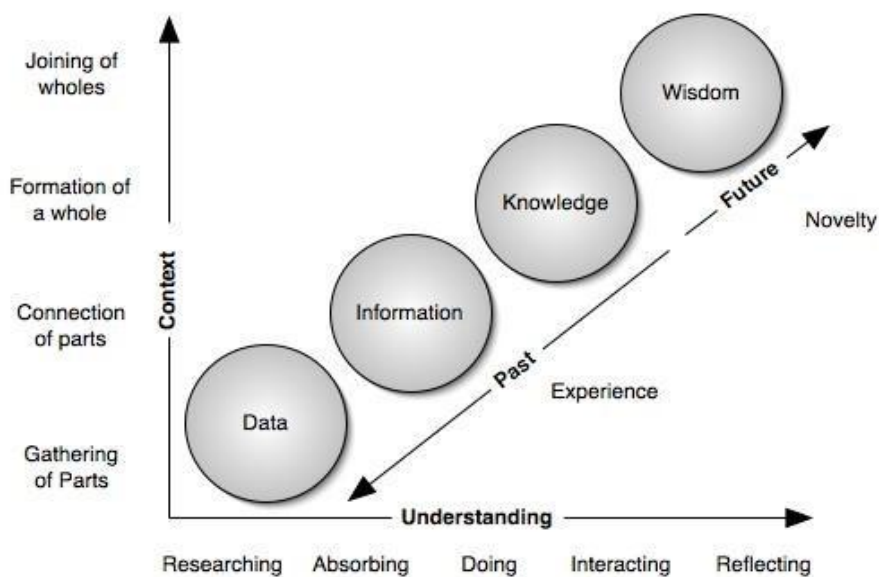


Рис. 1

Особое значение приобретает информатика в подготовке специалистов экономического профиля. Это связано с тем, что развитие информационных технологий свидетельствует о том, что выпускнику экономического вуза приходится:

- работать как конечному пользователю на персональном компьютере (ПК) (автоматизированном рабочем месте – АРМ, рабочей станции и т.п.) в

условиях «электронного офиса», интегрированной информационной системы, электронной почты, в локальных и глобальных телекоммуникационных компьютерных сетях;

- совершенствовать технологические и управленческие процессы на своем рабочем месте на основе автоматизации управленческих задач с использованием новейших технических и программных средств.

Комплекс этих условий отражает социальный заказ на специалиста и диктует требования необходимого уровня его информационной культуры. От того, насколько специалисты хорошо знают и владеют современными методами и средствами информатики, зависит эффективность функционирования производственного подразделения (организации) в целом.

В Концепции информатизации сферы образования Российской Федерации, утвержденной 10 июля 1998 года Министерством образования Российской Федерации, отмечено, что в настоящее время «в обществе существует достаточно серьезная и устойчивая социальная потребность в изучении информатики, а главное – в использовании информационных технологий в повседневной деятельности. Эта потребность обусловлена стремительным развитием средств информационно-вычислительной техники и связи, проникновением информационных технологий практически во все сферы социальной практики и настоятельной необходимостью их эффективного использования в интересах решения целого ряда актуальных социально-экономических проблем».

Учебный курс «Информатика» ориентирован на то, чтобы в результате его освоения:

- сформировать у студента фундамент современной информационной культуры;
- обеспечить устойчивые навыки работы на персональном компьютере в условиях локальных и глобальных вычислительных сетей и систем телекоммуникаций, новых информационных технологий в экономической деятельности;
- освоить основы современной методологии разработки компьютерных информационных систем и практической реализации ее основных элементов с использованием персональных компьютеров и типовых программных продуктов.

В результате изучения дисциплины студент должен:

- иметь представление о национальных и мировых информационных ресурсах как экономической категории;

- знать основы новых информационных технологий и их влияние на успех в бизнесе;
- знать современное состояние и направление развития компьютерной техники и программных средств;
- владеть основами автоматизации решения экономических задач; □ уверенно работать на персональном компьютере в качестве конечного пользователя информационных систем;
- знать основы создания информационных систем и использования передовых информационных технологий переработки информации, в т.ч. уметь работать с программными средствами, соответствующими современным требованиям мирового рынка программных продуктов;
- иметь представление о работе в локальных и глобальных компьютерных сетях, иметь навыки работы с электронной почтой, телеконференциями, электронной доской объявлений, средствами создания электронного офиса;
- знать наиболее важные информационные ресурсы, программные средства информационного поиска, а также экономические и правовые основы использования информационных ресурсов.

1.7 Технические средства обработки информации

Технологический процесс обработки данных в информационных системах осуществляется при помощи:

- технических средств сбора и регистрации данных;
- средств телекоммуникаций;
- систем хранения, поиска и выборки данных; □ средств вычислительной обработки данных; □ технических средств оргтехники.

В современных информационных системах технические средства обработки данных используются комплексно, на основе технико-экономического расчета целесообразности их применения, с учетом соотношения «цена/качество» и надежности работы технических средств.

Тема 2 Технические и программные средства реализации информационных процессов

2.1 Основные сведения об Информационной системе

Информационная система (ИС) предназначена для сбора, хранения, обработки и поиска информации, используемой для целей управления и удовлетворения информационных потребностей. ИС содержит:

- функциональные подсистемы для поддержки функций управления,
- обеспечивающие подсистемы для реализации операций обработки информации.

Функциональные подсистемы соответствуют целям управления, учитывают доступные ресурсы для их достижения, возможный эффект автоматизации управления. В составе функциональных подсистем – совокупность комплексов задач и операций. Основные функции управления:

- планирование и (или) прогнозирование;
- учет, контроль, анализ;
- координация и (или) регулирование.

Согласно ГОСТ 24.103-84 «Автоматизированные системы управления. Общие положения типовой набор обеспечивающих подсистем» выделены: информационное, программное, техническое, организационное, метрологическое, правовое и лингвистическое, математическое обеспечение.

- В состав информационного обеспечения ИС входят классификаторы технико-экономической информации, унифицированные формы документов, массивы нормативно-справочной и оперативной информации для решения функциональных задач.
- В состав программного обеспечения ИС входят программные средства, необходимые для реализации всех функций ИС.
- В состав технического обеспечения ИС входят технические средства, необходимые для получения, ввода, подготовки, обработки, хранения (накопления), регистрации, вывода, отображения, использования, передачи информации, реализации управляющих воздействий.
- В состав организационного обеспечения ИС входят документы, определяющие функции подразделений управления, действия и взаимодействие персонала ИС.

- В состав метрологического обеспечения ИС входят метрологические средства и инструкции по их применению (актуально для ИС технологическими процессами, так называемых АСУ ТП).
- В состав правового обеспечения ИС входят нормативные документы, определяющие правовой статус ИС, персонала ИС, правил функционирования ИС и нормативы на автоматически формируемые документы, в том числе на машинных носителях информации. Правовое обеспечение ИС в составе функционирующей системы реализуется в виде документов организационного обеспечения ИС.
- В состав лингвистического обеспечения ИС входят тезаурусы и языки описания и манипулирования данными. Лингвистическое обеспечение функционирующей ИС может присутствовать в ней самостоятельно или в виде решений по информационному обеспечению ИС и в документах организационного обеспечения ИС.
- В состав математического обеспечения ИС входят методы решения задач управления, модели и алгоритмы. В функционирующей системе математическое обеспечение реализовано в составе программного обеспечения.

2.2 Структура и архитектура информационной системы

Существуют различные виды структур ИС:

- функциональная (элементы – функции, задачи, операции; связи – информационные);
- техническая (элементы-устройства; связи – линии связи);
- организационная (элементы – коллективы людей и отдельные исполнители; связи – информационные, соподчинения и взаимодействия);
- алгоритмическая (элементы – алгоритмы; связи – информационные);
программная (элементы – программные модули; связи – информационные и управляющие);
- информационная (элементы – формы существования и представления информации в системе; связи – операции преобразования информации в системе).

Структура ИС учитывает особенности объекта управления, виды деятельности, технико-экономические и другие параметры и характеристики информационных технологий (ИТ). Типовая структура

ИС представлена на рис. 2. Состав и содержание функциональных подсистем полностью определяется функциями системы управления, учитывает принятые стандарты управления и организационную структуру объекта управления.

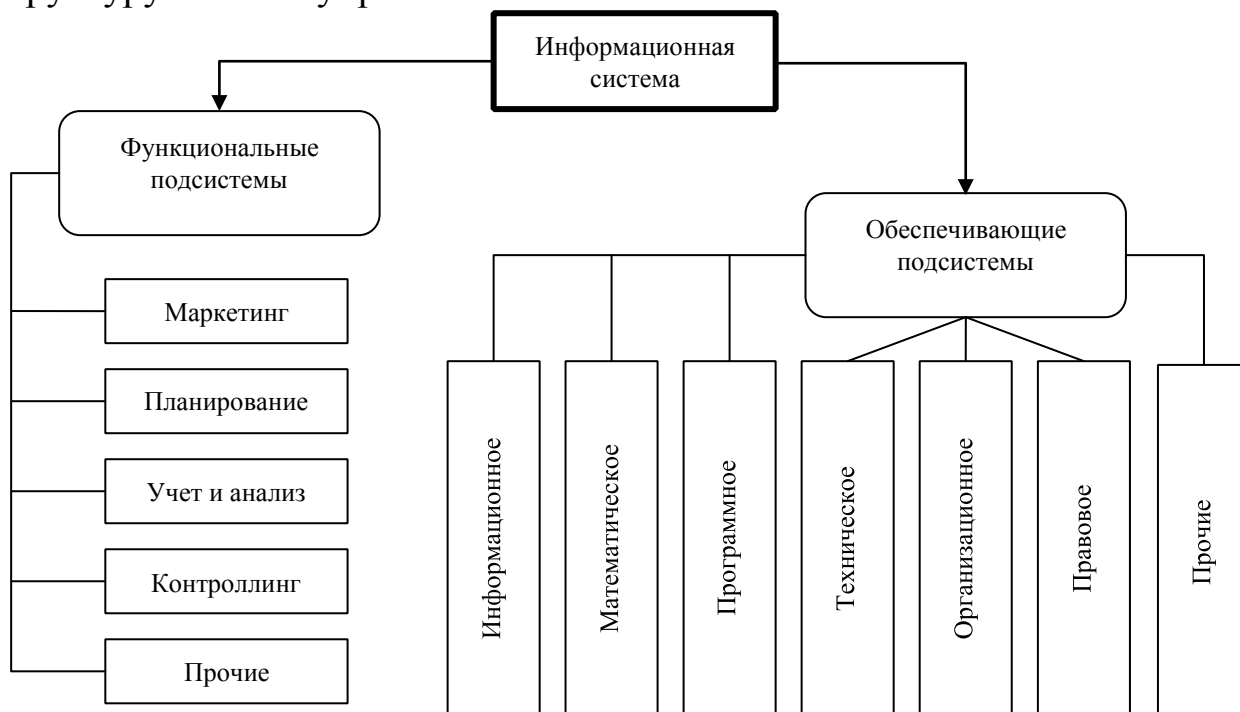


Рис. 2

Архитектура – организационная структура любой системы. (IEEE Std. 610.12-1990). Выделяют четыре области архитектур применительно к системе и объекту управления:

- Бизнес-архитектура.
- Архитектура программных систем.
- Технологическая или инфраструктурная архитектура.
- Информационная архитектура.

Архитектура ИС связана с концепцией построения, выполняемыми функциями и взаимосвязью компонентов ИС. Можно выделить следующие виды архитектур ИС:

1. Централизованная архитектура, которая предполагает наличие единого центра для хранения интегрированных данных (базы данных, БД), которая используется множеством приложений (пользователей) в условиях существования компьютерной сети; варианты централизованной архитектуры ИС:
 - а) файл-серверная архитектура компьютерной сети, хранение данных на файловом сервере;

- b) клиент-серверная двухуровневая архитектура, наличие интегрированной базы данных на сервере, варианты «тонкий» и «толстый» клиент, в зависимости от распределения функций обработки данных между сервером и рабочей станцией;
 - c) клиент-серверная трехуровневая архитектура, наличие интегрированной базы данных на сервере, наличие сервисориентированная архитектура (COA), которая предполагает интеграцию и многократное использование разнородных модулей на основе стандартных интерфейсов, взаимную независимость компонентов (операционной системы, платформы, языка программирования, вендора).
2. Облачная архитектура ИС, использующая виртуализацию информационных и инфраструктурных ресурсов, в том числе вебсервисов. Разновидности облачной архитектуры ИС:
- 1) данные как услуга;
 - 2) инфраструктура как услуга;
 - 3) программное обеспечение как услуга;
 - 4) платформа как услуга; 5) рабочее место как услуга; 6) всё как услуга.

2.3 Классификация информационных систем

(Материал из Википедии – свободной энциклопедии.)

Термин информационная система используется как в широком, так и в узком смысле.

В широком смысле информационная система есть совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией.

Так же в достаточно широком смысле трактует понятие информационной системы Федеральный закон РФ от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации»: «...информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств».

Одно из наиболее широких определений ИС дал М.Р. Когаловский: «...информационной системой называется комплекс, включающий вычислительное и коммуникационное оборудование, программное

обеспечение, лингвистические средства и информационные ресурсы, а также системный персонал и обеспечивающий поддержку динамической информационной модели некоторой части реального мира для удовлетворения информационных потребностей пользователей».

В узком смысле информационной системой называют только подмножество компонентов ИС в широком смысле, включающее базы данных, СУБД и специализированные прикладные программы.

ИС в узком смысле рассматривают как программно-аппаратную систему, предназначенную для автоматизации целенаправленной деятельности конечных пользователей, обеспечивающую, в соответствии с заложенной в нее логикой обработки, возможность получения, модификации и хранения информации.

В любом случае основной задачей ИС является удовлетворение конкретных информационных потребностей в рамках конкретной предметной области. Современные ИС де-факто немислимы без использования баз данных и СУБД, поэтому термин «информационная система» на практике сливается по смыслу с термином «система баз данных».

В идеале в рамках предприятия должна функционировать единая корпоративная информационная система, удовлетворяющая все существующие информационные потребности всех сотрудников, служб и подразделений. Однако на практике создание такой всеобъемлющей ИС слишком затруднено или даже невозможно, вследствие чего на предприятии обычно функционируют несколько различных ИС, решающих отдельные группы задач: управление производством, финансово-хозяйственная деятельность и т.д. Часть задач бывает «покрыта» одновременно несколькими ИС, часть задач – вовсе не автоматизирована. Такая ситуация получила название «лоскутной автоматизации» и является довольно типичной для многих предприятий.

2.4 Классификация по архитектуре

По степени распределённости отличают:

- настольные, или локальные ИС, в которых все компоненты находятся на одном компьютере;
- распределённые ИС, в которых компоненты распределены по нескольким компьютерам.

Распределённые ИС, в свою очередь, разделяют на: □
файл-серверные ИС;

- клиент-серверные ИС.

В файл-серверных ИС база данных находится на файловом сервере, а СУБД и клиентские приложения находятся на рабочих станциях.

В клиент-серверных ИС база данных и СУБД находятся на сервере, а на рабочих станциях находятся клиентские приложения.

В свою очередь, клиент-серверные ИС разделяют на двухзвенные и многозвенные.

В двухзвенных ИС всего два типа «звеньев»: сервер баз данных, на котором находятся БД и СУБД, и рабочие станции, на которых находятся клиентские приложения. Клиентские приложения обращаются к СУБД напрямую.

В многозвенных ИС добавляются промежуточные «звенья»: серверы приложений. Пользовательские клиентские приложения не обращаются к СУБД напрямую, они взаимодействуют с промежуточными звеньями. Типичный пример применения многозвенности – современные веб-приложения, использующие базы данных. В таких приложениях, помимо звена СУБД и клиентского звена, выполняющегося в веб-браузере, имеется как минимум одно промежуточное звено – веб-сервер с соответствующим серверным ПО.

2.5 Классификация по степени автоматизации

По степени автоматизации ИС делятся на:

- автоматизированные: информационные системы, в которых автоматизация может быть неполной;
- автоматические: информационные системы, в которых автоматизация является полной, то есть вмешательство персонала не требуется или требуется только эпизодически.

«Ручные ИС» существовать не могут, поскольку существующие определения предписывают обязательное наличие в составе ИС аппаратно-программных средств. Вследствие этого понятия «автоматизированная информационная система», «компьютерная информационная система» и просто «информационная система» являются синонимами.

2.6 Классификация по характеру обработки данных

По характеру обработки данных ИС делятся на:

- информационно-справочные, или информационно-поисковые ИС, в которых нет сложных алгоритмов обработки данных, а целью системы является поиск и выдача информации в удобном виде;
- ИС обработки данных, или решающие ИС, в которых данные подвергаются обработке по сложным алгоритмам. К таким системам в первую очередь относят автоматизированные системы управления и системы поддержки принятия решений.

2.7 Классификация по сфере применения

Поскольку ИС создаются для удовлетворения информационных потребностей в рамках конкретной предметной области, то каждой предметной области соответствует свой тип ИС. Перечислять все эти типы не имеет смысла, так как количество предметных областей велико, но можно указать в качестве примера следующие типы ИС:

- Экономическая информационная система – информационная система, предназначенная для выполнения функций управления на предприятии.
- Медицинская информационная система – информационная система, предназначенная для использования в лечебном или лечебно-профилактическом учреждении.
- Географическая информационная система – информационная система, обеспечивающая сбор, хранение, обработку, доступ, отображение и распространение пространственно-координированных данных.

2.8 Классификация по охвату задач

- Персональная ИС предназначена для решения некоторого круга задач одного человека.
- Групповая ИС ориентирована на коллективное использование информации членами рабочей группы или подразделения.
- Корпоративная ИС в идеале охватывает все информационные процессы целого предприятия, достигая их полной согласованности,

безызбыточности и прозрачности. Такие системы иногда называют системами комплексной автоматизации предприятия:

1. Системы электронной обработки данных (СОД), которые обеспечивают технологический процесс обработки данных.
2. Информационные системы управления (ИСУ). Используются для решения различного вида управленческих задач, использующих данные базы данных (БД).
3. Системы поддержки принятия решений (СППР) для решения в режиме диалога плохо структурированных задач, для которых характерна неполнота входных данных, неопределенность целей и ограничений, использующих как БД, так и базу моделей алгоритмов принятия решений.
4. Экспертные системы (ЭС) основываются на моделировании процесса принятия решения человеком-экспертом с использованием компьютера и разработок в области искусственного интеллекта. ЭС основываются на использовании как БД, так и знаний, улучшающих принимаемое человеком решение.

ИС можно разделить также на 2 класса: фактографические (единица хранения и обработки – «факт», информационная совокупность в виде реквизитов, показателей, структурных единиц информации) и документографические (единица хранения и обработки – «документ»).

К фактографическим ИС относятся:

- СОД – системы обработки данных, в которых преобладают технологические процессы и процедуры обработки данных.
- АИС – автоматизированные ИС.
- АСУ – автоматизированные системы управления объектом, в которых преобладают задачи для реализации функций управления.

К документографическим ИС относятся:

- ИПС – информационно-поисковые системы (АСНТИ – автоматизированная система научно-технической информации).
- ИСС – информационно-семантические системы на основе ИПЯ – информационно-поисковых языков (ЭС – экспертные системы).

2.9 Создание информационной системы

В соответствии с ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания» определены следующие стадии и этапы создания ИС (см. табл. 2):

Табл. 2

Стадии	Этапы работ
1. Формирование требований к ИС	1.1. Обследование объекта и обоснование необходимости создания ИС. 1.2. Формирование требований пользователя к ИС. 1.3. Оформление отчёта о выполненной работе и заявки на разработку ИС (тактико-технического задания)
2. Разработка концепции ИС	2.1. Изучение объекта. 2.2. Проведение необходимых научно-исследовательских работ. 2.3. Разработка варианта концепции ИС, удовлетворяющего требованиям пользователя. 2.4. Оформление отчёта о выполненной работе
3. Техническое задание	Разработка и утверждение технического задания на создание ИС
4. Эскизный проект	4.1. Разработка предварительных проектных решений по системе и её частям. 4.2. Разработка документации на ИС и её части
5. Технический проект	5.1. Разработка проектных решений по системе и её частям. 5.2. Разработка документации на ИС и её части. 5.3. Разработка и оформление документации на поставку изделий для комплектования ИС и (или) технических требований (технических заданий) на их разработку.

Окончание табл. 2

Стадии	Этапы работ
	5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации

6. Рабочая документация	6.1. Разработка рабочей документации на систему и её части. 6.2. Разработка или адаптация программ
7. Ввод в действие	7.1. Подготовка объекта автоматизации к вводу ИС в действие. 7.2. Подготовка персонала. 7.3. Комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями). 7.4. Строительно-монтажные работы. 7.5. Пусконаладочные работы. 7.6. Проведение предварительных испытаний. 7.7. Проведение опытной эксплуатации. 7.8. Проведение приёмочных испытаний
8. Сопровождение ИС	8.1. Выполнение работ в соответствии с гарантийными обязательствами. 8.2. Послегарантийное обслуживание

На этапе «Обследование объекта и обоснование необходимости создания в ИС» в общем случае проводят:

- 1) сбор данных об объекте автоматизации и осуществляемых видах деятельности;
- 2) оценку качества функционирования объекта и осуществляемых видов деятельности, выявление проблем, решение которых возможно средствами автоматизации;
- 3) оценку (технико-экономической, социальной) целесообразности создания ИС.

На этапе «Формирование требований пользователя к ИС» проводят:

- 1) подготовку исходных данных для формирования требований ИС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в действие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования системы);
- 2) формулировку и оформление требований пользователя к ИС.

На этапе «Оформление отчёта о выполненной работе и заявки на разработку ИС (технико-технического задания)» проводят оформление отчета о выполненных работах на данной стадии и оформление заявки на

разработку ИС (тактико-технического задания) или другого заменяющего её документа с аналогичным содержанием.

На этапах «Изучение объекта» и «Проведение научно-исследовательских работ» организация-разработчик проводит детальное изучение объекта автоматизации и необходимые научноисследовательские работы (НИР), связанные с поиском путей и оценкой возможности реализации требований пользователя, оформляют и утверждают отчёты о НИР.

На этапе «Разработка вариантов концепции ИС и выбор варианта концепции ИС, удовлетворяющего требованиям пользователя» в общем случае, проводят разработку альтернативных вариантов концепции создаваемой ИС и планов их реализации; оценку необходимых ресурсов на их реализацию и обеспечение функционирования; оценку преимуществ и недостатков каждого варианта; **определение порядка оценки качества и условий приёмки системы; оценку эффектов, получаемых от системы.**

На этапе «Оформление отчёта о выполненной работе» подготавливают и оформляют отчет, содержащий описание выполненных работ на стадии описания и обоснования предлагаемого варианта концепции системы.

На этапе «Разработка и утверждение технического задания на создание ИС» проводят разработку, оформление, согласование и утверждение технического задания на ИС и, при необходимости, технических заданий на части ИС.

На этапе «Разработка предварительных проектных решений по системе и её частям» определяют: функции ИС; функции подсистем, их цели и эффекты; состав комплексов задач и отдельных задач; концепция информационной базы, её укрупнённая структура; функции системы управления базой данных; состав вычислительной системы; функции и параметры основных программных средств.

На этапе «Разработка проектных решений по системе и её частям» обеспечивают разработку общих решений по системе и её частям, функционально-алгоритмической структуре системы, по функциям персонала и организационной структуре, по структуре технических средств, по алгоритмам решения задач и применяемым языкам, по организации и ведению информационной базы, системе классификации и кодирования информации, по программному обеспечению.

На этапе «Разработка документации на ИС и её части» проводят разработку, оформление, согласование и утверждение документации в объёме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию ИС.

На этапе «Разработка и оформление документации на поставку изделий для комплектования ИС и (или) технических требований (технических заданий) на их разработку» проводят: подготовку и оформление документации на поставку изделий для комплектования ИС; определение технических требований и составление ТЗ на разработку изделий, не изготавливаемых серийно.

На этапе «Разработка заданий на проектирование в смежных частях проекта объекта автоматизации» осуществляют разработку, оформление, согласование и утверждение заданий на проектирование в смежных частях проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием ИС.

На этапе «Разработка рабочей документации на систему и её части» осуществляют разработку рабочей документации, содержащей все необходимые и достаточные сведения для обеспечения выполнения работ по вводу ИС в действие и её эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы в соответствии с принятыми проектными решениями, её оформление, согласование и утверждение.

На этапе «Разработка или адаптация программ» проводят разработку программ и программных средств системы, выбор, адаптацию и (или) привязку приобретаемых программных средств, разработку программной документации.

На этапе «Подготовка объекта автоматизации к вводу ИС в действие» проводят работы по организационной подготовке объекта автоматизации к вводу ИС в действие, в том числе:

- реализацию проектных решений по организационной структуре ИС;
- обеспечение подразделений объекта управления инструктивно-методическими материалами; □ внедрение классификаторов информации.

На этапе «Подготовка персонала» проводят обучение персонала и проверку его способности обеспечить функционирование ИС.

На этапе «Комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)» обеспечивают получение комплектующих изделий серийного и единичного производства, материалов и монтажных изделий, проводят входной контроль их качества.

На этапе «Строительно-монтажные работы» проводят:

- выполнение работ по строительству специализированных зданий (помещений) для размещения технических средств и персонала ИС;
- сооружение кабельных каналов;
- выполнение работ по монтажу технических средств и линий связи;
- испытание смонтированных технических средств;
- сдачу технических средств для проведения пусконаладочных работ.

На этапе «Пусконаладочные работы» проводят:

- автономную наладку технических и программных средств, □ загрузку информации в базу данных и проверку системы её ведения; □ комплексную наладку всех средств системы.

На этапе «Проведение предварительных испытаний» осуществляют:

- а) испытания ИС на работоспособность и соответствие техническому заданию в соответствии с программой и методикой предварительных испытаний;
- б) устранение неисправностей и внесение изменений в документацию на ИС, в том числе эксплуатационную в соответствии с протоколом испытаний; □ в) оформление акта о приёмке ИС в опытную эксплуатацию.

На этапе «Проведение опытной эксплуатации» проводят:

- опытную эксплуатацию ИС;
- анализ результатов опытной эксплуатации ИС;
- доработку (при необходимости) программного обеспечения ИС; □ дополнительную наладку (при необходимости) технических средств ИС; □ оформление акта о завершении опытной эксплуатации.

На этапе «Проведение приёмочных испытаний» проводят: □ испытания на соответствие техническому заданию в соответствии с программой и методикой приёмочных испытаний;

- анализ результатов испытания ИС и устранение недостатков, выявленных при испытаниях; □ оформление акта о приёме ИС в постоянную эксплуатацию.

На этапе «Выполнение работ в соответствии с гарантийными обязательствами» осуществляются работы по устранению недостатков, выявленных при эксплуатации ИС в течение установленных гарантийных сроков, внесению необходимых изменений в документацию по ИС.

На этапе «Послегарантийное обслуживание» осуществляют работы по:

- анализу функционирования системы; □ выявлению отклонений фактических эксплуатационных характеристик ИС от проектных значений;
- установлению причин этих отклонений;
- устранению выявленных недостатков и обеспечению стабильности эксплуатационных характеристик ИС;
- внесению необходимых изменений в документацию на ИС.

2.10 Информационные технологии Информационные технологии (*information technology, IT*) – широкий класс дисциплин и областей деятельности, относящихся к технологиям управления и обработки данных с применением вычислительной техники.

Информационные технологии (ИТ) – это обобщённое название технологий, отвечающих за хранение, передачу, обработку, защиту и воспроизведение информации с использованием компьютеров. Организация объединённых наций по вопросам образования, науки и культуры ЮНЕСКО рассматривает ИТ как комплекс взаимосвязанных, научных, технологических, инженерных дисциплин, изучающих методы эффективной организации труда людей, занятых обработкой и хранением информации; вычислительную технику и методы организации и взаимодействия с людьми и производственным оборудованием, их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы.

Информационные процессы взаимодействия объектов материального мира можно структурировать (см. рис. 3):

- 1) перенос (запись) информации источником на носитель – кодирование;
- 2) передача сигнала по каналу связи;
- 3) преобразования кода сигнала в код получателя – декодирование;
- 4) обработка преобразованного кода получателем.

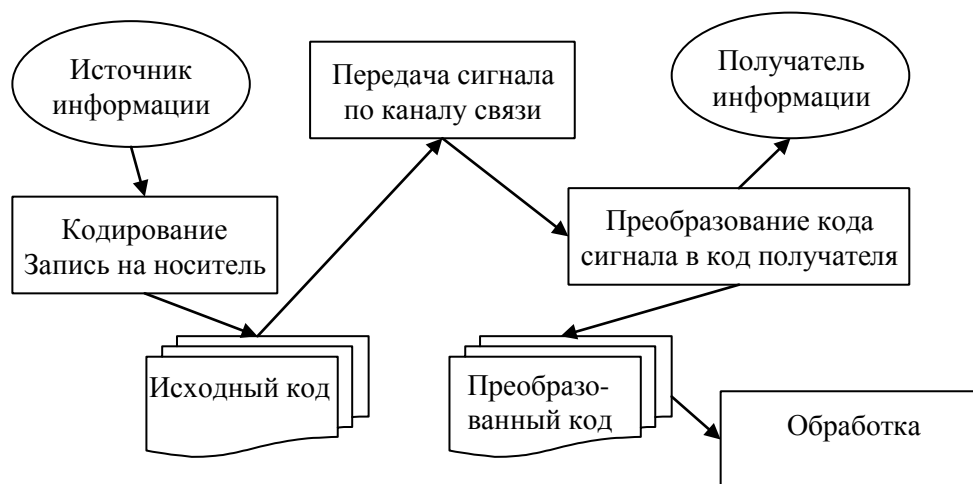


Рис. 3

Для каждого процесса существуют свои ИТ. Характерными чертами современных ИТ являются:

- сокращение трудозатрат на выполнение процедур обработки информации, повышение качества обработки информации;
- интерактивный характер обработки информации, ориентация на широкий круг пользователей и коллективный характер работы с информационными и вычислительными ресурсами;
- обеспечение единого информационного пространства ИТ, коллективная работа с информационными и вычислительными ресурсами на основе компьютерных сетей и систем телекоммуникаций;
- поддержка многосредовости (мультимедийности) ИТ, безбумажной технологии.

На какие классы можно разделить информационные технологии?

Информационные технологии можно разделить на классы:

1. Общего назначения ИТ (работа с текстовыми документами, вычисления в электронных таблицах, ведение баз данных, работа с компьютерной графикой и т.п.).
2. Методоориентированные ИТ, обеспечивающие применение особых моделей и алгоритмов для решения задач (математического аппарата, статистики, управления проектами и т.п.).
3. Проблемно ориентированные ИТ, учитывающие специфику предметной области, информационных потребностей пользователей.

Информационные технологии развиваются в следующих направлениях:

- компьютерная техника;
- средства связи и коммуникации;
- программные средства;
- методология организации проектных работ по созданию ИС.

Развитие ИТ связано:

- с прогрессом в области аппаратных средств обработки данных (ЭВМ, носителей информации, средств коммуникации и связи и прочего), промышленных технологий производства элементной базы компьютеров;
- развитием методов и инструментальных средств разработки программного обеспечения, способов хранения и поиска данных на машинных носителях;
- совершенствованием методов организацией труда коллективов разработчиков ИТ;
- ростом информационных потребностей конечных пользователей, требований повышения эффективности бизнес-систем.

Основные сведения о компьютере

2.11 Краткая история развития вычислительной техники

История вычислительной техники началась с попыток автоматизировать расчетные операции с помощью механических приспособлений. Полагают, что первыми «вычислительными» машинами были русские счеты (16 в.).

В течение 17-го и 18-го веков многие пытались найти простые способы вычисления. Шотландец Д. Напье изобрел механический способ умножения и деления, по принципу которого работают современные логарифмические линейки. Г. Бриггс использовал идеи Д. Напье для выпуска таблиц логарифмов, которые сегодня используются всеми математиками. Исчисления – другая область математики, она была изобретена независимо друг от друга англичанином сэром Исааком Ньютоном и немецким математиком Лейбницем.

В 1642 году французский ученый Блез Паскаль изобрел первую счетную машину, которая выполняла сложение чисел. В 1673 году Готфрид Вильгельм Лейбниц сконструировал арифмометр, позволяющий механически выполнять сложение, вычитание, умножение и деление. Начиная с 19-го века арифмометры получили самое широкое распространение. На них выполняли очень сложные расчеты, например расчеты баллистических таблиц для артиллерийских стрельб. Существовала и специальная профессия – счетчик – человек, работающий с арифмометром, быстро и точно выполняющий определенную последовательность инструкций (потом эту последовательность инструкций стали называть программой). Но многие расчеты производились очень медленно. Причина проста: при таких расчетах выбор выполняемых действий и запись результатов производились человеком, а скорость его работы весьма ограничена.

В 1830 году английский математик Чарльз Бэббидж попытался построить универсальное вычислительное устройство – «Аналитическую машину», которая должна была выполнить вычисления без участия человека. Для этого она должна была исполнять программы, вводимые с помощью перфокарт (они применялись уже довольно широко в ткацких станках), и иметь «склад» для запоминания данных и промежуточных результатов (память). Ч. Бэббидж не смог довести до конца работу по созданию «Аналитической машины» – она оказалась слишком сложной для техники того времени. Однако многие его идеи послужили основой для построения современных компьютеров.

В 1930 году американцем В. Бушем был построен первый аналоговый компьютер. Этот прибор использовался во Второй мировой войне для наводки орудий.

В 1944 году был закончен первый цифровой компьютер Марк 1. Создатели этого компьютера – профессор Г. Айкен и сотрудники фирмы IBM.

В 1946 году двое инженеров Д. Эккерт и Д. Мошли в Университете Пенсильвании построили первый цифровой компьютер на вакуумных лампах. Они назвали свое изобретение ЭНИАК.

В 1950 году появилось первое поколение компьютеров, использующих вакуумные лампы (например, УНИВАК). Они могли производить тысячи операций в секунду.

Развитие вычислительной техники в СССР тесно связано с С.А. Лебедевым, под руководством которого была создана первая отечественная ЭВМ: в 1951 году в Киеве – МЭСМ (Малая Электронная Счетная Машина) и в 1952 году – БЭСМ (Быстродействующая ЭСМ).

В 1960 году было разработано второе поколение компьютеров, которые были меньше, быстрее и надежнее, так как в них вместо вакуумных ламп использовались транзисторы. В нашей стране была создана БЭСМ-6, уровень которой, по мнению экспертов, опередил уровень зарубежных аналогов. Она широко использовалась в научных расчетах, при разработке и реализации отечественных космических программ.

Компьютеры третьего поколения появились на рынке в 1965 году, могли производить миллион операций в секунду и отличались более высокой надежностью и меньшими габаритами в связи с тем, что работали на малых интегральных схемах. В этих машинах впервые стали использовать дисплей, прямой доступ к машине получили ученые, инженеры, экономисты, студенты и т.д.

Наиболее типичными представителями машин третьего поколения являются IBM-360 и IBM-370 (США). В нашей стране созданы аналоги этих ЭВМ – Машины единой системы (ЕС ЭВМ) – ЕС-1022, ЕС-1035, ЕС-1066.

Немногим позже появились компьютеры четвертого поколения на основе больших интегральных микросхем (БИС), за счет чего удалось уменьшить размеры компьютеров, а также повысить их быстродействие и надежность. К ним относятся всякого рода микро- и мини-ЭВМ. Одним из достижений в области вычислительной техники явилось создание персональной ЭВМ, которую можно отнести к отдельному классу машин четвертого поколения. Именно с этого момента в нашем языке вместо ЭВМ утвердился термин «персональный компьютер» – ПК.

В последние годы развитие компьютеров шло очень стремительно. И теперь мы повсеместно встречаем компьютеры уже пятого поколения. Такие компьютеры могут разместиться у вас на коленях, и они способны помочь вам в решении самых различных задач. Кроме того, они надежнее своих предшественников, а скорость обработки информации составляет миллионы операций в секунду.

Таким образом, последующие модели ЭВМ имеют по сравнению с предыдущим более высокую скорость обработки информации, большую надежность и меньшие габариты.

История вычислительной техники уникальна, прежде всего, фантастическими темпами развития аппаратных и программных средств. До сих пор работают некоторые программисты, начинавшие еще на ламповых ЭВМ. И никто не возьмется предсказать, какой будет информационная технология через 10-15 лет.

2.12 Единицы измерения информации

Для цифровых ЭВМ элементарной единицей измерения количества информации является двоичная единица, которая называется – бит (сокращение от англ. *binary digit*). Бит – это количество информации, получаемой в результате однократного выбора из двух равновероятных событий. Значением бита, т.е. двоичного разряда, может быть 0 или 1, что соответствует двоичной системе счисления.

Общепринятый способ кодирования символов 8-битовыми числами называется ASCII (American Standard Coding for Information Interchange). В России принят стандарт кодировки КОИ-8, совместимый с ASCII.

Для измерения больших объемов информации используются следующие единицы измерения:

1 Килобайт = 1024, или 2^{10} байт;

1 Мегабайт = 1048576, или 2^{20} байт, или 1024 Кбайт; 1

Гигабайт = 1073741824, или 2^{30} байт, или 1024 Мбайт;

1 Терабайт = 1099511627776, или 2^{40} байт, или 1024 Гбайт.

Принято сокращенно записывать эти единицы измерения в виде:

Килобайт – Кбайт или Кб,

Мегабайт – Мбайт или Мб,

Гигабайт – Гбайт или Гб,

Терабайт – Тбайт или Тб.

Существует и такая единица информации, как слово. Слово составляется из 16 бит, т.е. из 2 байт. Если 8 бит (1 байт) позволяют записать целые положительные числа от 0 до 255, то 16-разрядным словом (2 байтами) можно выразить целые положительные числа от 0 до 65535.

2.13 Принцип действия ЭВМ

Электронная цифровая вычислительная машина (ЭВМ) – это устройство или система, способная выполнять заданную, четко определенную последовательность операций. Это чаще всего операции математических расчетов, манипулирование массивами данных, операции вводавывода.

Современные ЭВМ, созданные для различных областей применения, во многом отличаются друг от друга. Однако все они построены на основе принципа программного управления, один из способов реализации которого был предложен в 1945 году Джоном фон Нейманом. В результате реализации идей фон Неймана была создана архитектура ЭВМ, во многих чертах сохранившаяся до настоящего времени.

2.14 Общая структурная схема ЭВМ

На основе неймановского принципа программного управления определяется структурный состав ЭВМ.

Арифметико-логическое устройство (АЛУ) выполняет арифметические операции – сложение, вычитание, умножение и логические операции (обработка кодов чисел с помощью Булевой алгебры – «И», ИЛИ»).

Устройство управления (УУ) является организующим и направляющим устройством ЭВМ. Оно обеспечивает управление и контроль всех устройств, входящих в ЭВМ. Арифметико-логическое устройство и устройство управления образуют процессор ЭВМ. Процессор на одной или нескольких интегральных схемах называется микропроцессором. Назначение процессора – реализация программного управления, т.е. выборка команд из памяти и их выполнение.

Запоминающие устройства (память) обеспечивают хранение исходных и промежуточных данных, результатов счислений, а также программ. Они делятся на оперативные (ОЗУ), постоянные (ПЗУ) и внешние (ВЗУ).

Постоянные запоминающие устройства (ПЗУ), называемые также статической памятью, служат для постоянного хранения информации, которая записывается при изготовлении и не подлежит изменению пользователем (например, BIOS). Следовательно, прочитать эту информацию можно, а изменить нельзя. Даже при выключении питания информация в ПЗУ остается, в этом состоит отличие ПЗУ от ОЗУ.

Оперативные запоминающие устройства (ОЗУ), называемые также динамической памятью, предназначены для записи, хранения и считывания информации в процессе работы ПК, при выключении питания вся информация из ОЗУ разрушается. ОЗУ – устройство, способное работать с высоким быстродействием. Они обычно дороги и не обладают необходимыми характеристиками по объему хранимой информации. Поэтому для хранения

больших объемов информации приходится применять более дешевые, но и значительно менее быстродействующие устройства – внешние запоминающие устройства (ВЗУ).

В качестве ВЗУ (внешней памяти) используются магнитные носители – ленты и диски. Они предназначены для длительного хранения больших объемов информации, а также для переноса информации с одного компьютера на другой. ВЗУ не обладают нужными характеристиками по скорости и поэтому не могут согласованно работать вместе с процессором. Поэтому в ЭВМ применяется многоуровневая память. Непосредственно доступна процессору только информация, хранящаяся в ОЗУ. При необходимости использования информации из ВЗУ делается ее пересылка в ОЗУ. ВЗУ имеют много общего с устройствами ввода – вывода (УВВ). Поэтому в дальнейшем устройства этих двух классов назовем единым термином – периферийные устройства (ПУ).

Как правило, память ОЗУ и микропроцессор конструируются в виде электронных микросхем и информация в них представляется в виде логических сигналов, выраженных посредством электрических уровней напряжения. ПЗУ и ВЗУ в качестве носителей информации могут использовать такие физические носители, как электрический заряд, направление намагниченности используемого магнитного материала (доменные структуры), значение оптической поляризации (магнитооптические диски) и т.д.

Устройства ввода используются для ввода в ЭВМ данных, необходимых для вычислительного процесса, а также программ, в соответствии с которыми выполняется решение задач. Для ввода информации используются клавиатура, дисковод (для считывания информации с магнитных дисков), магнитофон, сканер, CD-диск, стримеры и т.д.

Устройство вывода обеспечивает выдачу результатов решения задачи на ЭВМ в форме, удобной для человека-оператора. Для вывода информации используются дисплей (монитор), принтер, графопостроитель, дисковод (для записи информации на магнитные диски) и т.д.

Таким образом, в состав ЭВМ входят устройства трех основных классов:

- 1) операционные, предназначенные для выполнения обработки информацией;
- 2) запоминающие, предназначенные для хранения информации (команд и данных),
- 3) ввода – вывода, предназначенные для связи ЭВМ с окружающей ее средой, в том числе человеком.

Основным операционным устройством ЭВМ является процессор. Исходя из всего вышесказанного следует, что, несмотря на имеющиеся различия при решении тех или иных задач, все типы ЭВМ выполняют один набор основных функций: ввод информации, хранение, арифметические и логические

преобразования, вывод информации и управление работой всех устройств, входящих в состав ЭВМ

При рассмотрении ЭВМ принято различать их архитектуру и структуру. Архитектура ЭВМ – понятие, охватывающее общую логическую организацию ЭВМ, состав и назначение ее функциональных средств, принципы кодирования и т.п., т.е. все то, что однозначно определяет принцип обработки информации на данной ЭВМ.

Структура ЭВМ – набор элементов и связей между ними. Совместное функционирование элементов, представляемое совокупностью физических процессов, приводит к реализации заданных функций ЭВМ, т.е. к вычислениям на основе заданных алгоритмов.

Ввиду большой сложности современных ЭВМ принято представлять их структуру иерархически, т.е. понятие «элемент» жестко не фиксируется. Так, на самом высоком уровне сама ЭВМ может считаться элементом. На следующем (программном) уровне иерархии элементами структуры ЭВМ являются память, процессор и другие операционные устройства, устройства ввода – вывода. На более низком уровне (микропрограммном) элементами являются узлы и блоки, из которых строятся память, процессор и т.д. Наконец, на самых низких уровнях элементами являются интегральные логические микросхемы и электрорадиоэлементы. Любой элемент ЭВМ, точно так же как и сама ЭВМ, характеризуется функциональным назначением и структурным построением.

Иерархичность функций и структур облегчает проектирование, использование и изучение ЭВМ и находит отражение в модульном принципе построения самой ЭВМ и ее программного обеспечения.

При проектировании современных ЭВМ используется модульный принцип построения. Суть этого принципа сводится к тому, что ЭВМ состоит из набора устройств и блоков – модулей, реализующих законченные функции и обладающих свойством независимости от других модулей в функциональном смысле. В конструктивном отношении модуль также представляет собой законченный конструктивный элемент. Отдельные модули могут быть соединены между собой в необходимую конфигурацию без изменения схем (функций) отдельных модулей.

Основное преимущество модульного принципа – возможность совершенствования ЭВМ без изменения ее функциональной организации путем замены отдельных блоков на новые (более быстродействующие, меньшие по размерам, потребляющие меньше энергии, более дешевые) или путем добавления новых модулей создание большого количества разных по характеристикам ЭВМ.

Кроме этого, такие ЭВМ наилучшим образом приспособлены к восстановлению работоспособности при отказах за счет упрощения поиска неисправности и ремонта.

2.15 Классификация вычислительной техники

Одна из основных характеристик ЭВМ – количество реализуемых в ней операций обработки, хранения и ввода – вывода информации – операционные ресурсы. Операционные ресурсы тем больше, чем более разнообразны способы представления информации и шире система команд.

Следующая важная характеристика ЭВМ – емкость памяти. Емкость памяти измеряется в двоичных единицах информации – битах, или кратных единицах: байтах, Кбайтах, Мбайтах, словах. Байт равен 8 битам.

1 Кбайт = 1024 байт, 1 Мбайт = 1024 Кбайт.

Быстродействие ЭВМ – число операций, выполняемых в 1 с. Поскольку разные операции выполняются с разной скоростью, то говорят о среднем быстродействии ЭВМ на разных классах задач, которое определяется с учетом вероятностей использования каждой операции при решении заданного класса задач.

Производительность ЭВМ, в отличие от быстродействия, измеряется средним числом задач, решаемых на ней за единицу времени. Производительность зависит от быстродействия процессора, устройств ввода – вывода, порядка прохождения задач в ЭВМ и т.п. Она увеличивается при совмещении операций ввода – вывода с обработкой, использовании мультипрограммного и мультипроцессорного режимов работы.

Надежность – свойство ЭВМ выполнять возложенные на нее функции в течение заданного времени. Надежность характеризуется такими величинами, как интенсивность отказов и наработка на отказ. Отказ – случайное событие, возникающее из-за неисправности элементов, из которых собрана ЭВМ, и соединений между ними. Интенсивность отказов – среднее число отказов за единицу времени. Нарработка на отказ – средний промежуток времени между отказами.

Увеличение надежности осуществляется за счет усовершенствования технологии производства компонентов, тестирования комплектующих и модулей на специальных стендах, а также резервирования, заключающегося во введении в состав ЭВМ отдельных устройств избыточного (дублирующего) оборудования.

Стоимость ЭВМ равна суммарной стоимости всего оборудования и программного обеспечения, входящего в ее состав. Улучшение любой из

рассмотренных характеристик ЭВМ при данном уровне технологии в конечном счете приводит к увеличению стоимости.

Конкретные области применения ЭВМ предъявляют к ним вполне определенные требования по каждой из перечисленных характеристик. Поэтому в конкретном случае существует оптимум, при котором требуемые характеристики обеспечиваются при минимальной стоимости ЭВМ.

Многообразие свойств и характеристик ЭВМ порождает и многообразие классификаций, отличающихся выделением разных признаков в качестве главных или второстепенных. Если в качестве основного признака взять размеры системы, то ЭВМ принято делить на сверхбольшие, большие, средние, малые и микроЭВМ.

Однако быстрое развитие технологии производства ЭВМ, элементной базы, методологии и средств разработки аппаратуры и программного обеспечения приводит к тому, что за короткий период существенно меняются характеристики новых ЭВМ, а именно повышается их производительность и надежность при сохранении стоимости. Так, современные микроЭВМ по многим своим характеристикам превосходят средние и даже большие ЭВМ первого поколения. Таким образом, удовлетворяются потребности пользователей в более эффективных машинах определенного типа.

Наиболее существенным образом характеристики ЭВМ определяются областью применения. В самом общем виде по областям применения ЭВМ разделяются на ЭВМ общего назначения, проблемно ориентированные, специализированные.

2.16 ЭВМ общего назначения

ЭВМ общего назначения имеют большие операционные ресурсы, приспособленные для обработки разнообразных числовых и текстовых данных, и предназначены для использования в вычислительных центрах (ВЦ). Они комплектуются большой оперативной и внешней памятью, широкой номенклатурой устройств отображения и устройств ввода – вывода и способны выполнять широкий спектр работ (выполнение научных, инженерных и экономических расчетов, обработка текстовой информации, решение задач моделирования). На их основе возможно создание крупных банков данных, систем автоматизированного проектирования (САПР), автоматизированных систем управления (АСУ) отраслью, крупным предприятием и т.д. ЭВМ общего назначения имеют большую стоимость, требуют специально приспособленных помещений, обслуживаются большим штатом программистов и инженерно-технических работников, выполняющих техническое обслуживание.

2.17 Проблемно ориентированные ЭВМ

Проблемно ориентированные ЭВМ приспособлены для решения ограниченного круга задач. Эти ЭВМ сравнительно дешевы, просты в эксплуатации и обслуживании и рассчитаны на массовое применение в качестве управляющих, а также для регистрации и обработки информации. Нужные характеристики по стоимости достигаются в этих ЭВМ в первую очередь за счет разумного ограничения операционных ресурсов, применительно к данному классу задач. В качестве проблемно ориентированных ЭВМ обычно используются мини- и микроЭВМ. По функциональной и структурной организации мини- и микроЭВМ мало отличаются друг от друга. Большинство мини- и микроЭВМ строятся на основе единого интерфейса. Характерная черта микроЭВМ – применение в них блоков, изготовленных на основе технологии больших и сверхбольших интегральных схем (БИС и СБИС). Некоторые наиболее удачные миниЭВМ позже были изготовлены в виде микроЭВМ, что позволило сохранить разработанное ранее программное обеспечение и расширить область применения.

2.18 Специализированные ЭВМ

Специализированные ЭВМ создаются для реализации какого-либо конкретного алгоритма: преобразования Фурье, вычисления корреляционных функций, преобразования координат и т.п. Узкая специализация позволяет существенно увеличить быстродействие ЭВМ, что очень важно при ее использовании для управления объектами в реальном масштабе времени. В настоящее время специализированные ЭВМ постепенно вытесняются универсальными мини- и особенно микро-ЭВМ, которые приспособляются к специальным применениям подключением к ним дополнительных обрабатывающих устройств, ускоряющих выполнение часто встречающихся операций.

Поскольку в настоящее время отсутствует точная классификация ЭВМ, одна и та же вычислительная машина может быть отнесена к разным классам. Более того, вполне реально использовать для одних и тех же целей ЭВМ, отличающиеся друг от друга размерами и стоимостью. В то же время арифметические и логические основы построения ЭВМ, принципы работы отдельных блоков и ЭВМ в целом, принципы программирования не зависят от того, как и по каким признакам разделены ЭВМ на классы.

2.19 Персональный компьютер

Распространение персональных компьютеров к концу 70-х годов привело к снижению спроса на ЭВМ и мини-ЭВМ. Это стало предметом серьезного беспокойства фирмы IBM international Business Machines Corporation), которая являлась основным поставщиком оборудования данного класса.

Термин ПК (PC) был придуман фирмой IBM для компьютера нового типа, разработку которого фирма начала в 1979 году. Термин расшифровывается как персональный компьютер (personal computer). С 1981 г., т.е. с момента официального представления первого IBM PC, он приобрел большую популярность у пользователей, а через несколько лет компьютер IBM PC занял ведущее место на рынке и фактически стал стандартом персонального компьютера. В настоящее время ПК, совместимые с IBM PC, занимают более 10% всех выпускаемых персональных компьютеров.

Поскольку термин PC сильно связан с оборудованием именно фирмы IBM, то владельцы фирмы Macintosh предпочитают называть свою продукцию по-другому. Они называют свои компьютеры Маками, даже если они всего-навсего разновидность ПК.

Второй IBM PC назывался IBM PC XT. XT означает extended Technology (расширенная технология). По большому счету это был обычный PC с 10-мегабайтным жестким диском. Затем появился IBM PC AT, имевший больший диск, более высокое быстродействие и улучшенный дизайн AT означает Advanced Technology (усовершенствованная технология). В действительности большинство современных компьютеров моделированы после AT. В 1987 г. IBM выпустила следующее поколение персональных компьютеров серии PS/2. PS означает Personal System (персональная система), а /2 – второе поколение. Так же как и PC, PS/2 является зарегистрированным названием.

Только фирма IBM может производить IBM PC. Обо всех прочих компьютерах, которые выглядят и функционируют как оригинальные модели, говорят как о клоне. Если у вас не оригинальная продукция IBM, то у вас клон.

Совместимость имеет несколько отличный от клона смысл. Понятие совместимости также относится к любым другим видам компьютеров, не имеющих марки IBM. Однако совместимость рассматривается в несколько ином смысле. Клоны считаются дешевыми: они обычно собираются прямо на задворках местных компьютерных магазинов из самых разнообразных комплектующих. Совместимые же компьютеры производятся солидными электронными компаниями и широко поддерживаются.

Поэтому совместимый компьютер стоит дороже, чем клон.

За последние несколько лет клоны PC быстро обогнали оригинальные PC и по объему продаж, и по массовости. Если о компьютере нельзя сказать, что он производства IBM, то это – клон.

Кроме всего прочего, совместимость означает, что вы можете использовать то же программное обеспечение, что и программное обеспечение компьютеров IBM. Вы сможете также использовать то же аппаратное обеспечение: принтеры, устройства ввода и пр. Сегодня понятие совместимости применимо более к DOS, чем к IBM, т.е. PC подразумевает скорее DOS-совместимость (системную совместимость), чем IBMсовместимость.

О некоторых компьютерах говорят, прибегая к маркетинговым названиям. IBM-компьютеры, например, носят названия IBM PC, IBM PC AT и IBM PS/2.

Большинство компьютеров носят имена их производителей плюс какой-нибудь номер, например, Mondo PC Mach 5, SmartFast i486, Testosterone 486/33. Это все те же ПК, клоны, совместимые компьютеры.

Компьютер может потребоваться по ряду причин: для постановки задачи, ее выполнения, для обучения или развлечения. Компьютер позволит выполнить любое дело намного проще. Однако он не имеет специального назначения, и потому довольно сложно сказать, что он конкретно сможет сделать. В этом смысле он не похож на прочие электронные приспособления, с которыми обычно приходится иметь дело.

Каждая компьютерная система состоит из двух частей. Необходимо кое-что знать о каждой из них, а также их названия: hardware (аппаратное обеспечение) и software (программное обеспечение).

Операционная система (ОС) является неотъемлемой частью современных ЭВМ, обеспечивая поддержку работы программ, организацию их взаимодействия с аппаратурой и предоставление пользователям возможностей общего управления компьютером. В настоящее время известно несколько ОС для ПК, отражающих как развитие технических средств, так и стремление разработчиков улучшить функциональные и эксплуатационные возможности, повысить степень комфортности ОС по отношению к пользователям.

Одной из представителей ОС, ещё широко используемой в мире является MS DOS фирмы Microsoft. К основным достоинствам MS DOS относятся: развитый командный язык, возможность организации многоуровневых каталогов, возможность подключения пользователем дополнительных драйверов внешних устройств и т.д.

Если компьютер использует DOS как свою главную управляющую программу, то он называется DOS-компьютер. Здесь речь идет об операционной системе MS DOS или Microsoft Disk Operating System (дискетной операционной системе фирмы Microsoft), которая устанавливается на многих

ПК как за рубежом, так и у нас в стране. Существуют и другие разновидности DOS: PC DOS, COMPAQ DOS, Tandy DOS и др.

Одно из направлений развития связано с созданием программных надстроек, так называемых операционных оболочек, которые обеспечивают удобный интерфейс (связь, общение) пользователей с прикладными программами и некоторые дополнительные функции.

Наиболее популярной оболочкой в настоящее время является Windows, фактически являющаяся операционной системой.

Компьютер предназначен для того, чтобы что-то на нем делать, работать с текстом или вести расчеты. Компьютер сам по себе – аппаратное обеспечение – абсолютно ничего не знает об этих вещах. Поэтому необходимо программное обеспечение, чтобы с его помощью настроить компьютер на выполнение определенного класса задач.

2.20 Типичный ПК

Типичный персональный компьютер семейства IBM PC состоит из следующих частей:

- системный блок,
- монитор (дисплей), □ клавиатура и мышь.

Дополнительно к компьютеру можно подключать различное периферийное оборудование: принтер, сканер, факс-модем, стример, CD-ROM, CD-Recorder, DVD-приводы, магнитооптика, графопостроитель, джойстик, внешний HDD, UPS, сетевое оборудование, цифровые камеры.

Системный блок содержит почти всю аппаратную часть компьютера, а именно: блок питания, видеоадаптер, накопители на жестком и гибких магнитных дисках, порты ввода/вывода, а также такие важные электронные устройства, как центральный процессор и основная «оперативная» память, как правило, расположенные на единой системной «материнской» плате. Кроме того, в системном блоке могут находиться дополнительные электронные устройства, такие как: факс-модем (внутренний), средства мультимедиа (CD дисковод, звуковая карта), сетевая карта.

Клавиатура представляет собой гибрид клавиатур печатной машинки и калькулятора. Она используется для ввода информации в компьютер и для управления. Есть клавиши для ввода символов, как на пишущей машинке, специальные функциональные клавиши, клавиши управляющие перемещением курсора и цифровые клавиши, похожие на кнопки калькулятора. Обычно на клавиатуре 104 (101) клавиша: 26 алфавитных, 10 с цифрами и знаками препинания, 12 (10) управляющих и несколько служебных.

Разъем для подключения клавиатуры может быть двух видов: АТ (DIN) или PS/2. В последние годы получили распространение эргономические и мультимедийные, позволяющие управлять CD-приводом клавиатуры, а также ориентированные на работу в среде Windows.

Клавиатура компьютеров типа laptop имеет сходные клавиши, но они размещены более плотно.

2.21 Персональный компьютер

Почти все ПК выглядят снаружи одинаково, потому что большинство клонов и IBM-совместимых компьютеров повторяют конструкцию и оформление оригинальных компьютеров PC и PC/XT/AT фирмы IBM. Однако имеются некоторые внешние различия, относящиеся к корпусам. Имеются два типа конструкций корпусов ПК: тип (стиль) DeskTop и АТХ(АТ)-тип. В настоящее время наиболее популярным является АТХ-тип. В пределах данного типа корпуса различаются по размеру MiniTower, MiddleTower (MidiTower), FullTower.

Так как современная технология дисковых накопительных устройств предлагает дисководы меньшего размера по сравнению с дисководами, которые существовали во время выпуска оригинального IBM PC, корпуса АТ-типа рассчитаны на применение «полугабаритных» накопителей, которые по высоте в два раза меньше накопителей, устанавливавшихся в оригинальных IBM PC.

Этот корпус оказался более гибким и практичным по сравнению со стандартным корпусом PC (ПК Macintosh использует аналогичную конструкцию).

В настоящее время используются 3-дюймовые накопители, а 5-дюймовые накопители не используются.

Имеется множество вариаций в конфигурации лицевой панели корпуса. Обычно панель содержит некоторую комбинацию следующих элементов:

- индикатор электропитания;
- индикатор доступа к накопителю на жестком диске (HDD);
- индикатор режима «турбо»;
- переключатель режима (режимов) «турбо»; □ кнопка перезагрузки (Reset); □ сетевой выключатель (Power).

Как накопители на жестком диске, так и накопители на гибком диске имеют индикатор рабочего режима (активности), который вспыхивает и горит, когда компьютер осуществляет доступ к накопителю.

Для хранения программ и данных в IBM-совместимых персональных компьютерах используются различного рода накопители. 5 1/4-дюймовый дисковод и 3 1/2 дюймовый дисковод.

По способу записи и чтения информации дисковые накопители можно разделить на магнитные, оптические и магнитооптические. Наиболее широко в настоящее время используются только магнитные накопители. Дисковые накопители делят обычно на следующие группы:

- накопители на флоппи-дисках;
- винчестеры;
- накопители на сменных жестких дисках;
- накопители, использующие эффект Бернулли;
- накопители на магнитооптических дисках;
- накопители на оптических дисках типа WORM (однократная запись – многократная запись);
- накопители на оптических компакт-дисках CD-ROM (Compact Disk Rom).

Накопители на дисках (или дисководы) являются устройствами внешней памяти и размещаются, как правило, на лицевой части системного блока. Дисководы устанавливаются один над другим, либо рядом один возле одного (тип PC/XT). Дисковод может быть вынесен за пределы системного блока и подсоединен специальным кабелем.

2.22 Накопители на компакт-дисках

Бурное развитие лазерной информационной технологии, и прежде всего лазерных компакт-дисков, привело к появлению новых компьютерных систем и направлений в создании носителей информации. Возможность записи на маленьком компакт-диске массой всего 15 граммов и диаметром 12 см огромного количества информации – около 640 Мб – привело к качественным изменениям в построении информационнопоисковых систем, внедрении компьютерных процессов обучения, создании тренажеров и т.п.

Этот прорыв произошел именно в результате применения лазерных дисководов CD-ROM. Не следует их путать с оптическими накопителями. Дисководы CD-ROM относительно дешевы (в зависимости от модели от \$50 и выше). Себестоимость изготовления лазерных дисков к ним тоже мала – около \$1–2 (и это при объеме диска 640 МБ, что позволяет разместить на нем объем информации, эквивалентный 275 томам энциклопедии по 1000 страниц в каждой). Эти дисководы аналогичны выпускающимся лазерным проигрывателям компакт-дисков. Поэтому, если на вашем компьютере установлен CD-ROM, вы можете слушать стандартные музыкальные компакт-диски.

Цифровой способ записи, применяемый на компакт-дисках CD-ROM, позволяет записывать на них и воспроизводить с одного и того же носителя совершенно разную информацию – буквенно-цифровую, звуковую (музыка и речь), графическую, а также цветные неподвижные и подвижные изображения

Накопители – устройства, обеспечивающие запись и считывание с носителей информации, в частности с магнитных дисков. Основными частями НГМД являются дисководное устройство, хотя их может быть несколько, и плата контроллера. Они связаны между собой кабелем. Кроме того, имеется кабель питания, подключаемый к задней части дисковода.

Сам по себе дисковод – это легкое и компактное устройство. Его размеры: ширина около 6 дюймов, высота 1 1/2 дюйма и длина 8 дюймов. Передняя часть дисковода содержит следующие компоненты:

- отверстие, через которое диск вставляется в устройство;
- лампочка рабочего режима (светодиод),
- замок-защелка или кнопка выброса (для 3 1/2 дюймовых дисководов).

Ориентация дисковода важна потому, что необходимо вставлять дискету в накопитель только определенным образом ориентированную – этикеткой кверху.

Гибкие диски (дискеты) позволяют переносить документы и программы с одного компьютера на другой, хранить информацию, не используемую постоянно на компьютере, делать архивные копии информации, содержащейся на жестком диске. Это настолько удобный способ хранения, что большинство компаний, занимающихся разработкой и распространением программного обеспечения, активно используют этот способ транспортировки своих программных продуктов как основной.

История гибкого магнитного (флоппи) диска началась с того момента, когда магнитный слой нанесли на тонкую майларовую (лавсановую) основу, примерно подобную той, что используется в магнитной ленте. Чтобы не поцарапать и не испачкать поверхность носителя, диск поместили в достаточно жесткий пластиковый конверт, внутри которого он мог свободно вращаться.

Чтобы компьютер смог работать с информацией на дискете, необходимо вставить ее в дисковод. Компьютер не может работать с данными прямо на диске. Вместо этого он сначала загружает информацию в память. Затем обрабатывает эти данные. Поскольку компьютер работает только с копией данных, то информация на диске остается неизменной. После завершения работы вы можете записать результаты своего труда обратно на диск, вместо старых данных или организовать новый файл с обновленной информацией.

подавляющее большинство дискет бывает двух размеров: 5 1/4 и 3 1/2 дюйма. Кроме того, каждая дискета может быть большой или малой емкости, так что существует четыре основных разновидности дискет.

Существуют и другие названия для дискет: высокой плотности, 1.2 Мбайт, DS/HD (double-side/high-density – две стороны, высокая плотность) и 96 TPI.

Емкость дискет этого типа 1.2 Мбайт, что почти в четыре раза больше, чем 360 Кбайт. Но с виду дискеты этих двух типов абсолютно одинаковы. Как же их различать? Посмотрите на отверстие в центре дискеты. На дискетах малой емкости установлено специальное усиливающее кольцо, кусочек пластика, вдоль края дорожки. А на дискетах высокой емкости такого кольца нет. Обычно дискеты емкостью 1,2 Мбайта имеют более темное магнитное покрытие. Только специальные дисководы могут работать с дискетами высокой емкости. На старых дисководах, рассчитанных на дискеты 360 Кбайт, работать с дискетами высокой емкости невозможно. Если вы все же попытаетесь, то получите сообщение об ошибке "General failure" (общий отказ).

В настоящее время дискеты такого формата используются редко. Достаточно обратить внимание на то, что компьютеры, предлагаемые к продаже сегодня, как правило, дисководами для 5 1/4-дюймовых дискет не комплектуются. Широко используются другие типы дисководов, рассчитанные на 3 1/2-дюймовые дискеты.

3-дюймовая дискета состоит из:

- Корпус диска.
- Магнитный диск.
- Место наклейки.
- Переключатель защиты от записи.
- Отверстие для головки записи-чтения.
- Металлический сердечник.
- Центрирующие отверстия.
- Замок отверстий под головки.
- Замок отверстий под головки в открытом состоянии.
- Отверстия, информирующие, что дискета высокой плотности.
- Скошенный угол, не позволяющий вставить диск не той стороной.

Другие названия: высокой плотности, 1,44 Мбайт, DS/HD (doubleside/high-density – две стороны, высокая плотность). Эти дискеты по 1,44 Мбайт довольно легко отличить от их родственниц с малой емкостью. Взгляните на край дискеты, противоположный металлической пластине. В одном углу вы увидите такое же отверстие с задвижкой (4), как и на 720-килобайтных дискетах. Но в другом углу тоже будет отверстие (10). Как уже сказано, отверстие с задвижкой служит для возможности установки защиты от записи. Эти дискеты часто помечены буквами "HD", что означает high-density – высокая плотность. 3 1/2-дюймовая дискета расширенной емкости. Другие названия. 2.8 Мбайт, DS/ED (double-side/extended-density – две стороны, расширенная плотность). Эти диски не получили большого

распространения. Отличить их от дискет с меньшей емкостью можно по буквам "ED" в одном из углов дискеты.

Запись информации на дискету производится на магнитную дорожку, которая должна быть определена заранее. Существуют и более профессиональные термины, как сектор, кластер, цилиндр, байты циклического контроля четности (CRC), корневой каталог и FAT.

Процесс разметки компьютером этих специальных дорожек и называется форматированием (или разметкой). Сегодня только IBMсовместимые компьютеры продолжают использовать 5 1/4-дюймовые дискеты. Они выпускаются в основном неразмеченными. (Действительно, можно встретить и форматированные дискеты, но стоят они дороже.) Прежде чем вы сможете использовать дискеты, все они должны быть форматированы.

Рекомендации. Хранится информация на диске в виде бинарных кодов, записанных на магнитном покрытии. При более детальном рассмотрении физики процесса записи и хранения информации можно сделать вывод, что, расположив близко к дискете магнит либо поместив носитель вблизи источника электромагнитного излучения, вы можете навсегда потерять то, что было записано на дискете. Помните о скрытых магнитах. Они могут находиться в телефонных трубках, в радио- и телевизионных динамиках, в специальных держателях бумаги, в настольных вентиляторах, усилителях электрогитар и многих других, на первый взгляд, безобидных устройствах.

Не стоит класть книги и другие тяжелые предметы сверху на дискету. Их вес может вдавить частички пыли в диск. Избегайте экстремальных температур. Не оставляйте дискеты на приборной доске автомобиля и даже на подоконнике. Постарайтесь всегда прятать дискеты обратно в защитные пакетики. Это особенно важно для старых 5 1/4-дюймовых дискет.

Не касайтесь поверхности самого диска, берите дискету только за пластиковую оболочку. Даже не пытайтесь чем-нибудь смазать или побрызгать дискету, если она издает слишком много шума при вращении. (Скорее всего, этот шум производит сам дисковод, но смазать его самостоятельно тоже лучше не пытайтесь.)

При пересылке 5 1/4-дюймовых дискет лучше пользоваться специальными упаковками, не стоит сгибать дискету пополам, чтобы вложить ее в обычный пакет.

Не пользуйтесь шариковыми ручками, чтобы писать на дискете. Твердый шарик из пишущего блока может повредить диск. Лучше пользуйтесь фломастерами или мягкими карандашами и обязательно наклеивайте на дискеты специальные этикетки.

Чтобы работать с дискетами расширенной емкости, требуется специальный дисковод. Они не получили пока что широкого распространения, поэтому программное обеспечение них встречается довольно редко.

На дисководах для дискет расширенной емкости прочие типы 3 1/2дюймовых дискет читаются без проблем.

Кроме всего выше перечисленного, для обеспечения надежности и долговечности работы дисковода и дискет рекомендуется пользоваться специальными чистящими дискетами, которые удаляют загрязнение головок механизма дисковода.

На сегодняшний день можно сказать, что наиболее важным узлом любого персонального компьютера является его жесткий диск – винчестер.

В 1973 году фирмой IBM по новой технологии был разработан первый жесткий диск, который мог хранить до 16 Кбайт информации. Поскольку этот диск имел 30 цилиндров (дорожек), каждый из которых разбит на 30 секторов, то сначала ему присвоили незамысловатое название – 30/30. По аналогии с популярными винтовками, имеющими калибр 30/30, такие жесткие диски стали называться винчестерами. С тех пор конструкция винчестера изменилась, его емкость, надежность и быстродействие значительно выросли. Однако название «прижилось» и остается понятным любому, несмотря на то что аналогия с ружьем «винчестер» осталась далеко в прошлом.

Итак, накопитель содержит один или несколько дисков (platters), т.е. носителей, которые смонтированы на оси-шпинделе и приводятся в движение специальным приводом. Скорость вращения двигателя для обычных моделей составляла около 3600 об./мин. Понятно, что чем выше скорость вращения, тем быстрее считывается информация с диска (разумеется, при постоянной плотности записи), однако пластины носителя при больших оборотах могут просто физически разрушиться. Тем не менее, в современных моделях винчестеров скорость вращения достигает уже 4500, 5400, 7200 об./мин. и даже выше.

Сами диски представляют собой обработанные с высокой точностью керамические или алюминиевые пластины, на которые и нанесен специальный магнитный слой (покрытие). Надо отметить, что за последние 25 лет технология изготовления этих деталей ушла далеко вперед. В старых накопителях в качестве магнитного покрытия обычно использовался оксид железа. В настоящее время для покрытий используются гамма-феррит-оксид, изотропный оксид и феррит бария, однако наиболее широкое распространение получили диски с напыленным магнитным слоем, а точнее, с металлической пленкой (например, кобальта).

Количество дисков может быть различным – от одного до пяти и выше. Число рабочих поверхностей соответственно в два раза больше, правда, не

всегда. Иногда наружные поверхности крайних дисков или одного из них не используются для хранения данных, при этом число рабочих поверхностей уменьшается и может оказаться нечетным.

Наиболее важной частью любого накопителя являются головки чтения-записи (read-write head). Как правило, они находятся на специальном позиционере, который напоминает рычаг звукоснимателя на проигрывателе грампластинок (тонарм). Это и есть поворачивающийся позиционер головок (head actuator). Существуют также и линейные позиционеры, по принципу движения напоминающие тангенциальные тонармы.

Кроме перечисленного, на любом винчестере обязательно находится печатная плата с электронными компонентами, которые необходимы для нормального функционирования устройства привода. Так, например, электроника расшифровывает команды контроллера жесткого диска, стабилизирует скорость вращения двигателя, генерирует сигналы для головок записи и усиливает их от головок чтения и т.п.

Непременными компонентами большинства винчестеров являются барометрические фильтры, выравнивающие внутреннее и наружное давление, а также обычные воздушные фильтры. По понятным причинам большое значение для работы жестких дисков имеет чистота окружающего воздуха, поскольку грязь или пыль могут вызвать соударение головки с диском, что однозначно приведет к выходу его из строя.

Принципы, на которых основано хранение информации накопителями на жестких и гибких дисках, во многом схожи. В современных винчестерах головки как бы «летят» на расстоянии доли микрона (обычно около 0.13 мкр) от поверхности диска, разумеется, не касаясь их. С помощью механизма позиционирования головка точно устанавливается на места записи или чтения информации.

Жесткие диски стационарны, вы не можете их снимать или хранить так же легко, как гибкие диски. Со временем, были созданы съемные жесткие диски. Но обычно диски фиксируются внутри компьютеров (почему их иногда называют несъемными или стационарными дисками).

Поскольку НЖМД находится в почти герметичной среде, никогда не пытайтесь вскрыть его защитную оболочку. На начальном уровне знаний и опыта в области модернизации и технического обслуживания ПК, если возникают проблемы с НЖМД, лучше заменить накопитель целиком. Учтите, что для страховки от неприятностей, лучше всего иметь резервные копии всей информации вашего НЖМД. Вы можете серьезно повредить диск, если попытаетесь проникнуть в герметизированный корпус, чтобы починить что-то, в чем разбираются только специалисты.

2.23 Мониторы и видеоадаптеры

Монитор (дисплей) предназначен для вывода на экран текстовой и графической информации. Монитор работает под управлением специального аппаратного устройства видеоадаптера, который предусматривает два возможных режима – текстовый и графический.

В текстовом режиме экран разбивается на 25 строк по 80 позиций в каждой строке (всего 2000 позиций). В каждую позицию (знакоместо) может быть выведен любой из символов кодовой таблицы – прописная или строчная буква латинского или русского алфавита, служебный знак («плюс», «минус», «точка» и т.д.), символ псевдографики, а также графический образ почти каждого управляющего символа.

В графическом режиме изображение формируется почти так же, как и на экране телевизора, – мозаикой, совокупностью точек, каждая из которых окрашена в тот или иной цвет. В обычно EGA выводит на экран 640 точек по горизонтали и 350 строк, VGA – 640 точек и 480 строк, SVGA – 640*480, 800*600, 1024*768 и более. Число возможных цветов каждой точки («палитра») зависит от типа адаптера, и от его разрешения, и от объема видеопамати, расположенной на видеоадаптере. Минимальный элемент изображения на экране (точка) называется пикселем – от английского «picture element».

С параметрами видеоадаптера не следует путать физические характеристики монитора – размер зерна люминофора (например, 0,39 мм, 0,31 мм, 0,28 мм и меньше) и размер экрана по диагонали (например, 14, 15, 17, 19, 21 дюймов, 1 дюйм = 25,4 мм).

Четкость изображения на экране определяется и физическими данными монитора, и разрешающей способностью видеоадаптера, а также качеством исполнения аппаратных элементов видеосистемы (например, качеством слоя люминофора).

Одной из наиболее важных составных частей любого персонального компьютера является его видеоподсистема. Под этим понятием обычно подразумевают монитор, плату видеоадаптера и набор соответствующих программ-драйверов, поставляемых в комплекте с адаптером или в составе прикладных пакетов. Оба вышеназванные устройства (монитор и видеоадаптер) очень плотно взаимодействуют между собой, поэтому, говоря об одном из них, приходится упоминать и другое.

Были распространены мониторы с видеоадаптерами EGA (Enhanced Graphic Adapter улучшенный графический адаптер), VGA (Video Graphic Array – видеографическая матрица) и SVGA (SuperVGA). Адаптер EGA сохранился на старых машинах, в современных компьютерах уже не используется.

Основные характеристики изображения в графическом режиме – разрешающая способность видеоадаптера, т.е. количество точек, выводимых по горизонтали и вертикали, и число возможных цветов каждой точки. За исключением портативных, подавляющее большинство современных настольных компьютеров используют мониторы на базе электронно-лучевых трубок (ЭЛТ). Принцип действия подобных мониторов мало чем отличается от обычного телевизора и заключается в том, что испускаемый электродом (электронной пушкой) пучок электронов, попадая на экран, покрытый люминофором, вызывает его свечение. На пути пучка электронов обычно находятся дополнительные электроды: отклоняющая система, позволяющая изменять направление пучка, и модулятор, регулирующий яркость получаемого изображения. Заметим, что любое текстовое или графическое изображение на экране монитора компьютера (как, впрочем, и телевизора) состоит, вообще говоря, из множества светящихся дискретных точек люминофора, называемых также пикселями (pixel – picture element). Электронный луч в этом случае периодически сканирует весь экран, образуя на нем близко расположенные строки развертки. Именно этот шаблон и называется растром, поэтому такие дисплеи называют еще растровыми. По мере движения луча по строкам видеосигнал, подаваемый на модулятор, изменяет яркость определенных пикселей, образуя некоторое видимое изображение. Разрешающая способность монитора определяется числом элементов изображения, которые воспроизводятся по горизонтали и вертикали, например, 640x480 или 1024x768 пикселей. Сразу же заметим, что существуют несколько стандартных типоразмеров экранов мониторов, используемых для IBM-совместимых персональных компьютеров: 14, 15, 17, 19, 20 и 21 дюйм (по диагонали).

Для формирования раstra в мониторе используются специальные управляющие сигналы. В цикле сканирования луч движется по зигзагообразной траектории от левого верхнего угла экрана до нижнего правого. Прямой ход луча по горизонтали осуществляется сигналом строчной (горизонтальной) развертки, а по вертикали – (вертикальной) кадровой. Перевод луча из крайней правой точки строки в крайнюю левую точку следующей строки (обратный горизонтальный ход луча) и из крайней правой позиции последней строки экрана в крайнюю левую позицию первой строки (вертикальный обратный ход луча) осуществляется специальными сигналами обратного хода.

Наиболее важными для монитора являются следующие параметры: частота вертикальной синхронизации (кадровая развертка), частота горизонтальной синхронизации (строчная развертка) и полоса пропускания видеосигнала. Кадровая частота определяется обычно в герцах и во многом определяет устойчивость изображения (отсутствие мерцаний). Как известно, человеческий глаз воспринимает смену изображений с частотой выше 20-25 Гц практически как непрерывное движение. Чем выше частота кадров, тем устойчивее

изображение. Однако повышение этой частоты требует увеличения частоты строчкой развертки, так как уменьшается время, отводимое на формирование каждой точки изображения. Частота строк в килогерцах, вообще говоря, определяется произведением частоты вертикальной развертки на количество выводимых строк в одном кадре (разрешающая способность по вертикали). Полоса видеосигнала, измеряемая в мегагерцах, определяет самые высокие частоты в видеосигнале. Приблизительно эта величина может быть определена произведением количества точек в строке (разрешающая способность по горизонтали) на частоту строчной развертки. Например, организация VESA (Video Electronics Standards Association) рекомендовала использовать следующие частоты кадровой развертки (см. табл. 3):

Табл. 3

Разрешение	Частота
640x480	72 Гц
800x600	72 Гц
1024x768	70 Гц

Не менее значимым фактором, чем частота кадровой развертки, является способ формирования изображения на экране монитора в режимах высокого разрешения – строчный или чересстрочный. При строчном способе формирования изображения все строки кадра выводятся в течение одного периода кадровой развертки, при чересстрочном – за один период кадровой развертки выводятся четные строки изображения, а за следующий – нечетные. Таким образом, говорят, что один кадр делится на два поля. Последний способ, кстати, используется и в телевизорах. Совершенно нетрудно заметить, что в случае чересстрочной развертки эквивалентная частота кадров снижается вдвое. Это позволяет достаточно легко увеличивать разрешающую способность монитора, хотя и в ущерб качеству изображения.

Некоторые исследования, проведенные различными организациями здравоохранения, показали, что излучения, сопровождающие работу монитора, могут существенно сказываться на здоровье человека. Заметим, что спектр этого излучения достаточно широк, это и мягкое рентгеновское излучение, и инфракрасное, и радиоизлучение, а также электростатические поля. Мониторы должны соответствовать тестам SSI 3/92 Шведского национального института защиты от излучения.

2.24 Клавиатура

Клавиатура является пока одним из основных устройств ввода информации в компьютер. В техническом аспекте компьютерная клавиатура представляет совокупность механических датчиков, воспринимающих давление на клавиши и замыкающих тем или иным образом определенную электрическую цепь. В настоящее время наиболее распространены два типа клавиатур: с механическими и мембранными переключателями. В первом случае датчик представляет собой традиционный механизм с контактами из драгоценного металла, а во втором – тонкие посеребренные листки пластика, между которыми с небольшим воздушным зазором находится, например, проводящая жидкость.

Внутри корпуса клавиатуры, помимо датчиков клавиш, расположены электронные схемы дешифрации. Сам же контроллер клавиатуры находится на системной плате и выполнен обычно в виде отдельной микросхемы. Для различных моделей компьютеров тип контроллера клавиатуры может отличаться. Большинство современных клавиатур либо имеют переключатель режимов (например, XT/AT), либо автоматически определяют тип контроллера.

Основной принцип работы клавиатуры вместе с микросхемой контроллера заключается в сканировании (постоянной проверке состояния с большой скоростью) переключателей клавиш. Замыканию и размыканию любого из этих переключателей (т.е. нажатию или отпусканию клавиши) соответствует уникальный цифровой код – скан-код (размером один байт). Кроме того, при каждом размыкании и замыкании переключателей контроллер клавиатуры инициирует аппаратный запрос, который обслуживается специальной подпрограммой, входящей в состав ROM BIOS. Процедура обработки запроса включает, в частности, такие операции, как считывание скан-кода из порта (адресуемая ячейка) контроллера клавиатуры, трансляцию этого кода в расширенный код ASCII (два байта: старший – скан-код, младший – ASCII), сохранение оттранслированного кода в циклическом буфере клавиатуры, слежение за состоянием служебных клавиш (например, Shift, Alt, Ctrl), детектирование специальных комбинаций клавиш (например, Ctrl+Alt+Del, Ctrl+Break) и т.п. Заметим, что микросхема контроллера клавиатуры имеет собственный встроенный буфер.

Привлекательность той или иной клавиатуры в основном зависит от расположения клавиш, тактильных ощущений и усилия при нажатии клавиши. Независимо от используемой технологии, сила, требуемая для нажатия клавиши, составляет около 20–50 г, а рабочий ход – около 4 мм.

подавляющее большинство современных IBM-совместимых компьютеров используют так называемую улучшенную (enhanced) или расширенную клавиатуру. Основное улучшение по сравнению со стандартной клавиатурой AT касается общего числа (101 и более) и расположения клавиш. Наиболее

распространенным стандартом расположения клавиш является QWERTY (ЙЦУКЕН). Есть около 60 клавиш с буквами, цифрами, знаками пунктуации и другими символами, встречающимися в печатных текстах, и еще около 40 клавиш, предназначенных для управления компьютером и исполнением программ. Продублированы клавиши управления курсором, а также клавиши Ctrl и Alt. Функциональные клавиши F1-F10 перенесены в верхний ряд и к ним добавлены две новые (F11 и F12). Габаритные размеры стандартной клавиатуры составляют примерно 30*190*450 мм, а вес – около 1 кг.

Соединители в кабелях для клавиатуры – это обычно трубчатые штекеры с пятью контактными выводами.

Говоря о клавиатуре, нельзя не упомянуть о том, каким образом осуществляется ввод в компьютер символов русского алфавита. Для этого используются специальные программы, называемые драйверами клавиатуры, загрузка которых в оперативную память происходит каждый раз при включении компьютера. Написано множество программ руссификаторов клавиатуры.

Широко используются драйверы, в которых для переключения клавиатуры из режима ввода латинских символов в режим ввода русских и наоборот используется одновременное нажатие правой и левой клавиш {Shift}, либо правой клавиши {Ctrl}. Удобными в использовании являются драйверы, отображающие переключение из одного режима в другой цветной рамкой по периметру экрана, (что приемлемо для текстовых редакторов, работающих под DOS).

2.25 Принтеры

Пожалуй, самым популярным устройством вывода информации для IBM-совместимых компьютеров является принтер. Все печатающие устройства (принтеры) можно подразделить на последовательные, строчные и страничные. Принадлежность принтера к той или иной из перечисленных групп зависит от того, формирует он на бумаге символ за символом, сразу всю строку или целую страницу.

Идея матричных печатающих устройств заключается в том, что все мыслимые (и немыслимые) знаки воспроизводятся ими из набора отдельных точек, наносимых на бумагу тем или иным способом. Все печатающие устройства (за исключением, пожалуй, страничных) по принципу действия могут быть ударными (impact) или безударными (non-impact).

Большинство принтеров, работающих (и продаваемых) сейчас с IBM-совместимыми компьютерами в нашей стране, могут быть причислены к группе последовательных, ударных матричных печатающих устройств:

вертикальный ряд (или два ряда) игл «вколачивает» краситель с ленты прямо в бумагу, формируя последовательно символ за символом. Такое засилье игольчатых вполне объясняется приемлемым качеством их печати, невысокой ценой расходных материалов (красящей ленты) и используемой бумаги. Кстати, обычно возможно использование как форматной, так и рулонной бумаги. Головка принтера может быть оснащена 9, 18 или 24 иглами. Существуют модели принтеров как с широкой (формат А3), так и с узкой (формат А4) кареткой. Высокое качество печати достигается в режимах NLQ (Near Letter Quality) для 9-игольчатых (почти машинописное) и LQ (Letter Quality) для 24-игольчатых принтеров. Как правило, современные принтеры оснащены резидентными или загружаемыми масштабируемыми шрифтами. Скорость печати для высокопроизводительных моделей может составлять до 380 знаков в секунду. Одна из причин выбора игольчатых принтеров заключается в том, что они могут оставлять отпечатки букв на бумаге, что важно при составлении контрактов или официальных писем. На рынке ударных матричных принтеров лидируют фирмы Epson, Star Micronics, Okidata. Объем продажи подобных устройств постоянно сужается за счет роста продажи струйных принтеров, которые относятся к безударным устройствам.

Обычно безударными принтерами называются такие устройства, у которых исполнительное устройство, печатающее изображение, не касается бумаги. Не требуется, конечно, пояснять, что безударные печатающие устройства работают практически бесшумно, что является одним из их несомненных преимуществ. Струйные чернильные принтеры относятся, как правило, к классу последовательных, матричных безударных печатающих устройств. Если продолжить уточнение признаков принадлежности печатающих устройств к отдельным группам, можно сказать, что последовательные безударные матричные струйные чернильные принтеры, в свою очередь, подразделяются на устройства непрерывного (continuous drop, continuous jet) и дискретного (drop-on-demand) действия. Последние в своей работе опять же могут использовать либо пузырьковую технологию (bubble-jet), либо пьезоэффект. Кстати, первопроходцами этих технологий стали фирмы Canon и Epson. Почти все современные устройства этого класса используют именно две последних технологии печати.

У чернильных устройств, как, впрочем, и у ударных матричных принтеров, печатающая головка движется только в горизонтальной плоскости, а бумага подается вертикально. Сопла (канальные отверстия) на печатающей головке, через которые разбрызгиваются чернила, соответствуют ударным иглам. Количество сопел у разных моделей принтеров, как правило, может варьироваться от 12 до 64. Размер каждого сопла существенно меньше диаметра иглы ударного матричного принтера, поэтому получаемое изображение (теоретически) должно быть четче. К сожалению, это не всегда

так, и очень многое зависит от качества используемой бумаги (все-таки чернила!).

При использовании метода bubble-jet в каждом сопле находится маленький нагревательный элемент (обычно это тонкопленочный резистор), который при пропускании тока за несколько микросекунд нагревается до температуры около 500°C и отдает выделяемое тепло непосредственно окружающим его чернилам. При резком нагревании образуется чернильный паровой пузырь, который старается вытолкнуть через выходное отверстие сопла необходимую порцию (каплю) жидких чернил. Поскольку при отключении тока тонкопленочный резистор также быстро остывает, паровой пузырь, уменьшаясь в размерах, «подсасывает» через входное отверстие сопла новую порцию чернил, которые занимают место «выстеленной» капли.

Как уже было сказано, второй метод для управления соплом использует пьезоэлектрический элемент. Как известно, обратный пьезоэффект заключается в деформации пьезокристалла под воздействием электрического поля. Изменение размеров пьезоэлемента, расположенного сбоку выходного отверстия сопла, приводит к выбрасыванию капли и приливу через входное отверстие новой порции чернил. При печати высокого качества скорость вывода не превосходит обычно 2–3 страниц в минуту (около 200 знаков в секунду). Печатающие головки некоторых моделей подобных устройств имеют до 64 сопел, через которые чернила в виде капель распыляются на поверхность бумаги. Это обеспечивает максимальную разрешающую способность – около 360 точек на дюйм. Тем не менее качество печати не всегда высокое. Как правило, струйные принтеры позволяют эмулировать работу наиболее популярных моделей ударных устройств и поддерживают соответствующее программное обеспечение.

Изображение в термопринтере создается путем сконцентрированного в нужном месте локального нагревания бумаги. Бумага в данном случае должна иметь очень тонкое термочувствительное покрытие с двумя отдельными бесцветными компонентами. При локальном нагревании цветоформирователь (первый компонент), смешиваясь с предварительно обесцвеченным красителем (второй компонент), образует видимое на бумаге пятнышко. В данном случае возможна печать различными цветами. Как правило, покрытие, обеспечивающее при печати изображение черного цвета, требует более высоких температур и давления печатающей головки. Но на бумаге с покрытием, обеспечивающим печать голубым цветом, хотя она и допускает обычно более высокие скорости вывода, полученное изображение со временем бледнеет и становится непригодным для практического использования.

Основными составными частями печатающей головки термопринтера являются несколько крошечных нагревательных элементов, которые расположены примерно так же, как иглы в обычном матричном ударном

принтере: один над другим в два ряда. Наиболее часто печатающие головки используют два ряда нагревательных элементов, по 12 в каждом, немного смещенных относительно друг друга. Нагрев элементов осуществляется путем пропускания через них электрического тока определенной величины. Поскольку сами термоэлементы очень маленькие, то и печатающая головка принтера, как правило, имеет небольшие габаритные размеры (всего несколько миллиметров в толщину).

Как у игольчатых и струйных принтеров, печатающая головка термопринтера позиционируется только в горизонтальном направлении, а подача бумаги осуществляется в вертикальном (последовательные принтеры). Не надо, видимо, пояснять, что термопринтеры относятся к группе матричных устройств. Поскольку, вообще говоря, между печатающей головкой и бумагой (подчеркиваю, самой бумагой) механический контакт отсутствует, то термопринтеры – это безударные устройства. К основным недостаткам термопечати следует отнести необходимость использования специальной (и, естественно, дорогой) бумаги.

Относительно недавно (по некоторым данным, в 1982 г.) появился новый класс термопринтеров, печатающих уже на обыкновенной бумаге. Такие устройства называются принтерами с термопереносом или термографическими печатающими устройствами с подачей красящего вещества. Как и обычные термопринтеры, они имеют печатающую головку с нагревательными элементами, правда, эти элементы стали еще меньше. Термопластичное красящее вещество, нанесенное на тонкой подложке (лента с красителем), попадает на бумагу именно в том месте, где элементами печатающей головки обеспечивается должный нагрев. Конструктивно такой способ печати достаточно прост, к тому же он обеспечивает практически бесшумную работу. Достаточно часто принтеры с термопереносом имеют печатающую головку шириной максимально допустимого размера бумаги, используемой в таком принтере (так называемые строчные или страничные устройства). Лента с красителем в этом случае, разумеется, также должна иметь соответствующую ширину. Понятно, что при такой ситуации отпадает потребность в механике, служащей для перемещения печатающей головки вдоль бумаги.

Еще один очень интересный метод печати используется в так называемых принтерах с термосублимацией. Хотя, возможно, термин «термосублимация» не очень удачен, но он достаточно четко поясняет, каким образом красящему веществу передается необходимая порция энергии сублимации. Напомним, что под сублимацией понимают переход вещества из твердого состояния в газообразное, минуя жидкообразную стадию (например, сублимируют при нагревании кристаллы йода).

Иначе этот процесс называют еще возгонкой.

Принтеры с термопереносом и термосублимацией красящего вещества также относятся к группе матричных безударных печатающих устройств. Понятно, что практически все преимущества матричных принтеров автоматически можно распространить и на эти типы принтеров. Как правило, не представляет проблемы создание практически любых типов шрифтов с различными атрибутами, нет проблем и при работе в графическом режиме. Кстати, достаточно просто реализуется здесь и негатив изображения (например, белое на черном). По четкости и резкости изображения некоторые модели таких принтеров могут поспорить с хорошими лазерными устройствами. Их разрешающая способность составляет обычно 300 точек на дюйм. По скорости печати принтеры с термопереносом, как правило, уступают современным моделям чернильных струйных печатающих устройств. Физические ограничения по скорости обусловлены в основном задержкой нагрева термоэлементов печатающей головки и процессом переноса красителя.

В принтерах с термосублимацией красителя имеется возможность точного определения необходимого количества красителя, переносимого на бумагу. Комбинацией цветов красителей можно подобрать практически любую цветовую палитру. Принтеры с термосублимацией выполнены обычно только как цветные, поскольку, пожалуй, ни один тип печатающих устройств не обеспечивает лучшей цветопередачи. Наиболее популярными моделями печатающих устройств с термопереносом красящего вещества являются портативные принтеры.

В лазерных принтерах используется электрографический принцип создания изображения – примерно такой же, как и в копировальных машинах. Этот процесс, в частности, включает в себя создание рельефа электростатического потенциала в слое полупроводника с последующей визуализацией этого рельефа. Собственно визуализация осуществляется с помощью частиц сухого порошка – тонера, наносимого на бумагу. Наиболее важными частями лазерного принтера можно считать фотопроводящий цилиндр (печатающий барабан), полупроводниковый лазер и прецизионную оптико-механическую систему, перемещающую луч.

Мощный полупроводниковый лазер генерирует тонкий световой луч, который, отражаясь от вращающегося зеркала, формирует электронное изображение на светочувствительном фотоприемном барабане. Барабану предварительно сообщается некий статический заряд. Для получения изображения лазер должен включаться и выключаться, что обеспечивается специальной управляющей электроникой принтера. Вращающееся зеркало служит для разворота луча лазера на новую строку, формируемую на поверхности печатающего барабана. Когда луч лазера попадает на предварительно заряженный барабан, заряд стекает с освещенной поверхности. Таким образом, освещаемые и неосвещаемые лазером участки барабана имеют

разные заряды. В зависимости от того, как (положительно или отрицательно) заряжены частицы порошкообразного тонера, они будут притягиваться и прилипать к барабану только в областях с разноименным зарядом. После формирования каждой строки специальный прецизионный шаговый двигатель поворачивает барабан так, чтобы можно было формировать следующую строку. Это смещение равняется разрешающей способности принтера и может составлять, например 300, 600 или 1200 точек на дюйм. Данный этап работы во многом напоминает построение изображения на экране монитора (растрирование).

Когда изображение на барабане построено и он покрыт тонером, подаваемый лист заряжается таким образом, чтобы тонер с барабана притягивался к бумаге. После этого изображение закрепляется на ней за счет нагрева частиц тонера до температуры плавления. Окончательную фиксацию изображения осуществляют специальные резиновые валики, прижимающие расплавленный тонер к бумаге.

На рынке лазерных принтеров можно выделить печатающие устройства малого быстродействия (скорость вывода 4-6 страниц в минуту), принтеры среднего быстродействия (7-11 страниц в минуту) и принтеры коллективного использования, так называемые «сетевые» принтеры (более 12 страниц в минуту). Для лазерных принтеров, работающих с бумагой формата А4, стандартом де-факто становится разрешающая способность 600 точек на дюйм. Принтеры, способные работать с бумагой формата А3, как правило, имеют разрешающую способность 1200 точек на дюйм и невысокую скорость вывода – 3-4 страницы в минуту.

К наиболее важным функциональным возможностям принтеров относятся такие, как поддержка технологии повышения разрешающей способности, наличие масштабируемых шрифтов (PostScript, TrueType), объем оперативной памяти и т.п. Безусловным лидером на рынке лазерных принтеров малого быстродействия является фирма Hewlett Packard.

Кроме лазерных принтеров, существуют так называемые принтеры LED (Light Emitting Diode), которые получили свое название из-за того, что полупроводниковый лазер в них был заменен «гребенкой» мельчайших светодиодов. Разумеется, в данном случае не требуется сложной оптической системы вращающихся зеркал и линз. Изображение одной строки на светочувствительном барабане формируется одновременно. Одним из лидеров на рынке LED-принтеров можно назвать фирму Okidata.

2.26 Сканер

Сканером называется устройство, позволяющее вводить в компьютер образы изображений, представленных в виде текста, рисунков, фотографий или

другой графической информации. В настоящее время известно два основных типа сканеров – ручной (hand-held) и настольный (desktop). Для того чтобы ввести в компьютер какой-либо документ с помощью этого устройства, надо без резких движений провести сканирующей головкой по соответствующему изображению. Ширина вводимого изображения для ручных сканеров не превышает обычно 4 дюймов (около 10 см), а длина, очевидно, ограничивается только используемым программным обеспечением. В некоторых моделях ручных сканеров, в угоду повышению разрешающей способности, уменьшают ширину вводимого изображения. Современные ручные сканеры могут обеспечивать автоматическую склейку вводимого изображения, т.е. формируют целое изображение из отдельно вводимых его частей.

Настольные сканеры называют и страничными, и планшетными, и даже автосканерами. Такие сканеры позволяют обычно вводить изображения размерами 8,5x11 или 8,5x14 дюймов. Существует три разновидности настольных сканеров: flatbed, sheet-fed и overhead.

Flatbed-сканеры – обычно достаточно дорогие устройства, но, пожалуй, и наиболее «способные». Внешне они чем-то могут напоминать копировальные машины – ксероксы. Для сканирования изображения необходимо открыть крышку сканера, положить сканируемый лист на стеклянную пластину изображением вниз, после чего закрыть крышку. Все дальнейшее управление процессом сканирования осуществляется с клавиатуры компьютера – при работе с одной из специальных программ, поставляемых вместе с таким сканером. Понятно, что рассмотренная конструкция сканера позволяет (подобно ксероксу) сканировать не только отдельные листы, но и страницы журнала или книги.

Работа sheet-fed-сканеров чем-то напоминает работу обыкновенной факс-машины. Отдельные листы документов протягиваются через такое устройство, при этом и осуществляется их сканирование. Понятно, конечно, что в этом случае копирование страниц книг и журналов просто невозможно. Рассматриваемые сканеры достаточно широко используются в областях, связанных с оптическим распознаванием символов (Optical Character Recognition, OCR). Для удобства работы sheet-fed-сканеры обычно оснащаются устройствами и для автоматической подачи страниц.

Третья разновидность настольных сканеров – overhead сканеры, которые больше всего напоминают несколько своеобразный overhead-проектор. Вводимый документ кладется на поверхность сканера изображением вверх, блок сканирования находится при этом также сверху.

Первые модели черно-белых сканеров могли работать только в двухуровневом (bilevel) режиме, воспринимая или черный, или белый цвет. Таким образом, сканироваться могли только либо штриховые рисунки

(например, чертежи), либо двухтоновые изображения. Хотя эти сканеры и не могли работать с действительными оттенками серого цвета, выход для сканирования полутоновых изображений такими сканерами был найден. Псевдополутоновой режим (dithering) работы такого сканера имитирует оттенки серого цвета, группируя несколько точек вводимого изображения в так называемые gray-scale-пиксели. Такие пиксели могут иметь размеры 2x2 (4 точки), 3x3 (9 точек) или 4x4 (16 точек) и т.д. Отношение количества черных точек к белым и определяет уровень серого цвета. Например, gray-scale-пиксель размером 4x4 позволяет воспроизводить 16 уровней серого (включая и полностью белый цвет). Не следует, правда, забывать, что разрешающая способность сканера при использовании gray-scale-пикселей снижается (в последнем случае в 4 раза).

Полутоновые сканеры используют максимальную разрешающую способность, как правило, только в двухуровневом режиме. Обычно такие сканеры поддерживают 16, 64 или 256 оттенков серого цвета для 4-, 6- или 8-разрядного кода, который ставится при этом в соответствие каждой точке изображения. Разрешающая способность сканера измеряется количеством различаемых точек на дюйм изображения – dpi (dot per inch). Если в первых моделях сканеров разрешающая способность была обычно 200-300 dpi, то в современных моделях это, как правило, 400, а то и 800 dpi. Обычно разрешение сканера может устанавливаться программным путем в процессе работы из ряда значений 75, 100, 150, 200, 300 и 400 dpi.

Надо сказать, что, благодаря операции интерполяции, выполняемой обычно программно, современные сканеры могут иметь разрешение 800 и даже 1600 dpi. В результате интерполяции на получаемом при сканировании изображении сглаживаются кривые линии и исчезают неровности диагональных линий. Для тех, кто не помнит, поясним. Интерполяция позволяет отыскивать значения промежуточных величин по уже известным значениям. Например, в результате сканирования один из пикселей имеет значение уровня серого цвета 48, а соседний с ним – 76. Использование простейшей линейной интерполяции позволяет сделать предположение о том, что значение уровня серого цвета для промежуточного пикселя могло бы быть равно $62 [(48+76)/2]$. Если вставить все оценочные значения пикселей в файл отсканированного изображения, то разрешающая способность сканера как бы удвоится, т.е. вместо обычных 400 dpi станет равной 800 dpi.

В настоящее время существует несколько технологий для получения цветных сканируемых изображений. Поясним принцип работы цветного сканера в общем. Сканируемое изображение освещается уже не белым светом, а через вращающийся RGB-светофильтр. Для каждого из основных цветов (красного, зеленого и синего) последовательность операций практически не отличается от последовательности операций при сканировании черно-белого

изображения. Исключение составляет, пожалуй, только этап предварительной обработки и гамма коррекции цветов, перед тем как информация передается в компьютер. Понятно, что этот этап является общим для всех цветных сканеров.

В результате трех проходов сканирования получается файл, содержащий образ изображения в трех основных цветах – RGB (образ композитного сигнала). Если используется 8-разрядный АЦП, который поддерживает 256 оттенков одного цвета, то каждой точке изображения ставится в соответствие один из 16,7 миллиона возможных цветов (24 разряда).

Надо отметить, что наиболее существенным недостатком описанного выше метода является увеличение времени сканирования в три раза. Проблему может представлять также «выравнивание» пикселей при каждом из трех проходов, так как в противном случае возможно размывание оттенков и «смазывание» цветов.

В сканерах известных японских фирм Epson и Sharp, как правило, вместо одного источника света используются три – для каждого цвета отдельно. Это позволяет сканировать изображение всего за один проход и исключает неправильное «выравнивание» пикселей. Сложности этого метода заключаются обычно в подборе источников света со стабильными характеристиками.

Другая японская фирма Seiko Instruments разработала цветной flatbed-сканер SpectraPoint, в котором элементы ПЗС были заменены фототранзисторами. На ширине 8,5 дюймов размещено 10 200 фототранзисторов, которые расположены в три колонки, по 3400 в каждой. Три цветных фильтра (RGB) расположены так, что каждая колонка фототранзисторов воспринимает только один основной цвет. Высокая плотность интегральных фототранзисторов позволяет достигать хорошей разрешающей способности – 400 dpi – без использования редуцирующей линзы. Применение такой технологии делает сканер SpectraPoint, пожалуй, самым быстрым из всех существующих.

Как правило, образы изображений в компьютере хранятся в графических файлах формата TIFF (Tagged Image File Format) или в файлах не менее популярного формата PCX. Надо иметь в виду, что при сканировании изображений файлы получаются достаточно громоздкими. Так, при вводе полутонового черно-белого изображения размером 8x10 дюймов с 256 оттенками серого цвета и при разрешающей способности 400 dpi будет создан файл размером более 12 Мбайт. Уменьшение объема хранимой информации осуществляется обычно «сжатием» таких графических файлов специальными программами-архиваторами (RAR) или в специальных графических форматах (JPG).

2.27 Мышь

Мыши и трекболы являются манипуляторами ввода информации в компьютер. Разумеется, полностью заменить клавиатуру они не могут. Первую компьютерную мышь создал молодой талантливый ученый Дуглас Энджелбарт. Произошло это событие в 1963 г. в Стенфордском исследовательском центре. Первый трекбол (trackball) был создан значительно позже фирмой Logitech. За прошедшие годы мышь значительно изменилась, однако неизменно элегантным и эргономичным остается ее «мышиный» дизайн.

Надо заметить, что, хотя до сих пор никаких официальных стандартов на компьютерные мыши не разработано, де-факто на рынке существуют три благородных породы мыши: Microsoft Mouse, Logitech Mouse и Mouse System. Все остальные фирмы всячески стремятся обеспечить совместимость своих «подопечных» с одной или сразу несколькими из породистых мышей.

Не секрет, что своей популярностью мышь обязана, главным образом, громадному спросу на прикладные графические программные системы, а также широкому распространению графического интерфейса пользователя, чему в немалой степени способствует экспансия Windows. Мышь делает очень удобным манипулирование такими широко распространенными в графических пакетах объектами, как окна, меню, кнопки, пиктограммы и т.д. При работе в такой среде мышь, по сравнению с клавиатурой, существенно облегчает работу, делая ее более простой и эффективной.

Как известно, «первобытная» мышь каталась на двух колесиках, которые были связаны с осями переменных резисторов. Перемещение такой мыши было прямо пропорционально изменению сопротивления переменных резисторов. В дальнейшем конструкция мыши претерпела значительные изменения. Колесики (ролики) были перенесены внутрь корпуса, а с поверхностью стал соприкасаться тяжелый обрешиненный или просто сделанный из твердой резины шарик сравнительно большого диаметра. Оси вращения роликов взаимно перпендикулярны. Ролики, прижатые к поверхности шарика, установлены на оси с датчиками, с помощью которых и определяются направление и скорость перемещения мыши. Некоторое время в качестве датчиков использовались непроводящие диски с нанесенными печатным способом контактами, которые поочередно могли соприкасаться с одним неподвижным контактом. При таком внутреннем устройстве мышь была практически полностью «механической».

Но, как известно, механика – вещь не очень надежная, поэтому впоследствии (да и до сих пор) подавляющее число компьютерных мышей стали использовать оптико-механический принцип кодирования перемещения. На смену механическим шифраторам пришли оптопары: светодиод-фотодиод, или фоторезистор, а в некоторых случаях даже фототранзистор. Такая пара располагается по разные стороны от диска с прорезями. Порядок, в котором

освещаются фоточувствительные элементы, определяет направление перемещения мыши, а частота приходящих от них импульсов – скорость.

Сегодня не все мыши используют одинаковый способ перемещения. Например мышь Honeywell имеет уникальный запатентованный дизайн. Вместо обычного шара эта мышь имеет две «ножки». Эти «ножки» являются частью X-Y-механизма оптико-механического шифратора.

Напомним, что наиболее распространенные мыши имеют либо две, либо три кнопки. Мыши от Microsoft, начиная со своей зеленоглазой прародительницы (первая мышь имела две кнопки зеленого цвета), имеют только две кнопки, а вот мыши Mouse System – три. Стоит, правда, отметить, что большинство прикладных программных систем ориентировано, вообще говоря, на некую однокнопочную мышь, поскольку оставшиеся кнопки либо вообще не используются, либо эмулируют определенные клавиши клавиатуры.

В настоящее время можно выделить три различных способа подключения мыши (мы не будем рассматривать «бесхвостых» радио- и иных редких мышей). Самым распространенным для настольных IBMсовместимых компьютеров является подключение через последовательный порт (интерфейс RS232). Таким образом происходит подключение подавляющего количества мышей различных фирм-производителей. Менее распространены так называемые мыши с шинным интерфейсом (bus-mouse) от фирмы Microsoft, которые для своего подключения требуют специальную плату, то есть особый «мышиный» порт. Сразу заметим, что мыши этого вида представляют у нас в стране некий раритет. Ну и третьей разновидностью можно считать мышей «в стиле PS/2», которые используются преимущественно в компьютерах аналогичной серии. Таким образом, наибольший интерес для массового пользователя представляют только те мыши, которые подключаются через последовательный порт компьютера.

Физически каждая такая мышь на своем «хвосте» имеет разъем типа DB-9. В некоторых случаях в комплекте с мышью поставляется и переходное устройство с DB-9 на DB25, поскольку на некоторых компьютерах последовательный порт может иметь именно такой разъем.

Неотъемлемой частью комплекта поставки мыши является дискета, на которой записаны программы установки и тестирования, а также драйвер (обычно в виде файлов типа COM и SYS). Часто в комплекте поставляются программы – генераторы меню. Они позволяют пользователю создавать на экране одно или несколько меню и «начинять» их пункты управления различными командами. Прикладная программа может получить указанную мышью команду точно так же, как если бы эта команда была введена с клавиатуры. Некоторые «благородные» мыши комплектуются также программами рисования типа упрощенной PaintBrush.

Мыши от Microsoft, имеющие последовательный интерфейс, и, разумеется, им подобные для передачи процессору используют 3-байтовый формат, содержащий информацию о позиции мыши и состоянии кнопок.

Не все мыши используют формат передачи, предложенный фирмой Microsoft. Например, трёхклавишная мышь Mouse System и совместимые с ней системы передают данные в 5-байтовом формате. Разница в форматах приводит к тому, что драйвер (программа управления) одной мыши не работает с другой.

2.28 Модемы и факс-модемы

Без модема немислима система электронных коммуникаций. Это позволяет вам включиться в увлекательный, а сегодня уже и просто жизненно необходимый мир информационных потоков, электронных баз данных, электронной почты, электронных справочников, электронных досок объявлений и т.д.

Если вы хотите оперативно передать файл (с программой, картинкой или сообщением) вашему сотруднику, то с помощью модема это делается элементарно. Используя специальную информационную программу, вы звоните по телефону своему партнеру, модемы на ваших компьютерах «договариваются» друг с другом об установлении соединения, и после этого, используя специальный протокол передачи данных, вы передаете файл на удаленный компьютер.

Вторая услуга (необходимо отметить, что она является бесплатной, конечно, если не учитывать плату за занятую телефонную линию) – обмен информацией с BBS (Buletен Board System) электронной доски объявлений. Сейчас на территории бывшего СССР имеется уже несколько тысяч BBS. Физически BBS представляют собой ПК со специальным программным обеспечением, который через модем подключен к обычной телефонной сети. Программное обеспечение предоставляет возможность любому дозвонившемуся зарегистрироваться в BBS и работать в ней.

BBS могут быть связаны между собой. При этом есть возможность посылать электронные сообщения (и даже бинарные файлы) пользователю, зарегистрированному на другом узле сети BBS. Система BBS имеет свою конституцию – правила поведения членов системы. Узел BBS содержит большое количество файлов, разбитых по темам. Работая в системе, вы можете просмотреть список файлов и «перекачать» на свой компьютер файлы, которые вас заинтересовали.

Наряду с этим, стремительно развивается другое направление компьютерной технологии – общение компьютеров между собой через мировые сети с помощью различных коммуникаций.

Крупнейшей глобальной информационной системой является сеть Internet, которая в действительности не имеет определенной организационной структуры и представляет собой некий конгломерат самостоятельных компьютерных сетей, созданных усилиями различных правительств, научных, коммерческих и некоммерческих организаций.

За 1996 год в Москве количество WWW (World Wide Web – Всемирная паутина) серверов (узлов) возросло приблизительно с 30 на начало года до 500 к концу года.

Основной формой сетевых коммуникаций в настоящее время является электронная почта (E-mail). Каждый пользователь, подключенный к E-mail, имеет свой электронный адрес, например: grigri@fines.ru (адрес преподавателя нашего университета). С помощью специального программного обеспечения ваше послание будет отправлено в любую точку мира, где находится получатель, электронный адрес которого вы указали. Эта операция занимает минуты, даже если адресат находится на другом континенте.

Однако возможности Internet намного шире, чем простой обмен сообщениями между отдельными людьми. С помощью этой сети Вы можете получить доступ к океану информации, циркулирующей по планете, объем которой невозможно ни перечислить, ни классифицировать.

Когда компьютер используется для обмена информацией по телефонной сети, необходимо устройство, которое может принять сигнал из телефонной сети и преобразовать его в цифровую информацию. На выходе этого устройства информация подвергается МОДуляции, а на входе ДЕМодуляции, отсюда и название МОДЕМ. Назначение модема заключается в замене сигнала, поступающего из компьютера (сочетание нулей и единиц), электрическим сигналом с частотой, соответствующей рабочему диапазону телефонной линии.

Прежде чем перейти к описанию факс-модемов для компьютеров, необходимо пояснить, что такое факс или факсимильный аппарат в частности.

Факсимильный аппарат (далее просто факс или телефакс) служит для передачи или приема графических и текстовых черно-белых изображений по телефонным линиям. Выглядит как большой телефонный аппарат, и им можно пользоваться как обычным телефоном. Но, в отличие от обычного телефона, в него устанавливается рулон термобумаги (для распечатки получаемого изображения), есть также автоответчик. Если вы хотите пользоваться факсом одновременно и как телефоном, и как автоответчиком, и для приема/отправки факсовых изображений, то вам надо установить автоматическое срабатывание автоответчика и факса на пятый-шестой звонок телефона. Если вы успеете поднять трубку раньше, то будете разговаривать как по обычному телефону (впрочем, ничто не мешает вам при этом в любой момент включить факс на прием или отправку изображений, нажав соответствующую кнопку). Стандартный пример такого рода факса – факс Panasonic KX-F110B.

Факсимильные аппараты, в основном, используются организациями для отправки и приема счетов, документов. Те же самые функции позволит выполнить и факс-модемная плата совместно с компьютером (обычный телефон при этом включается параллельно). Но такой вариант факсового устройства менее удобен. Во-первых, для автоматического приема компьютер должен быть всегда включен, что не всегда приемлемо. Во-вторых, если вы даже находитесь у телефона, а компьютер не включен, вам придется его включать и запускать соответствующую программу. В-третьих, для отправки изображения его необходимо сперва перевести в компьютер, а для кого нужен сканер. Правда, вы можете просто перепечатать текст и отправить его, но при этом подписи или печать (организации) передать невозможно.

Из всего этого следует вывод: факс-модемную плату (в качестве факса) можно предпочесть факсовому аппарату, только если вы испытываете финансовые затруднения (ее стоимость от \$50) или собираетесь пользоваться ею, в основном, на прием, или если вы вообще очень редко будете ею пользоваться (по части факса).

По части же модема факс-модемная плата ничем не отличается от модемной платы. И также может быть выполнена в виде внутреннего (вставляется в любой слот на «материнской» плате) или в виде отдельного блока (соединяется по RS-232). Факс-модемная плата обычно стоит чуть-чуть дороже модемной.

Все модемы и факсы начинают передавать информацию в телефонную линию с максимальной скоростью, на которую они рассчитаны. Если в линии большие помехи или у другого устройства ниже максимальная скорость, то происходит автоматический переход на более низкую скорость и так далее.

Модемы хотя и предназначены для передачи цифробуквенных сообщений, но с их помощью можно передать и графические изображения. Например, передать игровую программу. С одной стороны, она будет передаваться в виде цифробуквенной информации (текст программы), с другой стороны, будут переданы и графические картинки, которые зашифрованы в этой цифробуквенной информации. Однако оказывается, что для цифробуквенной информации существуют свои более плотные методы сжатия, а для графического изображены – свои эффективные методы сжатия (которые и применяются в факсовых аппаратах). Поэтому более быстро передавать графические изображения можно только с помощью факсимильной связи.

2.29 Компьютерные сети

Под локальной вычислительной сетью (ЛВС) обычно понимают коммуникационную систему, состоящую из двух или более компьютеров и

включающую в себя специальное аппаратное и программное обеспечение. В такой системе любое из подключенных устройств может использовать ЛВС для передачи или получения информации. Небольшие ЛВС, или LAN (Local Area Network), обычно охватывают какое-либо отделение предприятия и не выходят за пределы одного здания. Достаточно эффективно можно использовать ЛВС и в малом офисе. Стоит, видимо, перечислить ряд веских причин для объединения отдельных персональных компьютеров в ЛВС.

Во-первых, совместный доступ к внешним высококачественным устройствам снижает затраты на каждого отдельного пользователя. К тому же правильная реализация совместного доступа повышает надежность всей системы, так как при отказе одного устройства его функции может взять на себя другое. Поскольку дисковая память является достаточно дорогим ресурсом, можно, например, организовать коллективный доступ к дискам одного или нескольких компьютеров, причем в этом случае на ряде компьютеров дисковые подсистемы могут вообще отсутствовать.

Кроме того, средства сети легко обеспечивают доступ к одному (дорогому) лазерному принтеру от нескольких компьютеров.

Во-вторых, кроме совместной эксплуатации дорогостоящих периферийных устройств, ЛВС позволяет аналогично использовать сетевые версии прикладного программного обеспечения. Затраты на покупку и поддержку новых программных пакетов в этом случае также сокращаются.

В-третьих, ЛВС обеспечивают новые формы взаимодействия пользователей в одном коллективе, например при работе над общим проектом. Особое значение имеет организация распределенной обработки данных. В локальной сети можно организовать доступ всех пользователей к большой базе данных, расположенной на одном мощном компьютере. Стоит отметить, что в случае централизованного хранения информации значительно упрощаются процессы обеспечения.

Для подготовки проекта установки и запуска локальной сети для вашего офиса лучше всего обратиться в специализированную фирму, благо их сегодня существует достаточно много. Тем не менее надо всетаки представлять проблему хотя бы в общих чертах.

Довольно часто компьютер, подключенный к локальной сети, называют рабочей станцией (workstation), однако не следует путать этот термин с другим, который обозначает мощную высокопроизводительную машину, предназначенную для решения сложных задач, например графических САПР и т.д. Впрочем, мощный компьютер также может быть включен в ЛВС, но в сети для малого офиса его наличие маловероятно. Кстати, в случае коллективного доступа к дискам одного или нескольких компьютеров речь может идти даже о бездисковых рабочих станциях.

Каждый компьютер в ЛВС должен иметь встраиваемую плату, которая позволяет ему взаимодействовать с другими устройствами данной сети. Такая плата называется сетевым адаптером, или NIC (Network Interface Card). Небольшие низкоскоростные сети, выполненные на базе стандартных последовательных портов IBM PC-совместимых компьютеров, называют ЛВС «нулевым слотом» (zero slot LAN).

По определению, сервер – это некоторое обслуживающее устройство, которое в ЛВС выполняет, например, роль управляющего центра и концентратора данных. Под сервером, вообще говоря, понимается комбинация аппаратных и программных средств, которая служит для управления сетевыми ресурсами общего доступа. Среди ЛВС на базе персональных компьютеров различают сети с выделенным сервером (централизованным управлением) и так называемые одноранговые сети (peer-to-peer). В последних при совместном использовании информации каждая станция может выступать и как клиент, и как сервер. Таким образом, функции управления сетью передаются от одной станции к другой. Разумеется, в одноранговых сетях пользователям также предоставляется возможность совместного использования принтеров, накопителей, модемов и других ресурсов ЛВС.

Оба типа сетей имеют свои достоинства и недостатки. Одноранговые ЛВС достаточно дешевы и просты в обслуживании, однако не могут обеспечить должной защиты информации при большом размере сети. ЛВС с выделенным сервером имеют хорошие возможности для расширения, но требуют постоянного квалифицированного обслуживания. Существует большое количество сетевых операционных систем (Network Operation System, NOS), рассчитанных на ЛВС с централизованным управлением (Novell Netware, Microsoft LAN Manager), и программного обеспечения для одноранговых сетей (Artisoft LANtastic, Novell NetWare Lite, Microsoft Windows for Workgroups).

Способ объединения компьютеров в сети между собой называют топологией. Классическими топологиями ЛВС являются «звезда» («дерево»), «общая шина» и «кольцо». В случае использования топологии типа «звезда» компьютеры объединяются посредством специального устройства, называемого концентратором, или хабом (hub). Топология «общая шина» предполагает применение одного кабеля, к которому подключаются все компьютеры данной сети. Для топологии типа «кольцо» данные передаются от одного компьютера к другому как бы по эстафете.

2.30 Мультимедиа

Развитие интерактивных компьютерных систем обеспечивает работу со всеми этими видами информации в диалоговом режиме. Такие системы

называются латинским словом мультимедиа (multimedia), т.е. системы со многими средами, со многими видами информации.

Число доступных на CD-ROM прикладных программ достигло ныне критической массы. Сегодня практически каждый компьютер продается вместе со встроенным лазерным дисководом, а те, кто купил ПК ранее, покупают CD-ROM и самостоятельно его устанавливают.

Что же нам может дать применение лазерных дисков? Прежде всего это качественно новые подходы к подаче информации (особенно это относится к всевозможным справочникам, словарям, энциклопедиям).

При работе с энциклопедиями часто приходится разыскивать одну за другой взаимосвязанные статьи, следуя указаниям «см. также...», в результате на столе собирается гора тяжелых томов. Но если вся энциклопедия умещается на компакт-диске, она становится гипермедиадokumentом и читатель может изучать тот или иной вопрос, просто нажимая кнопку «мыши», а не перелистывая страницы. Более того, поскольку такой документ содержит, помимо текста, рисунки, карты, графики, звуковые записи, анимацию и видеоролики, среда CD-ROM позволяет превратить старую добрую энциклопедию в нечто совершенно новое – интерактивный обучающий мультимедиацентр.

Особое преимущество системы мультимедиа имеют там, где требуется поиск, в весьма ограниченные сроки, специальной информации среди огромного объема однотипных данных, например во врачебной практике.

Особую популярность приобрели на Западе обучающие системы. На дисках записывается курс обучения по одному из предметов. Звучат пояснения учителя, высвечиваются тексты, графики, цветные иллюстрации, демонстрируется динамика проведения опытов. При изучении какой-либо темы обучающийся может вызывать на свой компьютерный монитор любой учебный материал, скажем тот, который он хуже знает или который хотел бы изучить глубже. Это позволяет использовать учебное время максимально продуктивно и удовлетворить индивидуальные запросы каждого обучающегося.

Большими тиражами выходят каталоги о произведениях искусства, хранящихся в различных музеях и галереях, в том числе в их запасниках, с информацией об авторах. Можно назвать десятки областей деятельности, где компакт-диски нашли применение: это различные отрасли науки и техники, налоговые процедуры, законодательство, стандарты и многое другое.

Возможность размещения на одном компакт-диске технических руководств, конструкторской документации или архивных материалов позволяет тонущим в бумажном потоке компаниям выйти на передовые рубежи информации. Поэтому нет ничего удивительного в том, что CDROM быстро приобретает популярность в мире бизнеса.

В последнее время многие зарубежные фирмы выпускают игровые программы на лазерных дисках. Сложные красочные игры занимают, как правило, объем в несколько десятков Мбайт. Это значит, что на средний по объему винчестер можно записать в лучшем случае 5-6 таких игр. А на одном лазерном диске их поместится гораздо больше.

Применение технологии CD-ROM сегодня наиболее ярко отразилось на индустрии компьютерных игр. К настоящему времени уже существуют сотни игр на этих носителях. Среди игр, распространяемых на CD-ROM, можно четко выделить три направления: игры, просто перенесенные на другой носитель (т.н. Shovelware), игры с расширенными возможностями и оригинальные игры, созданные специально для CD.

Технология CD-ROM пока еще используется не на полную мощность: в основном CD-версии игр отличаются от оригинальных введением музыки и голосового сопровождения, либо улучшением качества графики. При этом игровой компонент как таковой практически не меняется: приключенческие игры как были линейными (то есть вы не можете сделать следующего шага, не завершив текущего), так и остаются таковыми даже на CD. Хотя, казалось бы, наличие большого объема для хранения данных позволяет делать игры с альтернативными сценариями: сделал это – попал туда, сделал то – попал куда-то еще, чуть задержался – и сюжет начал развиваться по-другому, и т.п. Очевидно, что именно с появлением специальных версий, рассчитанных только на CDROM, мы сможем в полной мере говорить о новом поколении компьютерных игр.

2.31 Другие периферийные устройства

1. Устройства защиты

Самую простейшую защиту по питанию обычно обеспечивают так называемые ограничители перенапряжений. Эти устройства предохраняют нагрузку от различного рода выбросов и всплесков питающего напряжения электросети, а также радиочастотных шумов. Некоторые из таких приборов гарантируют максимальный рабочий ток до 13 А (при напряжении 220–240 В). Примерами подобных устройств могут быть модели SurgeArrest фирмы American Power Conversion (APC), Isobar фирмы TrippLite, SpikeFree фирмы Best Power Technology.

Более высокий уровень защиты обеспечивают устройства нормализации, которые надежно «очищают» питающее напряжение от всевозможных шумов и позволяют регулировать его в достаточно широком диапазоне. Некоторые модели таких устройств могут предотвращать даже кратковременные провалы в питающем напряжении. Мощность нагрузки, подключаемой к таким

устройствам, может варьироваться (в зависимости от модели) от 250 ВА до 2кВА.

2. Бесперебойные источники питания

Обеспечить работу нагрузки при полном отключении электропитания (blackout) могут только устройства, называемые UPS (Uninterruptible Power Supply), или БИП (Бесперебойные источники питания). Функционально такой прибор практически всегда состоит из устройства подавления помех, зарядного устройства, батареи аккумуляторов и преобразователя напряжения (постоянное-переменное). Все предлагаемые в настоящее время БИП можно условно подразделить на несколько групп.

К самой немногочисленной группе относятся так называемые встраиваемые (internal) UPS. Встраиваемые БИП – это самый дешевый по стоимости и самый простой тип бесперебойных источников питания. Конструктивно эти устройства выглядят как отдельная плата расширения, вставляемая в соответствующий разъем на системной плате компьютера. В БИП этого типа отсутствует преобразователь напряжения, так как цепи питания подключены непосредственно к батарее аккумуляторов, которая и выдает необходимые уровни постоянного напряжения. Сами же аккумуляторы подзаряжаются, конечно, от электросети. Все это является несомненным преимуществом таких БИП и сказывается на их невысокой стоимости. Однако заметим, что подобные БИП не могут защитить от неприятностей в сети электропитания ни монитор, ни внешние накопители, ни другие периферийные устройства. Установка встраиваемых БИП не очень сложна, но требует внимания и аккуратности. В качестве примера подобных устройств можно привести модели AccuCard и Internal Power System фирм Emerson Computer Power и Powercard.

Наиболее многочисленные группы БИП представляют устройства, имеющие топологию on-line (постоянно включенные) и off-line или standby (резервные). БИП, относящиеся к последней группе, в свою очередь, могут быть поделены на две подгруппы: standby hybrid UPS и standby-ferro UPS (гибридные и феррорезонансные). Подгруппа устройств, выполненных по топологии line-interactive (интерактивные БИП), выглядит несколько обособленно, хотя чаще всего подобные устройства относят к типу standby (или hybrid) UPS.

Как уже говорилось, в состав БИП обычно входят стабилизатор (устройство подавления перенапряжения и фильтр), зарядное устройство, аккумуляторы (возможно кислотные), преобразователь постоянного напряжения в переменное (DC/AC) и переключатель выходного напряжения.

Постоянно включенные БИП (работающие в режиме on-line) обеспечивают энергоснабжение подключенных устройств от батареи аккумуляторов через преобразователь напряжения независимо от состояния электросети, в то время

как резервные UPS переходят на такой режим работы только при полном отключении внешнего питающего напряжения.

Большинство источников бесперебойного питания могут автономно работать от 5 до 20 минут. Они рассчитаны в основном на то, что вы немедленно сохраните свою информацию и выключите компьютер. Запаситесь достаточным терпением и ждите, пока снова появится электричество. Самые современные и очень дорогие источники бесперебойного питания могут работать часами и даже сутками.

Тема 3 Операционные системы

3.1 Уровни компьютерных систем

Работая с компьютером на бытовом уровне, мы часто не задумываемся, что же скрывается за встречающим нас дружелюбным интерфейсом, а ведь современная компьютерная система состоит из одного или нескольких процессоров, оперативной памяти, дисков, клавиатуры, монитора, принтеров, сетевого интерфейса и других устройств, то есть является сложной комплексной системой. Написание программ, которые следят за всеми компонентами, корректно используют их и при этом работают оптимально, представляет собой крайне трудную задачу. По этой причине компьютеры оснащаются специальным уровнем программного обеспечения, называемым операционной системой. Операционная система отвечает за управление всеми перечисленными устройствами и обеспечивает пользователя имеющими простой, доступный интерфейс программами для работы с аппаратурой.

Выделяют пять основных уровней компьютерных систем (КС), среди которых:

1. Самый нижний уровень содержит физические устройства, состоящие из интегральных микросхем, проводников, источников питания, электронно-лучевых трубок и т.п. То, как они устроены и как работают, относится к сфере деятельности инженеров, специалистов по электронике.

2. Выше расположен микроархитектурный уровень, на котором физические устройства рассматриваются с точки зрения функциональных единиц. Обычно на этом уровне находятся внутренние регистры центрального процессора (CPU – Central Processing Unit) и арифметикологическое устройство. На каждом такте процессора из регистра выбирается один или два операнда, которые обрабатываются в арифметикологическом устройстве (например, действием операции сложения или логического И). Результат сохраняется в одном или нескольких регистрах. В некоторых машинах операции над данными контролируются программными приложениями,

которые называются микропрограммами, в других такой контроль выполняется напрямую аппаратными цепями.

В микропрограммах используется:

- обработка прерываний,
- управление различными типами структур данных,
- примитивы, синхронизация, координирующий доступ к общим данным,
- переключение контекста,
- последовательности вызова из процедур и возврата в них.

3. Определенная система команд передается по маршруту передачи данных. Некоторые команды могут быть выполнены за один цикл передачи данных, другие требуют нескольких циклов. Такие команды могут использовать регистры или другие возможности аппаратуры. Команды, видимые для работающего на ассемблере программиста, формируют уровень ISA (Instruction Set Architecture – архитектура системы команд), часто называемый машинным языком.

Обычно машинный язык содержит от 50 до 300 команд, служащих преимущественно для перемещения данных по компьютеру, выполнения арифметических операций и сравнения величин. Управление устройствами на этом уровне осуществляется с помощью загрузки определенных величин в специальные регистры устройств.

Примером работы на таком уровне может служить обработка команды чтения с диска. Для этого нужно выполнить запись в его регистры: адреса места на диске, адреса в основной памяти, число байтов для чтения и еще множество параметров. Кроме того, нужно учитывать статус операции, возвращаемый диском.

4. Операционная система предназначена для того, чтобы скрыть от пользователя и даже от прикладного программиста все эти сложности. Она состоит из уровня программного обеспечения, который частично избавляет от необходимости общения с аппаратурой напрямую, вместо этого предоставляя программисту более удобную систему команд. В продолжение предыдущего примера, можно сказать, что чтение блока с диска из файла в этом случае представляется ему намного более простым, чем когда нужно заботиться о перемещении головок диска, ждать, пока они установятся на нужное место и т.д.

5. Над операционной системой на нашем рисунке расположены остальные системные программы. Здесь находятся интерпретатор команд (оболочка), системы окон, компиляторы, редакторы и т.д. Важно понимать, что подобные программы не являются частью операционной системы, хотя обычно поставщики компьютеров устанавливают их на машины. Это очень важное замечание. Под

операционной системой обычно понимается то программное обеспечение, которое запускается в режиме ядра или, как его еще называют, режиме супервизора. Она защищена от вмешательства пользователя с помощью аппаратных средств. Компиляторы и редакторы запускаются в пользовательском режиме. Если пользователю не нравится какой-либо компилятор, он при желании может написать свой собственный, но он не может написать собственный обработчик прерываний системных часов, являющийся частью операционной системы и обычно защищенный аппаратно от попыток его модифицировать.

Существуют системы, в которых это различие размыто. К ним относятся встроенные системы, они могут не иметь режима ядра, или интерпретируемые системы, подобные основанным на Java операционным системам, в которых для разделения компонентов используется интерпретация, а не оборудование. Но в традиционных компьютерах операционная система представляет собой набор программ, запускающихся в режиме ядра.

Во многих системах есть программы, которые работают в пользовательском режиме, но помогают операционной системе или выполняют специализированные функции. Например, часто встречаются программы, позволяющие пользователям изменять свои пароли. Они не являются частью операционной системы и запускаются не в режиме ядра, но выполняемые ими функции влияют на работу системы, и такие программы должны быть определенным способом защищены от воздействия пользователя.

6. Над системными программами расположены прикладные программы. Обычно они покупаются или пишутся пользователем для решения собственных проблем – обработки текста, электронных таблиц, технических расчетов или сохранения информации в базе данных.

Первые три уровня можно объединить под общим названием «технический уровень». Четвертый и пятый можно условно назвать уровень «системных программ». Последний, шестой, можно назвать уровнем «прикладных программ».

Представив компьютерную систему таким образом, мы готовы привести универсальное определение операционной системы.

Операционная система (ОС) – набор программ, как обычных так и микропрограмм, которые обеспечивают возможность использования аппаратуры компьютера, при этом аппаратура компьютера представляет собой «сырой» материал, а задача ОС заключается в том, чтобы сделать аппаратуру доступной и, по возможности, удобной для пользователя.

Исходя из задач ОС, неотделимо связанных с данным определением, можно выделить основные функции ОС, среди которых:

- определение интерфейса пользователя,
- обеспечение разделения аппаратных ресурсов между пользователями,
- предоставление возможности работы с общими данными,
- планирование доступа пользователей к общим ресурсам,
- обеспечение эффективного выполнения операций ввода/вывода,
- осуществление восстановления информации и вычислительного процесса в случае ошибок.

Необходимо заметить, что в распоряжение ОС предоставляются как пассивные, так и активные ресурсы. Поэтому классификация ресурсов ОС видится следующим образом:

Пассивные ресурсы – управляемые ресурсы:

- процессоры,
- память,
- устройства ввода/вывода, □ данные.

Активные ресурсы – управляющие ресурсы:

- операторы ЭВМ,
- различные программисты (прикладные, системные), □ административный персонал, □ программы пользователя.

Можно сказать, что ОС, основываясь на потребностях и возможностях своего активного ресурса и с учетом предоставленного в распоряжение пассивного ресурса, выполняет поставленные перед ней задачи посредника.

Следующим шагом в рассмотрении ОС можно считать приведение классификации ОС с различных точек зрения.

3.2 Особенности алгоритмов управления ресурсами

От эффективности алгоритмов управления пассивными ресурсами компьютера во многом зависит эффективность всей ОС в целом. Поэтому, характеризуя ОС, часто приводят важнейшие особенности реализации функций ОС по управлению процессорами, памятью, внешними устройствами. Так, в зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят на многозадачные и однозадачные, многопользовательские и однопользовательские, на многопроцессорные и однопроцессорные системы.

1. Поддержка многозадачности По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

- однозадачные (например, MS-DOS, MSX) и □ многозадачные (ОС ЕС, OS/2, UNIX, Windows 95).

Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс

взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Многозадачные ОС, кроме вышеперечисленных функций, управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства.

Для полноты рассмотрения данного класса ОС необходимо упомянуть, что существует вытесняющая и невытесняющая многозадачность.

Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами во многом определяет специфику ОС. Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов: □ невытесняющая многозадачность (NetWare, Windows 3.x);

□ вытесняющая многозадачность (Windows NT, OS/2, UNIX).

Основным различием между вытесняющим и невытесняющим вариантами многозадачности является степень централизации механизма планирования процессов. В первом случае механизм планирования процессов целиком сосредоточен в операционной системе, а во втором – распределен между системой и прикладными программами. При невытесняющей многозадачности активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление операционной системе для того, чтобы та выбрала из очереди другой готовый к выполнению процесс. При вытесняющей многозадачности решение о переключении процессора с одного процесса на другой принимается операционной системой, а не самим активным процессом.

Поддержка многопользовательского режима

По числу одновременно работающих пользователей ОС делятся на: □ однопользовательские (MS-DOS, Windows 3.x, ранние версии

OS/2); □ многопользовательские (UNIX, Windows NT).

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

Многопроцессорная обработка

Другим важным свойством ОС является отсутствие или наличие в ней средств поддержки многопроцессорной обработки – мультипроцессирование.

Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами.

Функции поддержки многопроцессорной обработки данных имеются в таких операционных системах, как: OS/2 фирмы IBM, Windows NT фирмы Microsoft, NetWare 4.1 фирмы Novell и некоторых других.

Многопроцессорные ОС в свою очередь можно классифицировать по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС. Асимметричная ОС целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. Симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Выше были рассмотрены характеристики ОС, связанные с управлением только одним типом ресурсов – процессором. Важное влияние на облик операционной системы в целом, на возможности ее использования в той или иной области оказывают особенности и других подсистем управления локальными ресурсами – подсистем управления памятью, файлами, устройствами ввода-вывода.

3.3 Особенности аппаратных платформ

На свойства операционной системы непосредственное влияние оказывают аппаратные средства, на которые она ориентирована. По типу аппаратуры различают операционные системы персональных компьютеров, мини-компьютеров, мейнфреймов, кластеров, встроенные ОС и ОС сетей ЭВМ.

1. Первую категорию составляют операционные системы для персональных компьютеров. Их работа заключается в предоставлении удобного интерфейса для одного пользователя. Такие системы широко используются для работы с текстом, электронными таблицами и доступа в Интернет. Наиболее яркие примеры – это Windows XP, MacOS и Linux.

2. Очевидно, что ОС большой машины (мини-компьютера) является более сложной и функциональной, чем ОС персонального компьютера.

Так, в ОС больших машин функции по планированию потока выполняемых задач, очевидно, реализуются путем использования сложных приоритетных дисциплин и требуют большей вычислительной мощности, чем в ОС персональных компьютеров. Аналогично обстоит дело и с другими функциями.

Так, сетевая ОС имеет в своем составе средства передачи сообщений между компьютерами по линиям связи, которые совершенно не нужны в автономной ОС. На основе этих сообщений сетевая ОС поддерживает разделение ресурсов компьютера между удаленными пользователями, подключенными к сети. Для поддержания функций передачи сообщений сетевые ОС содержат специальные программные компоненты, реализующие

популярные коммуникационные протоколы, такие как IP, IPX, Ethernet и другие.

Многопроцессорные системы требуют от операционной системы особой организации, с помощью которой сама операционная система, а также поддерживаемые ею приложения могли бы выполняться параллельно отдельными процессорами системы. Параллельная работа отдельных частей ОС создает дополнительные проблемы для разработчиков ОС, так как в этом случае гораздо сложнее обеспечить согласованный доступ отдельных процессов к общим системным таблицам, исключить эффект гонок и прочие нежелательные последствия асинхронного выполнения работ.

3. Операционные системы для мэйнфреймов. Эти компьютеры размером с комнату все еще можно встретить в центрах данных больших корпораций. Мэйнфреймы отличаются от персональных компьютеров по своим возможностям ввода-вывода. Довольно часто встречаются мэйнфреймы с тысячами дисков и терабайтами данных, а персональный компьютер с такими параметрами показался бы действительно необычным. Мэйнфреймы возвращаются в виде мощных WEB-серверов, серверов для крупномасштабных электронно-коммерческих сайтов и серверов для транзакций в бизнесе.

Операционные системы для мэйнфреймов в основном ориентированы на обработку множества одновременных заданий по операциям ввода-вывода. Примером операционной системы для мэйнфрейма является OS/390, произошедшая от OS/360.

4. Другие требования предъявляются к операционным системам кластеров. Кластер – слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений и представляющихся пользователю единой системой. Наряду со специальной аппаратурой для функционирования кластерных систем, необходима и программная поддержка со стороны операционной системы, которая сводится в основном к синхронизации доступа к разделяемым ресурсам, обнаружению отказов и динамической реконфигурации системы.

3.4 Операционные системы для мобильных устройств

Следует понимать, что на сегодняшний день рынок мобильных устройств, а следовательно и операционных систем для них (равно как и мобильных приложений), – самый быстроразвивающийся среди высокотехнологичных рынков. Поэтому рейтинг тех или иных мобильных ОС может меняться.

Symbian OS

Была самой популярной ОС для мобильных устройств благодаря поддержке фирмы Nokia. Важную роль также сыграло то, что система имеет небольшой размер, а также то, что графический интерфейс и ядро системы

отделены друг от друга. Это позволило легко портировать ее для различных мобильных устройств. Позднее была добавлена многозадачность.

Каждый разработчик создавал свой дистрибутив этой операционной системы в зависимости от ограничений аппаратной платформы, под которую она разрабатывалась. Так появились версии Series 60, Series 80, Series 90, UIQ и MOAP. Каждая версия обладала своими особенностями, что делало необходимым под каждую версию разрабатывать свои приложения. Это было неудобно, поэтому после появления Windows Mobile, Android и iPhoneOS утратила свою популярность среди производителей мобильных девайсов. Так, в этом году компании Sony Ericsson и Samsung объявили, что не будут больше поддерживать эту операционную систему. На данный момент из крупных производителей мобильных девайсов только компания Nokia использует эту ОС для своих смартфонов.

Достоинства:

- низкие требования к памяти и процессору;
- функция освобождения неиспользуемой памяти; □ стабильность;
- малое количество вирусов для этой платформы;
- быстро выходят новые версии и исправляются нестабильности; □ большое количество программ.

Недостатки:

- для связи с ПК нужно устанавливать дополнительный софт; □ несовместимость программ для старых и новых версий.

Windows Mobile

Эта операционная система разработана компанией Microsoft. Эта система использует такой же программный интерфейс, что и настольная версия. Это делает написание программ более знакомым, а пользователям нравится удобный и понятный интерфейс, знакомый им с настольной Windows. Windows Mobile является компонентной, многозадачной, многопоточной и многоплатформенной операционной системой. Благодаря этому она снискала широкое распространение на мобильных устройствах.

Достоинства:

- схожесть с настольной версией;
- удобная синхронизация;
- в комплекте идут офисные программы; □ многозадачность.

Недостатки:

- высокие требования к оборудованию, □ наличие большого числа вирусов, □ нестабильность в работе.

Android

Это одна из ОС, основанная на базе операционной системы Linux и разрабатываемая Open Handset Alliance (ОНА) при поддержке Google. Исходный код находится в открытом доступе, благодаря чему любой разработчик может создать свою версию этой мобильной ОС. Разработчикам приложений выдвинуто небольшое количество ограничений, благодаря чему существует множество как платных, так и бесплатных приложений, которые можно удобно загрузить с Android Market.

Достоинства:

- гибкость;
- открытые исходные коды;
- множество программ;
- высокое быстродействие;
- удобное взаимодействие с сервисами от Google; □ многозадачность.

Недостатки:

- множество актуальных версий – для многих устройств новая версия входит слишком поздно или не появляется вовсе, поэтому разработчикам приходится разрабатывать приложения, ориентируясь на более старые версии;
- высокая предрасположенность к хакерским атакам из-за открытости кода;
- почти всегда требует доработок.

iOS

Мобильная операционная система от компании Apple. Данная система получила распространение только на продуктах компании Apple. Применяется в iPhone, iPod, iPad а также телевизионной приставке AppleTV.

Достоинства:

- удобное использование;
- качественная служба поддержки;
- регулярные обновления, устраняющие многие проблемы в работе; □ возможность купить в App Store множество различных программ.

Недостатки:

- заблокированный характер ОС;
- отсутствие многозадачности;
- нет встроенного редактора документов.

Palm OS

Данная операционная система появилась в 1996 году. Применялась в КПК. Была очень распространена из-за широких возможностей и удобства пользователей. К настоящему моменту практически не применяется, но недавно разработчика поглотила компания HP. Благодаря этому появились надежды на воскрешение некогда популярной КПК ОС. Достоинства:

- не требовательна к ресурсам;
- очень удобный интерфейс пользователя; □ удобная синхронизация с ПК; □ надежность.

Недостатки:

- отсутствует полноценная многозадачность;
- не развиты мультимедийные функции;
- система не развивается (хотя возможно компания HP сможет это преодолеть).

BlackBerry OS

Операционная система работает исключительно на устройствах, выпускаемых компанией Research In Motion Limited (RIM). Ориентирована на корпоративных пользователей. Свое название получила от смартфонов, для которых создавалась. Смартфоны с этой операционной системой получили распространение в корпоративной среде, благодаря сложности перехвата сообщений.

Достоинства:

- удобное пользование электронной почтой;
- легкая синхронизация с ПК;
- широкие возможности настроек безопасности.

Недостатки:

- оптимизирована для вывода только текстовой информации; □ не очень удобный браузер.

Технические характеристики устройства отнюдь не главный параметр при выборе мобильного устройства.

Самые маленькие операционные системы работают на смарткартах, представляющих собой устройство размером с кредитную карту, содержащее центральный процессор. На такие операционные системы накладываются крайне жесткие ограничения по мощности процессора и памяти. Некоторые из них могут управлять только одной операцией, например электронным платежом, но другие операционные системы на тех же самых смарт-картах выполняют сложные функции. Зачастую они являются патентованными системами.

Некоторые смарт-карты являются Java-ориентированными. Это означает, что ПЗУ (ROM, Read Only Memory – память только для чтения) смарт-карт содержит интерпретатор виртуальной машины Java (Java Virtual Machine). Апплеты Java (маленькие программы) загружаются на карту и выполняются JVM-интерпретатором. Некоторые из таких карт могут одновременно управлять несколькими апплетами Java, что приводит к многозадачности и необходимости планирования. Из-за одновременной работы двух и более программ возникает необходимость в управлении ресурсами и защитой. Соответственно, все эти задачи выполняет обычно крайне примитивная операционная система, находящаяся на смарт-карте.

Серверы представляют собой либо очень большие персональные компьютеры, либо рабочие станции, или даже мэйнфреймы. ОС, работающие на них, одновременно обслуживают множество пользователей и позволяют делить между собой программные и аппаратные ресурсы. Серверы предоставляют возможность работы с печатающими устройствами, файлами или в сети Интернет. Интернет-провайдеры обычно запускают в работу несколько серверов для того, чтобы поддерживать одновременный доступ к сети множества клиентов.

3.5 Особенности областей использования

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- системы пакетной обработки (например, ОС ЕС);
- системы разделения времени (UNIX, VMS);
- системы реального времени (QNX, RT/11).

Системы пакетной обработки предназначены для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. Главной целью и критерием эффективности систем пакетной обработки является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени. Для достижения этой цели в системах пакетной обработки используются следующая схема функционирования: в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется мультипрограммная смесь, то есть множество одновременно выполняемых задач. Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины; так, например, в мультипрограммной смеси желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом. Таким образом, выбор нового задания из пакета заданий

зависит от внутренней ситуации, складывающейся в системе, то есть выбирается «выгодное» задание. Следовательно, в таких ОС невозможно гарантировать выполнение того или иного задания в течение определенного периода времени. В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач. Таким образом, взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Системы разделения времени призваны исправить основной недостаток систем пакетной обработки – изоляцию пользователя-программиста от процесса выполнения его задач. Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой. Так как в системах разделения времени каждой задаче выделяется только квант процессорного времени, ни одна задача не занимает процессор надолго и время ответа оказывается приемлемым. Если квант выбран достаточно небольшим, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину. Ясно, что системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как на выполнение принимается каждая запущенная пользователем задача, а не та, которая «выгодна» системе, и кроме того, имеются накладные расходы вычислительной мощности на более частое переключение процессора с задачи на задачу. Критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.

Системы реального времени применяются для управления различными техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка или технологические процессы, такими, как гальваническая линия, доменный процесс и т.п. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария: спутник выйдет из зоны видимости, экспериментальные данные, поступающие с датчиков, будут потеряны, толщина гальванического покрытия не будет соответствовать норме. Таким образом, критерием эффективности для систем реального времени является их

способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия). Это время называется временем реакции системы, а соответствующее свойство системы – реактивностью. Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют фоновым режимом.

3.6 Особенности методов построения ОС

При описании операционной системы часто указываются особенности ее структурной организации и основные концепции, положенные в ее основу.

К таким базовым концепциям относятся:

Способы построения ядра системы

Монолитное ядро или микроядерный подход. Большинство ОС использует монолитное ядро, которое компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский, и наоборот. Альтернативой является построение ОС на базе микроядра, работающего также в привилегированном режиме и выполняющего только минимум функций по управлению аппаратурой, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС – серверы, работающие в пользовательском режиме. При таком построении ОС работает более медленно, так как часто выполняются переходы между привилегированным режимом и пользовательским, зато система получается более гибкой: ее функции можно наращивать, модифицировать или сужать, добавляя, модифицируя или исключая серверы пользовательского режима. Кроме того, серверы хорошо защищены друг от друга, как и любые пользовательские процессы.

Построение ОС на базе объектно ориентированного подхода дает возможность использовать все его достоинства, хорошо зарекомендовавшие себя на уровне приложений, внутри операционной системы, а именно: аккумуляцию удачных решений в форме стандартных объектов, возможность создания новых объектов на базе имеющихся с помощью механизма наследования, хорошую защиту данных за счет их инкапсуляции во внутренние структуры объекта, что делает данные недоступными для

несанкционированного использования извне, структурированность системы, состоящей из набора хорошо определенных объектов.

Наличие нескольких прикладных сред дает возможность в рамках одной ОС одновременно выполнять приложения, разработанные для нескольких ОС. Многие современные операционные системы поддерживают одновременно прикладные среды MS-DOS, Windows, UNIX (POSIX), OS/2 или хотя бы некоторого подмножества из этого популярного набора. Концепция множественных прикладных сред наиболее просто реализуется в ОС на базе микроядра, над которым работают различные серверы, часть которых реализуют прикладную среду той или иной операционной системы.

Распределенная организация операционной системы позволяет упростить работу пользователей и программистов в сетевых средах. В распределенной ОС реализованы механизмы, которые дают возможность пользователю представлять и воспринимать сеть в виде традиционного однопроцессорного компьютера. Характерными признаками распределенной организации ОС являются: наличие единой справочной службы разделяемых ресурсов, единой службы времени, использование механизма вызова удаленных процедур (RPC) для прозрачного распределения программных процедур по машинам, многократной обработки, позволяющей распараллеливать вычисления в рамках одной задачи и выполнять эту задачу сразу на нескольких компьютерах сети, а также наличие других распределенных служб.

Подсистема управления процессами

Важнейшей частью операционной системы, непосредственно влияющей на функционирование вычислительной машины, является подсистема управления процессами.

Процесс (или по-другому, задача) – абстракция, описывающая выполняющуюся программу. Для операционной системы процесс представляет собой единицу работы, заявку на потребление системных ресурсов. Подсистема управления процессами планирует выполнение процессов, то есть распределяет процессорное время между несколькими одновременно существующими в системе процессами, а также занимается созданием и уничтожением процессов, обеспечивает процессы необходимыми системными ресурсами, поддерживает взаимодействие между процессами.

Некоторые из ресурсов выделяются процессу при его создании, а некоторые – динамически по запросам во время выполнения. Ресурсы могут быть приписаны процессу на все время его жизни или только на определенный период. При выполнении этих функций подсистема управления процессами взаимодействует с другими подсистемами ОС, ответственными за управление ресурсами, такими как подсистема управления памятью, подсистема ввода-вывода, файловая система. Когда в системе одновременно выполняется

несколько независимых задач, то возникают дополнительные проблемы. Хотя процессы возникают и выполняются асинхронно, у них может возникнуть необходимость во взаимодействии, например при обмене данными. Согласование скоростей процессов также очень важно для предотвращения эффекта «гонок» – когда несколько процессов пытаются изменить один и тот же файл – взаимных блокировок или других коллизий, которые возникают при совместном использовании ресурсов. Синхронизация процессов является одной из важных функций подсистемы управления процессами.

Каждый раз, когда процесс завершается, ОС предпринимает шаги, чтобы «зачистить следы» его пребывания в системе. Подсистема управления процессами закрывает все файлы, с которыми работал процесс, освобождает области оперативной памяти, отведенные под коды, данные и системные информационные структуры процесса. Выполняется коррекция всевозможных очередей ОС и списков ресурсов, в которых имелись ссылки на завершаемый процесс.

Чтобы поддерживать мультипрограммирование, ОС должна определить и оформить для себя те внутренние единицы работы, между которыми будет разделяться процессор и другие ресурсы компьютера. Необходимо напомнить, что мультипрограммирование это способ организации вычислительного процесса, при котором в памяти компьютера находится несколько программ, попеременно выполняющиеся на одном процессоре.

В настоящее время в большинстве операционных систем определены два типа единиц работы. Более крупная единица работы, обычно носящая название процесса, или задачи, требует для своего выполнения нескольких более мелких работ, для обозначения которых используют термины «поток» или «нить».

Очевидно, что любая работа вычислительной системы заключается в выполнении некоторой программы. Поэтому и с процессом, и с потоком связывается определенный программный код, который для этих целей оформляется в виде исполняемого модуля. Чтобы этот программный код мог быть выполнен, его необходимо загрузить в оперативную память, возможно, выделить некоторое место на диске для хранения данных, предоставить доступ к устройствам ввода-вывода, например, к последовательному порту. В ходе выполнения программе может также понадобиться доступ к информационным ресурсам, например, файлам. И, конечно же, невозможно выполнение программы без предоставления ей процессорного времени, то есть времени, в течение которого процессор выполняет коды данной программы.

В операционных системах, где существуют и процессы, и потоки, процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами

работы – потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд.

В простейшем случае процесс состоит из одного потока, и именно таким образом трактовалось понятие «процесс» до середины 80-х годов, и в таком же виде оно сохранилось в некоторых современных ОС. В таких системах понятие «поток» полностью поглощается понятием «процесс», то есть остается только одна единица работы и потребления ресурсов – процесс. Мультипрограммирование осуществляется в таких ОС на уровне процессов.

Существует 3 основных состояния процесса:

- готов; □
- выполняется; □
- блокирован.

В однопроцессорной системе только один процесс может быть в стадии выполнения. В это же время может быть несколько готовых процессов, может быть несколько заблокированных процессов, ожидающих появления некоторых событий (например, ввода/вывода). При этом список готовых процессов упорядочен по приоритетам.

Для того чтобы процессы не могли вмешаться в распределение ресурсов, а также не могли повредить коды и данные друг друга, важнейшей задачей ОС является изоляция одного процесса от другого. Для этого операционная система обеспечивает каждый процесс отдельным виртуальным адресным пространством, так что ни один процесс не может получить прямого доступа к командам и данным другого процесса.

Виртуальное адресное пространство процесса – это совокупность адресов, которыми может манипулировать программный модуль процесса. Операционная система отображает виртуальное адресное пространство процесса на отведенную процессу физическую память.

При необходимости взаимодействия процессы обращаются к операционной системе, которая, выполняя функции посредника, предоставляет им средства межпроцессной связи – конвейеры, почтовые ящики, разделяемые секции памяти и некоторые другие.

Однако в системах, в которых отсутствует понятие потока, возникают проблемы при организации параллельных вычислений в рамках процесса. А такая необходимость может возникать. Действительно, при мультипрограммировании повышается пропускная способность системы, но отдельный процесс никогда не может быть выполнен быстрее, чем в однопрограммном режиме (всякое разделение ресурсов только замедляет работу одного из участников за счет дополнительных затрат времени на ожидание освобождения ресурса). Однако приложение, выполняемое в рамках одного процесса, может обладать внутренним параллелизмом, который в

принципе мог бы позволить ускорить его решение. Если, например, в программе предусмотрено обращение к внешнему устройству, то на время этой операции можно не блокировать выполнение всего процесса, а продолжить вычисления по другой ветви программы. Параллельное выполнение нескольких работ в рамках одного интерактивного приложения повышает эффективность работы пользователя. Так, при работе с текстовым редактором желательно иметь возможность совмещать набор нового текста с такими продолжительными по времени операциями, как переформатирование значительной части текста, печать документа или его сохранение на локальном или удаленном диске.

Потоки возникли в операционных системах как средство распараллеливания вычислений. Конечно, задача распараллеливания вычислений в рамках одного приложения может быть решена и традиционными способами.

Во-первых, прикладной программист может взять на себя сложную задачу организации параллелизма, выделив в приложении некоторую подпрограмму-диспетчер, которая периодически передает управление той или иной ветви вычислений. При этом программа получается логически весьма запутанной, с многочисленными передачами управления, что существенно затрудняет ее отладку и модификацию.

Во-вторых, решением является создание для одного приложения нескольких процессов для каждой из параллельных работ. Однако использование для создания процессов стандартных средств ОС не позволяет учесть тот факт, что эти процессы решают единую задачу, а значит, имеют много общего между собой – они могут работать с одними и теми же данными, использовать один и тот же кодовый сегмент, наделяться одними и теми же правами доступа к ресурсам вычислительной системы.

В операционной системе, наряду с процессами, нужен другой механизм распараллеливания вычислений, который учитывал бы тесные связи между отдельными ветвями вычислений одного и того же приложения. Для этих целей современные ОС предлагают механизм многопоточной обработки (multithreading). При этом вводится новая единица работы – поток выполнения, а понятие «процесс» в значительной степени меняет смысл. Понятию «поток» соответствует последовательный переход процессора от одной команды программы к другой. ОС распределяет процессорное время между потоками. Процессу ОС назначает адресное пространство и набор ресурсов, которые совместно используются всеми его потоками.

Заметим, что в однопрограммных системах не возникает необходимости введения понятия, обозначающего единицу работы, так как там не существует проблемы разделения ресурсов.

Создание потоков требует от ОС меньших накладных расходов, чем процессов. В отличие от процессов, которые принадлежат разным, вообще говоря, конкурирующим приложениям, все потоки одного процесса всегда принадлежат одному приложению, поэтому ОС изолирует потоки в гораздо меньшей степени, нежели процессы в традиционной мультипрограммной системе. Все потоки одного процесса используют общие файлы, таймеры, устройства, одну и ту же область оперативной памяти, одно и то же адресное пространство. Это означает, что они разделяют одни и те же глобальные переменные. Поскольку каждый поток может иметь доступ к любому виртуальному адресу процесса, один поток может использовать стек другого потока. Между потоками одного процесса нет полной защиты, потому что, во-первых, это невозможно, а во-вторых, не нужно. Чтобы организовать взаимодействие и обмен данными, потокам вовсе не требуется обращаться к ОС, им достаточно использовать общую память – один поток записывает данные, а другой читает их. С другой стороны, потоки разных процессов по-прежнему хорошо защищены друг от друга.

Итак, мультипрограммирование более эффективно на уровне потоков, а не процессов. Каждый поток имеет собственный счетчик команд и стек. Задача, оформленная в виде нескольких потоков в рамках одного процесса, может быть выполнена быстрее за счет псевдопараллельного (или параллельного в мультипроцессорной системе) выполнения ее отдельных частей. Особенно эффективно можно использовать многопоточность для выполнения распределенных приложений, например, многопоточный сервер может параллельно выполнять запросы сразу нескольких клиентов.

Использование потоков связано не только со стремлением повысить производительность системы за счет параллельных вычислений, но и с целью создания более читабельных, логичных программ. Введение нескольких потоков выполнения упрощает программирование. Например, в задачах типа «писатель-читатель» один поток выполняет запись в буфер, а другой считывает записи из него. Поскольку они разделяют общий буфер, не стоит их делать отдельными процессами. Другой пример использования потоков – управление сигналами, такими как прерывание с клавиатуры (del или break). Вместо обработки сигнала прерывания один поток назначается для постоянного ожидания поступления сигналов. Таким образом, использование потоков может сократить необходимость в прерываниях пользовательского уровня. В этих примерах не столь важно параллельное выполнение, сколь важна ясность программы.

Наибольший эффект от введения многопоточной обработки достигается в мультипроцессорных системах, в которых потоки, в том числе и принадлежащие одному процессу, могут выполняться на разных процессорах действительно параллельно (а не псевдопараллельно).

Создать процесс – это, прежде всего, означает создать описатель процесса, в качестве которого выступает одна или несколько информационных структур, содержащих все сведения о процессе, необходимые операционной системе для управления им. В число таких сведений могут входить, например, идентификатор процесса, данные о расположении в памяти исполняемого модуля, степень привилегированности процесса (приоритет и права доступа) и т.п.

Создание описателя процесса знаменует собой появление в системе еще одного претендента на вычислительные ресурсы. Начиная с этого момента при распределении ресурсов ОС должна принимать во внимание потребности нового процесса.

Создание процесса включает загрузку кодов и данных исполняемой программы данного процесса с диска в оперативную память. Для этого ОС должна обнаружить местоположение такой программы на диске, перераспределить оперативную память и выделить память исполняемой программе нового процесса. Затем необходимо считать программу в выделенные для нее участки памяти и, возможно, изменить параметры программы в зависимости от размещения в памяти.

В системах с виртуальной памятью в начальный момент может загружаться только часть кодов и данных процесса с тем, чтобы «подкачивать» остальные по мере необходимости. Существуют системы, в которых на этапе создания процесса не требуется непременно загружать коды и данные в оперативную память, вместо этого исполняемый модуль копируется из того каталога файловой системы, в котором он изначально находился, в область подкачки – специальную область диска, отведенную для хранения кодов и данных процессов. При выполнении всех этих действий подсистема управления процессами тесно взаимодействует с подсистемой управления памятью и файловой системой.

В многопоточной системе при создании процесса ОС создает для каждого процесса как минимум один поток выполнения. При создании потока так же, как и при создании процесса, операционная система генерирует специальную информационную структуру – описатель потока, который содержит идентификатор потока, данные о правах доступа и приоритете, о состоянии потока и другую информацию. В исходном состоянии поток (или процесс, если речь идет о системе, в которой понятие «поток» не определяется) находится в приостановленном состоянии. Момент выборки потока на выполнение осуществляется в соответствии с принятым в данной системе правилом предоставления процессорного времени и с учетом всех существующих в данный момент потоков и процессов. В случае если коды и данные процесса находятся в области подкачки, необходимым условием активизации потока

процесса является также наличие места в оперативной памяти для загрузки его исполняемого модуля.

Во многих системах поток может обратиться к ОС с запросом на создание так называемых потоков-потомков. В разных ОС по-разному строятся отношения между потоками-потомками и их родителями. Например, в одних ОС выполнение родительского потока синхронизируется с его потомками, в частности после завершения родительского потока ОС может снимать с выполнения всех его потомков. В других системах потоки-потомки могут выполняться асинхронно по отношению к родительскому потоку. Потомки, как правило, наследуют многие свойства родительских потоков. Во многих системах порождение потомков является основным механизмом создания процессов и потоков.

3.7 Алгоритмы планирования процессов

Планирование процессов включает в себя решение следующих задач:

- определение момента времени для смены выполняемого процесса; □
- выбор процесса на выполнение из очереди готовых процессов; □
- переключение контекстов «старого» и «нового» процессов.

Первые две задачи решаются программными средствами, а последняя в значительной степени аппаратно.

Существует множество различных алгоритмов планирования процессов, по-разному решающих вышеперечисленные задачи, преследующих различные цели и обеспечивающих различное качество мультипрограммирования. Среди этого множества алгоритмов рассмотрим подробнее две группы наиболее часто встречающихся алгоритмов: алгоритмы, основанные на квантовании, и алгоритмы, основанные на приоритетах.

В соответствии с алгоритмами, основанными на квантовании, смена активного процесса происходит, если:

- процесс завершился и покинул систему,
- произошла ошибка,
- процесс перешел в состояние ОЖИДАНИЕ,
- исчерпан квант процессорного времени, отведенный данному процессу.

Процесс, который исчерпал свой квант, переводится в состояние ГОТОВНОСТЬ и ожидает, когда ему будет предоставлен новый квант процессорного времени, а на выполнение в соответствии с определенным правилом выбирается новый процесс из очереди готовых. Таким образом, ни один процесс не занимает процессор надолго, поэтому квантование широко используется в системах разделения времени.

Кванты, выделяемые процессам, могут быть одинаковыми для всех процессов или различными. Кванты, выделяемые одному процессу, могут быть

фиксированной величины или изменяться в разные периоды жизни процесса. Процессы, которые не полностью использовали выделенный им квант (например, из-за ухода на выполнение операций ввода-вывода), могут получить или не получить компенсацию в виде привилегий при последующем обслуживании. По-разному может быть организована очередь готовых процессов: циклически, по правилу «первый пришел – первый обслужился» (FIFO) или по правилу «последний пришел – первый обслужился» (LIFO).

Другая группа алгоритмов использует понятие «приоритет» процесса. Приоритет – это число, характеризующее степень привилегированности процесса при использовании ресурсов вычислительной машины, в частности процессорного времени: чем выше приоритет, тем выше привилегии.

Приоритет может выражаться целыми или дробными, положительным или отрицательным значением. Чем выше привилегии процесса, тем меньше времени он будет проводить в очередях. Приоритет может назначаться директивно администратором системы в зависимости от важности работы или внесенной платы, либо вычисляться самой ОС по определенным правилам, он может оставаться фиксированным на протяжении всей жизни процесса либо изменяться во времени в соответствии с некоторым законом. В последнем случае приоритеты называются динамическими.

Существует две разновидности приоритетных алгоритмов: алгоритмы, использующие относительные приоритеты, и алгоритмы, использующие абсолютные приоритеты.

В обоих случаях выбор процесса на выполнение из очереди готовых осуществляется одинаково: выбирается процесс, имеющий наивысший приоритет. По-разному решается проблема определения момента смены активного процесса. В системах с относительными приоритетами активный процесс выполняется до тех пор, пока он сам не покинет процессор, перейдя в состояние БЛОКИРОВКИ (или же произойдет ошибка, или процесс завершится). В системах с абсолютными приоритетами выполнение активного процесса прерывается еще при одном условии: если в очереди готовых процессов появился процесс, приоритет которого выше приоритета активного процесса. В этом случае прерванный процесс переходит в состояние готовности.

Во многих операционных системах алгоритмы планирования построены с использованием как квантования, так и приоритетов. Например, в основе планирования лежит квантование, но величина кванта и/или порядок выбора процесса из очереди готовых определяется приоритетами процессов.

3.8 Вытесняющие и невытесняющие алгоритмы

Существует два основных типа процедур планирования процессов – вытесняющие (preemptive) и невытесняющие (non-preemptive).

Non-preemptive multitasking – невывесняющая многозадачность – это способ планирования процессов, при котором активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление планировщику операционной системы для того, чтобы тот выбрал из очереди другой, готовый к выполнению процесс.

Preemptive multitasking – вытесняющая многозадачность – это такой способ, при котором решение о переключении процессора с выполнения одного процесса на выполнение другого процесса принимается планировщиком операционной системы, а не самой активной задачей.

Понятия preemptive и non-preemptive иногда отождествляются с понятиями приоритетных и бесприоритетных дисциплин, что совершенно неверно, а также с понятиями абсолютных и относительных приоритетов, что неверно отчасти. Вытесняющая и невывесняющая многозадачность – это более широкие понятия, чем типы приоритетности. Приоритеты задач могут как использоваться, так и не использоваться и при вытесняющих, и при невывесняющих способах планирования. Так в случае использования приоритетов дисциплина относительных приоритетов может быть отнесена к классу систем с невывесняющей многозадачностью, а дисциплина абсолютных приоритетов – к классу систем с вытесняющей многозадачностью. А бесприоритетная дисциплина планирования, основанная на выделении равных квантов времени для всех задач, относится к вытесняющим алгоритмам.

Основным различием между preemptive и non-preemptive вариантами многозадачности является степень централизации механизма планирования задач. При вытесняющей многозадачности механизм планирования задач целиком сосредоточен в операционной системе, и программист пишет свое приложение, не заботясь о том, что оно будет выполняться параллельно с другими задачами. При этом операционная система выполняет следующие функции: определяет момент снятия с выполнения активной задачи, запоминает ее контекст, выбирает из очереди готовых задач следующую и запускает ее на выполнение, загружая ее контекст.

При невывесняющей многозадачности механизм планирования распределен между системой и прикладными программами. Прикладная программа, получив управление от операционной системы, сама определяет момент завершения своей очередной итерации и передает управление ОС с помощью какого-либо системного вызова, а ОС формирует очереди задач и выбирает в соответствии с некоторым алгоритмом (например, с учетом приоритетов) следующую задачу на выполнение. Такой механизм создает проблемы как для пользователей, так и для разработчиков.

Для пользователей это означает, что управление системой теряется на произвольный период времени, который определяется приложением (а не пользователем). Если приложение тратит слишком много времени на

выполнение какой-либо работы, например на форматирование диска, пользователь не может переключиться с этой задачи на другую задачу, например на текстовый редактор, в то время как форматирование продолжалось бы в фоновом режиме. Эта ситуация нежелательна, так как пользователи обычно не хотят долго ждать, когда машина завершит свою задачу.

Поэтому разработчики приложений для non-preemptive операционной среды, возлагая на себя функции планировщика, должны создавать приложения так, чтобы они выполняли свои задачи небольшими частями. Например, программа форматирования может отформатировать одну дорожку дискеты и вернуть управление системе. После выполнения других задач система возвратит управление программе форматирования, чтобы та отформатировала следующую дорожку. Подобный метод разделения времени между задачами работает, но он существенно затрудняет разработку программ и предъявляет повышенные требования к квалификации программиста. Программист должен обеспечить «дружественное» отношение своей программы к другим выполняемым одновременно с ней программам, достаточно часто отдавая им управление. Крайним проявлением «недружественности» приложения является его зависание, которое приводит к общему краху системы. В системах с вытесняющей многозадачностью такие ситуации, как правило, исключены, так как центральный планирующий механизм снимет зависшую задачу с выполнения.

Однако распределение функций планировщика между системой и приложениями не всегда является недостатком, а при определенных условиях может быть и преимуществом, потому что дает возможность разработчику приложений самому проектировать алгоритм планирования, наиболее подходящий для данного фиксированного набора задач. Так как разработчик сам определяет в программе момент времени отдачи управления, то при этом исключаются нерациональные прерывания программ в «неудобные» для них моменты времени. Кроме того, легко разрешаются проблемы совместного использования данных: задача во время каждой итерации использует их монопольно и уверена, что на протяжении этого периода никто другой не изменит эти данные. Существенным преимуществом non-preemptive систем является более высокая скорость переключения с задачи на задачу.

Примером эффективного использования невытесняющей многозадачности является файл-сервер NetWare, в котором, в значительной степени благодаря этому, достигнута высокая скорость выполнения файловых операций.

Однако почти во всех современных операционных системах, ориентированных на высокопроизводительное выполнение приложений (UNIX, Windows NT, OS/2, VAX/VMS), реализована вытесняющая многозадачность. В последнее время дошла очередь и до ОС класса настольных систем, например,

OS/2 Warp и Windows 95. Возможно в связи с этим вытесняющую многозадачность часто называют истинной многозадачностью.

3.9 Управление памятью

Память является важнейшим ресурсом, требующим тщательного управления со стороны мультипрограммной операционной системы. Распределению подлежит вся оперативная память, не занятая операционной системой. Обычно ОС располагается в самых младших адресах, однако может занимать и самые старшие адреса. Функциями ОС по управлению памятью являются: отслеживание свободной и занятой памяти, выделение памяти процессам и освобождение памяти при завершении процессов, вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место, а также настройка адресов программы на конкретную область физической памяти.

Для идентификации переменных и команд используются символьные имена (метки), виртуальные адреса и физические адреса. Символьные имена присваивает пользователь при написании программы на алгоритмическом языке или ассемблере. Виртуальные адреса вырабатывает транслятор, переводящий программу на машинный язык. Так как во время трансляции в общем случае не известно, в какое место оперативной памяти будет загружена программа, то транслятор присваивает переменным и командам виртуальные (условные) адреса, обычно считая по умолчанию, что программа будет размещена, начиная с нулевого адреса. Совокупность виртуальных адресов процесса называется виртуальным адресным пространством. Каждый процесс имеет собственное виртуальное адресное пространство. Максимальный размер виртуального адресного пространства ограничивается разрядностью адреса, присущей данной архитектуре компьютера, и, как правило, не совпадает с объемом физической памяти, имеющимся в компьютере.

Физические адреса соответствуют номерам ячеек оперативной памяти, где в действительности расположены или будут расположены переменные и команды. Переход от виртуальных адресов к физическим может осуществляться двумя способами. В первом случае замену виртуальных адресов на физические делает специальная системная программа – перемещающий загрузчик. Перемещающий загрузчик на основании имеющихся у него исходных данных о начальном адресе физической памяти, в которую предстоит загружать программу, и информации, предоставленной транслятором об адресно-зависимых константах программы, выполняет загрузку программы, совмещая ее с заменой виртуальных адресов физическими.

Второй способ заключается в том, что программа загружается в память в неизменном виде в виртуальных адресах, при этом операционная система фиксирует смещение действительного расположения программного кода относительно виртуального адресного пространства. Во время выполнения программы при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический. Второй способ является более гибким, он допускает перемещение программы во время ее выполнения, в то время как перемещающий загрузчик жестко привязывает программу к первоначально выделенному ей участку памяти. Вместе с тем использование перемещающего загрузчика уменьшает накладные расходы, так как преобразование каждого виртуального адреса происходит только один раз во время загрузки, а во втором случае – каждый раз при обращении по данному адресу.

В некоторых случаях (обычно в специализированных системах), когда заранее точно известно, в какой области оперативной памяти будет выполняться программа, транслятор выдает исполняемый код сразу в физических адресах.

Все методы управления памятью могут быть разделены на два класса: методы, которые используют перемещение процессов между оперативной памятью и диском, и методы, которые не делают этого.

Самым простым способом управления оперативной памятью является разделение ее на несколько разделов фиксированной величины. Это может быть выполнено вручную оператором во время старта системы или во время ее генерации. Очередная задача, поступившая на выполнение, помещается либо в общую очередь, либо в очередь к некоторому разделу.

Подсистема управления памятью в этом случае выполняет следующие задачи:

- сравнивая размер программы, поступившей на выполнение, и свободных разделов, выбирает подходящий раздел,
- осуществляет загрузку программы и настройку адресов.

При очевидном преимуществе – простоте реализации – данный метод имеет существенный недостаток – жесткость. Так как в каждом разделе может выполняться только одна программа, то уровень мультипрограммирования заранее ограничен числом разделов не зависимо от того, какой размер имеют программы. Даже если программа имеет небольшой объем, она будет занимать весь раздел, что приводит к неэффективному использованию памяти. С другой стороны, даже если объем оперативной памяти машины позволяет выполнить некоторую программу, разбиение памяти на разделы не позволяет сделать этого.

Задачами операционной системы при реализации данного метода управления памятью является:

- ведение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти,
- при поступлении новой задачи – анализ запроса, просмотр таблицы свободных областей и выбор раздела, размер которого достаточен для размещения поступившей задачи,
- загрузка задачи в выделенный ей раздел и корректировка таблиц свободных и занятых областей,
- после завершения задачи корректировка таблиц свободных и занятых областей.

Программный код не перемещается во время выполнения, то есть может быть проведена единовременная настройка адресов посредством использования перемещающего загрузчика.

Выбор раздела для вновь поступившей задачи может осуществляться по разным правилам, таким, например, как «первый попавшийся раздел достаточного размера», или «раздел, имеющий наименьший достаточный размер», или «раздел, имеющий наибольший достаточный размер». Все эти правила имеют свои преимущества и недостатки.

По сравнению с методом распределения памяти фиксированными разделами данный метод обладает гораздо большей гибкостью, но ему присущ очень серьезный недостаток – фрагментация памяти. Фрагментация – это наличие большого числа несмежных участков свободной памяти очень маленького размера (фрагментов). Настолько маленького, что ни одна из вновь поступающих программ не может поместиться ни в одном из участков, хотя суммарный объем фрагментов может составить значительную величину, намного превышающую требуемый объем памяти.

Одним из методов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших либо в сторону младших адресов, так, чтобы вся свободная память образовывала единую свободную область. В дополнение к функциям, которые выполняет ОС при распределении памяти переменными разделами, в данном случае она должна еще время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей. Эта процедура называется «сжатием». Сжатие может выполняться либо при каждом завершении задачи, либо только тогда, когда для вновь поступившей задачи нет свободного раздела достаточного размера. В первом случае требуется меньше вычислительной работы при корректировке таблиц, а во втором – реже выполняется процедура сжатия. Так как программы перемещаются по оперативной памяти в ходе своего выполнения, то преобразование адресов из виртуальной формы в физическую должно выполняться динамическим способом.

Хотя процедура сжатия и приводит к более эффективному использованию памяти, она может потребовать значительного времени, что часто перевешивает преимущества данного метода.

Уже достаточно давно пользователи столкнулись с проблемой размещения в памяти программ, размер которых превышал имеющуюся в наличии свободную память. Решением было разбиение программы на части, называемые оверлеями. 0-й оверлей начинал выполняться первым. Когда он заканчивал свое выполнение, он вызывал другой оверлей. Все оверлеи хранились на диске и перемещались между памятью и диском средствами операционной системы. Однако разбиение программы на части и планирование их загрузки в оперативную память должен был осуществлять программист.

Развитие методов организации вычислительного процесса в этом направлении привело к появлению метода, известного под названием «виртуальная память». Виртуальным называется ресурс, который пользователю или пользовательской программе представляется обладающим свойствами, которыми он в действительности не обладает. Так, например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную оперативную память. Пользователь пишет программы так, как будто в его распоряжении имеется однородная оперативная память большого объема, но в действительности все данные, используемые программой, хранятся на одном или нескольких разнородных запоминающих устройствах, обычно на дисках, и при необходимости частями отображаются в реальную память.

Таким образом, виртуальная память – это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся оперативную память; для этого виртуальная память решает следующие задачи:

- размещает данные в запоминающих устройствах разного типа, например, часть программы в оперативной памяти, а часть на диске;
- перемещает по мере необходимости данные между запоминающими устройствами разного типа, например, подгружает нужную часть программы с диска в оперативную память; □ преобразует виртуальные адреса в физические.

Все эти действия выполняются автоматически, без участия программиста, то есть механизм виртуальной памяти является прозрачным по отношению к пользователю.

Наиболее распространенными реализациями виртуальной памяти является страничное, сегментное и странично-сегментное распределение памяти, а также свопинг.

Виртуальное адресное пространство каждого процесса при страничном распределении делится на части одинакового, фиксированного для данной системы размера, называемые виртуальными страницами. В общем случае размер виртуального адресного пространства не является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.

Вся оперативная память машины также делится на части такого же размера, называемые физическими страницами (или блоками).

Размер страницы обычно выбирается равным степени двойки: 512, 1024 и т.д., это позволяет упростить механизм преобразования адресов.

При загрузке процесса часть его виртуальных страниц помещается в оперативную память, а остальные – на диск. Смежные виртуальные страницы не обязательно располагаются в смежных физических страницах. При загрузке операционная система создает для каждого процесса информационную структуру – таблицу страниц, в которой устанавливается соответствие между номерами виртуальных и физических страниц для страниц, загруженных в оперативную память, или делается отметка о том, что виртуальная страница выгружена на диск. Кроме того, в таблице страниц содержится управляющая информация, такая как признак модификации страницы, признак невыгружаемости (выгрузка некоторых страниц может быть запрещена), признак обращения к странице (используется для подсчета числа обращений за определенный период времени) и другие данные, формируемые и используемые механизмом виртуальной памяти.

При активизации очередного процесса в специальный регистр процессора загружается адрес таблицы страниц данного процесса.

При каждом обращении к памяти происходит чтение из таблицы страниц информации о виртуальной странице, к которой произошло обращение. Если данная виртуальная страница находится в оперативной памяти, то выполняется преобразование виртуального адреса в физический. Если же нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое страничное прерывание. Выполняющийся процесс переводится в состояние ожидания, и активизируется другой процесс из очереди готовых. Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее в оперативную память. Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, если же свободных страниц нет, то решается вопрос, какую страницу следует выгрузить из оперативной памяти.

В данной ситуации может быть использовано много разных критериев выбора, наиболее популярные из них следующие:

- дольше всего не использовавшаяся страница,

- первая попавшаяся страница,
- страница, к которой в последнее время было меньше всего обращений.

В некоторых системах используется понятие рабочего множества страниц. Рабочее множество определяется для каждого процесса и представляет собой перечень наиболее часто используемых страниц, которые должны постоянно находиться в оперативной памяти и поэтому не подлежат выгрузке.

После того как выбрана страница, которая должна покинуть оперативную память, анализируется ее признак модификации (из таблицы страниц). Если вытаскиваемая страница с момента загрузки была модифицирована, то ее новая версия должна быть переписана на диск. Если нет, то она может быть просто уничтожена, то есть соответствующая физическая страница объявляется свободной.

При страничной организации виртуальное адресное пространство процесса делится механически на равные части. Это не позволяет дифференцировать способы доступа к разным частям программы (сегментам), а это свойство часто может быть очень полезным. Например, можно запретить обращаться с операциями записи и чтения в кодовый сегмент программы, а для сегмента данных разрешить только чтение. Кроме того, разбиение программы на «осмысленные» части делает принципиально возможным разделение одного сегмента несколькими процессами. Например, если два процесса используют одну и ту же математическую подпрограмму, то в оперативную память может быть загружена только одна копия этой подпрограммы.

Рассмотрим, каким образом сегментное распределение памяти реализует эти возможности. Виртуальное адресное пространство процесса делится на сегменты, размер которых определяется программистом с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п. Иногда сегментация программы выполняется по умолчанию компилятором.

При загрузке процесса часть сегментов помещается в оперативную память (при этом для каждого из этих сегментов операционная система подыскивает подходящий участок свободной памяти), а часть сегментов размещается в дисковой памяти. Сегменты одной программы могут занимать в оперативной памяти несмежные участки. Во время загрузки система создает таблицу сегментов процесса (аналогичную таблице страниц), в которой для каждого сегмента указывается начальный физический адрес сегмента в оперативной памяти, размер сегмента, правила доступа, признак модификации, признак обращения к данному сегменту за последний интервал времени и некоторая другая информация. Если виртуальные адресные пространства нескольких процессов включают один и тот же сегмент, то в таблицах сегментов этих процессов делаются ссылки на один и тот же участок оперативной памяти, в который данный сегмент загружается в единственном экземпляре.

Система с сегментной организацией функционирует аналогично системе со страничной организацией: время от времени происходят прерывания, связанные с отсутствием нужных сегментов в памяти, при необходимости освобождения памяти некоторые сегменты выгружаются, при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический. Кроме того, при обращении к памяти проверяется, разрешен ли доступ требуемого типа к данному сегменту.

Виртуальный адрес при сегментной организации памяти может быть представлен парой (g, s) , где g – номер сегмента, а s – смещение в сегменте. Физический адрес получается путем сложения начального физического адреса сегмента, найденного в таблице сегментов по номеру g , и смещения s .

Недостатком данного метода распределения памяти является фрагментация на уровне сегментов и более медленное, по сравнению со страничной организацией, преобразование адреса.

Как видно из названия, странично-сегментный метод представляет собой комбинацию страничного и сегментного распределения памяти и вследствие этого сочетает в себе достоинства обоих подходов. Виртуальное пространство процесса делится на сегменты, а каждый сегмент в свою очередь делится на виртуальные страницы, которые нумеруются в пределах сегмента. Оперативная память делится на физические страницы. Загрузка процесса выполняется операционной системой постранично, при этом часть страниц размещается в оперативной памяти, а часть на диске. Для каждого сегмента создается своя таблица страниц, структура которой полностью совпадает со структурой таблицы страниц, используемой при страничном распределении. Для каждого процесса создается таблица сегментов, в которой указываются адреса таблиц страниц для всех сегментов данного процесса. Адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс.

Разновидностью виртуальной памяти является свопинг. При свопинге, в отличие от рассмотренных ранее методов реализации виртуальной памяти, процесс перемещается между памятью и диском целиком, то есть в течение некоторого времени процесс может полностью отсутствовать в оперативной памяти. Существуют различные алгоритмы выбора процессов на загрузку и выгрузку, а также различные способы выделения оперативной и дисковой памяти загружаемому процессу.

Лекция 3. Информационные технологии электронного офиса

Тема 4 Текстовые редакторы, подготовка документов в среде MS Word. Обработка текстовых данных

В общем случае под *офисом* понимается организация типа канцелярии, функция которой – получать и выпускать документы, а также отслеживать их движение. В основе офисной деятельности лежит работа с документами. Обобщенное понятие «документ» приведено в федеральном законе «Об обязательном экземпляре документов» № 77-ФЗ от 29 декабря 1994 года:

«Документ – материальный объект с зафиксированной на нем информацией в виде текста, звукозаписи или изображения, предназначенный для передачи во времени и пространстве в целях хранения и общественного использования».

К *офисным* относятся следующие задачи: делопроизводство, управление, контроль управления, создание отчетов, поиск, ввод и обновление информации, составление расписаний, обмен информацией между отделами офиса, между офисами предприятия и между предприятиями.

К видам деятельности, осуществляемым в офисе, которые требуют информационной поддержки, главным образом, относятся¹:

- основная деятельность офиса – принятие решений разных уровней;
- составление контрактов, планирование и контроль выполнения производственных заданий и др.;
- стратегическое и тактическое планирование основной деятельности;
- собственно документооборот и организация делопроизводства;
- обслуживание входных информационных потоков;
- техническая поддержка бизнес-процессов как процессов специфического рода (длительных, вариантных, распределенных и др.). *Типовые процедуры*, выполняемые в перечисленных выше задачах:
- обработка входящей и исходящей информации (чтение и ответы на письма, написание отчетов, циркуляров и прочей документации, которая может включать также рисунки и диаграммы);
- редактирование (набор, проверка, оформление) и печать документов;
- контроль прохождения и маршрутизация документов;

¹ Зиндер Е. Корпорация LVS Group. www.citforum.ru : Реинжиниринг бизнес-процессов и автоматизация офиса. Материалы конференции «Офисные информационные системы 96».

- контроль исполнения поручений;
- сбор и последующий анализ данных (отчетность за определенные периоды времени по различным подразделениям в соответствии с различными критериями выбора);
- хранение поступившей информации (быстрый доступ к информации и поиск необходимых данных) и т.п.

Документами, используемыми при выполнении указанных выше процедур, могут быть:

- письмо или заключение, фиксирующее решение, принятое должностным лицом или организацией в целом в ответ на запрос к этой организации (письмо или заявление частного лица, распоряжение государственного органа и др.), форма: документ или виза на другом документе;
- договор (контракт, соглашение, дополнительное соглашение и т.п.) с некоторым юридическим или физическим лицом;
- распорядительный или учетный документ, выпущенный для фиксации и юридического оформления действий в рамках договора или свободной инициативы (заказ на покупку, счет на оплату, запрос на оказание услуги и др.) и т.п.

Электронным офисом (ЭО) называется программно-аппаратный комплекс, предназначенный для обработки документов и автоматизации работы пользователей в системах управления.

В электронном офисе методы ручной обработки документов заменены автоматизированными процедурами. Как любая информационная система, ЭО имеет техническое, информационное и программное обеспечение.

В состав ***технического обеспечения*** ЭО входят следующие аппаратные средства: один или несколько компьютеров, возможно, объединенные в сеть; печатающие устройства; средства копирования документов; модем (если компьютер подключен к глобальной сети или к территориально удаленной ЭВМ). Дополнительно в состав аппаратных средств могут входить сканеры, используемые для автоматического ввода текстовой и графической информации непосредственно с первичных документов; стримеры, предназначенные для создания архивов на мини-кассетах на магнитной ленте; проекционное оборудование для проведения презентаций.

Информационное обеспечение ЭО предоставляет возможность производить интегрированную обработку всех видов информации, циркулирующей в организации, в том числе документов, порожденных электронным и бумажным документооборотом: внешней и внутренней переписки, осуществляемой как в электронной, так и в бумажной форме. База данных, хранящая все документы ЭО, становится элементом централизованной

базы данных организации и формируется как централизованный электронный архив документов, включающий в том числе и бумажные оригиналы, и электронные копии оригиналов бумажных документов¹.

Разработчики программного обеспечения ввели понятие электронного документа, под которым понимают сообщения, полученные по факсу или посредством электронной почты; отчет, созданный с использованием текстового редактора, звуковой файл или видеоклип и т.д.

Система управления базой данных документов обеспечивает:

- централизованную регистрацию всех документов, которые циркулируют в организации;
- хранение документов в электронном виде в различных форматах;
- ведение централизованного каталога документов организации, обеспечивающего возможность их поиска (по ключевым атрибутам, с использованием полнотекстового поиска и т.д.);
- хранение полной истории работы с документами (кто, когда и как работал с документом), а также различных версий документов;
- надежную систему защиты документов, регламентацию доступа персонала к документам различного назначения;
- возможность поддержки архивов документов на всех видах внешних устройств.

Прикладное программное обеспечение ЭО в общем случае включает в себя следующие ключевые компоненты:

- набор стандартных *бизнес-приложений*: текстовых процессоров, электронных таблиц и т.п. – и специализированных функциональных приложений, используемых сотрудниками организации для подготовки документов;
- систему управления хранением документов – программное обеспечение, реализующее функции управления единым документарным фондом организации (централизованным архивом);
- систему управления документооборотом – программное обеспечение, реализующее администрирование документооборота, управление маршрутизацией и движением документов, координацией документопотоков, контролем за передвижением документов, за своевременной их обработкой и т.д.;
- систему экспорта / импорта документов.

¹ Карабутов В. ЛАНИТ. www.citforum.ru : Разработка CASE-модели офисного документооборота: методология, технология, практическая реализация. Материалы конференции “Офисные информационные системы 96”.

На рис. 4 приведена архитектура электронного офиса (см. сноску на предыдущей странице).

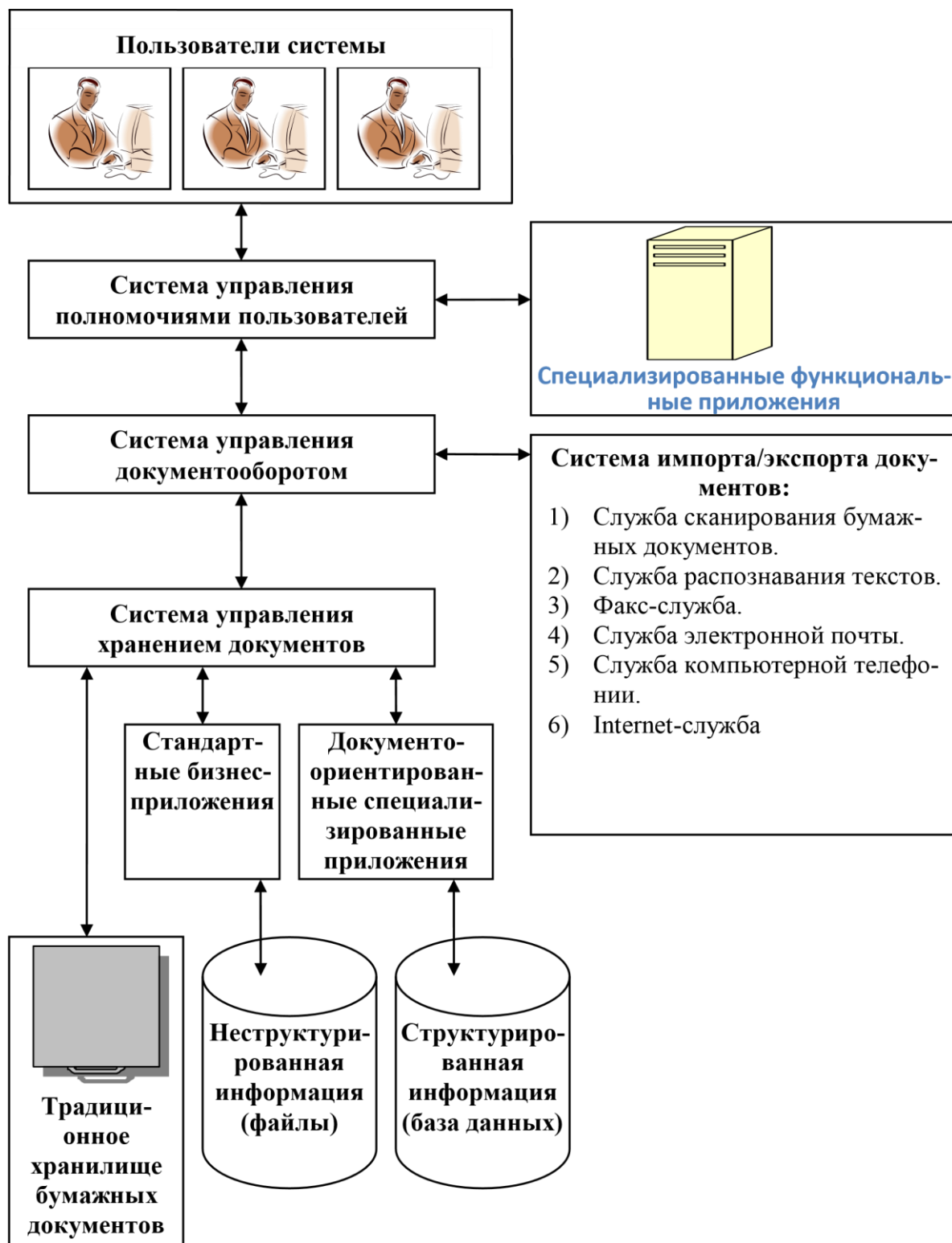


Рис. 4. Архитектура электронного офиса

4.1 Общие сведения

Бизнес-приложение – это программа или ППП, предназначенные для управления **бизнес-процессом**. М. Хаммер и Дж. Чампи¹ определяют бизнес-процесс как «совокупность видов деятельности (activities), которая имеет один или более видов входных потоков и создает выход, имеющий ценность для клиента». На рис. 5 приведен пример бизнеспроцесса покупки материалов². Он состоит из получения счета, его оплаты, получения материалов по накладной и их оприходования на склад. Этому сопутствует оформление и подписание определенного набора документов. Из подобных бизнес-процессов и состоит весь оперативный документооборот предприятия.

В состав программного обеспечения электронного офиса входят: □ функциональные и интегрированные пакеты офисной автоматизации;

- системы для организации групповой работы; □ системы управления электронными документами; □ средства управления документооборотом.

Основными программными продуктами, входящими в офис, являются:

- текстовый редактор;
- электронная таблица;
- система управления базами данных (СУБД).

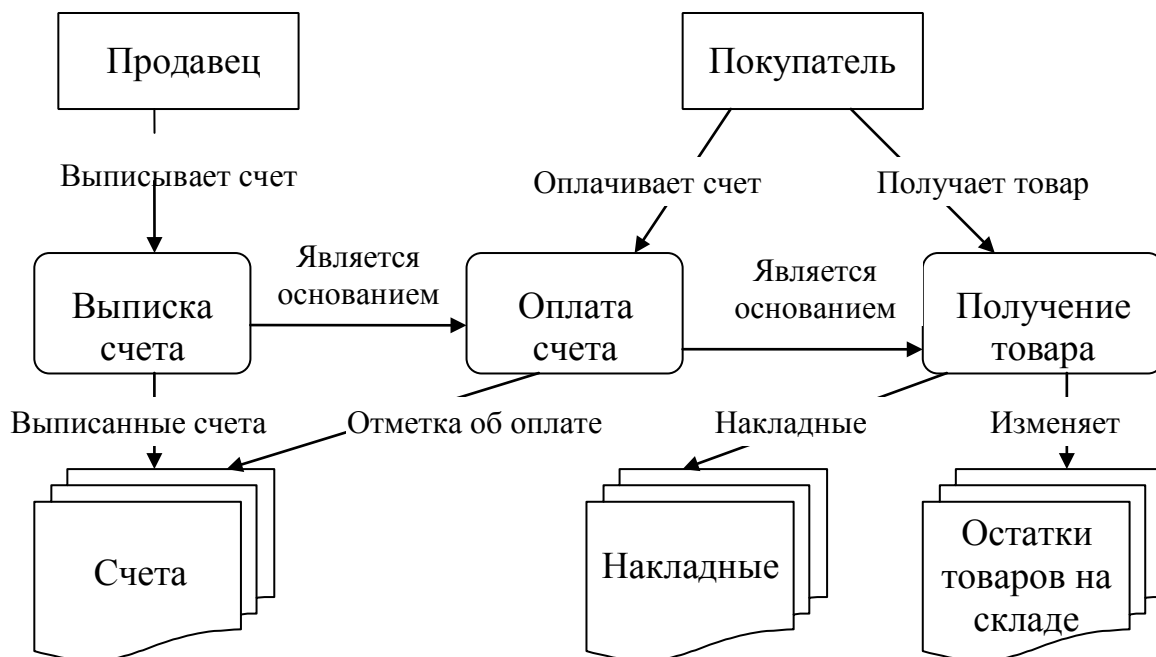


Рис. 5. Бизнес-процесс покупки материалов

¹ Hammer M., Champy J. Reengineering the Corporation. A Manifesto for Business Revolutions. HarperBusiness, 1993.

² Вольфман Б. ЭЛКО Технологии. www.citforum.ru: Технология разработки корпоративных систем с использованием современных инструментальных средств. Материалы конференции «Офисные информационные системы 96».

В состав программного обеспечения офиса могут также входить:

- программа анализа и составления расписаний;
- программа презентации;
- графический редактор;
- программа обслуживания факс-модема; □ сетевое программное обеспечение; □ программы перевода.

Офисные программные продукты используются как самостоятельно, так и в составе интегрированных пакетов.

4.2 Интегрированные пакеты для офиса

В интегрированный пакет (ИП) для офиса входят взаимодействующие между собой программные продукты. Основу пакета составляют текстовый редактор, электронная таблица и СУБД. Кроме них в интегрированный пакет могут входить и другие офисные продукты, перечисленные выше. Главной отличительной чертой программ, составляющих интегрированный пакет, является общий интерфейс пользователя, позволяющий применять одни и те же (или похожие) приемы работы с различными приложениями пакета. Взаимодействие программ осуществляется на уровне документов. Это означает, что документ, созданный в одном приложении, можно вставить в другое приложение и при необходимости изменить его. Общность интерфейса уменьшает затраты на обучение пользователей. Кроме того, цена комплекта из трех и более приложений, поддерживаемых одним и тем же производителем, значительно ниже, чем суммарная цена, если приобретать их по отдельности.

В настоящее время в России и других странах СНГ на рынке офисных программных продуктов доминирует пакет Microsoft Office. Фирма Microsoft постоянно совершенствует этот пакет, и в настоящее время в эксплуатации можно встретить три последние его *версии*: Microsoft Office 97, Microsoft Office 2000 и Microsoft Office XP. Каждая версия имеет несколько *выпусков*, наиболее распространенными из которых являются стандартный выпуск и профессиональный выпуск.

В *стандартный выпуск* Microsoft Office XP входят:

- **Microsoft Word 2002** – текстовый процессор;
- **Microsoft Excel 2002** – электронная таблица;
- **Microsoft PowerPoint 2002** – программа презентационной графики;
- **Microsoft Outlook 2002** – система управления электронной почтой и информацией.

В *профессиональный выпуск* дополнительно входит система управления базами данных (СУБД) **Microsoft Access 2002**.

Текстовый процессор Word, электронная таблица Excel и СУБД Access подробно рассмотрены в других лекциях.

Программа презентационной графики Microsoft PowerPoint позволяет создавать слайды и прозрачные пленки для проведения семинаров, конференций и т.д. Этот пакет позволяет указать шаблон презентации, т.е. однозначно определить ее стиль: шрифты, фоновую заставку, цветовую палитру и т.д. В PowerPoint включен целый набор масштабируемых иллюстраций, разбитых по тематике. Пользователь имеет возможность их редактировать и совмещать. В этом приложении есть готовая библиотека элементов мультимедиа. Оно позволяет включать в презентации таблицы, диаграммы, математические формулы и даже фрагменты видеоклипов. По готовому докладу можно предписать PowerPoint создать слайд-фильм. Пользователю остается добавить лишь оформительские детали. Дополнительно PowerPoint позволяет подготавливать заметки – информационные материалы, содержащие как слайды, так и пояснительный текст к ним, раздаваемые аудитории для лучшего усвоения материала.

Система управления электронной почтой и информацией Microsoft Outlook 2002 позволяет:

- работать с электронной почтой;
- спроектировать и вести личный календарь и систему группового планирования;
- организовать и вести хранилище персональной информации, включая контакты и задачи;
- разработать специализированные приложения для сотрудничества и совместного использования информации.

Microsoft Outlook 2002 помогает систематизировать, искать и просматривать всю эту информацию, сведенную воедино, используя для этого стандартный интерфейс.

В Microsoft Outlook 2002 предусмотрено пять различных типов представления информации – таблица, календарь, карточка, значки и временная шкала. Однако при необходимости на их основе можно создать неограниченное количество личных и разделяемых представлений. Таблицы, календарь и значки, по-видимому, хорошо знакомы большинству пользователей. Карточки напоминают набор визитных или каталожных карточек, кратко отображающих ключевую информацию, а временная шкала представляет собой горизонтальную ось времени, по которой в хронологическом порядке распределены события или задачи.

Microsoft Outlook 2002 позволяет пользователю:

- бегло просматривать список полученных сообщений, записи в календаре и текущие задачи; □ просматривать приглашения на собрания, поиск занятых или свободных интервалов времени;
- публиковать и загружать данные об интервалах занятости для планирования встреч, а также посылать, получать и отвечать на приглашения через Интернет;
- заказывать конференц-залы и другие ресурсы, необходимые для проведения собраний, без помощи специально выделенных компьютеров управления ресурсами;
- следить за всеми сообщениями электронной почты, задачами, встречами и документами, связанными с каждым контактом;
- бегло ознакомиться с самой свежей информацией: списком намеченных собраний, задачами, которые должны быть выполнены в текущие сутки, а также количеством почтовых сообщений, поступивших за ночь. Все эти данные выводятся в одном окне, что избавляет от необходимости переключаться между разными видами представления информации.

Подробные сведения об использовании Microsoft Outlook 2002 при работе с электронной почтой приведены в лекции «Глобальная компьютерная сеть Internet».

4.3 Организация документооборота в электронном офисе

В настоящее время большая часть информации на машинных носителях хранится в виде файлов различного формата, и для доступа к ним требуется применить программы, с помощью которых они были созданы. Этой информацией можно воспользоваться в полной мере лишь при условии, если имеются средства быстрого доступа к необходимым документам, возможность построения связей между различными материалами и эффективные механизмы поиска данных. Для решения этой задачи предназначены средства подготовки электронных документов и программы документооборота, позволяющие автоматизировать ручные операции и поиск документов, автоматически передавать и отслеживать перемещение документов и контролировать выполнение поручений, связанных с документами.

4.4 Средства подготовки электронных документов

Главной целью этих средств является создание документов, которые можно просматривать, аннотировать и распространять с помощью компакт-дисков, сетей или служб оперативного доступа к информации. В зависимости

от используемой ими технологии эти продукты можно разделить на четыре категории.

1. Средства подготовки документов на базе гипертекстового языка HyperText Markup Language (HTML). Именно эта технология лежит в основе системы доступа к информации World Wide Web.

2. Программные продукты на базе промышленного стандарта Standart Generalized Markup Language (SGML). Он представляет собой набор правил для описания структуры любого электронного документа.

3. Программные продукты, которые преобразуют документы из любого приложения в переносимый формат, благодаря чему можно читать, печатать и распространять документы без применения приложений, с помощью которых они были созданы. К данной группе принадлежат такие популярные системы создания электронных документов, как Acrobat компании Adobe и Command Ground фирмы No Hands Software.

4. Системы, позволяющие собрать из различных файлов информацию, записанную в свободной форме, и объединить ее в единую структуру, предоставляя в то же время возможность организовывать связи между соответствующими материалами и легко просматривать документы.

5. Для подготовки электронных документов используются специальные языки: HTML и SGML; специальные средства переноса документов, ППП создания гипертекстовых документов; пакеты для работы с формами документов.

6. Язык HTML – это стандарт, на котором основана World Wide Web. Благодаря ему можно форматировать документы и осуществлять связь текста и изображений с документом, находящимся на другом сервере WWW. HTML позволяет создавать документы путем вставки управляющих кодов (тегов) в текст для обозначения заголовков, названий, графических изображений и гипертекстовых связей, а также для форматирования текста. Помимо изображений, HTML поддерживает и формы, которые позволяют собирать и обрабатывать информацию, вводимую пользователями. Пользователь может ввести текст, выбрать требуемые элементы из списка, представленного в окне, и передать эту информацию на сервер.

7. Средствами доступа к информации службы WWW являются браузеры, которые обеспечивают просмотр документов и файлов в различных форматах и имеют средства обработки крупных мультимедиаобъектов, такие как звуковые, графические и видеофайлы.

8. Язык SGML является наиболее зрелым стандартом подготовки электронных документов. SGML предназначен для обеспечения быстрого доступа к информации, содержащейся в электронном виде в библиотеках, хранилищах данных, базах данных или размещенной в дисковых файлах. SGML

представляет собой набор правил для описания структуры и управления содержанием электронных документов.

9. Пакеты подготовки переносимых документов предназначены для пользователей, совместно применяющих документы, созданные различными приложениями. Все эти пакеты выполняют сходные функции: создание документа посредством какого-нибудь приложения и просмотр документа его автором или любым другим пользователем на экране компьютера с помощью специальной программы просмотра. Программа просмотра позволяет выводить документы на печать и копировать текст и графику в буфер, предоставляя тем самым возможность другому пользователю импортировать документ в свое приложение. Самым популярным пакетом подготовки переносимых документов является Acrobat компании Adobe.

10. Пакеты создания гипертекстовых документов предназначены для просмотра многочисленных и/или больших документов в поисках нужной информации, позволяя быстро переходить от одной части документа к другой и от документа к документу. Это достигается за счет гиперсвязей, т.е. структуры, объединяющей информацию, записанную в свободной форме в различных файлах. Можно создавать, поддерживать и обновлять любое количество гиперсвязей, объединяющих определения, графику и текст в любых распространенных форматах. Гипертекстовые системы, как правило, включают в себя два модуля, один из которых предназначен для редактирования, а другой для просмотра. В отличие от описанных выше средств создания электронных документов, эти системы не только позволяют просматривать документы, но и обладают развитыми средствами поиска информации по ключевым словам и отношениям, по полному или неполному совпадению и предоставляют возможность упорядочивать, отображать и создавать двунаправленные связи.

11. Распределение гипертекстовых документов в вычислительной системе выполняется посредством различных носителей, включая магнитные и оптические диски и сети, причем при работе в сети в системах подготовки гипертекстовых документов можно использовать средства, ограничивающие права пользователя на редактирование документов.

12. Основное преимущество гипертекстовых систем перед средствами создания переносимых документов состоит в том, что с их помощью легче строить составные документы из информации, записанной в различных файлах.

4.5 Методы автоматизации офиса

Основными фазами жизни информации в офисе являются¹:

- ввод информации в систему;
- хранение, навигация, поиск и фильтрация документов; □ коллективная работа с документами; □ вывод информации из системы.

Способы ввода информации в систему 1.

Сканирование документов.

В системах первого поколения графические образы введенных документов идентифицируются с помощью ключевых слов для последующего поиска необходимой информации (например, система SoftSolutions). Позднее стала применяться *технология оптического распознавания символов* OCR (Optical Character Recognition). После сканирования и ввода документа в систему его графический образ «переводится» в символьный текст, а затем следует исправление ошибок распознавания.

2. Ввод при помощи электронных форм.

Форма – это структурированный документ с незаполненными областями, в которые вводятся данные. Пользователь может создать форму и заполнить ее на бумаге или в приложении. В Microsoft Office можно создать формы, просматриваемые и заполняемые пользователями в Microsoft Word, Excel или Access. Этот вид форм можно пересылать посредством электронной почты или по сети. При создании таких форм можно использовать текстовые поля, флажки и раскрывающиеся списки.

Электронные формы используются при массовом ручном вводе однотипных документов. Они обеспечивают структуризацию документа путем выделения частей текста и добавления полей (атрибутов), что позволяет упростить заполнение документов и выполнить необходимые вычисления.

К преимуществам относятся также автоматическая проверка введенных данных (например, табельного номера сотрудника), обновление зависимых полей (например, полей города и области при вводе почтового индекса) и наличие подсказок, которые облегчают заполнение формы. Импорт файлов с магнитных носителей или по телекоммуникациям (факсы, сообщения электронной почты).

¹ Артемьев В. ЦБ РФ. www.citforum.ru : Методы и средства автоматизации учрежденческой деятельности. Материалы конференции «Офисные информационные системы 96».

4.6 Хранение, навигация, поиск и фильтрация документов

Под *навигацией* документов понимается организация и отслеживание перемещения документов. *Фильтрация* – это отбор документов по заданным признакам. Документы могут храниться в отдельных файлах, а средством группирования и навигации в хранилище документов является система каталогов (папок). Средства поиска и фильтрации файлов по их параметрам имеются в современных операционных системах и утилитах (Norton Commander, FAR и т.п.).

В системах, основанных на электронной почте, документы хранятся в *почтовых ящиках* в виде *почтовых сообщений с присоединенными файлами*.

Многие современные системы электронных документов используют в дополнение к файловой системе так называемые *библиотеки документов*, содержащие в базе данных (БД) карточки документов с атрибутами и ключевыми словами.

Поиск и фильтрация документов производится по запросам на основе *контекстного поиска*¹: по атрибутам, по ключевым словам и по полному содержанию текста.

Хранение информации во всемирной компьютерной сети *Internet* и в корпоративной сети *интранет* осуществляется в виде совокупности *гипертекстовых страниц*, распределенных по узлам сети. Каждая страница размещается в отдельном файле и представляет собой текст, размеченный с помощью языка гипертекстовой разметки HTML. Структуризация документа осуществляется путем форматирования, выделения полей, создания форм для диалогового заполнения документа и организации внутренних гипертекстовых ссылок. Допускается включение любой мультимедиа-информации: растровая графика, аудио, видео. Навигация по хранилищу гипертекста осуществляется с помощью внешних *гипертекстовых ссылок* на документы, расположенные на различных узлах сети (Web-серверах). Кроме того, для определения местонахождения документов служит *контекстный поиск*. Для ускорения поиска информации в Internet применяются специальные программы-браузеры, просматривающие Web-серверы и строящие некое подобие указателя. Использование гипертекста позволяет создать информационную инфраструктуру территориально распределенного учреждения и упростить

¹ *Контекст* – законченный в смысловом отношении отрывок письменной или устной речи, необходимый для определения смысла отдельного входящего в него слова или фразы.

диалоговый интерфейс пользователя, что наиболее важно при разработке информационных приложений для руководителей. Более подробные сведения о Web-технологиях изложены в Лекции «Глобальная компьютерная сеть Internet».

4.7 Коллективная работа с документами

Коллективная работа с документами строится на технологиях *groupware* и *workflow*.

Технологии *groupware* ориентированы на небольшие рабочие группы и характеризуются поддержкой выполнения одной коллективной задачи и отсутствием структуризации в организации работ. Поддержка ограничивается обеспечением коллективного доступа к информации с помощью различных **способов доступа**:

- сетевой доступ к файлам и базе данных;
- локальная и глобальная электронная почта (включая конференции и дискуссии);
- терминальный доступ, пересылка файлов и электронная доска объявлений;
- просмотр и интерпретация гипертекста (гипермедиа).

Технологии *workflow* служат для автоматизации документооборота в средних и крупных офисах, для них характерны:

- поддержка многопользовательской работы с несколькими задачами одновременно;
- четкая структуризация выполнения работ по ролям и документам с контролем исполнения.

Деловой процесс формализуется как совокупность состояний и переходов, необходимых для описания взаимодействия, как минимум, двух субъектов (например, сотрудников предприятия) для достижения выполнения заранее заданного условия. Частным случаем такого взаимодействия является простая пересылка документа из точки в точку.

Одной из реализаций технологии *workflow* является так называемая «система графов», где каждый шаг отражает движение задания, связанного с документом, или просто передвижение документа от одного субъекта к другому. При этом на человека, отвечающего за правильность функционирования схемы, ложится ответственность учета всевозможных непредвиденных (или отказных) ситуаций, которые могут возникнуть на пути движения документа.

Другая реализация основывается на понятии «цикл». В этом случае подразумевается, что наименьшим элементом в схеме взаимодействия является

цикл, учитывающий взаимодействие между двумя произвольными субъектами. При этом система сама отслеживает правильность протекания процесса и в случае ошибки показывает место некорректности с указанием ее причины, после чего прекращается воспроизведение нового процесса.

Установление правил взаимоотношений субъектов документооборота дополняется заданием безусловной и условной маршрутизации документов (по электронной почте) и времени обработки документа для контроля и учета исполнения.

Вывод информации осуществляют путем печати документов, публикации их на Web-серверах, в общих почтовых папках и электронных досках объявлений или рассылки по телекоммуникациям.

4.8 Обзор средств автоматизации учреждений

Информационно-программные средства автоматизации учреждений делятся на следующие категории:

- функциональные и интегрированные пакеты офисной автоматизации;
- системы для организации групповой работы; □ системы управления электронными документами; □ средства управления документооборотом.

4.9 Средства офисной автоматизации и коллективной работы в сети

Одним из наиболее распространенных средств офисной автоматизации является описанный выше интегрированный пакет Microsoft Office.

Для разработки и размещения прикладных программ, ориентированных на совместное использование информации, предназначена система *Lotus Notes*, представляющая собой платформу типа клиентсервер. Система Lotus Notes позволяет пользователям получать, отслеживать, совместно использовать и создавать информацию, предназначенную для документов. Эта информация может поступать в различных форматах, таких как тексты, изображения, видео и звук, и от различных источников, таких как компьютерные прикладные системы, оперативные системы, сканеры или факс-аппараты. Пользователям система Lotus Notes обеспечивает доступ к сети через любой применяемый ими графический пользовательский интерфейс (Windows, Mac, OS/2, Unix).

База документов Notes представляет собой средство хранения объектов, при помощи которого пользователи могут вызывать, отслеживать, хранить и преобразовывать информацию в своей сети. База документов может совместно эксплуатироваться пользователями, присоединенными к одной и той же сети.

4.10 Средства управления электронными документами

Стержнем любой системы управления электронными документами является архив, где документы находятся в процессе работы над ними и где они остаются до тех пор, пока содержащаяся в них информация представляет интерес. Под электронным архивом понимается совокупность аппаратно-программных средств и технологий для создания хранилища электронных документов и обеспечения доступа к ним из систем управления электронными документами. Электронные документы – это все документы, созданные электронными средствами в виде текстовых или графических файлов.

4.11 Система управления документами DOCS OPEN

Программный продукт DOCS OPEN (компания PC DOCS Inc.) позволяет организовать электронный архив на предприятии. Система предназначена для хранения, поиска и обработки информации, хранящейся в распределенной гетерогенной среде на накопителях различной природы.

Система построена по архитектуре «клиент-сервер». DOCS Open имеет минимум два сервера: сервер библиотеки, который хранит карточки документов, и сервер документов, хранящий сами документы; оба сервера могут с успехом функционировать на одной машине. Сервер документов может строиться на основе любой сетевой операционной системы. Основное внимание фирма уделяет поддержке Novell NetWare и Windows NT.

Схемы хранения документов в DOCS Open основаны на сетевой файловой системе и иерархической системе управления хранением файлов. Документы хранятся в файлах, которые размещаются на файловом сервере. Предусмотрена возможность полуавтоматического удаления редко используемых документов. Документы редактируются непосредственно по месту хранения. Каждый документ в системе DOCS Open снабжается учетной карточкой. DOCS Open позволяет находить документы по содержанию документа.

Поисковые возможности DOCS Open основаны на интерфейсе запроса по образцу QBE (Query By Example). Для того чтобы найти документ, необходимо заполнить предполагаемую карточку документа. В ответ система выдает список документов, соответствующих введенным данным. Пользователи имеют возможность объединять документы в папки.

DOCS Open поддерживает распределенную обработку документов. Поэтому документы всегда находятся на тех серверах, куда они первоначально были положены. Документ передается пользователю только в тот момент, когда он ему нужен для обработки.

При редактировании документ видоизменяется прямо по месту своего хранения. DOCS Open позволяет редактировать с временным копированием документов на локальный диск.

Наиболее сильной стороной DOCS Open является возможность ведения распределенных и удаленных архивов информации.

DOCS Open обладает развитыми *средствами защиты документов*. Система может определять права доступа к карточке документа и к самому телу документа.

4.12 Система управления документами Excalibur EFS

Другим примером системы управления электронными документами является продукт Excalibur EFS фирмы Excalibur Technologies Corp. Серверное программное обеспечение данной системы функционирует в среде операционной системы Unix, а клиентские рабочие места могут работать под MS Windows. В качестве базы данных учетных карточек могут применяться СУБД Oracle, Informix, Sybase и Ingres. Эта система использует новейшие технологии нейронных сетей и искусственного интеллекта и основанные на них метод нечеткого поиска по полному содержанию документа и «компактный» способ индексирования (30% исходного текста против 70–100% для полнотекстового поиска). Нечеткий поиск сокращает до минимума влияние ошибок распознавания символов, ошибок набора на клавиатуре при вводе данных, а также ошибок правописания в запросах поиска. Пакет базируется на оригинальной технологии адаптивного распознавания образов APRP и реализован с помощью механизма нейронных сетей. Эта технология обеспечивает автоматическую индексацию всего содержания документа, что исключает необходимость выбирать вручную ключевые слова и дает возможность нечеткого поиска любого слова в документе.

4.13 Средства автоматизации документооборота

В последнее время в зарубежных странах пользуются особой популярностью автоматизированные системы построения и управления деловыми процессами в организациях. С помощью таких систем можно организовать систему электронного документооборота на предприятии, а также систему контроля выполнения заданий и загрузки сотрудников. Данные системы относятся к типу workflow. Следует отметить, что они ставят целью не полный отказ от бумажных документов (что невозможно по ряду причин, главными из которых являются причины юридического характера), а сведение к минимуму перемещение бумаг внутри предприятия.

Пакет управления документооборотом Action Workflow

Зачастую возникает потребность в организации определенных маршрутов работы над документами (маршрутизация) и контроля исполнительской дисциплины. Когда нет устоявшихся маршрутов обработки информации и желательно осуществить последовательную или параллельную рассылку необходимой документации, можно воспользоваться «свободным маршрутизатором», созданным и интегрированным в систему DOCS Open. Посредством этого маршрутизатора можно из списка пользователей системы или групп пользователей выбрать адресатов корреспондентов, назначить длительность работы, просмотреть статус работ (получена / прочитана / выполнена и т.д.), проследить историю работы над той или иной информацией. В случае же, когда имеются относительно стабильные маршруты хождения документов, предпочтительнее использовать возможности продукта Action Workflow компании Action Technologies.

Пакет электронного документооборота Staffware

Этот пакет работает по технологии «клиент-сервер», относится к классу workflow и характеризуется следующими особенностями:

- поддержка коллективной работы с множеством заданий большого числа исполнителей;
- динамическое управление и контроль исполнения работ; □
интегрируемость с разными приложениями под Windows и Unix; □
использование различных платформ.

StaffWare позволяет простым способом маршрутизировать документопоток и контролировать исполнение документов по времени и дате исполнения. StaffWare способна интегрироваться с другими продуктами, поддерживающими автоматизированный ввод входящих документов, ведение архива документов, криптографическую защиту информации и проч. Для мощных систем документооборота имеются версии StaffWare on Oracle и StaffWare on Infirmix.

В StaffWare встроен удобный графический построитель процедур, наглядно отражающий маршрут прохождения документов и алгоритм их обработки. Система электронного документооборота с использованием инструментальной среды StaffWare представляет собой множество процедур, которое может модифицироваться и пополняться. Каждая процедура связывает совокупность документов, из которых одни являются родителями, другие – потомками. Отдельный документ может быть родителем одних документов и потомком других. Отношения между документами, с одной стороны, их связь с процедурами и внутреннее описание каждого документа, с другой стороны, однозначно идентифицируют его в общем потоке документов. Этот идентификатор может использоваться для организации архива документов с целью быстрого поиска. Документы-потомки порождаются при исполнении

шагов процедур StaffWare в любых доступных приложениях MS Windows, включая текстовые процессоры, электронные таблицы, процессоры мультимедиа и т.п.

В заключение можно сказать, что комбинация методов и средств офисной автоматизации, систем управления базами данных (ввод, хранение и поиск структурированной информации), систем workflow (управление, маршрутизация и координация передвижения документов, контроль за своевременностью их обработки) и систем управления электронными документами (ввод, хранение, поиск неструктурированной информации), а также интеграция программных продуктов, реализующих эти методы, обеспечивает комплексную автоматизацию учрежденческой деятельности.

4.14 Контрольные вопросы

1. Офисные задачи.
2. Виды деятельности, осуществляемые в офисе.
3. Типовые процедуры, выполняемые при решении офисных задач.
4. Документы, используемые в офисе.
5. Понятие электронного офиса.
6. Техническое, информационное и программное обеспечение электронного офиса.
7. Интегрированные пакеты для офиса; пакет Microsoft Office.
8. Организация документооборота в электронном офисе.
9. Методы автоматизации офиса.
10. Коллективная работа с документами.
11. Средства управления электронными документами.
12. Средства автоматизации документооборота.

Лекция . Текстовые редакторы и процессоры

4.15 6.01 Назначение и классификация текстовых процессоров

Каждый пользователь компьютера встречается с необходимостью подготовки тех или иных документов – писем, статей, служебных записок, отчетов, рекламных материалов, полной подготовки к типографскому изданию книг и журналов (издательские системы). Удобство и эффективность применения компьютеров для подготовки текстов привели к созданию множества различных программ для обработки документов. Такие программы называются текстовыми процессорами (Word Processors). Основные функции этого класса прикладных программ заключаются в вводе и редактировании текстов. Дополнительные функции состоят в автоматизации процессов ввода и редактирования.

Текстовый процессор – компьютерная программа, используемая для написания и модификации документов, компоновки макета текста и предварительного просмотра документов в том виде, в котором они будут напечатаны (свойство, известное как WYSIWYG)

Современные текстовые процессоры, помимо форматирования шрифтов и абзацев и проверки орфографии, включают возможности, ранее присущим лишь настольным издательским системам, в том числе создание таблиц и вставку графических изображений. Наиболее известными примерами текстовых процессоров являются Microsoft Word, Apple iWork Pages, OpenOffice.org Writer.

Более совершенные текстовые процессоры имеют целый спектр возможностей по созданию документов (например, поиск и замена символов, средства проверки орфографии, вставка таблиц и др.), их называют иногда текстовыми процессорами. Примером такой программы является Word из офисного пакета Microsoft Office. Хороший интерфейс, привлекательный и понятный пользователю, во многом определяет успех предлагаемого решения. Диалог с пользователем – важная часть интерфейса. Во многих случаях для организации диалога достаточно обычных диалоговых окон, открываемых в нужный момент с выдачей подходящего сообщения.

Мощные программы обработки текста – настольные издательские системы – предназначены для подготовки документов к публикации. Пример подобной системы – Adobe PageMaker.

Объектно ориентированный подход дает возможность реализовать механизм встраивания и внедрения объектов (OLE – Object Linking Embedding). Этот механизм позволяет копировать и вставлять объекты из одного

приложения в другое. Например, работая с документом в текстовом редакторе Word, в него можно встроить изображения, анимацию, звук и даже видеофрагменты и таким образом из обычного текстового документа получить мультимедиа-документ.

Текстовые процессоры имеют много возможностей, например форматирование – преобразование, изменяющее форму представления документа. В начале работы над документом целесообразно задать параметры страницы: ее формат (размер), ориентацию, размер полей.

Формат файла определяет способ хранения текста в файле. Простейший формат текстового файла (TXT) содержит только символы (числовые коды символов), другие же форматы (DOC, RTF) содержат дополнительные управляющие числовые коды, которые обеспечивают форматирование текста.

Основное назначение текстовых редакторов – создавать текстовые файлы, редактировать тексты, просматривать их на экране, изменять формат текстового документа, распечатывать его на принтере.

1. Качество печатной машинки, небольшой набор возможностей по работе с текстом: Norton Editor, Фотон, Лексикон, MultiEdit, Chiwriter.

2. Издательское качество. Реализация принципа WYSIWYG – What You See Is What You Get: Microsoft Word, Ventura Publishers, Aldus Page Maker.

3. Технические редакторы, например Tex, Latex.

Помимо средств оформления текста, текстовые редакторы часто снабжают дополнительными утилитами, облегчающими работу с документом: средствами поиска и замены; проверки орфографии, пунктуации; средствами работы с буфером обмена; справочной системой по программе; средствами автоматизации (написание сценариев или макросов).

Таким образом, мощный текстовый процессор состоит из текстового окна для ввода текста, библиотеки элементов форматирования, интерпретатора этих элементов, ряда вспомогательных программ для создания и форматирования внешних объектов и набора утилит, облегчающих работу с документом. Мощные процессоры снабжают набором конверторов, которые переводят элементы другого текстового редактора в команды данного.

Наиболее известные ныне редакторы текстов можно условно разделить «по специализации» на три группы:

- процессоры общего назначения (Лексикон, Microsoft Word, WordPerfect и др.);
- редакторы научных документов (ChiWriter, TEX и др.);
- редакторы исходных текстов программ (Multi-Edit и встроенные редакторы систем программирования BASIC, Pascal, Си).

6.02

Основные функции текстовых процессоров:

- создание документов;
- редактирование документов: перемещение по тексту, вставками замена символов, удаление, перемещение, копирование, поиск и замена фрагментов текста, отмена команд; вставка фрагментов других документов или целых документов и т.д.;
- сохранение документов во внешней памяти (на дисках);
- чтение из внешней памяти;
- форматирование документов, т.е. выполнение преобразований, изменяющих форму (внешний вид) документа: оформление отдельных символов и абзацев, страниц, документа в целом – изменение длины строки, межстрочного расстояния, выравнивания текста, изменение шрифта, его размера, применение различного начертания шрифтов и т.д.;
- печать документов (или их некоторой части);
- автоматическое составление оглавлений и указателей в документе;
- создание и форматирование таблиц;
- внедрение в документ рисунков, формул и др.; □ проверка пунктуации и орфографии.

Другие функции обработки текста включает проверку орфографии (на самом деле проверки в отношении Словарные), «грамматики» (чеки на, казалось бы, простые грамматические ошибки), и «Тезаурус» функции (находит слова с аналогичными или противоположными значениями). Другие общие функции включают в себя совместное редактирование, комментарии и аннотации, поддержку изображений и диаграмм и внутренних перекрестных ссылок.

Текстовые процессоры можно отличить от ряда других связанных с ними форм программного обеспечения.

Текстовые редакторы были предшественниками текстовых процессоров. Текстовые редакторы в настоящее время используются в основном для программистов, веб-сайт дизайнеров, администраторов компьютерной системы и, в случае LaTeX, математиков и ученых (для сложных формул и ссылок в редких языках).

Почти все текстовые процессоры позволяют пользователям использовать стили, которые используются для автоматизации форматирования текста тела, заголовков, подзаголовков, выделенного текста.

Стили позволяют значительно упростить управление форматированием больших документов, так как изменение стиля автоматически изменяет весь текст. Даже в коротких документах применение стилей может сэкономить много времени при форматировании.

Кроме того, современные программы позволяют включать в текст, графические объекты: рисунки, диаграммы, фотографии. Таким образом, от понятия «простой текст как объект обработки текстового редактора» мы приходим к понятию «текстовый документ как объект обработки текстового процессора». Файл, представляющий собой текстовый документ, содержит, помимо алфавитно-цифровых символов, обширную двоичную информацию о форматировании текста, а также графические объекты.

Начинающих пользователей иногда запутывает различное толкование английских терминов «Text Editor» (Редактор текстов), «Text Processor» (Текстовый процессор) и «Word Processor» («Процессор слов»). Текстовым процессором обычно называют мощный текстовый редактор, располагающий развитыми возможностями по обработке текстовых документов.

При выборе текстового редактора для работы вы должны учитывать многие факторы: и сложность ваших документов, и масштаб (объемы) текстов, и требования к качеству документа на бумаге, и характер материалов (например, простая беллетристика или таблицы, формулы, уравнения и т.п.).

4.16 6.03 AkeIpad + Portable

AkeIpad – текстовый редактор с открытым исходным кодом, созданный, чтобы быть маленьким и быстрым. Возможности:

- Однооконный режим (SDI), многооконный режим (MDI).
- Редактирование файлов размером более 64К (вообще, размер редактируемого файла теоретически не ограничен). □ Полная поддержка Unicode строк на Unicode системах.
- Работа с кодировками Unicode (UTF-16 little endian, UTF-16 big endian, UTF-8). □ Работа с любой кодовой страницей, установленной в системе.
- Работа с DOS/Windows и Unix форматами перевода строки (в том числе сохранение).
- Предпросмотр открытия файлов.
- Корректное отображение псевдографики.
- Редактирование файлов с атрибутом "Read Only".
- Предупреждение при попытке открыть бинарный файл.
- Многоуровневый откат действий.
- Поиск/замена строк текста, escape-последовательностей, а также поиск/замена по шестнадцатичному коду.
- Запоминание кодировки файла.
- Запоминание позиции каретки в файле.
- Список последних строк поиска/замены.
- Список последних открытых файлов.

- Поддержка плагинов.
- Поддержка языковых модулей.
- Печать.

Язык интерфейса: Многоязычный (в т.ч. русский). Тип лицензии: Open Source.

4.17 6.04 Популярные текстовые редакторы в России

Самым популярным текстовым редактором в России был до недавнего времени Лексикон, созданный более десяти лет назад Е.Н. Веселовым. Лексикон прост в обращении и эксплуатации, позволяет готовить сложные документы и обладает массой других достоинств, благодаря которым он вполне удовлетворяет большинство российских пользователей.

Разумеется, с помощью Лексикона можно подготовить и текст программы, а с помощью Multi-Edit – документ общего назначения. «Специализация» редактора заключается в том, что в нем добавлены (или оптимизированы) функции, которые необходимы для обслуживания документов определенного типа. Например, Multi-Edit позволяет выделять цветом смысловые сегменты исходных текстов программ, ChiWriter удобен для набора математических выражений и т.д. Кроме того, особое место в группе пакетов, работающих с документами, занимают так называемые издательские системы – Aldus PageMaker, Corel Ventura, QuarkXPress. Как правило, издательские системы используются только для подготовки набранного документа к тиражированию (верстка, макетирование издания). Для набора текста удобнее применять текстовые процессоры (типа Word), а для создания иллюстраций – графические системы (типа CorelDRAW!).

4.18 6.05 Текстовый процессор Word

1. Word – это приложение Windows

В нашем курсе мы изучаем основы редактирования документов на примере текстового процессора Word фирмы Microsoft. Это связано не только (и не столько) с исключительными возможностями этой программы. Дело в том, что Word – один из основных элементов офисной технологии.

Word – это приложение Windows, предназначенное для создания, просмотра, модификации и печати текстовых документов (особенности версии Word 2000, появившейся в 1999 году на российском рынке, не рассматривается в силу пока еще малой распространенности).

Word – одна из самых совершенных программ в классе текстовых процессоров, которая предусматривает выполнение сотен операций над текстовой и графической информацией. С помощью Word можно быстро и с высоким качеством подготовить любой документ – от простой записки до оригинал-макета сложного издания. Издательские возможности процессора Word так велики, что почти любую книгу можно подготовить к печати, не прибегая к услугам издательских систем.

Во-первых, Word дает возможность выполнять все без исключения традиционные операции над текстом, предусмотренные в современной компьютерной технологии:

- набор и модификация неформатированной алфавитно-цифровой информации;
- форматирование символов с применением множества шрифтов True Type разнообразных стилей и размеров;
- форматирование страниц (включая колонтитулы и сноски); □ форматирование документа в целом (автоматическое составление оглавления и разнообразных указателей);
- проверка правописания, подбор синонимов и автоматический перенос слов.

Во-вторых, в процессоре Word реализованы возможности новейшей технологии связывания и встраивания объектов, которая позволяет включать в документ текстовые фрагменты, таблицы, иллюстрации, подготовленные в других приложениях Windows. Встроенные объекты можно редактировать средствами этих приложений.

В-третьих, Word – одна из первых общедоступных программ, которая позволяет выполнять многие операции верстки, свойственные профессиональным издательским системам, и готовить полноценные оригинал-макеты для последующего тиражирования в типографии.

В-четвертых, Word – это уникальная коллекция оригинальных технологических решений, которые превращают нудную и кропотливую работу по отделке текста иногда в увлекательное, а иногда даже в успокаивающее занятие. Среди таких решений – система готовых шаблонов и стилей оформления, изящные приемы создания и модификации таблиц, функции автотекста и автокоррекции, форматная кисть, пользовательские панели инструментов, макроязык и многие другие.

2. Microsoft Word 2010

Microsoft Word 2010 – это средство для создания офисных документов с массой функций для их улучшения. В среде этого приложения вы можете

создать документ и добавить на него самые разнообразные объекты (рисунки, таблицы, диаграммы, фигуры, эффектный текст), придавая им неповторимый стиль. В связке с приложением Office Communicator вы и ваши друзья сможете редактировать документ в режиме on-line, а также общаться между собой, не выходя из Word. С помощью первоклассного, интуитивно понятного интерфейса вы легко овладеете всеми навыками работы в программе.

Особенностью Microsoft Office 2010 является Microsoft DirectAccess – Помощник (DCA) помогает организациям сократить расходы на поддержку пользователей и значительно улучшить их совместную работу над документами. Это является частью Windows Optimized Desktop Toolkit 2010 (WODT 2010 года). DCA – сообщает мобильным пользователям их статус соединения на все времена, предоставляет инструменты, чтобы помочь им восстановить на своих собственных, если возникают проблемы, и создает диагностики, чтобы помочь мобильным пользователям получать ключевую информацию, если нужно, – все, чтобы помочь клиентам работать с большей эффективностью и с меньшими затратами. DCA это новейшее дополнение к Windows Optimized Desktop Toolkit 2010 года.

Идея Помощников (Мастеров, Агентов) не нова. Сегодня эта технология приобрела широкую популярность. В процессоре Word пользователю отводится активная роль, и поэтому очень полезными оказались Помощники пользователя. Пользователь будет выбирать, какую задачу он хочет решить в тот или иной момент, а соответствующий Помощник поможет ему на этом пути, проводя пользователя через все этапы решения задачи, снабжая необходимой информацией, информируя о возникших ошибках, позволяя сделать откат и вернуться к предыдущему этапу. В процессоре Microsoft Word подобные Помощники или Мастера (Assistants, Wizards) сопровождают нас на каждом шагу.

Кроме того, что Помощники предназначены для обеспечения помощи в решении некоторой конкретной задачи и могут вести диалог с пользователем, они могут обладать еще двумя важными качествами:

- иметь видимый образ, возможно допускающий анимацию. Помощник с анимацией, делает приложение одушевленное;

- обладать определенной интеллектуальностью. «Интеллектуальный» Помощник должен уметь собирать факты в процессе работы пользователя, иметь базу знаний, содержащую факты, и правила, и иметь встроенную машину вывода. Такой Помощник может делать собственные выводы, брать на себя часть творческой работы, например по заданию части параметров, или приходить на помощь в нужный, по его мнению, момент.

Помощник, встроенный в справочную систему Office, и построенный на основе объекта Assistant, обладает всеми тремя качествами: выдает справки,

имеет видимый образ и обладает интеллектуальностью. Следует отметить, что его интеллектуальные способности оставляют желать лучшего.

Каркас документа Word строится автоматически в момент его создания. Из общей совокупности библиотек объектов, составляющих Office, в каркас документа входят те библиотеки, чьи объекты необходимы для построения документа данного типа. Часть из библиотек содержат общие объекты и входят в каркасы документов всех типов, создаваемых в Office. Часть библиотек каркаса определяют специфику документа.

Normal – проект, доступный для всех документов Word. Здесь могут храниться функции и классы, используемые всеми документами. Но при работе с документом этот каркас можно существенно изменить, добавив в документ новые классы, а значит, и новые свойства. Для этого достаточно включить в состав каркаса другие библиотеки классов из той совокупности, что входит в Office.

Технология работы в Office рекомендует создавать документы на основе некоторого шаблона. Поэтому, однажды создав документ с требуемым набором возможностей и сохранив его в качестве шаблона, в последующем можно открывать новые документы на основе этого шаблона, не задумываясь уже о том, как и какие библиотеки следует подключать.

Каркас любого документа Word, как уже говорилось, составляют объекты разных классов, объединенные в библиотеки. Для документов Word основной библиотекой, конечно же, является библиотека классов Word. Центральным объектом каркаса документа является объект этой библиотеки Application, определяющий само приложение. Поскольку в основе построения объектов Office лежит встраивание, то объекты Office являются, как правило, «толстыми» объектами, в каждый из которых встроены другие объекты.

Объект Application это один из самых «толстых» объектов, в него встроены объекты, задающие различные компоненты приложения Word. В программном проекте любого из открытых документов Word доступен корневой объект Application, определяющий само приложение. Одновременно становятся доступными и все встроены в него объекты. В частности, становится доступной коллекция Documents всех открытых документов Word, а тем самым и объект, задающий наш, например, только что открытый документ.

В приложении Word на разных уровнях иерархии определено около 200 объектов, совокупность которых и определяет мир объектов Word, его возможности, свойства и поведение. Сам корневой объект Word.Application имеет более сотни элементов: свойств, методов и событий. Объект Application, задающий приложение, естественно, определяет свойства и поведение приложения в целом.

На главной вкладке Word, помимо основных функций, появилось две новых – параметры анимации в группе «Шрифт» и дополнительные надстройки стилей в группе «Стили».

Параметры анимации. К выделенному тексту можно применить различные визуальные эффекты, такие как тень, свечение и отражение. В программе доступно 20 готовых шаблонов оформления текста, а также можно создавать свои.

3. Графическая технология Word

Следует помнить, что Word использует графическую технологию и шрифты TrueType, поэтому некоторые вопросы редактирования решаются в Word принципиально иначе, чем в простых текстовых редакторах. Например, для оформления абзацев, таблиц, рисунков используются линии, рамки, узоры и другие элементы графической природы, не имеющие аналогов в текстовом режиме.

4. Недостатки Word

Довольно длинный (хотя и неполный) перечень достоинств и преимуществ Word по традиции должен сопровождаться списком недостатков этой программы. К сожалению, общепризнанных изъянов Word не имеет, однако, как и Windows, он является слишком универсальной программой. А это означает, что некоторые процедуры проще и быстрее выполнить в менее мощных, но более специализированных приложениях.

Как считают многие пользователи, основной недостаток Word – низкая производительность при наборе чернового текста (по сравнению с простыми редакторами (MS-DOS)). По замыслу разработчиков, Word обречен обслуживать сотни действий пользователя и потому выглядит иногда «тугодумом» при исполнении даже простых функций. Если вы постоянно заняты подготовкой крупных блоков сравнительно однородной текстовой информации (например, статей или брошюр на гуманитарные темы), вас вполне устроит описанный выше основной набор операций (ввод и модификация текста, работа с фрагментами, поиск и замена). А эти операции гораздо проворнее выполняет простейший текстовый редактор (например, MS-DOS Editor, Norton Editor). Подготовленный текстовый файл можно преобразовать в файл формата Word, а затем форматировать документ средствами Word. Во многих случаях последняя операция займет у вас несколько минут. Другой «недостаток» Word – высокая трудоемкость при вводе сложных математических выражений и химических формул. Кроме того, Word не

предназначен для изготовления полиграфической продукции особо сложной структуры (атласов, альбомов, журнальных обложек), а также для редактирования высококачественных иллюстраций. В зависимости от своих потребностей и мощности ПК пользователь может выбрать один из нескольких вариантов установки Word: минимальный, выборочный, стандартный, полный. Мы предполагаем, что вы используете как минимум стандартную конфигурацию Word.

5. Создание, загрузка и сохранение файлов-документов

Если вы запустили Word без аргумента командной строки (т.е. не указав имя файла), процессор по умолчанию предлагает вам начать создание нового документа под условным наименованием Document1 (Документ1). Шаблон этого документа (Normal) хранится в стандартном файле NORMAL.DOT. Вы можете подготовить документ (или часть документа), а затем сохранить его на диске (команда (File-Save As)) в виде файла с произвольным именем и расширением .DOC.

Сохранение файлов-документов выполняется по стандартным правилам Windows. С помощью команды (Tools-Options) (Сервис-Опции) вы можете заранее указать промежуток времени (например, 20 минут), по истечении которого Word будет автоматически сохранять ваш документ. Если вы хотите продолжить обработку уже существующего файла документа, выберите команду (File-Open). При сохранении и загрузке документов не забывайте о правильной установке каталога, в который помещается или из которого читается файл DOC,

Иногда вы редактируете попеременно несколько файлов документов. Процессор запоминает имена нескольких файлов, которые вы обрабатывали за последнее время, и выдает их на экран в качестве команды пункта меню File. Для открытия любого из таких файлов достаточно щелкнуть мышью на его имени. Это очень удобно, так как избавляет вас от довольно нудной работы с файлером. Число запоминаемых файлов (например, 4) надо указать на вкладке General диалогового окна (Tools-Options).

Если вы решили создать новый документ, выберите команду (FileNew). На экране появится диалоговое окно New со списком Template (Шаблон), из которого вы можете выбрать шаблон для создаваемого документа (например, стандартный шаблон Normal).

В отличие от Paintbrush, Word является многооконным процессором. Это означает, что вы можете одновременно открыть несколько документов, причем часть из них может быть готовыми файлами, а часть – заготовками (без имени, но с обозначениями Document1, Document2 и т.д.). Максимальное число одновременно открытых документов определяется памятью машины, вашими

потребностями и способностью лавировать при работе между несколькими окнами. Напомним основные правила работы с окнами документов.

1. Для любого окна документа, открытого в приложении, можно избрать любой вариант представления (полный экран, нормальный, пиктограмма). Окно в нормальном варианте может иметь любой размер и размещаться в любом месте в пределах окна приложения. Пиктограмма может размещаться в любом месте окна приложения.

2. Одновременно вы можете разместить на экране для обозрения и работы любое число открытых документов: например, один (остальные окна будут закрыты), два и т.д.

3. В данный момент вы можете работать только с одним окном, которое называется окном активного документа (заголовок такого окна выделен синим цветом). Именно к активному документу относятся все команды, подаваемые вами. Выбирая такие команды, как Save, Print, Close и т.п., вы должны четко осознавать, какой документ вы сохраняете, печатаете, закрываете и т.п.

4. Чтобы переключиться на работу с другим окном, вы должны либо щелкнуть на нем мышью (если хотя бы краешек его виден на экране), либо выбрать команду системного меню окна документа, либо выбрать команду горизонтального меню (Window «окно»).

5. Для обмена текстовой и графической информацией между окнами открытых документов можно использовать буфер обмена, а также метод «Drag-and-Drop». Кроме того, Word позволяет разбить (расщепить) окно документа на два подокна (команда (Window-Split)) и одновременно работать с разными частями одного и того же документа.

6. Окно процессора Word

Окно приложения Word (совместно с окном документа) содержит все стандартные элементы, рассмотренные нами. Показаны зона заголовков, горизонтальное меню, четыре панели инструментов (три вверху и одна внизу), полосы прокрутки, линейки и строка состояния.

Перечислим элементы окна, специфичные для процессора Word.

На горизонтальной линейке «укреплены» треугольники. Верхний треугольник (слева) предназначен для установки отступа в красной строке абзаца, а пара нижних треугольников (слева и справа), как правило, применяются для установки отступов абзаца от границ страницы (т.е. левой и правой границ абзаца). Эти отступы можно установить, передвигая треугольники вдоль линейки методом «Drag-and-Drop». Кроме того, с помощью нижних треугольников можно переместить границы страницы и разделители столбцов таблицы.

В левой части горизонтальной полосы прокрутки находятся три кнопки выбора режима отображения Normal, Page Layout и Outline (слева направо).

Среди переключателей, расположенных в правой части строки, мы рассмотрим только один – OVR (в русской версии ЗАБ – забой). Если название этого переключателя имеет серый цвет, в Word установлен режим вставки символов; если цвет черный, Word работает в режиме замещения («забоя») символов.

7. Обзор функций горизонтального меню

В пункте File собраны команды, с помощью которых мы работаем с файлами-документами. Кроме уже известных нам стандартных команд Open, Close, Save, Save As, Print, в этом пункте имеется несколько команд, специфичных для процессора Word, например. Summary Info (библиографической карточки документа) Print Preview (Предварительный просмотр документа перед печатью) и др. В пункте Edit (Правка, редактирование), кроме уже известных нам стандартных команд Undo (отменить операцию), Cut, Copy, Paste (работа с буфером обмена), мы видим команды поиска и замены фрагментов текста (Find и Replace), команду перехода к заданному элементу текста (Go To – к странице, строке, сноске и т д), команду создания автотекста (Auto Text) и др.

Пункт View (Вид, просмотр) позволяет управлять отображением документа и его элементов на экране. Для нас наиболее важными в этом пункте являются:

- группа полей выбора, управляющая режимом отображения документа в целом Normal (Нормальный), Outline (Структура документа) и Page Layout (Разметка страницы);
- команда Toolbars (Панели инструментов), позволяющая управлять отображением и составом различных панелей инструментов,
- переключатель горизонтальной линейки Ruler;
- команды просмотра и редактирования колонтитулов и сносок (Footnotes).

Группа команд Insert (Вставка) предназначена для вставки в документ различных объектов:

- линий разрыва страниц, колонок, разделов (Break);
- номеров страниц в том или ином формате (Page Numbers);
- произвольных символов (Symbol);
- сносок (Footnote);
- указателей и оглавления (Index and Tables); □ рисунков (Picture); □ объектов обработки других программ (Object).

В пункте Format собраны очень важные команды, с помощью которых мы форматируем наш документ:

- форматирование символов и абзацев (Font и Paragraph); □ обрамление и заполнение абзацев, рисунков, ячеек таблиц (Borders and Shading);
- нумерация заголовков (Heading Numbering); □ работа со стилями оформления (Style).

В группе команд Tools (Сервис) вы должны познакомиться, прежде всего, с командами установки опций (Options) и создания элементов автокоррекции (AutoCorrect). «Грамматические» команды Spelling, Thesaurus мы рассмотрим конспективно.

В пункте Table собраны все команды работы с таблицами.

Наконец, пункт Window предназначен, в основном, для управления обработкой нескольких документов, одновременно открытых процессором Word.

8. Основные элементы текстового документа

В позицию текстового курсора может быть введен любой символ (с кодом 32 – 255) из любого шрифта, подключенного к Windows.

Символ – минимальная единица текстовой информации.

Слово – это произвольная последовательность букв и цифр, ограниченная с обоих концов служебными символами. Служебный символ – это пробел, точка, запятая, дефис и т.п.

Строка – произвольная последовательность символов между левой и правой границей абзаца.

Предложение – произвольная последовательность слов между двумя точками.

9. Абзац

В большинстве простых текстовых редакторов абзац – это группа смежных строк файла, первая из которых начинается с отступа (в начале строки находится несколько пробелов), а все последующие начинаются с отличающегося от пробела символа. Концом абзаца считается строка, за которой следует либо конец файла, либо строка с отступом. В процессоре Word абзац – это просто произвольная последовательность символов, замкнутая символом «Возврат каретки» (код 13). Допускаются и пустые абзацы (одиночные символы «Возврат каретки»).

Абзац – простейшее, но фундаментальное понятие Word. Абзац всегда начинается с новой строки, однако, в отличие от многих текстовых редакторов, Word не требует обязательного отступа вправо в первой (красной строке) абзаца. Первая строка может начинаться с отступом влево по отношению к остальным строкам абзаца («висячая строка»), с нулевым отступом или, как обычно, с отступом вправо. Word фактически рассматривает абзац как одну длинную «строку», размещая набранные слова в строке страницы и перенося слово на новую строку, если оно не уместилось в текущей.

10. Структура страницы

На странице мы различаем следующие структурные элементы:

- основной текст;
- верхний и нижний колонтитулы; □ сноски.

Основной текст страницы – это строки и абзацы, таблицы и рисунки. Некоторые из абзацев могут быть заголовками; рисунки могут иметь подписи.

Колонтитул размещается в верхней или нижней части страницы и содержит некоторую информацию, которая как-то идентифицирует данный документ (номер раздела, название документа или раздела, дату, марку фирмы и т.п.). В колонтитул входит номер страницы. Колонтитулы автоматически воспроизводятся на каждой странице документа.

Чтобы вставить и отредактировать верхний (Header) или нижний (Footer) колонтитул, необходимо выбрать команду (View-Header and Footer). Колонтитул связан с определенным стилем.

Сноска – это примечание к тексту, которое может находиться либо в нижней части страницы, либо в конце документа. Каждая сноска помечается либо автоматически (порядковым номером со сквозной нумерацией – 1, 2...), либо пометкой по вашему выбору (например, звездочкой или номером). Сноска и ссылка на нее (т.е. номер или пометка) также имеют стиль.

Чтобы вставить сноску, необходимо выбрать команду (Insert-Footer); чтобы отредактировать сноску, надо выбрать команду (ViewFootnotes).

Все сноски данной страницы, в отличие от колонтитула, располагаются в пределах места, отведенного основному тексту страницы. Поэтому чем больше сносок вы набрали на данной странице, тем меньше места останется для основного текста.

11. Структура документа

Каждый документ обычно делится на структурные разделы, которые называются по-разному: части, главы, параграфы, пункты, подпункты и т.д. Каждый раздел имеет заголовок, который снабжен каким-либо текстовым обозначением или номером, например: Глава 5, 3.2, 5.4.1 и т.п.

Структуру документа определяет его автор. Например, наше учебное пособие в целом разделено на главы (глава 1, глава 2... глава 9). Каждая глава в свою очередь разделена на пункты с нумерацией 1.2, 2.4, 6.5 и т.д., причем первая цифра номера пункта – это номер главы, а цифра (или две цифры) после точки – номер пункта в данной главе.

Word не знает, что такое глава, что такое параграф или пункт. Для этой программы существует только понятие «заголовок» (heading), причем допускаются заголовки девяти уровней – от первого (наивысшего) до девятого.

Каждый заголовок – это абзац. Любому абзацу вы можете присвоить статус заголовка того или иного уровня. Например, абзацу:

Глава 2. Персональный компьютер типа IBM PC можно присвоить статус «Heading 1» (Заголовок 1).

Так при использовании заголовков четырех уровней можно принять следующую организацию:

- Heading 1 (Заголовок 1) Глава 2. Персональный компьютер типа IBM PC
- Heading 2 (Заголовок 2) Организация и представление данных в ПК
- Heading 3 (Заголовок 3) 2.3.2. Байт
- Heading 4 (Заголовок 4) Магнитные диски

Обратите внимание, что в нумерации заголовков принят определенный порядок: высший уровень мы отмечаем названием «Глава», второй и третий уровень структурированным номером (2.3, 2.3.2), а четвертый уровень – никак не отмечаем (хотя можно было бы писать 2.3.2.2 и т.п.). Разделов четвертого уровня мало, поэтому загружать вид текста ненужными цифровыми обозначениями считаем нецелесообразным.

Таким образом, при этом подходе процессору Word совершенно безразлично конкретное содержание (включая номер) того абзаца, который мы объявили заголовком, скажем, второго уровня.

Процесс управления заголовками разделов средствами программы Word будем называть структурированием документа. Разумеется, выделять, нумеровать, форматировать заголовки и составлять оглавление можно и вручную, однако Word позволяет резко упростить эту работу.

Во-первых, мы можем одним щелчком мыши присвоить любому абзацу статус заголовка любого уровня.

Во-вторых, если нас не устраивает внешний вид заголовка того или иного уровня, можно изменить его стиль, и все заголовки того же уровня автоматически изменят свой внешний вид в соответствии с новым стилем.

В-третьих, при желании можно поручить процессору дополнительную работу – автоматически пронумеровать все заголовки в соответствии с их иерархией и выбранным вами форматом номера.

В-четвертых, с помощью Word можно составить оглавление документа (Table of Contents), причем и в этом случае заголовку конкретного уровня приписывается определенный стиль оформления (в общем случае – не совпадающий со стилем заголовка в тексте). При этом в оглавление можно включить только заголовки, уровень которых не ниже заданного (например, заголовки с первого по третий уровень). Это позволяет избежать слишком подробных и плохо читаемых оглавлений.

В целом такой механизм дает возможность в дальнейшем как угодно модифицировать документ (вставлять или удалять разделы, добавлять и удалять текст в разделах), при этом оглавление легко заменить или обновить.

Кроме того, Word позволяет сформировать и другие справочные разделы документа: алфавитный указатель терминов, список таблиц, список рисунков.

12. Таблицы и рисунки

На страницах документа могут размещаться специфические объекты Word – таблицы и рисунки.

Каждая таблица состоит из некоторого количества строк и столбцов; на пересечении строки и столбца находится ячейка таблицы. Все команды работы с таблицами собраны в пункте меню Table (Таблица).

В документ Word можно поместить рисунок, созданный в другом приложении и записанный в графическом файле формата .BMP, .PCX (Paintbrush), .WMF (специальный векторный формат Windows), TIFF (издательский формат фирмы Aldus) и некоторых других форматов. Microsoft предлагает вам специальную библиотеку картинок формата .WMF, которая размещена в подкаталоге CLIPART каталога Word и предназначена для вставки этих рисунков документы.

Для вставки готового рисунка применяется команда (Insert-Picture) (Вставка – Рисунок).

Кроме того, с помощью панели инструментов «Рисование» можно создавать в документе собственные картинки (рисованные объекты), а также редактировать «чужие» изображения. Во многом эта процедура похожа на процесс обработки картинок в графическом редакторе Paintbrush, поэтому в этой главе мы лишь упоминаем о ней.

6.06 Текстовый процессор OpenOffice.org Writer Writer – это текстовый процессор в составе OpenOffice.org (OOo). Является свободным

программным обеспечением, доступным любому пользователю. Применяется для создания текстовых документов самого разного рода: от простых писем и записок до многотомных отчетов и рекламных буклетов.

OpenOffice.org был переведен (локализован) на более чем 40 языков. Дополнительно есть для более чем 70 языков и диалектов словари орфографии, расстановки переносов и тезауруса, которые не имеют локализованного интерфейса программы.

В дополнение к обычным особенностям текстового процессора (проверка орфографии, тезаурус, расстановка переносов, автозамена, поиск и замена, автоматическое составление оглавлений и индексов, составление стандартных писем), Writer обеспечивает следующие важные возможности:

- Шаблоны и стили.
- Мощные методы разметки страниц (включая врезки, столбцы и таблицы).
- Встраивание или связывание графики, электронных таблиц и других объектов.
- Встроенные средства рисования.
- Главные документы, используемые для объединения набора документов в один документ.
- Отслеживание изменений в версиях документов.
- Интеграция с базами данных, включая базу данных библиографии. Экспорт в формат PDF, включая закладки.

OpenOffice.org Writer – текстовый редактор и редактор html. Бесплатный аналог Microsoft Word. Большая часть функционала Writer идентична Word и при этом может использоваться для редактирования файлов Word (обратной поддержки еще нет). Входит в пакет OpenOffice.

Возможности программы полностью отвечают требованиям для современного текстового процессора, их хватит с лихвой для создания любого текста: можно выбирать начертание и размер шрифта, применять различные стили оформления текста. В документ можно вставлять не только текст, таблицы или изображения, но также видео- и аудиоданные. Программа поддерживает автозамену и проверку орфографии, в том числе и для русского языка.

OpenOffice.org Writer поддерживает большое количество форматов документов, среди которых, в первую очередь, следует отметить хорошую поддержку документов, созданных в программе Microsoft Office (doc, xls и другие), документы rtf, а также набирающий популярность открытый стандарт Open Document Standard (odf), призванный стать универсальным стандартом, поддерживаемым всеми текстовыми процессорами. Также интересной особенностью программы OpenOffice.org Writer является возможность сохранять созданные документы в формате

Adobe pdf.

На сегодняшний день OpenOffice.org Writer является одним из самых мощных бесплатных текстовых редакторов, которым совершенно спокойно можно пользоваться на работе и дома. Начиная с версии 2.0.4, OpenOffice.org поддерживает XUL расширения в формате .oxt, которые легко добавляются, подобно тому, как это организовано в Mozilla Firefox. Наиболее интересные и достаточно популярные расширения для Writer:

- ComPad – позволяет делать математические и инженерные расчеты в форме ввода математических выражений с помощью редактора уравнения.
- Language Tool – расширение для проверки грамматики. Реализована возможность проверки для русского, английского, немецкого, польского и других языков.
- Типографика для ООО – расширение позволяет привести текст в соответствие с типографскими стандартами (кавычки, тире, лишние пробелы).
- AltSearch – расширение для улучшения параметров поиска и замены текста, поддерживает регулярные выражения.

Основные преимущества OpenOffice.org перед другими офисными пакетами:

- отсутствие лицензионной платы;
- межплатформенность – официально существуют сборки для следующих операционных систем и платформ: Microsoft Windows, GNU/Linux, Unix, Sun Solaris, MacOS;
- обширная языковая поддержка – интерфейс OpenOffice.org доступен на более чем 40 языках. Кроме этого, для 70 языков имеются орфографические словари, словари переносов, тезауруса и диалектов;
- совместимость файлов – в OpenOffice.org есть встроенная возможность экспорта в форматы PDF и Flash, а также поддержка открытия и сохранения файлов в форматах Microsoft Office, RTF, PDF, HTML, XML, WordPerfect, StarWriter и Lotus 123.

В отличие от таких редакторов, как Microsoft Word и Abiword, в Writer отсутствует проверка грамматики, хотя и присутствует проверка орфографии. Для включения проверки грамматики нужно установить специальное расширение. Возможность работы с электронной почтой, которая была полностью переписана в версии 2.0, всё ещё менее стабильна и расширяема по сравнению с другими текстовыми процессорами.

ООо поставляется с обширной справочной системой. Если всплывающие подсказки разрешены, при наведении указателя на любой из значков вы увидите маленький блок («всплывающую подсказку») с кратким объяснением

назначения значка. Для более детального объяснения выберите Справка > Что это такое? и удерживайте указатель на значке.

Сообщество OpenOffice.org не только развивает программное обеспечение, но и обеспечивает бесплатную поддержку на добровольной основе. Пользователи ООо могут получить всестороннюю сетевую поддержку от местных групп через группы новостей, форумы или списки рассылки. Есть также многочисленные сайты, выполненные пользователями, которые предлагают бесплатные подсказки и учебные пособия.

4.19 6.07 Apple IWork Pages

iWork – набор приложений, созданных Apple Inc, содержащий текстовый процессор, табличный редактор, а также программу для создания презентаций. Основным конкурентом iWork является пакет Microsoft Office for Mac. iWork дешевле этого пакета, но не содержит эквивалентов ряда программ, входящих в некоторые версии Microsoft Office.

iWork интегрируется с приложениями, входящими в пакет iLife. Последняя версия пакета, iWork'09, не поддерживает формат OpenDocument (используется в OpenOffice.org, признан стандартом в ряде стран). Эти форматы поддерживает стандартный редактор Mac OS X 10.5 Leopard, Textedit.

В комплекте имеется набор шаблонов, упрощающих подготовку документа, но при этом бо льшая часть шрифтов, использованная в стандартных шаблонах, не локализована на русский язык, то есть они непригодны для документов с кириллицей. Позволяет вставлять мультимедиа из интегрированного обозревателя iLife (музыка, фото, видео). Имеет функции полного экрана, возобновления, автосохранения, хранения версий документа. Поддерживается экспорт в следующие форматы:

- Документы Pages (.pages)
- Документы Microsoft Word (.doc)
- Файлы PDF (.pdf)
- Форматированный текст (.rtf)
- Простой текст (.txt)
- Открытый стандарт электронных книг (.epub)

В приложении Pages на iPad, iPhone или iPod touch вы можете создавать красивые письма, отчёты, листовки и другие документы.

Pages поддерживает технологию Multi-Touch, поэтому набор текста становится естественным и увлекательным. Новый горизонтальный режим на iPhone и iPod touch позволяет вводить текст двумя большими пальцами с максимальной скоростью, а функция интеллектуального масштабирования автоматически увеличивает текст, когда вы работаете над ним. Навигатор страниц отображает миниатюры каждой страницы, поэтому вы можете легко

перейти к любой части документа. Просто коснитесь кнопки микрофона на клавиатуре, произнесите слова – и они тут же появятся на экране. Писать письма и вести заметки теперь гораздо проще.

В Pages можно работать не только с текстом. Это отличный инструмент для работы с графикой. Одним касанием вставьте фотографии и фильмы из приложения «Фотографии». Кадрируйте изображения, примените эффекты отражения и добавьте рамки, чтобы подчеркнуть фотографии на странице. Добавьте впечатляющие трёхмерные диаграммы и динамические таблицы, чтобы проиллюстрировать свои мысли. Создайте фигуры, например звёздочки и многоугольники, и выделите их тенями. Потяните за угол объекта или изображения, чтобы изменить его размер, разверните жестом поворота или перетащите на новое место – направляющие линии помогут расположить его именно так, как вам нужно. Текст автоматически обтекает изображение, даже когда вы его перемещаете.

Pages сохраняет результаты вашей работы – прямо во время работы. Вы можете набирать текст на ходу, и при следующем открытии документа он будет выглядеть в точности так, каким вы его оставили, даже если вы откроете его на другом устройстве. Любые изменения можно отменить, независимо от того, когда вы их внесли.

Каждый шаблон в Pages имеет свой стиль, поэтому ваш документ выглядит великолепно с самого начала. А всего одним касанием линейки вы можете изменить шрифты и расположение текста. Коснитесь панели «Стили», чтобы применить единый стиль ко всем заголовкам, тексту и подписям в документе.

Расширенное форматирование в Pages – это легко. Коснитесь кнопки «Свойства», чтобы настроить межстрочный интервал и разбить текст на колонки. Простым перетаскиванием настройте табуляцию, отступы абзацев и поля страниц. Добавьте колонтитулы и номера страниц.

Pages помогает вам создавать списки множеством способов. Выберите маркированный, нумерованный или буквенный список – и он будет создаваться автоматически по мере ввода текста. Измените порядок элементов списка, просто перетащив маркер или число. Воспользуйтесь инструментами на панели «Свойства», чтобы изменить стиль списка в любой момент.

Если вы неправильно ввели слово, Pages сразу предупредит вас об этом. Не выходя из Pages, можно быстро посмотреть слово во встроенном словаре. Если вам нужно отслеживать каждую букву, абзац и страницу, можно включить режим статистики. Можно даже использовать сноски – внизу страницы или в конце документа.

Функции «Найти» и «Заменить» помогут вам внести быстрые правки, например вставить прописные буквы или изменить написание имён собственных. Просто введите слово, которое вы хотите заменить, и Pages

найдёт каждую вариацию этого слова в документе. Слова можно заменять по одному или все одновременно.

Ваши творения может прочитать каждый – в Интернете или по электронной почте, на Mac и PC. Pages хранит все ваши документы вместе, поэтому их легко найти и экспортировать. Если вам прислали документ Microsoft Word, его можно открыть в Pages. Вы можете экспортировать свой документ в файл Word или PDF, чтобы его могли открыть любые пользователи. Кроме того, в Pages можно открывать документы Pages'09, созданные на Mac. Вы можете публиковать и отправлять документы различными способами. Отправляйте и получайте файлы по электронной почте. Передавайте их через iTunes. И печатайте по беспроводной сети через AirPrint.

Создавать документы с Pages просто. Вверху страницы находится контекстная панель форматирования, которая позволяет выполнить основные действия – форматирование текста и настройку изображения – всего одним нажатием. Вы можете просмотреть и выбрать шрифты в меню шрифтов, где они показаны точно так же, как будут выглядеть в документе. Можно изменить размер и цвет текста, указать межстрочное расстояние, выровнять текст по одному или другому краю страницы, изменить стили абзацев и символов. При выборе фотографии, фигуры или таблицы в документе на панели форматирования отображаются инструменты, которые можно использовать для настройки изображения. Pages автоматически форматирует маркированные или нумерованные списки, проверяет орфографию, корректирует документ и составляет таблицу содержания.

Задать дополнительные параметры документа можно всего несколькими нажатиями мыши, можно добавить верхние и нижние колонтитулы, сноски и закладки. Можно вставить раздел, макет и разрыв страницы из раскрывающегося меню. А внизу страницы показана статистика по словам – нажмите на общую сумму слов, чтобы получить подробные сведения: количество символов, строк, абзацев. Когда же дело доходит до обработки текста, в качестве нового стандарта работы Pages предлагает только самые передовые и интеллектуальные методы.

Набрать текст можно в любой программе. А в Pages его можно значительно улучшить. Теперь вы сможете увидеть документ в полноэкранный режиме. Одно нажатие – и ненужные элементы исчезают, вы концентрируетесь на том, что пишете, и редактируете, не отвлекаясь. Организуйте свои идеи с помощью режима структуры. Создавайте структуры с несколькими уровнями, разворачивайте или сворачивайте темы, перетаскивайте элементы, чтобы переместить их на более высокий или низкий уровень. Функция автоматической рассылки писем позволяет взять данные или контактную информацию из Адресной книги для создания личных писем, счетов и факсов.

Если вы хотите получить комментарии и обеспечить обратную связь, функция отслеживания изменений сделает взаимодействие простым, понятным и точным. Найти нужное место в документе всегда легко: рядом с документом показаны миниатюры всех страниц и разделов, включая сделанные изменения. Можно быстро копировать, перемещать или удалять разделы. Для предварительного просмотра документа можно пролистать миниатюры или увеличить их, чтобы просмотреть внимательнее.

Pages позволяет легко обмениваться документами с коллегами. Вы можете открывать файлы Microsoft Word в Pages и сохранять документы Pages как файлы Word. Благодаря мощным графическим инструментам и средствам форматирования документы Word отлично выглядят в Pages. Документы Pages можно также сохранять как файлы RTF или в формате простого текста. Или экспортировать их в формат ePub или PDF. Оба этих формата совместимы с приложением iBooks на iPad, iPhone и iPod touch. Документы Pages, Word или PDF можно также отправлять по электронной почте прямо из Pages, используя приложение Mail, встроенное в Mac OS X.

Используя iBooks, вы можете сохранять и читать документы, созданные в приложении Pages на iPhone, iPad и iPod touch. Отчёты и книги с большим количеством текста можно экспортировать в формат ePub – это открытый стандарт, поддерживаемый почти любыми устройствами для чтения электронных книг. Вы можете отправлять документы ePub в приложение iBooks через iTunes, публиковать их в iBookstore или отправлять по электронной почте коллегам. В iBooks можно выбрать подходящую гарнитуру и размер шрифта. Внешний вид текста будет автоматически преобразован в зависимости от размера экрана и ориентации iPhone, iPad или iPod touch. Документы, насыщенные графикой и имеющие более сложное форматирование, например новости или брошюры, можно экспортировать как файлы PDF. Файлы в формате PDF всегда выглядят точно так же, как оригинал, вне зависимости от устройства, на котором вы их просматриваете.

4.20 6.08 Adobe Acrobat

Adobe Acrobat – пакет программ, выпускаемый компанией Adobe Systems для создания и просмотра электронных публикаций в формате PDF. Существует несколько версий пакета, отличающихся возможностями: Adobe Acrobat Standard, Adobe Acrobat Professional, Adobe Acrobat Professional Extended (бывший Adobe Acrobat 3D) и Adobe Acrobat Elements. Разработка Adobe Acrobat Professional Extended прекращена, поддержка для Acrobat 9 Pro Extended продлится до 26 июня 2013 года.

Десятая версия пакета программ Acrobat X вышла в 3-х редакциях: Acrobat X Standard, Acrobat X Pro и Acrobat X Suite. Acrobat Professional, начиная с версии 7.0, содержит в себе Adobe LiveCycle Designer. Acrobat Suite дополнительно содержит: Captivate (включая Media Encoder), Photoshop и Presenter. В версии XI Adobe LiveCycle Designer заменён более упрощённой программой – Adobe FormsCentral.

С версии XI в линейке Acrobat прекращается дальнейшее развитие Acrobat Suite. Пользователям, которым нужны мультимедийные и коммуникационные функции, предлагается приобрести Adobe eLearning Suite, который содержит в себе все программы, входившие ранее в Acrobat Suite. Для КПК и коммуникаторов существует версия Adobe Reader for Pocket PC.

4.21 6.09 Издательская система LaTeX

Система компьютерной верстки TeX (произносится «тех») была создана выдающимся американским математиком и программистом Дональдом Кнутом в конце 70-х годов XX века; издательские системы на ее базе до сих пор широко используются и сдавать позиции не собираются. LaTeX (произносится «латех» или «лэйтех», пишется также «LaTeX») – это созданная Лесли Лэмпортом (Leslie Lamport) издательская система на базе TeX-a.

Чем объясняется столь редкое в компьютерном мире долголетие? На первый взгляд, все свидетельствует против TeX'a. В самом деле, в отличие, допустим, от популярного ныне Microsoft Word'a, TeX не является системой типа WYSIWYG (What You See Is What You Get): чтобы посмотреть, как будет выглядеть на печати набираемый текст, надо запустить отдельную программу. И по структуре файлов TeX несовместим с Word'ом (что не удивительно: эта структура мало изменилась с начала 1980-х годов). Наконец, чтобы работать в TeX'e, надо потратить определенное время на его изучение: трудно представить себе книгу под названием "TeX for dummies" («TeX для болванов»).

Краткий перечень TeX'овских достоинств:

- Никакая другая из существующих в настоящее время издательских систем не может сравниться с TeX'ом в полиграфическом качестве текстов с математическими формулами.
- Система TeX реализована на всех современных компьютерных платформах, и все эти реализации действительно работают одинаково.
- Благодаря этому TeX стал международным языком для обмена математическими и физическими статьями: набрав свою статью в TeX'e, математик может послать ее по электронной почте своему коллеге, даже

если отправитель работает под Windows, а получатель – с UNIX'ом или, допустим, на Макинтоше.

- В Интернете существуют обширные «архивы препринтов», в которые каждый может послать (и из которых каждый может получить) статью; все эти статьи набраны опять-таки в TeX'e.
- Наконец, основные реализации TeX'a для всех платформ распространяются бесплатно.

Разумеется, у TeX'a есть и недостатки. Главный из них – в том, что с помощью TeX'a тяжело (хотя в принципе и возможно) готовить тексты со сложным расположением материала на странице (наподобие рекламных буклетов). Для таких приложений, практически не встречающихся в научно-технической литературе, TeX не предназначен.

Наряду с LaTeX'ом распространены также макропакеты Plain-TeX и AMS-TeX. Макропакет Plain-TeX был разработан самим Дональдом Кнудом, рассматривавшим его в качестве платформы для построения более сложных систем; на практике он используется и как средство для обмена текстами (текст, подготовленный для Plain TeX'a, сравнительно несложно переделать в исходный текст для того же LaTeX'a). Что касается AMS-TeX'a, то эта издательская система сориентирована на важный, но узкий круг приложений: верстку статей для математических журналов и книг, издаваемых Американским математическим обществом. Соответственно, в AMS-TeX'e предусмотрено большое количество весьма изощренных возможностей для создания сложных математических формул, но при этом нет многих вещей, которые естественно было бы ожидать в издательских системах общего назначения (например, автоматической нумерации частей документа). Современные версии LaTeX'a включают в себя AMS-TeX'овские возможности для набора формул (и используются Американским математическим обществом).

6.10 QuarkXPress

QuarkXPress – мощная издательская система, обладающая интуитивным интерфейсом и расширенным набором инструментов для обработки текста, управления цветом и графическими элементами и проектирования веб-страниц. Программа широко используется в книжных, газетных и журнальных издательствах, рекламных и маркетинговых агентствах, дизайнерских фирмах и типографиях. Содержит более 160 новых функций и усовершенствований, позволяющих объединить процессы макетирования, проектирования и создания информационного наполнения. Благодаря новым функциям, таким как инструменты управления прозрачностью объектов и технология Composition

Zones, а также улучшенным функциям печати и создания документов в формате PDF, приложение обеспечивает полную свободу в процессе проектирования и возможность совместной работы с документами. Язык интерфейса: Многоязычный (в т.ч. Русский). Тип лицензии: ShareWare. Цена: 799.00 USD.

Электронные таблицы

9.01 Общие сведения об электронных таблицах Электронные таблицы (или табличные процессоры) – это прикладные программы, предназначенные для проведения табличных расчетов.

Появление электронных таблиц исторически совпадает с началом распространения персональных компьютеров. Первая программа для работы с электронными таблицами – табличный процессор – была создана в 1979 году, она предназначалась для компьютеров типа Apple II и называлась VisiCalc. В 1982 году появляется знаменитый табличный процессор Lotus 1-2-3, предназначенный для IBM PC. Lotus объединял в себе вычислительные возможности электронных таблиц, деловую графику и функции реляционной СУБД. Популярность табличных процессоров росла очень быстро. Появлялись новые программные продукты этого класса: Multiplan, QuattroPro, SuperCalc и другие. Одним из самых популярных табличных процессоров на сегодняшний день является MS Excel, входящий в состав пакета Microsoft Office.

Электронная таблица (ЭТ) – это средство информационных технологий, позволяющее решать целый комплекс задач, прежде всего выполнение вычислений, так как многие расчеты выполняются в табличной форме, особенно в области делопроизводства: многочисленные расчетные ведомости, табуляграммы, сметы расходов и т.п. Кроме того, решение численными методами целого ряда математических задач удобно выполнять в табличной форме.

Математическое моделирование и использование математических формул в ЭТ позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. Основное свойство ЭТ – мгновенный пересчет формул при изменении значений входящих в них операндов. Благодаря этому свойству, таблица представляет собой удобный инструмент для реализации следующих задач:

- подбор параметров,
- прогноз поведения моделируемой системы, □ анализ зависимостей, □ планирование.

Дополнительные удобства для моделирования дают возможность графического представления данных (диаграммы).

Электронные таблицы могут быть использованы также в качестве базы данных. Конечно, по сравнению с СУБД электронные таблицы имеют меньшие возможности в этой области. Однако некоторые операции манипулирования данными, свойственные реляционным СУБД, в них реализованы. Это поиск информации по заданным условиям и сортировка информации.

В электронных таблицах предусмотрен также графический режим работы, который дает возможность графического представления (в виде графиков, диаграмм) числовой информации, содержащейся в таблице.

Тема 5 Электронные таблицы, их характеристики, выбор, применение. Обработка данных в электронных таблицах MS Excel

Программный продукт Excel, разработанный фирмой Microsoft, является популярным средством для работы с электронными таблицами. Функциональные возможности и вычислительные средства Excel позволяют решать многие инженерные и экономические задачи, представляя данные не только в табличном, но и в графическом виде.

Excel функционирует в операционной среде Windows, поэтому, работая с ним, можно реализовывать любые возможности Windows: сворачивать и разворачивать окна, использовать кнопки пиктограмм, работать одновременно с несколькими документами и т.п.

После запуска Excel на экране появляется окно, состоящее из нескольких элементов (см. рис. 23):

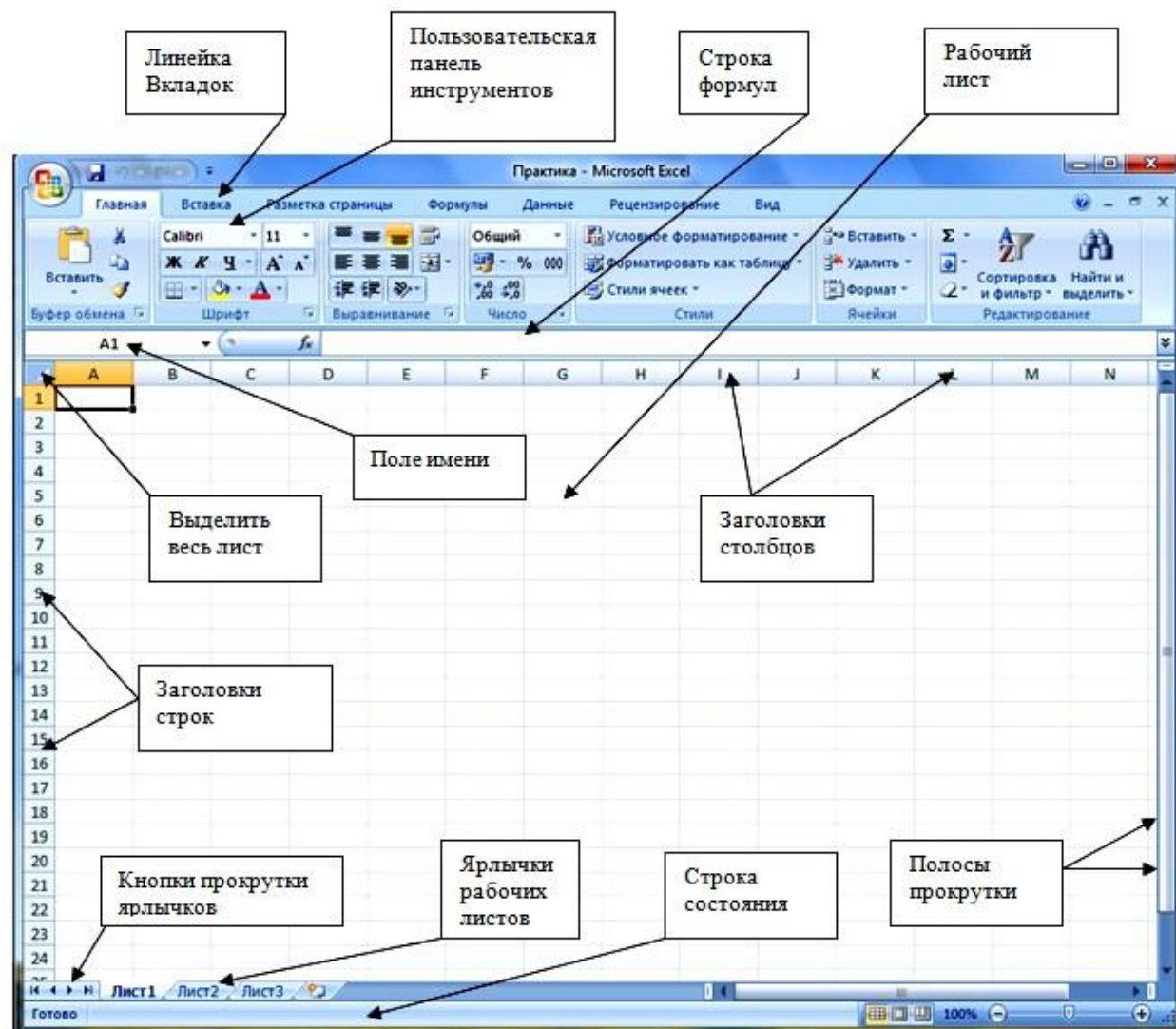


Рис. 23

Excel содержит так называемые **контекстные меню**, содержащие наиболее употребляемые команды для работы с ячейками, диаграммами или другими объектами.

Контекстное меню активизируется нажатием *правой* кнопки мыши (или Shift + F10). Для выхода из контекстного меню нужно щелкнуть кнопкой мыши вне его или нажать Esc.

5.1 1. Типы данных, используемых в Excel

Excel использует два основных типа данных:

- текст, то есть последовательность символов (при вводе они автоматически выравниваются по левому краю ячейки);
- числа, различая их как числовые константы, формулы, встроенные функции или даты. При вводе чисел Excel автоматически выравнивает

их по правому краю ячейки и производит над ними необходимые пользователю вычисления.

Данные текстового типа используются, как правило, для обозначения названий таблиц, заголовков столбцов, текстовой информации в строках и столбцах, а также для комментариев.

Данные числового типа используются для числовых величин (или ссылок на соответствующие ячейки) и связывающих их арифметических операций.

Например, данными числового типа в Excel являются следующие выражения:

- =46+55;
- =200*B5; □ =A7/B4.

Ввод формулы всегда начинается со знака «равно» (=) или «плюс» (+).

Примечания

▪ Формулу, содержащуюся в ячейке, по умолчанию можно увидеть в строке формул, когда данная ячейка станет активной, а в самой ячейке виден только результат вычислений. Для отображения формул в ячейках необходимо нажать кнопку **Office**, в открывшемся окне нажать кнопку **Параметры Excel**, в окне **Параметры Excel** выбрать параметр **Дополнительно**, в группе **Показать параметры для следующего листа** включить параметр **Показывать формулы, а не их значения**.

▪ Excel вычисляет формулу каждый раз, когда изменяется содержимое таблицы, ссылающееся на данную формулу.

▪ Если результат вычислений не помещается в ячейке таблицы, Excel может вывести на экран последовательность символов «#####», говорящую о том, что необходимо увеличить ширину столбца.

▪ При вводе десятичных чисел используется запятая (,) в качестве десятичного разделителя.

Функции Excel – это заранее определенные формулы, позволяющие производить расчеты в финансовой, статистической, математической, логической и других областях деятельности. Функции подразделяются на *встроенные* и *определенные пользователем*. Встроенные функции сгруппированы по категориям и имеют одинаковый синтаксис: имя функции и ее аргументы.

Функции задаются с помощью математических и других формул, в соответствии с которыми выполняются вычисления по заданным величинам, называемым аргументами, и в указанном порядке, определяемом синтаксисом. Синтаксис встроенных функций достаточно прост:

Имя функции(<аргумент1; аргумент2; аргумент3 и т.д.), где:

Имя функции показывает, о каких вычислениях идет речь. Примеры имен функции – СУММ, СРЗНАЧ и др.

Аргументы – значения, которые функция использует, вычисляя результат. Аргументы перечисляются в скобках следом за именем функции. В качестве аргументов могут выступать числовые значения, текст, логические значения, массивы, значения ошибок или ссылки, дата/время, а также другие функции и формулы. В Excel различают обязательные (которые всегда нужно задавать) и необязательные аргументы. Отдельные аргументы разделяются символом «точка с запятой» (;). **Результат** – значение, полученное при вычислении функции.

Многие математические операции в Excel производятся с использованием встроенных функций.

Функции могут быть выбраны из списка функций с помощью кнопки **Вставить функцию** из группы **Библиотека функций** на вкладке **Формулы** или с помощью нажатия пиктограммы **f_x** в строке формул.

После выбора требуемой функции подключается **Мастер функций**, позволяющий пользователю пошагово заполнять ее аргументы (см. рис. 24).

Функции в Excel подразделяются на следующие основные группы:

- Математические, арифметические и тригонометрические функции.
- Функции для работы с датами и временем.
- Финансовые функции.
- Логические функции.
- Функции для работы со ссылками и массивами.
- Функции для работы с базами данных.
- Статистические функции.
- Текстовые функции и др.

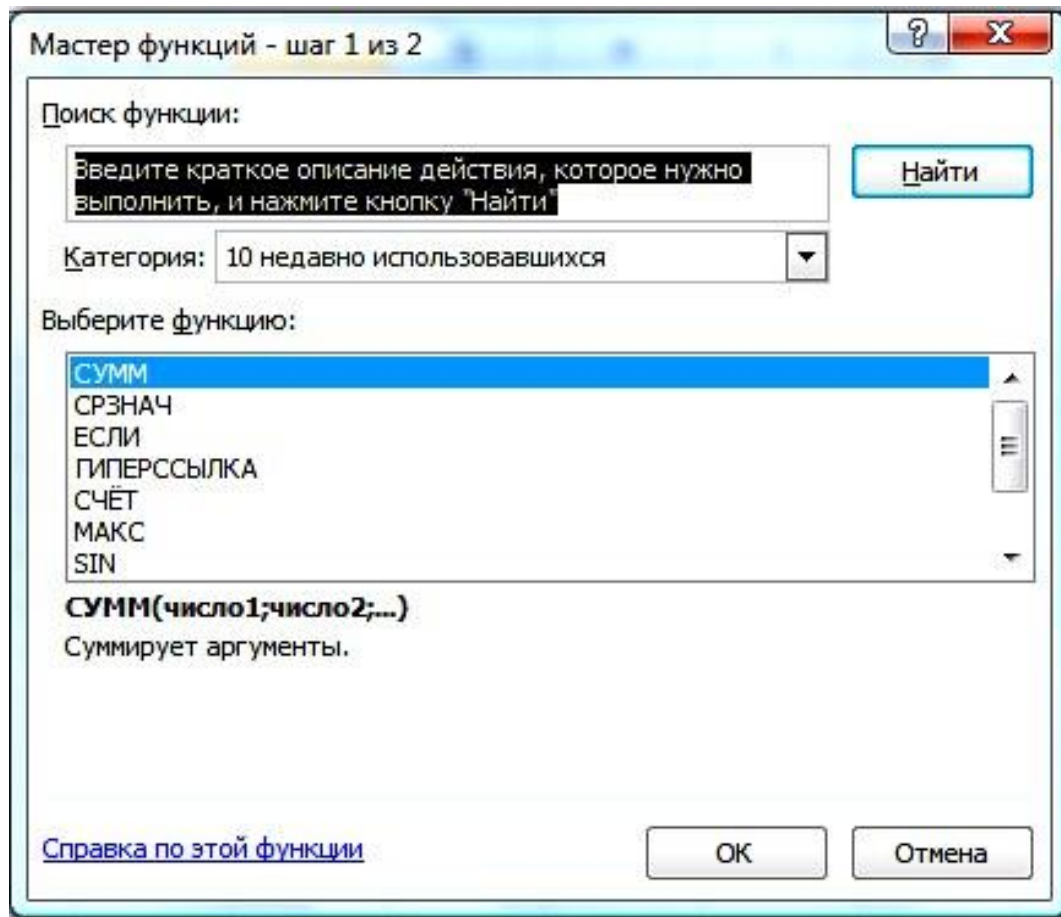


Рис. 24

Математические и тригонометрические функции используются для выполнения вычислительных операций (вычитания, сложения, умножения, деления), а также возведения в степень, округления, логарифмических вычислений, операций над случайными числами и подсчета количества. В группу тригонометрических функций объединены прямые и обратные тригонометрические функции.

Функции для работы с датой и временем позволяют анализировать и работать со значениями даты и времени в формулах. Значения даты и времени сохраняются и обрабатываются программой как числа. Начальной датой является 1 января 1900 г. Ей соответствует целое значение 1. Каждый следующий день представлен целочисленным значением, на единицу больше значения предыдущей даты.

Финансовые функции решают в основном задачи расчета амортизации и определения процентов по займам и инвестициям, а также анализируют операции с ценными бумагами. Оперирруя этими функциями, можно определить такие показатели, как текущая стоимость инвестиций, доходность ценных бумаг и прочие.

Логические функции оперируют с логическими значениями и результат их также представляет собой логическое значение – ИСТИНА или ЛОЖЬ.

Функции для работы со ссылками и массивами данных используют в качестве аргумента массив данных. Применяются для поиска в списках или таблицах, а также для нахождения ссылки к ячейке.

Функции для работы с базами данных применяются для работы с базами данных (списками) и таблицами с соответствующей структурой данных. С помощью этих функций можно выполнять анализ данных рабочего листа.

Статистические функции позволяют решать множество различных задач – как сложных профессиональных, так и простых, например определение арифметического среднего.

Текстовые функции используются при работе с текстом, позволяют искать, заменять или объединять определенные последовательности символов, а также подсчитывать количество символов и многое другое.

Десять последних функций, применяемых пользователем, Excel автоматически группирует в категорию «10 недавно использовавшихся».

2. Диагностика ошибок в формулах Excel

Если Excel не может выполнить обработку формулы в ячейке и вывести результат, то он генерирует сообщение об ошибке и выводит его в данной ячейке (вместо самой формулы или ее результата). Сообщение об ошибке всегда начинается со знака «#».

Сообщения об ошибках в Excel могут принадлежать к одному из 8 типов:

- ##### □ #ЗНАЧ!
- #ДЕЛ/0! □ #ИМЯ? □ #Н/Д □
#ССЫЛКА!
- #ЧИСЛО!
- #ПУСТО!

Как правило, ошибка ##### возникает, когда полученный результат не помещается в ячейке. В этом случае необходимо увеличить ширину столбца.

Ошибка #ЗНАЧ! возникает, когда используется недопустимый тип аргумента, например, пользователь пытается сложить текстовое и числовое значение.

Ошибка #ДЕЛ/0 появляется, когда в формуле делается попытка деления на ноль.

Сообщение об ошибке типа **#ИМЯ?** появляется, когда Excel не может найти имя, используемое в формуле. Например, такая ситуация возникнет, если:

- при наборе имени произошла опечатка;
- текст ошибочно не был заключен в двойные кавычки;
- в ссылке на диапазон ячеек пропущен знак двоеточия (:).

Ошибка **#Н/Д** является сокращением термина «Неопределенные Данные».

Ошибка **#ССЫЛКА!** появляется, когда при ссылке на ячейку указывается недопустимый адрес.

Сообщение об ошибке вида **#ЧИСЛО!** возникает в том случае, когда в формуле задан неприемлемый аргумент для функции.

Сообщение об ошибке типа **#ПУСТО!** появляется, когда используется ошибочная ссылка на ячейку или диапазон, например, задано пересечение двух областей, которые не имеют общих ячеек.

3. Ввод и обработка данных в Excel

Значительная часть работы в Excel приходится на ввод данных, их редактирование и обработку.

Рабочий лист в Excel 2007 состоит из 13384 столбцов и 1048576 строк.

В ячейку рабочего листа можно вводить число, текст, дату/время или формулу. Ввод всегда происходит в активную ячейку. Место появления вводимых символов указывает мигающий курсор.

Весь введенный текст и числа отображаются и в строке формул, и в самой ячейке. При вводе формул (по умолчанию) в ячейке отражается результат вычислений, а в строке формул видна сама вводимая формула или функция.

При необходимости можно исправить ошибки до фиксации ввода или изменить содержимое ячейки после того, как ввод зафиксирован.

4. Форматирование и защита рабочих листов

Вводимая в ячейки Excel информация может быть отображена на экране различными способами. Для изменения формы отображения и доступа к информации используются средства форматирования и защиты.

Неправильный формат представления данных может вызвать значительные проблемы, особенно если у пользователя отсутствует достаточный опыт. Например, если пользователь введет число 0.9, но в качестве десятичного разделителя в настройках компьютера используется запятая, то введенные данные будут восприниматься как текст. Опытный пользователь сразу это заметит, так как обычно по умолчанию текст выравнивается по левому краю, а числа – по правому краю ячейки.

Форматирование в Excel включает в себя решение следующих вопросов:

- изменение шрифта, размеров, начертания и цвета символов;
- выравнивание и изменение ориентации текста и чисел в ячейках;
- форматирование чисел, дат и времени;
- форматирование строк и столбцов;
- создание и использование пользовательских форматов;
- условное форматирование;
- защита ячеек, листов и рабочих книг; □ использование стилей при форматировании; □ применение автоформатов.

5.2 Работа с электронными таблицами

Электронные таблицы в Excel располагаются на рабочих листах рабочих книг, последние из которых представляют собой электронный эквивалент папки-скоросшивателя, «складывающей» документы. Количество рабочих листов в книге может регулировать пользователь. В рабочие книги можно дополнительно «подшивать» диаграммы, сводные таблицы, различные отчеты и т.п.

Рабочий лист электронной книги состоит из ячеек, каждая из которых имеет свой адрес: сочетание имени столбца и строки. **Столбцы** идентифицируются буквами латинского алфавита (A, B, C, D...), а **строки** – арабскими цифрами (1,2,3...). Ячейка, в которой находится курсор, считается активной, то есть предназначенной для ввода данных.

Например, адрес F10 говорит о том, что вводимая информация при активной ячейке F10 попадет именно в эту ячейку, а содержимое этой ячейки отразится в строке формул.

Многие команды Excel позволяют работать с блоками ячеек. **Блок ячеек** – это прямоугольник, задаваемый координатами противоположных углов, обычно, верхней левой и нижней правой ячеек. Имена ячеек в блоках разделяются двоеточием (:). Например, блок A1:B4 включает в себя ячейки A1, A2, A3, A4, B1, B2, B3 и B4. Выделение блока ячеек осуществляется протаскиванием курсора мыши на нужный диапазон. При этом ячейка, начиная с которой выделяется блок, остается с белым цветом, а остальные ячейки затемняются. Но блок ячеек не обязательно может включать смежные ячейки. Для выделения таких ячеек в блоке следует использовать клавишу <Ctrl>.

Ячейкам и блокам для удобства работы можно давать имена. Рабочие листы также могут быть переименованы, чтобы лучше отражать смысловое содержание решаемой задачи.

1. Ввод заголовка, шапки и исходных данных таблицы

После загрузки Excel на экран выводится стандартный шаблон таблицы, имеющий определенную ширину столбцов и высоту строк.

Ввод данных осуществляется в *активную* ячейку, т.е. в ту ячейку, в которую предварительно установлен табличный курсор и которая отличается от остальных контрастной рамкой вокруг нее.

Ввод исходных данных в ячейку будет завершен в результате выполнения одного из следующих действий:

- ❖ нажатия клавиши Enter;
- ❖ нажатия соответствующих клавиш управления курсором на клавиатуре;
- ❖ щелчка кнопкой мыши по следующей ячейке для ввода данных;
- ❖ нажатия клавиши табуляции.

При вводе данных в таблицу названия некоторых граф могут быть расположены в ячейке в несколько строк. Подобный ввод данных может быть осуществлен следующими способами:

1. Нажать клавишу **Alt** (левую или правую) и, удерживая ее, нажать коротко клавишу **Enter** после любого введенного в ячейку слова или словосочетания. Одновременное нажатие клавиш принято обозначать символом “+”, например **Alt+Enter**.

2. Использовать вкладку **Главная**, где в группе **Ячейки** нажать кнопку **Формат**, в раскрывшемся окне выбрать команду **Формат ячеек**, а затем в окне **Формат ячеек**, на вкладке **Выравнивание** установить переключатель **Переносить по словам** в области **Отображение**.

3. Нажать кнопку **Перенос текста** в группе **Выравнивание** на вкладке **Главная**.

2. Редактирование содержимого ячейки

Если при просмотре таблицы были обнаружены ошибки, их необходимо отредактировать.

Excel позволяет редактировать содержимое ячеек несколькими способами:

- набором в активной ячейке новой информации поверх ошибочной (используется, как правило, при полной замене информации);
- активизацией строки формул щелчком мыши для перехода в режим редактирования;
- нажатием функциональной клавиши **F2** в активной ячейке. Удаление ошибочной информации осуществляется нажатием клавиши **Delete** в активной ячейке.

3. Оформление электронной таблицы

Каждая электронная таблица требует своего оформления, то есть придания ей определенного эстетического вида.

Как правило, заголовки таблиц, названия их граф, итоговая строка могут быть выделены отличающимся от общего текста размером шрифта, его цветом, фоном. Кроме того, в таблицах принято использовать разделители столбцов и строк (границы) и прочие графические эффекты.

Большинство операций по оформлению таблиц сосредоточено на вкладке **Главная**. Наиболее употребляемые операции (цвет заливки, цвет шрифта, выравнивание по различным признакам, границы, разрядность и т.п.) выведены в качестве кнопок в группах **Шрифт**, **Выравнивание**, **Число**.

4. Сохранение таблиц на диске

Для многократного использования электронной таблицы в дальнейшем необходимо сохранить ее на диске.

При сохранении файла сохраняются и установки, сделанные для данной книги, включая конфигурацию окна и характеристики дисплея, формулы, функции, шрифты и стили.

Активизировав **Кнопку «Office»** в верхнем левом углу окна, вы увидите несколько команд, позволяющих сохранить файл EXCEL: **Сохранить**, **Сохранить как**, **Закреть**. Каждая из этих команд имеет свою специфику. Команда **Сохранить как** обычно используется при первом сохранении документа или для выбора способа его сохранения. Команда **Сохранить** применяется для сохранения изменений, сделанных в существующем документе. **Сохранить как Web-страницу** сохраняет документ на Web-странице. При выборе команды **Закреть**, во избежание случайной потери выполненной работы, Excel всегда спрашивает, не хотите ли вы сохранить свои изменения. Кроме того, нажав кнопку **Сохранить** на панели инструментов, вы можете быстро сохранить текущий документ точно так же, как с помощью команды **Сохранить** из меню **Office**. После сохранения файла книги, с помощью одной из перечисленных выше команд сохранения, книга остается открытой. **Файл** удаляется с экрана только при закрытии книги.

5.3 Загрузка рабочей книги

Если был осуществлен выход из EXCEL после записи на диск рабочей книги, необходимо выполнить следующие действия: загрузить Excel, нажать кнопку **«Office»** (в левом верхнем углу экрана), выбрать команду **Открыть**, в окне **Открытие документа** выбрать диск, на котором был сохранен файл,

папку, имя файла (или ввести его имя в поле **Имя файла**), а затем щелкнуть по кнопке **Открыть**. Можно дважды щелкнуть по значку файла в списке файлов в окне **Открытие документа**.

Обеспечение автоматизации и проверки вводимых данных.

С целью обеспечения автоматизации и проверки данных при вводе в соответствующие ячейки таблицы будут использованы кнопка **Проверка данных** из группы **Работа с данными** на вкладке **Данные** и функция **Просмотр**.

Ввод данных в ячейки может показаться довольно несложной и очевидной процедурой, особенно по сравнению с записью сложных формул. Однако если тип или диапазон данных будет некорректным, могут возникнуть проблемы, которые позволяет решить кнопка Проверка данных из группы Работа с данными. Применить средство проверки данных можно в выделенной ячейке или диапазоне ячеек, так же как и средство форматирования ячеек. При копировании или перемещении ячеек, для которых установлена проверка данных, ее критерии передаются новым ячейкам.

Для установки проверки данных необходимо выделить ячейку или диапазон ячеек, выбрать **Данные|Работа с данными|Проверка данных** для открытия окна **Проверка вводимых значений** и использовать вкладки этого окна, в которых задаются критерии проверки, или для ограничения диапазона значений, которые можно ввести в какую-либо ячейку путем выбора из *списка*, а не ввода с клавиатуры. Проверка вводимых данных позволяет избежать ошибок, особенно если с рабочей книгой будет работать не один пользователь.

1. Ввод формул и функций для табличных расчетов

Формулы в MS Excel должны начинаться со знака равно или плюс, который подтверждает, что последующие символы образуют формулу. Символы, введенные без этих знаков, воспринимаются как текст. По умолчанию текст выравнивается по левой границе ячеек, а числа – по правой. Необходимо помнить несколько правил:

1. В первую очередь вычисляются выражения внутри круглых скобок.
2. Умножение и деление выполняются раньше сложения и вычитания.
3. Операторы с одинаковым приоритетом выполняются слева направо.
4. Для изменения порядка выполнения операторов используют круглые скобки.
5. Если в формуле количество закрывающих и открывающих круглых скобок не совпадает, выводится сообщение «Несоответствие скобок» и выделяется ошибочная часть формулы.

6. Формула, содержащая ссылки на адреса ячеек, связана с ячейками рабочей книги, а значение формулы зависит от содержимого ячеек, на которые указывают ссылки, и оно изменяется при изменении содержимого этих ячеек.

2. Относительная и абсолютная адресация ячеек

Относительные адреса используются в формуле в том случае, когда нужно, чтобы при определенных операциях с ячейкой, содержащей эту формулу (например, при копировании на новое место), адреса изменялись бы в соответствии с новым расположением ячейки (имя столбца, номер строки).

Абсолютный адрес используется в формуле в том случае, когда нужно, чтобы при определенных операциях с ячейкой, содержащей эту формулу, данный адрес оставался бы неизменным.

Адрес ячейки называют также *ссылкой*; в этом случае используют термины *Относительная ссылка* и *Абсолютная ссылка*.

Адрес можно сделать абсолютным двумя способами:

1. Поместить *символ доллара(\$)* в строке формул перед именем столбца и номером ячейки, например **\$A\$5**, введя его непосредственно с клавиатуры, или установить курсор в строке формул на адресе ячейки и нажать клавишу **F4**.

2. Присвоить имя ячейке с помощью кнопки Присвоить имя из группы Определенные имена на вкладке Формулы. 3. Копирование формул в электронных таблицах

Экономические таблицы содержат в пределах одного столбца, как правило, однородные данные, то есть данные одного типа и структуры.

Excel предоставляет возможность один раз ввести формулу расчета в ячейку, а затем скопировать ее из одной ячейки в другие.

Для этого существует несколько приемов копирования:

- Использование кнопок **Копировать** и **Вставить** в группе **Буфер обмена** на вкладке **Главная**
- Использование правой кнопки мыши, то есть вызов контекстного меню, и выбор соответствующих пунктов: **Копировать** и **Вставить**.
- Протаскивание «маркера заполнения» через требуемые смежные ячейки. Маркером заполнения при этом называется черный квадратик, расположенный в правом нижнем углу активной ячейки.
- Другие способы копирования в среде Windows (например, сочетание «горячих клавиш» Ctrl+C, Ctrl+V и т.п.).

5.4 Объединение и связывание нескольких электронных таблиц

EXCEL дает возможность создания итоговых или сводных таблиц различными способами. Можно просто суммировать данные рабочих листов и помещать результаты на итоговый лист, можно использовать кнопку **Консолидация** из группы **Работа с данными** на вкладке **Данные**, которая может объединить информацию из исходных листов в одном итоговом листе.

Команду **Консолидация** можно использовать несколькими способами. Можно связать консолидированные данные с исходными данными, чтобы последующие изменения в исходных листах отражались в итоговом листе. Или можно просто консолидировать исходные данные без создания связей.

Объединять данные можно, используя некоторые функции, например **Сумма**, **Среднее значение**, **Максимум**, **Минимум**, **Произведение** и т.д., которые приведены в окне с раскрывающимся списком **Функция** в диалоговом окне **Консолидация**. По умолчанию используется функция **Сумма**, которая суммирует данные из каждого исходного листа и помещает результат в итоговый лист.

Консолидировать данные можно по расположению или по категории. При консолидации по расположению Excel применяет итоговую функцию к ячейкам с одинаковыми адресами в каждом исходном листе. Это простейший способ консолидации, при котором консолидируемые данные во всех исходных листах должны иметь совершенно одинаковое расположение. Консолидация по категории в качестве основы для объединения использует заголовки строк или столбцов. **9.06 Построение диаграмм в Excel**

Значительный набор возможностей предоставляет пользователю Microsoft Excel для графического представления данных. Для построения диаграмм в Excel нужно выделить на рабочем листе ячейки с данными, после чего на вкладке **Вставка** в группе **Диаграммы** раскрыть меню кнопки с названием одного из типов диаграмм и выбрать наиболее подходящую диаграмму. Или щелкнуть по кнопке **Создать диаграмму** (диагональная стрелка) и в открывшемся окне **Вставка диаграммы** выбрать нужный тип диаграммы, дважды щелкнув по нему или выделив его щелчком кнопки мыши, а затем нажав кнопку **ОК**. Если ячейки, по данным которых требуется построить диаграмму, не являются смежными, нужно выделить первую группу ячеек с данными, а затем нажать и, удерживая нажатой клавишу Ctrl, выделить все остальные ячейки, значения которых необходимо отразить на диаграмме.

В любую диаграмму можно вносить следующие изменения:

- Добавление к диаграмме названия и подписей осей.
- Изменение вида осей.

- Добавление легенды и таблицы данных.
- Применение специальных возможностей для диаграмм разных типов.

5.5 1. Элементы диаграммы

Основные компоненты диаграммы представлены на следующей схеме:

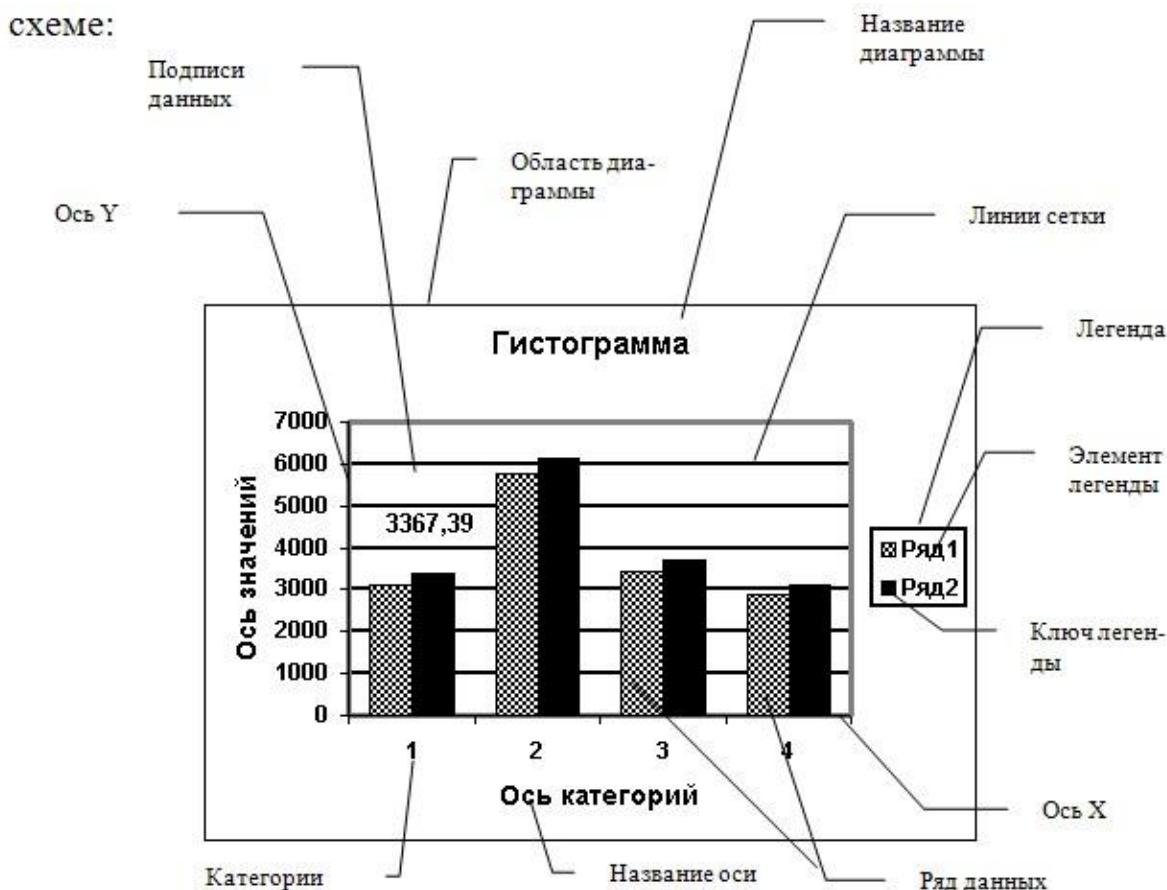


Рис. 25

Примечание: для объемной диаграммы составные части несколько отличаются.

2. Типы диаграмм

В зависимости от выбранного типа диаграммы можно получить различное отображение данных (см. рис. 25):

- **линейчатые диаграммы** и **гистограммы** могут быть использованы для иллюстрации соотношения отдельных значений или показа динамики изменения данных за определенный период времени;
- **график** отражает тенденции изменения данных за определенные промежутки времени;

- **круговые диаграммы** предназначены для наглядного отображения соотношения частей и целого;
- **точечная диаграмма** отображает взаимосвязь между числовыми значениями нескольких рядов данных и представляет две группы чисел в виде одного ряда точек, часто используется для представления данных научного характера;
- **диаграмма с областями** подчеркивает величину изменения данных во времени, показывая сумму введенных значений, а также демонстрирует вклад отдельных значений в общую сумму;
- **кольцевая диаграмма** показывает вклад каждого элемента в общую сумму, но, в отличие от круговой диаграммы, может содержать несколько рядов данных (каждое кольцо – отдельный ряд);
- **лепестковая диаграмма** позволяет сравнивать общие значения из нескольких рядов данных;
- **поверхностная диаграмма** используется для поиска наилучшего сочетания двух наборов данных;
- **пузырьковая диаграмма** представляет разновидность точечной диаграммы, где два значения определяют положение пузырька, а третье – его размер;
- **биржевая диаграмма** часто используется для демонстрации цен на акции, курсов валют, для определения изменения температуры, а также для научных данных.

Кроме того, можно строить диаграммы так называемого нестандартного типа, позволяющие совмещать в одной диаграмме различные типы представления данных. Такие диаграммы называются смешанными.

При работе с нестандартным типом диаграмм предусмотрена возможность быстрого просмотра диаграммы. Каждый нестандартный тип диаграммы основывается на стандартном типе и содержит дополнительные формат и параметры, такие как легенда, сетка, подписи данных, вспомогательная ось, цвета, шаблоны, заливки и места расположения различных элементов диаграммы.

Можно использовать либо один из встроенных нестандартных типов диаграмм, либо создать свой собственный. **3. Редактирование диаграмм**

Редактирование диаграммы осуществляется отдельно для каждого элемента после его выделения. Выделить элемент диаграммы можно следующим способом,

- Поместить на него указатель мыши и щелкнуть один раз по левой кнопке, при этом выделенный элемент диаграммы будет помещен в рамку с маркерами по периметру.
- Выделить всю диаграмму, щелкнув по ней в любом свободном месте, и выбрать элемент из списка, который открывается нажатием кнопки со стрелкой вниз справа от надписи **Область диаграммы** в группе **Текущий фрагмент** на вкладке **Макет** или **Формат** (см. рис. 26).

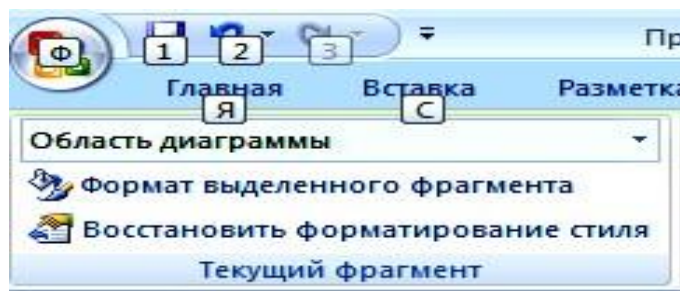


Рис. 26

После этого на вкладке **Макет** нужно нажать кнопку **Формат выделенного элемента** в группе **Текущий фрагмент** или вызвать контекстное меню с соответствующими командами для выделенного элемента нажатием правой кнопки мыши. Возможно редактирование изображения выделенного элемента и с помощью вкладки **Формат**.

4. Ввод текста названия диаграммы

Ввод текста названия диаграммы (если он не связан с ячейкой листа) можно осуществить одним из следующих способов:

- щелкнуть левой кнопкой мыши по данному элементу, чтобы выделить его, а затем щелкнуть его еще раз, чтобы поместить курсор внутрь текста, после чего ввести необходимое название диаграммы и щелкнуть левой кнопкой мыши за пределами данного элемента;
- щелкнуть левой кнопкой мыши по данному элементу диаграммы, чтобы выделить его, затем щелкнуть правой кнопкой мыши и в контекстном меню выбрать команду **Изменить текст**;
- щелкнув левой кнопкой мыши по названию, нажать функциональную клавишу **F2** на клавиатуре, ввести текст названия диаграммы в строку формул, после чего нажать клавишу **Enter**.

5. Добавление линии тренда к ряду данных

Для выявления общей тенденции изменения значений отдельного ряда на диаграмму выводится линия тренда.

Линия тренда – это линия регрессии, аппроксимирующая точки данных, или линия скользящего среднего. Линию тренда можно вывести за пределы, в которых данные уже известны, и показать тенденцию их изменения.

Линии тренда обычно используются в задачах прогнозирования, для сглаживания разброса цен, по линии тренда можно определить линейную корреляцию двух переменных и т.д.

Линии тренда различаются в зависимости от заданного математического соотношения между значениями ряда данных. Например, при сглаживании разброса цен на акции за определенный период может быть использован или линейный тип тренда, или скользящее среднее, при определении периода полураспада радиоактивного соединения понадобится экспоненциальный тип тренда и т.д. Выбор типа линии тренда всегда связан с предметной областью.

Линии тренда можно использовать не для всех типов диаграмм, а только для гистограмм, линейчатых диаграмм, графиков, XY-точечных диаграмм, диаграмм с областями.

Выведенная на диаграмму линия тренда сохраняет связь с исходным рядом, то есть при изменении данных соответственно изменяется линия тренда, а при удалении ряда линия тренда удаляется вместе с ним.

Тема 6 Основные понятия баз данных, структур данных и систем управления базами данных

Возможность работы с базами данных программ, предназначенных для обработки таблиц, является весьма важной в работе экономиста. Excel имеет различные инструментальные средства поддержки баз данных. Виды работ с ними можно разбить на 2 категории:

- Организация встроенных, расположенных непосредственно на рабочем листе, баз данных с помощью **СПИСКОВ**.
- Работа с внешними данными с помощью специальных запросов.

Современные базы данных представляют собой, как правило, набор взаимосвязанных таблиц. Каждая такая таблица состоит из множества строк, называемых **записями**. Каждая строка состоит из отдельных данных (реквизитов), называемых **ПОЛЯМИ**, для которых определены требования к типу содержащихся в них данных.

Таблица-список Excel должна удовлетворять следующим требованиям.

- Каждый столбец должен содержать информацию одного типа.
- Одна или две верхние строки списка должны содержать уникальные заголовки, каждый из которых описывает содержимое расположенного ниже столбца.
- Список не должен включать пустые строки и столбцы.
- Лучше всего, если для списка отводится отдельный лист. Если же это невозможно, то список должен быть отделен от других данных

рабочего листа, по крайней мере, одной пустой строкой и одним пустым столбцом.

- Не следует размещать данные слева и справа от списка, поскольку они могут быть скрыты в процессе фильтрации списка.

Над списками можно производить следующие действия:

- Сортировать список.
- Добавлять или изменять данные с помощью формы.
- Фильтровать список таким образом, чтобы были выбраны только те строки, которые удовлетворяют заданному критерию.
- Автоматически вычислять промежуточные итоги.
- Создавать для списка структуру.
- Создавать отчет, который в Excel называется Сводная таблица. С ее помощью можно мгновенно подытожить данные в списке.
- Проверять вводимые в ячейку данные.

6.1 Сортировка списков и диапазонов

Excel предоставляет разнообразные способы сортировки данных. Можно сортировать строки или столбцы в возрастающем или убывающем порядке данных, с учетом или без учета регистра букв. Можно задать и свой собственный пользовательский порядок сортировки. При сортировке строк изменяется порядок расположения строк, в то время как порядок столбцов остается прежним. При сортировке столбцов соответственно изменяется порядок расположения столбцов.

Стандартные средства Excel по умолчанию позволяют сортировать данные по нескольким признакам .

Перед тем как производить сортировку, необходимо установить курсор на любую ячейку сортируемой таблицы.

6.2 Промежуточные итоги

Довольно часто на практике приходится анализировать данные части таблицы или списка по определенным критериям. Для решения этой проблемы Excel предлагает операцию подведения промежуточных итогов. Только после выполнения сортировки данных таблицы можно использовать кнопку **Промежуточные итоги** из группы **Структура** на вкладке **Данные**, чтобы представить различную итоговую информацию. В результате этого в таблицу будут добавлены строки промежуточных итогов для каждой группы элементов списка, при этом можно использовать различные функции для вычисления итогов на уровне группы, а также общие итоги.

Команда Промежуточные итоги создает на листе структуру, где каждый уровень содержит одну из групп, для которых подсчитывается промежуточный итог. Вместо того чтобы рассматривать множество строк данных, можно закрыть любой из уровней и опустить тем самым ненужные детали.

6.3 Обеспечение поиска и фильтрации данных

Наиболее часто используемыми операциями над списками (базами данных) в Excel являются поиск и фильтрация данных.

Отфильтровать список – значит скрыть все строки, за исключением тех, которые удовлетворяют заданным условиям отбора. Для осуществления операций по фильтрации данных в Excel используются две команды: **Фильтр** – для простых условий отбора и **Расширенный фильтр** – для более сложных критериев.

Перед использованием команды **Фильтр** необходимо выделить любую ячейку в таблице. При этом Excel выведет кнопки со стрелками (кнопки фильтра) рядом с каждым заголовком столбца. Щелчок по кнопке со стрелкой рядом с заголовком столбца раскрывает список значений, которые можно использовать для того, чтобы задать условия отбора строк.

Фильтр можно применить к любому количеству столбцов. Для этого сначала нужно отфильтровать список по одному столбцу, затем полученный список отфильтровать по другому столбцу и т.д.

6.4 Применение расширенного фильтра

Основной особенностью, отличающей расширенный фильтр от автофильтра, является необходимость в создании вне таблицы вспомогательного списка, содержащего критерии условий поиска.

Команда **Расширенный фильтр** позволяет:

- Задать условия, соединенные логическим оператором **ИЛИ** или **И** для нескольких столбцов.
- Задать три или более условия для конкретного столбца с использованием, по крайней мере, одного логического оператора **ИЛИ**.
 - Задать вычисляемые условия.
- С помощью данной команды можно извлекать строки из списков, вставлять копии этих строк в другую часть текущего листа.

Команда **Расширенный фильтр** требует задания условий отбора строк в отдельном диапазоне рабочего листа. Поскольку при фильтрации скрываются целые строки, диапазон условий лучше поместить выше или ниже списка. Если предполагается, что список будет расширяться, то диапазон условий рекомендуется размещать выше таблицы. Диапазон условий должен

содержать, по крайней мере, две строки. За исключением вычисляемых условий, заголовки столбцов в верхней строке диапазона условий должны точно совпадать с заголовками столбцов таблицы. Для обеспечения точности эти заголовки лучше копировать из таблицы.

В диапазоне условий можно ввести любое количество условий, которые интерпретируются в соответствии со следующими правилами:

- Условия на одной строке считаются соединенными логическим оператором **И**.
- Условия на разных строках считаются соединенными логическим оператором **ИЛИ**.

6.5 Расширенный фильтр с использованием вычисляемых значений

Вычисляемые условия отличаются от обычных условий сравнения тем, что позволяют использовать сравнения со значениями, которые вычисляются формулой. Для правильного использования такой фильтрации необходимо выполнять следующие правила:

- Заголовок над вычисляемым условием должен обязательно отличаться от любого заголовка столбца в таблице. Заголовок условия может быть пустым или содержать произвольный текст. Это ограничение прямо противоположно требованию для обычных условий.
- Ссылки на ячейки, которые находятся вне таблицы (списка), должны быть абсолютными (при адресации используется знак доллара).
- Ссылки на ячейки в таблице должны быть относительными.

6.6 Анализ данных с помощью сводных таблиц

Средство Сводная таблица представляет собой еще один инструмент организации и подведения итогов данных. Этот инструмент комбинирует возможности работы со списками.

Сводная таблица является специальным типом таблицы, которая подытоживает информацию из конкретных полей списка или базы данных. При создании сводной таблицы можно задать нужные поля, организацию таблицы (ее макет) и тип выполняемых вычислений.

Сводные таблицы обеспечивают удобный интерфейс для анализа данных различной сложности и разного объема, а также возможность быстрой перестройки макета сводной таблицы фактически только с помощью мыши без применения сложного программирования.

Сводная таблица – это средство для упорядочивания информации. При создании сводной таблицы пользователь распределяет информацию, указывая, какие элементы и в каких полях будут содержаться.

Для создания сводных таблиц используются различные источники данных. Это могут быть списки и таблицы, расположенные на рабочих листах Excel, либо внешние источники данных (таблицы, созданные с помощью других программ).

Сводная таблица является многомерной и всегда связана с источником данных. Она предназначена только для чтения, а изменения можно вносить в исходные таблицы. При этом можно изменять форматирование сводных таблиц, выбирать различные параметры вычислений. Для большей наглядности на основе сводной таблицы можно создать диаграмму, которая будет перестраиваться при изменении сводной таблицы.

Пользователь может менять формат сводной таблицы, переименовывать поля или элементы, добавлять новые строки или столбцы, а также делать другие преобразования с внешним видом сводной таблицы.

Чтобы изменить название поля или элемента сводной таблицы, необходимо выделить нужную ячейку и набрать новое имя, которое не должно повторять существующие в исходной таблице имена.

1. Средства для анализа данных

- Подбор параметра.
- Создание таблицы подстановки с одной изменяющейся переменной.
- Создание таблицы подстановки с двумя изменяющимися переменными.
- Проверка результатов с помощью сценариев.

2. Подбор параметра

Средство **Подбор параметра** предназначено для получения заданного значения в целевой ячейке путем подбора значений в ячейкепараметре. С помощью данного средства можно найти аргумент, при котором получается задаваемое решение. При этом ячейка-параметр должна содержать значение, но не формулу и влиять на результат, который требуется получить.

3. Таблица подстановки

Таблицей подстановки называется диапазон ячеек, показывающий, как изменение значений подстановки влияет на возвращаемый формулой результат. Таблицы подстановки обеспечивают быстрый доступ к выполнению одной операции разными способами, а также возможность просмотра и сравнения полученных результатов. В Excel существует возможность

применения на одном рабочем листе нескольких таблиц подстановок, если необходимо:

- 1) изменять *одно исходное значение*, просматривая результаты одной или *нескольких формул*;
- 2) изменять *два исходных значения*, просматривая результаты *только одной формулы*.

В первом случае используется таблица подстановки с *одной изменяющейся переменной*, а во втором – с *двумя*.

6.7 Проверка результатов с помощью сценариев

В работе часто возникают задачи, имеющие множество исходных данных и множество результатов, причем необходимо четко представлять, как изменения первых влияют на последние. Ощутимую помощь в анализе такого рода задач могут оказать сценарии Excel.

Сценарий – это инструмент, позволяющий моделировать различные экономические, математические, физические и др. задачи. Он представляет собой зафиксированный в памяти компьютера набор значений ячеек рабочего листа. Использование сценариев позволяет одновременно манипулировать с множеством переменных. Создав сценарий, можно получить возможность узнать, что произойдет с результатом, если поменять исходные значения в некоторых ячейках листа. Кроме того, в случае необходимости всегда можно вернуться к одному из вариантов, рассмотренных ранее.

6.8 Системы управления базами данных

8.01 Общие сведения о СУБД

подавляющее большинство деловых операций сегодня записывается, отслеживается и анализируется в виде данных, хранящихся в системах управления базами данных.

Система управления базами данных – СУБД (Database Management System – DBMS) – является универсальным программным инструментом создания и обслуживания баз данных и приложений пользователя в самых разных предметных областях. СУБД обеспечивает создание, многоаспектный доступ к данным и использование одних и тех же данных различными задачами и приложениями пользователей.

СУБД поддерживаются различные модели данных. Модель данных – это метод логической организации данных, используемый СУБД. Наиболее известными являются иерархическая, сетевая и реляционная модели.

В настоящее время существует несколько сотен различных СУБД, которые работают на персональных компьютерах, майнфреймах или в сети из многих компьютеров. Появившиеся в конце 70-х – начале 80-х годов реляционные СУБД устранили недостатки предыдущих моделей и получили наибольшее распространение.

В настоящее время СУБД поддерживается преимущественно реляционная модель, которую отличает простота и единообразие представления данных простейшими двумерными таблицами. Реляционная модель обеспечивает возможность использования в разных СУБД операций обработки данных, имеющих единую основу – алгебру отношений (реляционную алгебру), и универсального языка структурированных запросов – SQL (Structured Query Language). Наиболее популярной реляционной СУБД для персональных компьютеров (настольной СУБД) является Microsoft Access. Среди многопользовательских СУБД можно выделить Microsoft SQL Server, Oracle, Informix.

Рассматривая функциональные возможности СУБД, будем обращаться к Microsoft Access, которая включает все необходимые для их реализации инструментальные средства. Access обеспечивает создание локальной базы данных и приложения пользователя, работающего с этой базой. База данных Access может быть размещена в локальной сети с файловым сервером и доступна для приложений нескольких пользователей. Кроме того СУБД Access включает средства, обеспечивающие подключение к базам данных на сервере Microsoft SQL Server, создание серверных баз данных и их объектов, а также

средства разработки пользовательского приложения для многопользовательской базы данных.

8.02 Реляционная база данных

1. Определения и понятия

База данных является организованной на машинном носителе совокупностью взаимосвязанных данных и содержит сведения о различных сущностях одной предметной области – реальных объектах, процессах, событиях или явлениях.

Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц – реляционных таблиц, называемых также отношениями, в каждой из которых содержатся сведения об одной сущности автоматизируемой предметной области – реальном объекте, процессе, событии или явлении.

В таблицах базы должны сохраняться все данные, необходимые для решения задач предметной области. Причем желательно, чтобы каждый элемент данных хранился в базе только один раз. Минимальное дублирование данных в реляционной базе обеспечивает высокую эффективность поддержания базы данных в актуальном и непротиворечивом состоянии, однократный ввод и корректировку данных. Для достижения этой цели в реляционной модели данных используется процесс, называемый нормализацией данных. Нормализация – это удаление из таблиц повторяющихся данных путем их переноса в новые таблицы, строки которых не содержат повторяющихся значений.

Структура реляционной таблицы определяется составом полей. Каждое поле отражает определенную характеристику сущности. Для поля указывается тип и размер элементарных данных, размещаемых в нем, и ряд других свойств. Содержимое поля отображается в столбце таблицы.

Столбец таблицы содержит данные одного типа.

Содержание таблицы заключено в ее строках, однотипных по структуре. Каждая строка таблицы содержит данные о конкретном экземпляре сущности и называется записью.

Для однозначного определения (идентификации) каждой записи таблица должна иметь уникальный (первичный) ключ. По значению ключа таблицы отыскивается единственная запись в таблице. Ключ может состоять из одного или нескольких полей таблицы. Значение уникального ключа не может повторяться в нескольких записях.

Обеспечивая рациональное хранение недублированных данных, необходимо предусмотреть возможность объединения данных из разных таблиц. Для объединения данных в соответствии с требованиями решаемых

задач между таблицами устанавливаются логические связи. Связь каждой пары таблиц обеспечивается одинаковыми полями в них – ключом связи.

В нормализованной реляционной базе данных связь двух таблиц характеризуется отношениями записей типа «один-к-одному» (1 — 1) или «один-ко-многим» (1 — М). Отношение 1 — 1 предполагает, что каждой записи одной таблицы соответствует одна запись в другой. Отношение 1 — М предполагает, что каждой записи первой таблицы соответствует много записей во второй, но каждой записи второй таблицы соответствует только одна запись в первой.

Для двух таблиц, находящихся в отношении типа 1 — М, связь устанавливается по уникальному ключу таблицы, представляющей в отношении сторону «один», – главной таблицы в связи. Во второй таблице, представляющей в отношении сторону «многие» и называемой подчиненной, этот ключ связи может быть либо частью уникального ключа, либо не входить в состав ключа. В подчиненной таблице ключ связи называется еще внешним ключом.

На рис. 14 показаны две таблицы со списком покупателей и перечнем заключенных договоров. Эти таблицы находятся в отношении типа 1 — М и логически связаны с помощью общего поля (столбца) Код покупателя – ключа связи. Это поле является уникальным ключом в главной таблице – ПОКУПАТЕЛЬ и неключевым полем в подчиненной таблице – ДОГОВОР.

Связь 1:М по одноименному поля

Уникальный ключ Поле Главная таблица Подчиненная таблица

ПОКУПАТЕЛЬ							
Код покупателя	ИНН	Наименование	Адрес	Телефон	Номер счета	БАНК	
П001	77895765111	Компьютер маркет	Москва	(812)345-23-45	76358509763264	Мост	
П002							
П004							
П005							
П006							
П007							
П008							

ДОГОВОР				
Номер договора	Дата заключения	Код покупателя	Сумма по договору	
Д111	11.01.2010	П001	6 047 100,00р.	
Д222	05.02.2010	П001	152 280,00р.	
Д333	01.01.2010	П002	74 920,00р.	
Д555	12.11.2010	П002	9 352,00р.	
Д777	14.06.2010	П001	1 579 920,00р.	
Д888	23.05.2010	П002	30 000,00р.	
Д999	12.09.2010	П002	450 000,00р.	
*			0,00р.	

Зачисления: 2 из 7

Зачисления: 3 из 7

Рис. 14. Взаимосвязанные таблицы реляционной базы данных

Размещение сведений о каждой сущности в отдельной таблице и связывание таблиц позволяет избежать повторения описательных данных в разных таблицах. При этом обеспечивается однократный ввод данных при

загрузке и корректировке базы данных. Если данные двух таблиц в приведенном примере разместить в одной таблице, то каждая запись должна соответствовать одному договору. Причем данные о покупателе (наименование, ИНН, адрес и др.) будут повторяться во всех записях о договорах одного покупателя, что усложнит ввод, корректировки и обеспечение актуального состояния базы данных. При хранении данных в двух таблицах, сведения о покупателе хранятся в единственном экземпляре, а в таблице договоров повторяются только значения ключевого поля с кодом покупателя.

В СУБД Access реализовано средство просмотра и редактирования связанных записей нескольких таблиц. При этом данные отображаются в иерархическом виде. При раскрытии одного уровня иерархии рядом с записью главной таблицы отображаются связанные записи подчиненной. Для записи подчиненной таблицы также могут быть открыты связанные записи. Например, для таблиц ПОКУПАТЕЛЬ, ДОГОВОР (рис. 15), связанных отношением 1 — М, для каждой записи таблицы ПОКУПАТЕЛЬ могут быть отображены и отредактированы связанные записи в таблице ДОГОВОР.

Код покупателя	ИНН	Наименование	Адрес	Телефон	Номер			
P001	77895765111	Компьютер маркет	Москва	(812)345-23-45	763			
Номер договора						Дата заключения	Сумма по дог.	Код исполнителя
D111						11.01.2010	6 047 100,00р.	6
D222						05.02.2010	152 280,00р.	1
D777						14.06.2010	1 579 920,00р.	1
*							0,00р.	
P002	78988979879	Перспектива	Москва	(995)345-67-89	293			
Номер договора						Дата заключения	Сумма по дог.	Код исполнителя
D333						01.01.2010	74 920,00р.	3
D555						12.11.2010	9 352,00р.	3
D888						23.05.2010	30 000,00р.	6
D999						12.09.2010	450 000,00р.	7
*							0,00р.	
P004	45657567567	Монитор	Санкт-Петерб ()	123-45-67	585			
P005	77124356782	Компьютер лэнд	Саратов	()123-56-23	597			
P006	58787987912	Компьютерная техни	Ярославль	()123-45-67	763			

Рис. 15. Отображение в записях главной таблицы связанных записей подчиненной таблицы

2. Схема данных

В СУБД Access процесс создания реляционной базы данных включает создание схемы данных. Схема данных наглядно отображает логическую структуру базы данных: таблицы и связи между ними, а также обеспечивает использование установленных в ней связей при обработке данных.

Для нормализованной базы данных, основанной на одно-многозначных и одно-однозначных отношениях между таблицами, в схеме данных для связей таких таблиц по первичному ключу или уникальному индексу главной таблицы могут устанавливаться параметры обеспечения связной целостности.

При поддержании целостности взаимосвязанных данных не допускается наличия записи в подчиненной таблице, если в главной таблице отсутствует связанная с ней запись. Соответственно при первоначальной загрузке базы данных, а также корректировке, добавлении и удалении записей система допускает выполнение операции только в том случае, если она не приводит к нарушению целостности.

Связи, определенные в схеме данных, автоматически используются для объединения таблиц при разработке многотабличных форм, запросов, отчетов, существенно упрощая процесс их конструирования.

В схеме данных связи могут устанавливаться для любой пары таблиц, имеющих одинаковое поле, позволяющее объединять эти таблицы.

3. Объекты базы данных

Все возрастающая сложность приложений баз данных привела к реализации в СУБД объектно ориентированного подхода, предполагающего создание программного обеспечения на основе стандартных и повторно используемых компонентов.

Расширение реляционных СУБД функциями объектно-ориентированного подхода определило включение понятия «объект» и реализацию возможности сохранения методов (процедур), так же как и данных в базе.

В качестве объектов в объектно ориентированном программировании рассматриваются любые программируемые элементы. Каждый объект обладает набором свойств, при изменении которых можно управлять объектом. С каждым объектом связывается ряд методов – процедур, которые работают с объектом. Один и тот же метод может использоваться для обработки различных объектов.

Все многообразие объектов, их свойств и методов обработки сводится в иерархически упорядоченные совокупности, что облегчает программирование.

Основными объектами Access, как и многих других СУБД, являются:

- таблицы, запросы, схемы данных, имеющие непосредственное отношение к базе данных («Объекты Access»);
- формы, отчеты, макросы и модули, называемые объектами приложения.

Формы и отчеты предназначены для типовых процессов обработки данных – просмотра, обновления, поиска по заданным критериям, получения отчетов.

Эти объекты приложений конструируются из графических элементов, называемых элементами управления. Основные элементы управления служат для отображения полей таблиц, являющихся источниками данных объекта.

Для автоматизации доступа к объектам и их взаимодействия используется программный код. Только с помощью программного кода получается полноценное приложение пользователя, функции которого доступны через меню, панели инструментов и формы. Для создания программного кода используются модули на языке VBA и макросы.

Каждый объект и элемент управления имеет свой сохраняемый набор свойств. Определяя свойства можно настраивать и управлять объектом. С каждым объектом и элементом управления связывается набор событий, которые могут обрабатываться макросами или процедурами обработки событий на VBA, входящими в состав модулей форм, отчетов.

Объекты представлены в области навигации окна базы данных Access. Все операции по работе с объектами собственно базы данных и приложений начинаются в этом окне.

Таблицы (Tables) создаются пользователем для хранения данных об одной сущности – одном информационном объекте модели данных предметной области. Таблица состоит из полей (столбцов) и записей (строк). Каждое поле содержит одну характеристику информационного объекта предметной области. В записи собраны сведения об одном экземпляре информационного объекта.

База данных Access может включать до 32 768 объектов (в том числе формы, отчеты и т.д.). Одновременно может открываться до 2048 таблиц.

Запросы (Queries). Запросы на выборку служат для выборки нужных данных из одной или нескольких связанных таблиц. Результатом выполнения запроса является виртуальная таблица. В запросе можно указать, какие поля исходных таблиц следует включить в запись таблицы запроса и как отобразить нужные записи. Таблица запроса может быть использована наряду с другими таблицами базы при обработке данных. Запрос может формироваться с помощью конструктора запросов или инструкции языка SQL. Запросы на изменение позволяют обновлять, удалять или добавлять данные в таблицы, а также создавать новые таблицы на основе существующих.

Схема данных (Relationships) определяет, с помощью каких полей таблицы связываются между собой, как будет выполняться объединение данных этих таблиц, нужно ли проверять связную целостность при добавлении и удалении записей, изменении ключей таблиц. Схемы данных в области навигации в окне базы данных отображаются только в проектах Access, работающих с базами данных сервера. Формы (Forms) являются основным средством создания диалогового интерфейса приложения пользователя. Форма может создаваться для работы с электронными документами, сохраняемыми в таблицах базы данных. Вид таких документов может соответствовать привычному для

пользователя бумажному документу. Форма используется для разработки интерфейса по управлению приложением. Включаемые в форму процедуры обработки событий позволяют управлять процессом обработки данных в приложении. Такие процедуры хранятся в модуле формы. В формы могут вставляться рисунки, диаграммы, звуковые фрагменты, видео. Возможна разработка форм с набором вкладок, с каждой из которых связано выполнение той или иной функции приложения.

Отчеты (Reports) предназначены для формирования на основе данных базы выходных документов любых форматов, содержащих результаты решения задач пользователя, и вывода их на печать. Как и формы, отчеты могут включать процедуры обработки событий. Использование графических объектов позволяет дополнять данные отчета иллюстрациями. Отчеты обеспечивают возможность анализа данных при использовании фильтрации, агрегирования и представления данных источника в различных разрезах.

Макросы (Macros) являются программами, состоящими из последовательности макрокоманд, которая выполняется по вызову или при наступлении некоторого события в объекте приложения или его элементе управления. Макросы позволяют автоматизировать некоторые действия в приложении пользователя. Создание макросов осуществляется в диалоговом режиме путем выбора нужных макрокоманд и задания параметров, используемых ими при выполнении. Новые возможности конструктора макросов в Access 2010 упрощают создание, редактирование макросов, позволяют сокращать количество ошибок кода и более эффективно создавать надежные приложения. В Access 2010 появилась новая возможность – макросы данных, позволяющие изменять данные на основе событий в исходных таблицах. Макросы данных используются для добавления логики к данным и сосредоточения ее в исходных таблицах. В Web-приложениях Access, базирующихся на базах данных, опубликованных в SharePoint, для программирования необходимо использовать только макросы, так как код VBA не совместим со средствами Webпубликации.

Модули (Modules) содержат процедуры на языке Visual Basic for Applications. Могут создаваться процедуры-подпрограммы, процедуры-функции, которые разрабатываются пользователем для реализации нестандартных функций в приложении пользователя, и процедуры для обработки событий. Использование процедур позволяет создать законченное приложение, которое имеет собственный графический интерфейс пользователя, позволяющий запросить выполнение всех функций приложения, обработать все ошибки и нестандартные ситуации.

В Access для удобства пользователя объекты в области навигации базы данных могут быть объединены в пользовательские группы по функциональному или иному признаку. Группы содержат ссылки на объекты

базы данных различных типов. Группы, в свою очередь, объединяются в категории. С помощью такой организации объектов базы данных может быть разработан интерфейс пользовательского приложения, обеспечивающий доступность только к категориям и группам, наглядно и понятно представляющим функциональность приложения.

Все объекты базы данных Access размещаются на диске в одном файле формата ACCDB («Файл базы данных»). Это упрощает технологию ведения базы данных и приложения пользователя. Обеспечивается высокая компактность размещения всех объектов базы данных на диске и эффективность обработки данных.

8.03 Оперативный анализ данных («Сводные диаграммы», «Сводные таблицы»)

В СУБД не остались без внимания и инструменты оперативного решения задач бизнес-аналитики, обеспечивающие анализ и представление информации, необходимой для принятия решений в удобном для понимания виде.

Эти инструменты реализованы с помощью отчетов сводных таблиц и диаграмм, которые используются для суммирования, статистического анализа, исследования большого количества исходных сводных данных, содержащих повторяющиеся значения, и представления их в удобном для пользователя виде. Строки и столбцы подобной интерактивной таблицы базируются на данных из нескольких столбцов исходной таблицы.

Следует отметить, что сводные таблицы диаграммы являются средством оперативной аналитической обработки данных кубов OLAP (On-Line Analytical processing), созданных с помощью Analysis Services Microsoft SQL Server.

Access позволяет открывать в режимах сводной таблицы и сводной диаграммы таблицы, запросы и формы.

Сводная таблица представляет собой интерактивную таблицу, с помощью которой можно анализировать данные, быстро объединяя большие объемы данных и рассчитывая итоги (рис. 16). С помощью сводных таблиц делается простым выполнение сложного анализа данных.

Отгрузка товаров						
Дата отгрузки по месяцам ▼ 2007						
Наименование покупателя ▼ Компьютер маркет			Номер договора ▼ Перспектива		Общие итоги	
Д111		Д222	Итоги	Д333	Итоги	
+-		+-	+-	+-	+-	+-
Наименование товара ▼	Сумма "Колич	Сумма "Колич	Сумма "Колич	Сумма "Колич	Сумма "Колич	Сумма "Колич
CD-ROM Panasonic IDE	4	16	20			20
DIMM 64M PC100	5			5	1	6
FDD 3,5		15	15			15
HDD Maxtor 20GB		8	8	3	3	11
Зв. Карта Genius Liv		10				13
Монитор 17LG	2	7				13
Сканер Acer		10				22

Значение: 8
Итог: Сумма "Количество отгружено"
Компонент строки: HDD Maxtor 20GB
Компонент столбца: Компьютер маркет - Итоги
Фильтр: Дата отгрузки по месяцам = 2007

Рис. 16. Сводная таблица для анализа суммарного количества отгруженного по любому из товаров по различным покупателям и договорам по всем или некоторым месяцам, кварталам, годам. Для получения различных итогов по исходным данным достаточно в созданном макете сводной таблицы выбрать значения в поле строк (Наименование товара ▼), поле столбцов (Наименование покупателя ▼, Номер договора ▼) и поле страницы (фильтра) (Дата отгрузки по месяцам ▼).

Сводные таблицы позволяют динамически изменять макет для всестороннего анализа данных. Существует возможность изменять заголовки строк, столбцов, а также полей, определяющих страницу. Создавать и быстро модифицировать макет можно, выбирая и перетаскивая поля из раскрывающегося списка полей в рабочую область. При каждом изменении макета сводная таблица немедленно выполняет вычисления заново в соответствии с новым расположением данных.

Сводные диаграммы служат для наглядного графического представления анализируемой информации, облегчая для пользователей сравнение и выявление тенденций и закономерностей в данных (рис. 17).

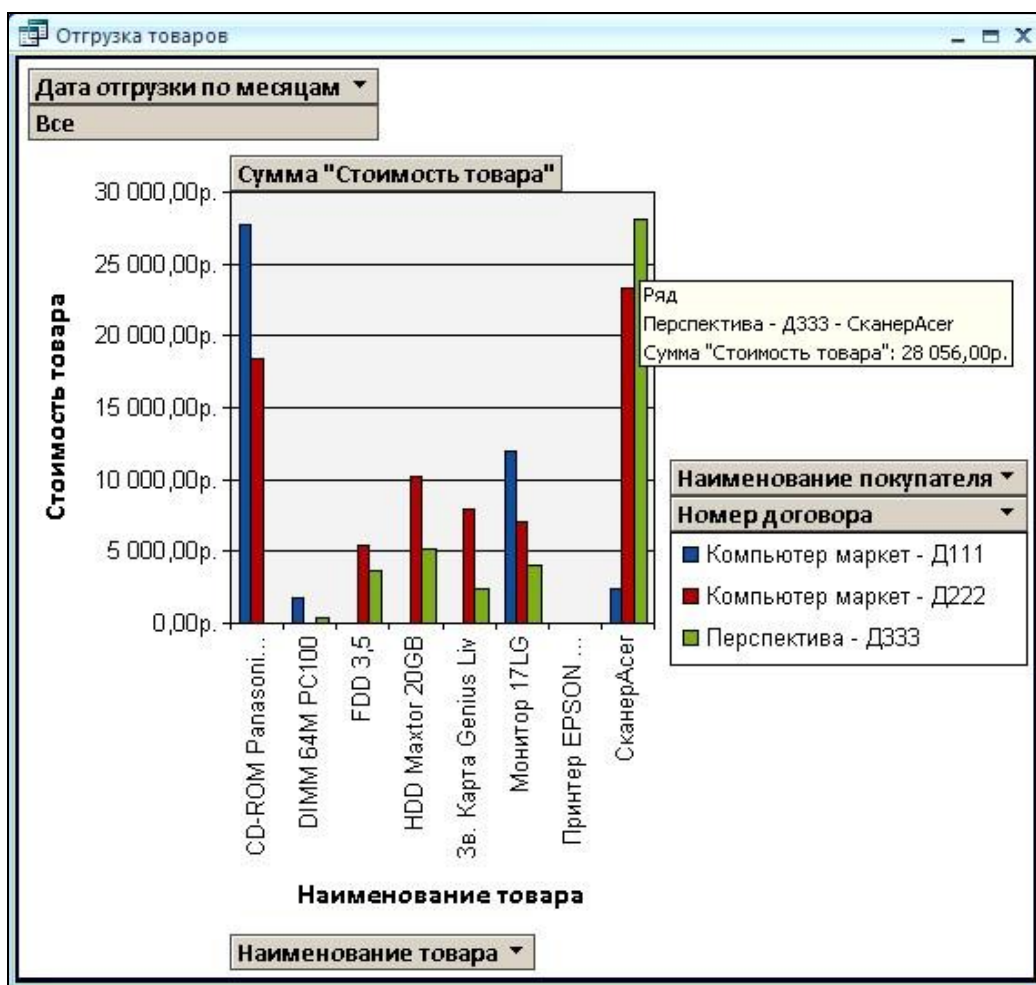


Рис. 17. Сводная диаграмма для анализа суммарного количества отгруженного по любому из товаров по различным покупателям и договорам по всем или некоторым месяцам, кварталам, годам. Таким образом, источником записей для режима сводной таблицы и режима сводной диаграммы может быть как таблица, так и базовый источник данных для формы. В базе данных Access источником записей может быть таблица, запрос или инструкция SQL; в проекте Access – таблица, представление, инструкция SQL, хранимая процедура или табличная функция.

8.04 Средства создания объектов базы данных

Access располагает средствами, которые позволяют создавать объекты базы данных и приложения пользователя с помощью мастеров и конструкторов, не прибегая к программированию.

1. Мастера Access

Access располагает разнообразными диалоговыми средствами, которые позволяют создавать объекты базы данных и приложения, не прибегая к программированию.

Множество мастеров Access позволяет автоматизировать процесс создания таблиц базы данных, запросов, форм, отчетов; анализировать таблицы базы данных и выполнять многие другие работы. Практически для любых работ имеется мастер (Wizard), который поможет их выполнить.

Мастера по созданию форм и отчетов упрощают и ускоряют процесс создания однотоабличных и многотоабличных форм и отчетов. Так, выбрав таблицу или запрос в области навигации базы данных, одним щелчком можно создать форму, отображающую поля только одной записи в столбец или отображающую несколько записей в виде таблицы или смешанное представление записей. В диалоге с другим мастером пользователю достаточно выбрать таблицы и поля, которые необходимо включить в форму, выбрать источник основной и подчиненной части формы. Мастера по разработке форм и отчетов автоматически создают инструкцию SQL, описывающую источник записей для формы или отчета, что избавляет пользователя от подготовки запроса.

Мастер подстановок (Lookup Wizard) создает в поле таблицы или формы раскрывающийся список значений из полей другой таблицы или запроса для выбора и ввода в поле нужного значения. Созданные в полях таблиц списки наследуются при включении этих полей в форму (поле со списком).

Мастера запросов позволяют создавать простые запросы на выборку или запросы на выборку, в которых выполняются групповые операции над данными из одной или нескольких таблиц.

Мастер перекрестных запросов (Crosstab Query Wizard) формирует из взаимосвязанных таблиц или запросов базы данных таблицу, подобную электронной, в которой одно поле используется в качестве заголовков строк, второе – столбцов, а на их пересечении размещаются итоговые данные, рассчитываемые по значениям третьего поля.

Мастер по созданию диаграмм (Chart Wizard) обеспечивает создание в формах и отчетах диаграмм, базирующихся на данных в таблицах или запросах. На рис. 18 приведена форма, позволяющая просматривать в справочнике данные о товаре и синхронно отображать диаграмму с количеством заказанного по договорам (в примере Д111, Д222, Д333). Вызывается мастер с помощью элемента управления Диаграмма (Chart) «Элементы управления: Вставить диаграмму», расположенного на вкладке ленты конструктора форм в группе Элементы управления (Controls). Ответив на ряд элементарных вопросов, легко можно получить диаграмму, связанную с выбранными в форме данными.

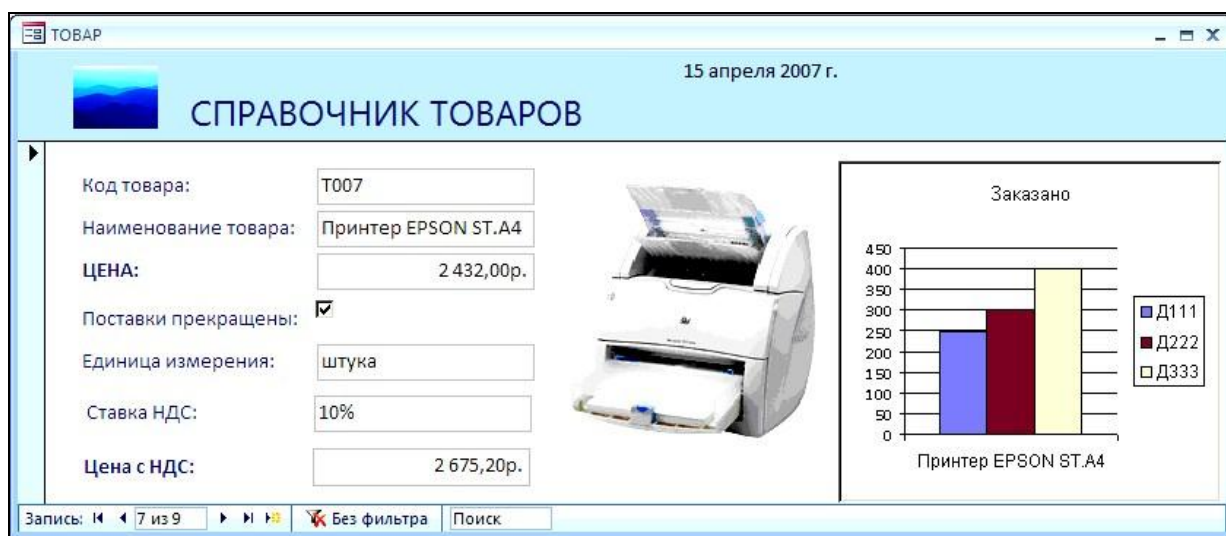


Рис. 18. Форма с диаграммой, построенной мастером

После добавления диаграммы или открытия существующей диаграммы с помощью двойного щелчка мышью может быть выполнен переход к ее редактированию с помощью Microsoft Graph.

Мастер кнопок (Command Button Wizard и Page Command Button Wizard) создает командные кнопки – элементы управления в форме. При построении кнопки мастер предлагает большой набор действий, которые могут быть выполнены при щелчке на кнопке. Для выбранного действия мастер создает встроенный макрос и связывает его с событием Нажатие кнопки (On Click). Формы с командными кнопками, связанными с макросами, позволяют реализовать удобный интерфейс пользователя для управления приложением.

Мастер по анализу таблиц (Table Analyzer Wizard) позволяет выполнить нормализацию данных базы. Мастер разделяет ненормализованную таблицу с дублированными данными на две или несколько таблиц меньшего размера, в которых данные сохраняются без повторений. Вызывается мастер с помощью элемента управления Анализ таблицы (Table Analyzer), расположенного на стандартной вкладке ленты Работа с базами данных в группе Анализ (Analyzer).

Мастер сводных таблиц и диаграмм (PivotTable Wizard) позволяет открыть существующие таблицы, запросы, формы в режиме сводной таблицы и создать макет.

Служебная программа (Database Utilities) Преобразовать базу данных (Convert Database), которая ранее позволяла конвертировать базу данных из предыдущих версий в текущую и наоборот, в Access 2010 заменена командами Сохранить базу данных как доступными в разделе Сохранить и опубликовать на вкладке Файл. Если база данных Access 2007/2010 использует возможности,

не поддерживаемые более ранними версиями, выполнение преобразования невозможно.

Служебная программа (Database Utilities) Сжать и восстановить базу данных (Compact and Repair Database) заменена одноименной командой, доступной в разделе Сведения на вкладке Файл (File). Команда реорганизует базу данных на диске с целью улучшения объемно-временных характеристик. Перезаписывает фрагментированный в результате удаления и добавления объектов файл базы данных и сокращает его размер, а также восстанавливает поврежденную базу данных.

2. Средства конструирования объектов

Для автоматизации процесса создания объектов базы данных – таблиц, схемы базы данных, запросов – и объектов приложения – форм, отчетов, макросов – используются специализированные диалоговые графические средства, называемые конструкторами (Design). Конструктор предоставляет пользователю набор элементов, с помощью которых быстро создается и модифицируется объект. Для конструирования макета формы, отчета используются элементы управления, которые доступны в режиме макета и режиме конструктора на вкладках ленты Работа с макетами форм/отчетов или Инструменты конструктора форм/отчетов.

Дополнительные возможности по изменению структуры форм и отчетов в режиме макета были включены в Access начиная с 2007. В режиме макета форма или отчет пользователь, изменяя макет, одновременно видит данные почти в том же виде, в каком они отобразятся в режиме формы или после вывода отчета на печать. Режим макета представляет собой наиболее наглядный режим для изменения макета форм и отчетов. Его можно использовать для внесения практически любых изменений. Поскольку при внесении изменений можно просматривать данные, в этом режиме очень удобно задавать размеры элементов управления и выполнять другие задачи, связанные с внешним видом и удобством формы или отчета. Некоторые задачи невозможно выполнить в режиме макета, и тогда необходимо переключиться в режим конструктора. Access выдает сообщение о том, что для внесения изменений следует переключиться в режим конструктора.

В Access 2007 формирование источника данных для формы или отчета может быть выполнено с помощью списка доступных полей. При этом оно сводится к простому перетаскиванию полей из таблиц базы данных в нужное место формы/отчета в режиме конструктора или макета.

В проекте Access, связанном с сервером баз данных MS SQL Server, конструкторы таблиц, представлений, хранимых процедур, определяемых пользователем функций, позволяют работать с объектами базы данных сервера

в простом привычном для пользователя локальных баз данных графическом режиме.

Для упрощения внесения изменений в объекты базы данных Access предлагает технологию интеллектуальной замены имен объектов («Автозамена имен объектов») в базе данных. При этом автоматически исправляются ошибки, вызванные переименованием таблиц, полей, форм, отчетов, запросов, текстовых блоков или других элементов управления. Реализуется эта технология за счет того, что каждый именуемый объект (или элемент) базы данных имеет внутренний уникальный идентификатор, а имя является только псевдонимом. При переименованиях изменяется лишь псевдоним, и при необходимости в других объектах корректируются все ссылки на объект. Для применения этой технологии следует установить параметры, отслеживать автозамену имен (Track name AutoCorrect info) и выполнять автозамену имен (Perform name AutoCorrect) в окне Параметры Access (Access Options) на вкладке Текущая база данных (Current Database) в разделе Параметры автозамены имен (Name AutoCorrect Options). Окно Параметры Access (Access Options) открывается соответствующей командой в представлении Backstage, открываемом на вкладке Файл.

3. Средства программирования

Наряду с диалоговыми средствами создания объектов базы данных и объектов приложения, которые позволяют решить многие задачи пользователя, Access располагает мощными средствами программирования. Эти средства могут использоваться как для доработки приложений, созданных диалоговыми средствами, так и для реализации сложных задач и создания приложений с необходимым пользователю интерфейсом в целом. Без использования программного кода практически невозможно получить законченное автоматизированное приложение пользователя.

Одним из средств программирования в Access является язык макрокоманд. Программы, созданные на этом языке, называемые макросами, позволяют легко связывать отдельные действия, реализуемые с помощью форм, запросов, отчетов. Макросы управляются событиями, которые вызываются действиями пользователя при диалоговой работе с данными через формы, например нажатие кнопки, изменение данного в поле, или системой.

Простой язык макрокоманд и новый конструктор макросов с ясным и понятным интерфейсом, поддерживающий функции IntelliSense, позволяют при малой трудоемкости интегрировать объекты приложения и организовать процесс обработки данных. Конструктор макросов упрощает создание, редактирование и автоматизацию логики базы данных, позволяет сокращать количество ошибок кода.

Помимо обычных макросов, новый конструктор макросов позволяет создавать макросы данных, которые являются новшеством в данной версии.

Макросы данных связываются с событиями в таблице и позволяют выполнять определенные действия при изменении, вставке или удалении записи. Например, с помощью макросов можно проверять данные или выполнять вычисления. Макросы данных позволяют добавлять логику к данным и сосредотачивать логику в исходных таблицах. По сути макросы данных позволяют реализовать триггеры в любой базе данных Access 2010.

С помощью усовершенствованного конструктора макросов и макросов данных можно распространить автоматизацию за пределы клиентского приложения Access на веб-базы данных SharePoint и другие приложения, обновляющие таблицы Access.

Наряду с языком макрокоманд, Access включает развитую интегрированную среду объектно ориентированного программирования Visual Basic for Applications (VBA), позволяющую реализовать любые программные решения. Программы на VBA реализуются процедурами, которые объединяются в объектах, называемых модулями.

В VBA база данных рассматривается как совокупность объектов (таблиц, форм, отчетов, их элементов и т.д.), имеющих свойства и методы, реализующие заранее определенные действия над объектами. Структурированность объектов базы данных упрощает освоение этого языка и создание приложений. Управление выполнением программ в диалоговых приложениях VBA осуществляется событиями, вызываемыми действиями пользователя или системы.

Среда VBA объединяет разнообразные наглядные графические инструменты: редактор VBA, окно разрабатываемого проекта, окно свойств объектов проекта, окно просмотра объектов, отладчик и др. Все инструменты унифицированы и являются общими для всех приложений Microsoft Office, для Visual Basic, а также продуктов ряда других фирм.

Приложения, разрабатываемые на VBA, могут выполняться только в той среде, в которой поддерживается VBA, в то время как Visual Basic ориентирован на полностью самостоятельную разработку автономно выполняющихся приложений. Язык VBA является производным от самостоятельной системы программирования Visual Basic и имеет с ним много общего. Их синтаксис и интерфейс практически одинаковы.

Заметим, что код VBA несовместим со средством веб-публикации, поэтому, если планируется опубликовать приложение как веб-приложение Access для выполнения задач программирования, необходимо использовать только макросы.

8.05 Интеграция и использование внешних данных

Access продолжает поддерживать технологию OLE (Object Linking and Embedding, связь и внедрение объектов), обеспечивающую возможность интеграции данных различных приложений в составном документе. С помощью OLE пользователь может внедрять объекты другого приложения в базу данных или устанавливать с ними связи. Активизация внедренного объекта запускает программу, которая его создала, и пользователь может изменить объект. При установлении связи с объектом он по-прежнему сохраняется в файле другого приложения, а не в базе данных. За счет этого он может обновляться независимо, а в базе данных всегда будет представлена последняя версия объекта. Надо иметь в виду, что при изменении местоположения файла, связь с ним должна обновляться пользователем.

Внедряемыми или связываемыми объектами могут быть документы различных приложений Windows – рисунки, графики, электронные таблицы, звуковые или видеофайлы. Например, в таблице, наряду с обычными реквизитами, характеризующими информационный объект, может храниться любая графическая информация о нем – схемы, чертежи, диаграммы, рисунки и т.п. Таким образом, в Access расширяется традиционное понятие данных, хранимых в базе и представляющих информационные объекты.

В Access 2007/2010 обеспечивается хранение одного или нескольких файлов разных типов – документов Word, презентаций PowerPoint, изображений и т.п. – в поле записи базы данных имеющем тип данного Вложение (Attachment). Вложения позволяют хранить данные более рационально.

Access может использовать данные различных внешних источников. Внешними источниками данных могут служить таблицы других баз данных Access, dBase, Paradox, Oracle и Microsoft SQL Server, электронные таблицы Microsoft Excel, Lotus 1-2-3, таблицы и списки HTML и файлы XML, списки Windows SharePoint Services, текстовые файлы и др., находящиеся на локальном компьютере или на сервере в локальной, корпоративной сети или Интернет.

Для взаимодействия с внешними источниками данных применяется специальное программное обеспечение – интерфейсы. Широко используемыми интерфейсами являются ODBC (Open Database Connectivity, открытый интерфейс подключения к базам данных), который служит, прежде всего, для доступа к базам данных, и интерфейс модели составного объекта (Component Object Model), называемые OLE DB, разработанные как средство универсального доступа к данным по сети. OLE DB позволяет подсоединяться к источникам данных многих типов, в том числе реляционным источникам

данных, почтовым файлам, неформатированным текстовым файлам и электронным таблицам.

К источникам данных ODBC могут отправляться запросы, таблицы таких источников данных могут использоваться в базе данных Access как связанные. Связанные таблицы отображаются в области навигации базы данных, оставаясь в исходном файле вне файла Access. Их можно использовать при создании запросов, форм, отчетов, причем можно объединять их данные с данными из локальных объектов Access. Связывание позволяет использовать данные другой программы, не импортируя их. В этом случае можно просматривать и изменять данные, как в исходной программе, так и в базе данных Access.

Данные внешних источников могут импортироваться в базу данных Access.

На вкладке Внешние данные в группе Импорт и связи представлены доступные для этой операции источники данных. Импорт и связывание могут быть выполнены для данных из таких источников, как Access, Excel, база данных ODBC, текстовый файл, XML-файл, список SharePoint, документ HTML, папка Outlook, файл dBase.

Возможен экспорт таблиц, запросов, форм и отчетов, выделенной части объекта в режиме таблицы из базы данных Access в форматы других приложений. Операции экспорта представлены в одноименной группе на вкладке Внешние данные.

Выполнение операций импорта или связывания данных, как и экспорта данных из базы для большинства форматов, требует лишь указания, где расположены данные, и выбора способа их хранения в базе данных.

8.06 Поддержка технологий корпоративных сетей

Корпоративные сети являются сетями уровня предприятия, которые базируются на клиент-серверных и интернет-технологиях. Эти сети могут подключаться или не подключаться к Интернету (рис. 19). Технологии Интернета позволяют получить доступ к информации всего предприятия со своего рабочего места, не заботясь о совместимости аппаратных и программных платформ, используя обычную программу просмотра – обозреватель Internet Explorer.

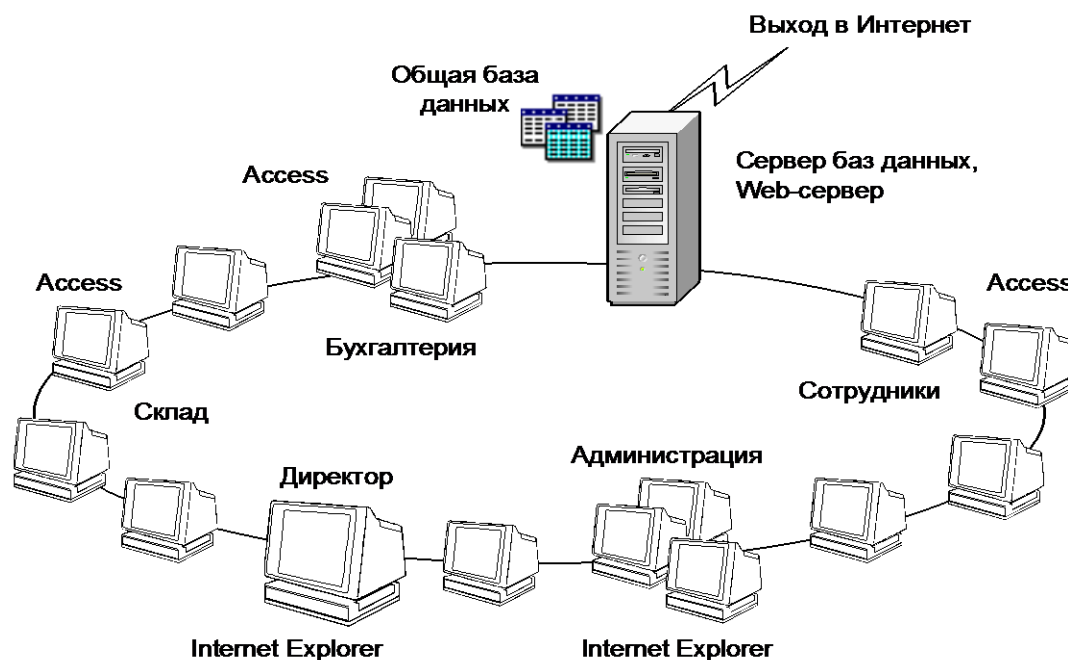


Рис. 19. Корпоративная сеть с SQL- и Web-серверами

Access включает развитые средства, ориентированные на клиентсерверные технологии, которые позволяют создавать клиентские приложения для работы с общими базами данных SQL Server.

Для организации совместной работы с данными и взаимодействия пользователей с помощью единой Web-среды могут быть использованы служба Windows SharePoint Services, специальные технологии взаимодействия и коммуникаций, которые легко интегрируются с приложениями Microsoft Office. Служба Windows SharePoint Services является компонентом Microsoft Windows Server, обеспечивает пользователей набором инструментов для организации информации, управления документами и эффективного взаимодействия.

С помощью Windows SharePoint Services можно создавать Webсайты для хранения общих данных и совместной работы с ними. Доступ пользователей к содержимому Web-сайтов осуществляется через Webобозреватель или приложения Microsoft Office из любой точки корпоративной сети.

Access предоставляет разнообразные простые и удобные средства размещения данных из баз на одном или нескольких сайтах SharePoint для доступна к ним пользователей корпоративной сети, а также средства использования списков SharePoint в приложениях Access.

1. Многопользовательская база данных Access

База данных, как правило, содержит данные, необходимые многим пользователям. Создание многопользовательской базы данных Access и

получение одновременного доступа нескольких пользователей к общей базе данных возможно в одноранговой локальной сети персональных компьютеров или в сети с файловым сервером.

Под одноранговой понимается сеть, каждый компьютер которой может предоставлять остальным подключенным к сети компьютерам доступ ко всем или некоторым своим ресурсам, т.е. являться сервером и рабочей станцией одновременно. Одноранговая сеть может управляться встроенной сетевой операционной системой Windows 9x или Windows NT/2000/XP/Vista/Windows 7.

Сети больших масштабов используют выделенные файловые серверы. В сети, поддерживающей концепцию файлового сервера, база данных Access размещается на компьютере, выделенном в качестве файлового сервера. СУБД Access может быть установлена или на файловом сервере или на каждой рабочей станции, но выполняется она всегда на рабочей станции пользователя. Обработка данных базы в обоих случаях также осуществляется на рабочих станциях пользователей. Поэтому по сети передаются с сервера на рабочие станции большие объемы данных, что сильно загружает ее и делает невозможным одновременное обслуживание большого числа пользователей.

Работа локальной сети с файловым сервером обеспечивается рядом сетевых операционных систем. Наиболее популярными являются локальная сеть Microsoft Windows Server и локальная сеть Novell NetWare. ОС Windows имеют версию Windows NT Server, предназначенную для управления файловым и другими серверами сети, и версию «Workstation» 2000/XP/Vista/Windows 7, которая устанавливается на рабочей станции и под управлением которой могут выполняться различные приложения, в том числе и Access.

Сеть обеспечивает аппаратную и программную поддержку обмена данными между компьютерами. Сетевая операционная система обеспечивает защиту базы данных, размещенной в сети, предоставляя пользователям доступ к определенным сетевым ресурсам, например папкам, файлам, только по предъявлению имени и пароля.

Access не следит за разграничением доступа разных пользователей к базе данных, но автоматически обеспечивает защиту данных от одновременной их корректировки несколькими пользователями. Для обеспечения защиты данных от одновременной их корректировки несколькими пользователями в Access предусматривается блокировка на уровне страниц, при которой блокируются все записи одной страницы. Дополнительно к блокировке страничного уровня Access поддерживает блокировку на уровне записи. Блокирование не допускает изменения записи другими пользователями, пока текущее изменение не будет закончено.

Выбор режима блокировки на уровне записей по умолчанию в текущей базе данных производится параметром Открытие баз данных с использованием блокировки на уровне записей (Open databases using record level locking) в окне

Параметры Access, «Блокировка на уровне записи» (Access Options) на вкладке Параметры клиента в группе Дополнительно (Advanced). Если флажок этого параметра сброшен, по умолчанию будет использована блокировка на уровне страницы. Выбранный вариант применяется к данным в формах, таблицах и программах, использующих объект Recordset для перебора записей. Этот параметр не применяется к запросам на изменение или программам, выполняющим массовые операции с использованием инструкций SQL.

2. Работа Access с базой данных SQL Server

Работа с общей базой данных в сети с файловым сервером становится неэффективной уже при одновременной работе 15 пользователей. На обеспечение эффективной работы большого числа пользователей с общей базой данных ориентирована технология «клиент – сервер». В этой технологии пользователь-клиент передает со своего компьютера запрос на компьютер сервера, там СУБД обрабатывает запрос и обратно посылает только результат выполнения запроса. Таким образом, значительно снижается объем передаваемых по сети данных.

Приложение пользователя разрабатывается и выполняется под управлением СУБД Access на компьютере клиента. Общая база данных размещается на мощном компьютере, где функционирует сервер баз данных, управляемый СУБД SQL Server (см. рис. 19). Эта СУБД выполняет обработку данных, размещенных на сервере, и отвечает за их целостность и сохранность. Для доступа к данным базы на сервере используется язык структурированных запросов SQL.

Широко известны серверы баз данных – SQL Server фирмы Microsoft и Oracle Server фирмы Oracle. SQL-серверы баз данных являются самым мощными приложениями для сетевой обработки данных.

Подключение из Access к серверам баз данных SQL может быть осуществлено с помощью драйверов ODBC. Каждому серверу баз данных соответствует свой драйвер ODBC. В комплект поставки MS Access включены драйверы ODBC для MS SQL Server и Oracle SQL Server.

Использование унифицированного языка запросов SQL позволяет работать с одной и той же базой данных сервера разным пользователям из различных приложений. Данные из базы могут получать Access, Excel, FoxPro и многие другие приложения, использующие протокол ODBC, посылая запросы на языке SQL серверу баз данных.

Приложение Access взаимодействует с данными, расположенными на сервере, несколькими способами. Можно посылать на сервер запросы на языке SQL-сервера. С помощью запроса можно получать необходимые данные в виде виртуальной таблицы – таблицы запроса и далее использовать эту таблицу в

качестве источника данных в форме или отчета. Можно посылать на сервер обновленные данные, а также создавать или изменять таблицы в базе данных сервера. Кроме того, в локальной базе данных могут быть созданы связанные таблицы, отображающие данные из таблиц SQL-сервера. Эти таблицы могут обрабатываться в приложении наряду с локальными таблицами базы.

В Access реализована возможность создания приложения-проекта, в котором хранятся только объекты, составляющие приложение пользователя, а база данных, с которой работает приложение, размещается на SQL-сервере. Причем в проекте Access обеспечивается работа с объектами базы данных (таблицами, представлениями, хранимыми процедурами, схемами данных) в диалоговом режиме через интерфейс, аналогичный интерфейсу SQL-сервера. Проект Access, являющийся клиентом SQL-сервера, позволяет напрямую, не используя ядро баз данных Access Database Engine, подключаться к базам данных Microsoft SQL Server с помощью интерфейсов OLE DB. Из проекта Access посредством OLE DB обеспечивается доступ не только к базам данных SQL Server, но и к базам данных Access, файлам электронной почты и источникам данных многих других типов.

Создается проект на вкладке Файл в окне представления Backstage, где при выборе местоположения файла новой базы данных в окне Файл новой базы данных задается имя файла и в поле Тип файла из списка выбирается Microsoft Access Проекты (*.adp).

MS SQL Server, начиная с версии 2005, представляет бесплатную версию SQL Server Express, которая позволяют создать проект, предназначенный для работы с базой данных SQL Server, работая на локальном компьютере, не подключенном к сети. При этом нет необходимости приобретать лицензию и устанавливать сетевой вариант сервера баз данных. Работа этой версии основывается на том же ядре базы данных, что SQL Server. Созданное с использованием этих версий приложение будет полностью пригодно для работы с базой данных на большом сетевом SQL-сервере. После завершения разработки, переместив базу с локального компьютера на сервер и модифицировав информацию о соединении, проект можно подключить к удаленному SQL-серверу.

Такой подход может быть удобен при разработке проекта для небольшого числа пользователей, когда в дальнейшем предполагается эксплуатация базы данных на SQL-сервере. Кроме того, использование локального варианта SQL-сервера позволяет изучить работу с базами данных сервера при наличии лишь одного компьютера.

В Access 2010 возможно преобразование базы данных в формат SQL Server. Для этого предназначена команда SQL Server в группе Перемещение данных на вкладке Работа с базами данных. Мастер может создать новую базу

данных SQL Server или воспользоваться существующей. При этом указывается имя SQL Server, имя базы данных и способ соединения.

3. Интернет-технологии

Пользователи баз данных все больше ориентируются на уникальные возможности быстрого сбора и совместного использования информации, предоставляемые интернет-технологиями. Базы данных широко используются как в интернет-публикациях, так и в электронной коммерции. Новые технологии в Access 2010 позволяют создавать специальные веб-приложения – веб-базы данных и публиковать их на сайтах Microsoft SharePoint Server, на котором выполняются службы Access. Базу данных можно опубликовать как на собственном сервере SharePoint в интранете, так и в Интернете. Пользователи могут использовать базу данных с помощью стандартного браузера, не устанавливая приложение Access на компьютере. Это делает простым совместное использование корпоративной информации в среде настольных систем. При этом пользователи могут с помощью браузера открывать веб-формы и отчеты. В то же время сохраняется возможность работы в Access с веб-базами данных автономно, можно вносить изменения в макеты и данные, а затем, восстановив подключение, синхронизировать их с сервером Microsoft SharePoint Server 2010.

Создается веб-база данных, как и база данных для настольных компьютеров, в окне представления Backstage после открытия Access или в любой момент после закрытия активной базы данных выбором Пустая веб-база данных. Для создания таблицы в веб-базе данных используется режим таблицы, режим конструктора отсутствует. Использовать связанные таблицы в веб-базе данных нельзя, схема данных недоступна. Для создания связи в веб-базе данных используется поле подстановки. Поле подстановки создается с помощью мастера в таблице на стороне связи многие и указывает на таблицу на стороне один. Обеспечение целостности данных в веб-базе данных можно реализовать с помощью макросов данных.

При публикации веб-базы данных службы Access создают сайт SharePoint, содержащий базу данных. Публикация веб-базы данных выполняется на вкладке Файл выбором раздела Сохранить и опубликовать и пункта Опубликовать в Access Services. Здесь заполняется поле URL-адрес сервера и поле Имя сайта, в которое вводится имя веб-базы данных. Это имя будет добавлено к URL-адресу сервера для получения URL-адреса приложения.

Все таблицы становятся списками SharePoint, а записи – элементами списка. Это позволяет использовать разрешения SharePoint для управления доступом к веб-базе данных, а также другие возможности SharePoint.

После публикации базы данных посетители сайта SharePoint, обладающие необходимыми разрешениями, могут с ней работать. Одни пользователи, имея полный доступ, могут вносить изменения в данные и структуру базы данных, другие только изменять данные, и, наконец, ряду пользователей разрешается только читать данные.

В браузере выполняются формы, отчеты и большинство макросов. Формы являются основным средством ввода и редактирования, а также просмотра данных в веб-базе данных. Отчеты предназначены для просмотра и печати данных из веб-базы данных. При открытии формы/отчета браузер получает необходимые данные с сервера SharePoint. Данные форм и отчетов можно фильтровать и сортировать без их повторного получения с сервера. Вся обработка SQL-кода выполняется на сервере, что повышает производительность сети за счет ограничения обмена данными с результирующими наборами.

В веб-браузере недоступна область навигации. Чтобы упростить работу с объектами базы данных, необходимо предоставить им средства навигации. Для этого можно создать форму навигации и указать, что она должна отображаться при открытии приложения в веб-браузере.

В веб-базе данных можно создать множество клиентских объектов, но их нельзя использовать в браузере. Тем не менее эти объекты являются частью веб-базы данных, и пользователи, открыв веб-базу данных в приложении Access на настольном компьютере, могут использовать клиентские объекты.

Некоторые функции, которые можно использовать в базах данных для настольных компьютеров, отсутствуют в службах Access. Однако ряд новых функций поддерживает многие из тех сценариев, которые доступны для настольных компьютеров.

SharePoint устраняет все проблемы, связанные с параллельным доступом к данным. Работа с приложением базы данных через браузер выполняется на основе разрешений SharePoint, определяющих возможность доступа к тем или иным объектам. Кроме того, средствами SharePoint можно воспользоваться для выполнения таких функций, как запись версий данных, подписка на получение предупреждений при внесении изменений. Таким образом обеспечивается эффективный общий доступ к базе данных, а также предоставляются новые возможности для совместной работы через Интернет.

8.07 Защита баз данных

Защита локальных баз данных в современных СУБД, как правило, строится на доверии к происхождению ее выполняемого содержимого. Дело в том, что некоторые компоненты баз данных, полученные из ненадежных источников,

выполняясь, могут повредить базу данных. Поэтому база данных открывается для работы с полным набором функциональных возможностей, только если она имеет состояние доверенной.

База данных представляет собой набор объектов – таблиц, форм, запросов, отчетов, макросов и т.д., которые, как правило, являются взаимозависимыми и, следовательно, выполнение некоторых объектов может повлечь порчу базы данных.

К небезопасным компонентам относятся:

- запросы на изменение, которые добавляют, удаляют или изменяют данные в таблицах;
- макрокоманды, изменяющие базу данных или получающие доступ к ресурсам вне базы данных;
- управляющие запросы (DDL-запросы), предназначенные для создания или изменения объектов базы данных, таких как таблицы и процедуры;
- SQL-запросы к серверу, которые передаются на сервер баз данных и работают с таблицами непосредственно на нем, минуя ядро базы данных Access;
- элементы управления ActiveX;
- выражения (функции, возвращающие одно значение); П код VBA.

База данных приобретет состояние доверенной, если поместить ее в надежное расположение, либо применить цифровую подпись, воспользовавшись сертификатом от надежного издателя. Издатель – это разработчик макроса, элемента управления ActiveX, надстройки или других расширений приложений, используемых потребителями. Подпись подтверждает, что выполняемые компоненты базы данных созданы издателем, и никто другой не изменял их после подписи.

Защита баз данных в Access 2010 реализуется Центром управления безопасностью. Центр управления безопасностью отображает диалоговое окно, в котором можно задавать и менять параметры безопасности и конфиденциальности, в том числе определять надежное расположение, надежных издателей. Чтобы открыть это окно, воспользуйтесь кнопкой Параметры (Options) в представлении Backstage на вкладке Файл (F), затем в окне Параметры Access откройте группу Центр управления безопасностью (Trust Center) и нажмите кнопку Параметры центра управления безопасностью (Trust Center Settings). Параметры, установленные в Центре управления безопасностью, определяют поведение новых и существующих баз данных при их открытии в Access. Для установки параметров безопасности открывать базу данных не требуется.

Для создания надежного расположения и добавления туда папки с базами данных выберите «Защита баз данных» Надежные расположения (Trusted

Locations), нажмите кнопку **Добавить новое расположение** (Add new Location), а затем в диалоговом окне **Надежное расположение Microsoft Office** (Microsoft Office Trusted Location) укажите путь к папке надежного расположения на жестком диске или в сети (рис. 20). Для перемещения или копирования файла базы данных в надежное расположение используются любые привычные способы. Например, средства проводника Windows. После размещения базы данных в надежном расположении при ее открытии не выполняется проверка средствами Центра управления безопасностью, не выводится предупреждение системы безопасности, а все выполнимое содержимое будет включено.

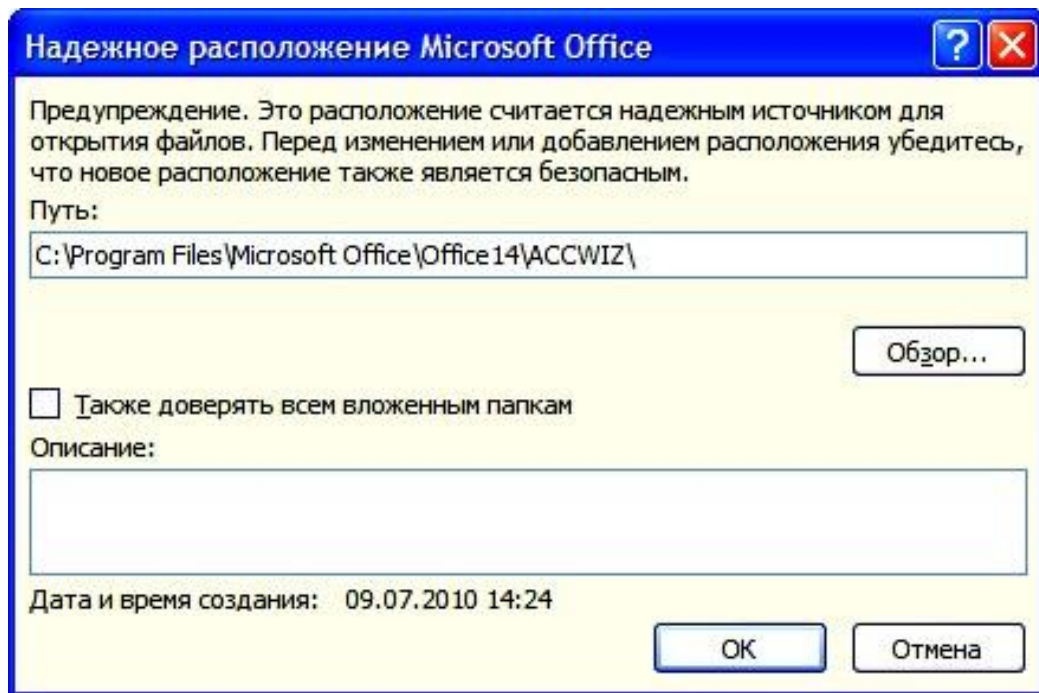


Рис. 20. Определение папки в качестве надежного источника для открытия файлов баз данных

Назначение в качестве надежного расположения всей папки Мои документы повышает риск атаки хакеров и увеличивает угрозу безопасности.

Любые внешние элементы, такие как сетевые папки, являются менее безопасными. Не рекомендуется назначать в качестве надежного расположения папку общего пользования сетевого ресурса. При создании надежного расположения, не являющегося локальным ресурсом компьютера, в диалоговом окне **Надежное расположение** (Trusted Locations) нужно установить **Разрешить надежные расположения в моей сети** (не рекомендуется) (Allow Trusted Locations on my network (not recommended)), напоминающее о ненадежности расположения.

Для применения цифровой подписи (шифрованной электронной подписи) к компонентам базы данных, прежде всего, необходимо иметь цифровой

сертификат. Если базы данных создаются для коммерческого распространения, нужно получить сертификат в коммерческом центре сертификации, например, VeriSign, Inc. или GTE. Центр сертификации наводит справки об издателе, чтобы удостовериться в его надежности.

Если базу данных планируется использовать в личных целях или в небольшой рабочей группе, можно воспользоваться предусмотренным в операционных системах Microsoft Windows средством создания сертификатов с собственной подписью, называемым SelfCert.exe. Этот сертификат следует добавить в список надежных источников, а затем подписать базу данных.

В Windows XP, Vista или Windows 7 для создания сертификата нажмите кнопку Пуск и выберите последовательно пункты Все программы, Microsoft Office, Средства Microsoft Office 2010 и Средство создания цифровых сертификатов для проектов VBA. Откроется диалоговое окно Создание цифрового сертификата. В поле Имя сертификата введите описательное имя сертификата. Нажмите ОК. При появлении сообщения "SelfCert: успех", нажмите ОК.

Для просмотра сертификата в хранилище личных сертификатов откройте браузер Internet Explorer. Выберите в меню Сервис команду Свойства обозревателя и откройте вкладку Содержание. Нажмите кнопку Сертификаты и откройте вкладку Личные.

При открытии accdb- или accde-файла Access 2010 сообщает расположение базы данных центру управления безопасностью. Если это расположение надежное, она работает с полным набором функциональных возможностей. При открытии базы данных из более ранней версии Access 2010 передает в Центр управления безопасностью расположение и цифровую подпись, если она имеется в базе данных.

Центр управления безопасностью проверяет подлинность этих сведений, чтобы определить, имеет ли база данных состояние доверенной, а затем информирует приложение Access о том, как следует ее открывать. Access либо отключает опасные компоненты базы данных, либо открывает с полным набором функциональных возможностей.

Если база данных не имеет состояния доверенной, программные средства центра управления безопасностью по умолчанию, не оценивая компоненты базы данных, отключает все выполняемое содержимое в базе данных и запрашивает у пользователя разрешение на его включение, отображая панель сообщений «Защита баз данных» (рис. 21).

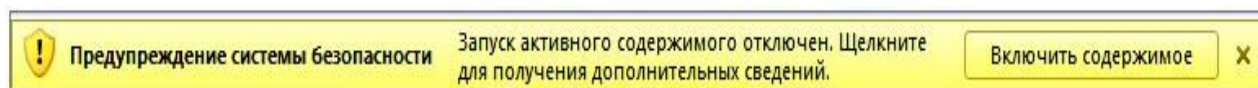


Рис. 21. Предупреждающее сообщение системы безопасности при открытии базы данных

Если пользователь доверяет источнику или берет на себя ответственность за безопасность базы данных, он разрешает использование опасного содержимого, щелкнув на кнопке Включить содержимое. При этом источник автоматически попадает в разряд надежных и при следующем открытии панель сообщений не будет выводиться. Для получения дополнительных сведений щелкните на соответствующей подписи на панели сообщений. В открывшемся представлении Backstage в разделе Сведений о базе данных будет отображено предупреждение системы безопасности Включить содержимое, дополненное списком (рис. 22). Первое значение списка переводит базу данных в разряд надежных. Второе позволяет установить защиту от неизвестного содержимого или разрешить его выполнение на время текущего сеанса работы. При повторном открытии такой базы данных будет выводиться запрос по поводу отключения опасного содержимого.



Рис. 22. Предупреждение системы безопасности при открытии файла базы данных

При выборе в диалоговом окне Центра управления безопасностью вкладки Надежное расположение (Trusted Locations) и варианта Отключить все надежные расположения (Disable all Trusted Locations) Access отключит все надежные расположения.

В Access 2007/10 панель сообщений является единым средством вывода предупреждений, которое появляется по умолчанию уже при открытии базы данных вне доверенного расположения. Это позволяет пользователям иметь дело со значительно меньшим числом предупреждающих сообщений в сравнении с предыдущими версиями Access.

Для включения панели сообщений откройте вкладку Файл. В левой части представления Backstage выберите Параметры.

На левой панели диалогового окна Параметры Access выберите Центр управления безопасностью. В окне Центр управления безопасностью, щелкните элемент Параметры центра управления безопасностью.

В следующем окне Центра управления безопасностью щелкните элемент Панель сообщений (Message Bar). Установите флажок Показывать панель сообщений во всех приложениях, если активное содержимое такое как элементы ActivX и макросы заблокированы и нажмите кнопку ОК.

После включения и отображения панели сообщений на ней можно включить содержимое.

Другим средством защиты является шифрование баз данных. Это средство объединяет два улучшенных средства прежних версий – кодирование и пароли баз данных. Чтобы зашифровать базу данных, достаточно установить пароль для шифрования базы данных. Стойкий алгоритм шифрования баз данных в формате Access 2010 исключает несанкционированный их просмотр. Все данные становятся нечитаемыми в других программных средствах, и для того чтобы использовать эту базу данных, пользователи должны вводить пароль.

Шифрование с использованием пароля выполняется для базы данных, открытой в монопольном режиме. Щелкните вкладку Файл, затем выберите команду Открыть (Open), в окне Открытие файла базы данных (Open) выделите файл, который нужно открыть, щелкните стрелку рядом с кнопкой Открыть (Open) и выберите Монопольно (Open Exclusive).

Для шифрования базы данных с использованием пароля на вкладке Файл нажмите кнопку Сведения и выберите Зашифровать паролем (Encrypt with Password).

В открывшемся диалоговом окне Задание пароля базы данных (Set Database Password) введите пароль и подтверждение и нажмите кнопку ОК. При открытии зашифрованной таким образом базы данных потребуется ввод пароля.

Выполнением команды Расшифровать базу данных (Decrypt Database), в которую преобразовалась команда Зашифровать паролем (Encrypt with Password) пароль базы данных удаляется. При этом также необходимо ввести пароль.

Пароли должны состоять не менее чем из 8 символов. Надежные пароли должны сочетать в себе прописные и строчные буквы, цифры и символы и состоять не менее чем из 14 символов. Важно помнить свой пароль, т.к. восстановить его невозможно.

8.08 Контрольные вопросы

1. Какие объекты составляют базу данных Access?
2. Чем определяется структура реляционной таблицы?
3. Для чего предназначен первичный ключ реляционной таблицы?
4. Какими отношениями характеризуется связь двух таблиц реляционной базы данных?
5. Для чего используется ключ связи?
6. Что предлагает технология интеллектуальной замены имен?
7. Выведется ли сообщение системы безопасности об отключении части функций при ее открытии из надежного расположения?
8. С помощью каких интерфейсов обеспечивается доступ к данным Microsoft SQL Server?
9. Какой интерфейс используется в проекте Access для доступа к данным Microsoft SQL Server?
10. С какой целью создаются в Access веб-базы данных?
11. Какой режим используется в веб-базе данных для создания таблиц?
12. С помощью каких программ пользователи получают возможность работать с веб-базой данных из Интернет?

Ответы

1. Таблицы, запросы, формы, отчеты, макросы и модули.
2. Составом полей.
3. Для однозначной идентификации каждой записи таблицы.
4. Отношениями записей типа «один-к-одному» (1 : 1) и «один-многим» (1 : M).
5. Для связи двух таблиц с помощью одинакового поля в них.
6. Отслеживать изменение имен таблиц, полей, запросов, элементов управления и др. и выполнять их автозамену в зависимых объектах.
7. Нет.
8. ODBC и OLE DB.
9. OLE DB.
10. Для публикации и совместного использования в среде SharePoint.
11. Режим таблицы.
12. Web-браузера и Access.

Модули являются программами на языке Visual Basic, которые служат для реализации нестандартных процедур обработки данных.

Все данные БД Microsoft Access и средства их отображения хранятся в одном файле с расширением MDB.

Таблицы

Таблицы составляют основу базы данных - именно в них хранятся все данные. Таблицы должны быть тщательно спланированы. Прежде всего, должна быть спланирована структура каждой таблицы. Структура таблиц определяется содержанием тех выходных форм и отчетов, которые должны быть затем получены. При планировании таблиц необходимо избежать дублирования информации в разных таблицах.

Таблица - это объект БД, который хранит данные определенной структуры. Таблица состоит из записей (строк), каждая из которых описывает одну сущность. Каждый столбец таблицы - это поле. Столбец содержит однотипную информацию.

Длина имени таблицы - не более 64 символов.

Длина имени поля - не более 64 символов.

Количество полей в одной таблице - не более 255.

Количество записей - неограниченно.

Суммарный объем информации во всей БД - не более 2 гигабайта.

Для каждого поля необходимо указать тип данных. Тип данных определяет вид и диапазон допустимых значений, которые могут быть введены в поле, а также объем памяти, выделяющийся для этого поля.

Таблица может содержать следующие типы полей (всего 8):

Текстовый Короткий текст. Текст и числа, например, имена и адреса, номера телефонов и почтовые индексы. Текстовое поле может содержать до 255 символов.

Поле Мемо Длинный текст и числа, например, комментарии и пояснения. Мемо-поле может содержать до 65 536 символов.

Числовой Общий тип для числовых данных, допускающих проведение математических расчетов, за исключением расчетов для денежных значений. Свойство Размер поля позволяет указать различные типы числовых данных. Длина - 8 байт. Точность - 15 знаков.

Дата/время Значения даты и времени. Пользователь имеет возможность выбрать один из многочисленных стандартных форматов или создать специальный формат. Длина - 8 байт.

Денежный Денежные значения. Числа представляются с двумя знаками после запятой. Не рекомендуется использовать для проведения денежных расчетов значения, принадлежащие к числовому типу данных, так как последние могут округляться при расчетах. Значения типа "Денежный" всегда

выводятся с указанным числом десятичных знаков после запятой. Длина - 8 байт.

Счетчик Автоматически вставляющиеся последовательные номера. Счетчик увеличивается на единицу для каждой следующей записи. Нумерация начинается с 1. Поле счетчика удобно для создания ключа. В таблице может быть только одно такое поле.

Длина - 4 байта.

Логический Значения "Да"/"Нет", "Истина"/"Ложь", "Вкл"/"Выкл", т.е. одно из двух возможных значений. Длина - 1 байт.

Поле объекта OLE Объекты, созданные в других программах, поддерживающих протокол OLE, например графики, рисунки и т.п. Объекты связываются или внедряются в базу данных Microsoft Access через элемент управления в форме или отчете.

Максимальный объем информации объекта OLE -1 Гбайт.

Гиперссылка. Поле, в котором сохраняются адреса гиперссылок, позволяющих переходить к файлам, фрагментам файлов или веб-страницам. Гиперссылка может иметь вид пути UNC либо адреса URL. Сохраняет до 64 000 знаков

Индексирование полей таблицы. Индексирование позволяет ускорить сортировку и поиск данных в таблице. Можно индексировать числовые, денежные, текстовые, логические поля, а также поля типа Счетчик и Дата. Не следует создавать слишком много индексов для одной таблицы, т.к. это замедлит ввод и редактирование ее данных.

Первичный ключ - это специальный тип индекса, который однозначно идентифицирует каждую запись. В первичный ключ могут входить несколько полей, но значение первичного ключа должно быть уникальным для каждой записи. Первичные ключи используются для установления связей между таблицами.

Связи между таблицами. Таблицы могут быть связаны отношениями один-к-одному, один-ко-многим и многие-кмногим. Access позволяет использовать только отношения первых двух типов.

При установлении связей нужно определить, какая таблица является главной, а какая - подчиненной.

Отношение один-к-одному означает, что одной записи подчиненной таблицы соответствует только одна запись в главной таблице. Такие отношения встречаются очень редко, т.к. требуют неоправданно много места в БД. Вместо них можно просто добавить поля подчиненной таблицы к полям главной.

Наиболее часто используются отношения один-ко-многим. В этом случае одной записи в главной таблице соответствует несколько записей в подчиненной таблице.

Для создания отношений необходимо указать поля в двух таблицах, которые содержат одни и те же данные. Обычно такое поле в одной из таблиц (главной) является ключевым. Имена связывающих полей могут отличаться, но типы и свойства должны совпадать. Возможна связь между полем типа Счетчик и полем типа Число с форматом Длинное целое.

Рекомендации для ввода данных в таблицы

Для ввода в поле текущей записи значения из того же поля предыдущей записи нажать клавиши <Ctrl> и <“>. (Двойной апостроф на русском регистре - на клавише “2”).

Для редактирования ранее введенного значения нажимайте клавишу F2.

Задание 2. Создайте базу данных «Академия» на основе инфологической модели, приведенной на рисунке. База данных должна содержать 4 взаимосвязанных таблицы: Студент, Группа, Специальность и Факультет.

Таблица Студент:

- N зачетной книжки – ключевое поле, длинное целое
- N группы – числовое поле, целое
- ФИО – текстовое поле 15 символов
- Дата рождения – поле типа «дата»
- Коммерческий – логическое поле (вкл/выкл) Таблица Группа:
- N группы – ключевое поле числового типа, целое - N специальности – числовое поле, длинное целое
- N факультета – числовое поле, байтовое
- Курс – числовое поле, байтовое Таблица Факультет:
- N факультета – ключевое поле числового типа, байтовое
- Наименование факультета – текстовое поле, 30 символов
- Декан - текстовое поле, 15 символов Таблица Специальность:
- N специальности – ключевое поле числового типа, длинное целое
- Наименование специальности – текстовое поле, 40 символов
- Стоимость обучения – денежного типа.

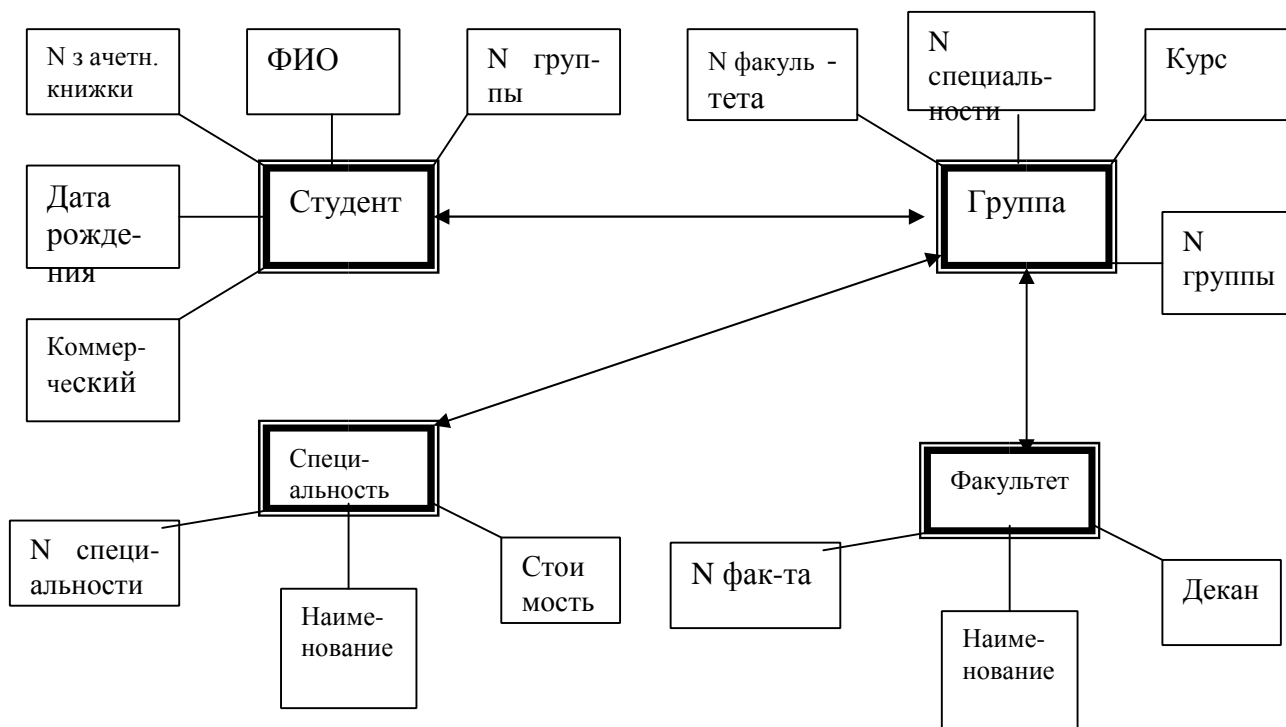


Рис.1.1 Инфологическая модель базы данных

6.9 Технология создания таблицы Студент

1. Создайте новую базу данных, щелкнув по соответствующей кнопке инструментальной панели. Появится Область задач с заголовком Создание файла. Выберите гиперссылку Новая база данных. В окне Файл новой базы данных выберите для файла папку Мои документы и присвойте файлу имя Академия2003. Щелкните по кнопке Создать. Появится окно базы данных с перечнем объектов.

2. В окне базы данных выберите объект Таблицы и выберите вариант создания с помощью мастера таблиц.

3. На 1-м шаге работы мастера выберите в качестве образца таблицу Студенты и, дважды щелкая по образцам полей, включите в создаваемую таблицу поля:

- код студента
- фамилия
- специализация
- Из таблицы Сотрудники:
- дату рождения
- код отдела

4. Переименуйте поля в соответствии с заданием:

- Фамилия - ФИО
- код студента - N зачетной книжки - специализация – коммерческий - код отдела – N группы.

5. Щелкните по кнопке Далее.
6. На шаге 2 в окне Создание таблиц дайте имя таблице Студент и сохраните включенным флажок Microsoft Access автоматически определяет ключ. Щелкните по кнопке Далее.
7. На шаге 3 нажмите на кнопку Готово и приступите к непосредственному вводу данных в таблицу.
8. Переключите таблицу в режим конструктора, щелкните по кнопке Вид инструментальной панели. Проведите корректировку типов данных в соответствии с заданием.
9. Вернитесь в режим таблицы с помощью аналогичной кнопки и приступите к вводу данных.
10. Введите данные для 3-х групп по 10 студентов в каждой с различными значениями полей.
11. После ввода данных сохраните базу данных.
12. Измените вид представления данных в столбце Коммерческий - замените флажки на значения логического типа - Да/Нет. Для этого в режиме конструктора выделите поле Коммерческий, и раскройте список на вкладке Подстановка. Выберите тип элемента управления Поле. Перейдите в режим таблицы и просмотрите ее. Верните назад тип элемента управления - Флажок.
13. Для того, чтобы сделать столбец уже, измените название столбца таблицы, соответствующего полю N зачетной книжки. Замените название столбца на сокращенное: НЗК. Для этого в режиме конструктора выделите указанное поле и введите на вкладке Общие в строку Подпись новое название. Просмотрите таблицу.
14. Предусмотрите контроль данных. Запретите ввод даты рождения меньше заданной, например, даты более ранней, чем 01.01.1970 г. При попытке ввода такой даты предусмотрите вывод сообщения: Слишком старый студент. Для этого в режиме конструктора установите окно свойств поля Дата рождения. Для свойства Условие на значение введите: >#01.01.1970#. Для свойства Сообщение об ошибке введите: Слишком старый студент. Проверьте правильность установленного контроля значений поля Дата рождения.

6.10 Освоение приемов работы с фильтрами в таблицах

MS Access позволяет применять 3 вида фильтров для работы с таблицами: Фильтр по выделенному, Фильтр для и Расширенный фильтр.

Задание 4. Найдите студентов, фамилия которых начинается на заданную букву, например на букву «В». Список найденных студентов должен быть упорядочен по алфавиту. Для поиска использовать расширенный фильтр.

Технология поиска с помощью фильтра по выделенному

1. Найдите в поле ФИО любую фамилию, начинающуюся на букву «В» и выделите мышкой эту букву.

Тема 7 Концептуальная и логическая модели БД.

Тема 8 Реляционная модель и реляционные СУБД. Физическая модель БД.

Тема 9 SQL - стандартный язык запросов к реляционным СУБД.

Тема 10 Алгоритмы, их свойства и средства описания

Типовые решения практических задач

Типовое решение практических задач с привлечением компьютера включает в себя следующие основные этапы:

1. **Корректная постановка практической задачи** (сбор информации о задаче; точная формулировка условия задачи; определение конечных целей решения; поиск возможных аналогов решения).

2. **Формализация задачи** (построение математической модели задачи; разработка ее структур данных).

3. **Разработка алгоритма решения задачи** (выбор метода проектирования алгоритма, составление алгоритма).

4. **Программирование** (выбор рабочей среды программирования, программная реализация алгоритма).

5. **Совершенствование программного решения** (отладка компьютерных программ, их проверка на тестовых примерах, уточнение при необходимости математической модели с повторением этапов 2-5).

6. **Передача разработанного решения в постоянную эксплуатацию** (документирование всех этапов решения, составление инструкций пользователю, сопровождение задачи в процессе опытной эксплуатации заказчиком).

Важным этапом освоения технологии решения задач на компьютере является развитие навыков разработки алгоритмов, их правильного представления в соответствии с общепринятыми стандартами, знание базовых алгоритмических структур.

Для повышения эффективности применения компьютера как инструмента решения задач необходимо освоение основной фундаментальной концепции подхода к использованию цифровых вычислительных средств. В информатике таким фундаментом является алгоритмизация возникающих задач. Алгоритмический стиль мышления позволяет связать воедино функционирование информации в конкретной прикладной среде с требованиями обработки информации на компьютере.

Алгоритмическое мышление помогает сформировать следующие основные навыки решения задач:

- умение правильно планировать структуру предстоящих действий для достижения заданной цели при помощи стандартного набора средств;
- строить информационные структуры для описания объектов и процессов в конкретной предметной области;
- правильно организовывать поиск информации, необходимой для решения задачи;
- четко и однозначно формулировать способ решения задачи в общепринятой форме и правильно понимать способ решения, предложенный другим разработчиком;
- формировать навыки анализа имеющейся информации, умения представлять ее в структурированном виде.

Развитие системного, логического мышления, освоение навыков оперирования формальными понятиями и объектами, характерными для используемых информационных технологий – это тот необходимый уровень, который формируется на базе правильного понимания алгоритмов и способов алгоритмизации.

10.1 Понятие алгоритма

Происхождение термина «алгоритм». Слово произошло от имени среднеазиатского ученого Аль-Хорезми (8–9 вв., Хорезм – историческая область на территории современного Узбекистана). В 1857 в библиотеке Кембриджского университета был найден перевод на латинский язык математической работы Аль-Хорезми, в котором имя Аль-Хорезми упоминается как Алгоритми, откуда и появилось слово «алгоритм». Оно стало особенно употребительным с появлением компьютеров для обозначения совокупности действий, составляющих какой-либо вычислительный процесс.

Термин «алгоритм» в бытовом понимании. В повседневной жизни выполнение каждой, даже простой задачи обычно осуществляется в несколько последовательных этапов (шагов). Например, такую цепочку шагов описывает инструкция для получения наличных денег со счета по банковской карте. Подобную инструкцию – четкую последовательность шагов в решении какой-либо жизненной задачи – принято называть алгоритмом. Каждое отдельное действие – это шаг алгоритма.

Формализация понятия алгоритма, теория алгоритмов. Приведенное выше общее описание понятия алгоритма не всегда позволяет сравнить какие-либо две таким образом определенные инструкции. Можно, например, сравнивать

два алгоритма решения системы уравнений и выбрать из них более подходящий, но невозможно сравнить алгоритм перехода через улицу с алгоритмом извлечения квадратного корня. С этой целью нужно формализовать понятие алгоритма, т.е. отвлечься от существа решаемой алгоритмом задачи и выделить свойства алгоритмов, привлекая к рассмотрению только форму их записи.

Задача нахождения единообразной формы записи алгоритмов, решающих различные задачи, является одной из важнейших в теории алгоритмов. В этой теории предполагается, что каждый шаг алгоритма должен быть таков, что его может выполнить достаточно простое устройство (например, машина). Так, для уточнения понятия «алгоритм» и получения возможности математического исследования алгоритмов в 30-х гг. XX века были предложены абстрактные вычислительные машины – машина Поста и машина Тьюринга. Эти механизмы, обладающие свойствами универсальности и простоты своей логической структуры, позволили решить проблему существования алгоритма решения любой практической задачи. В частности, было доказано, что если для решения задачи можно построить машину Поста-Тьюринга, то такая задача алгоритмически разрешима. Также была доказана возможность существования математических задач, для решения которых вообще не может существовать никакой алгоритм.

Вычислительный алгоритм. Это упорядоченный набор основных математических и логических действий, однозначно определяющий процесс перехода от допустимых исходных данных задачи к конечному результату ее решения и обладающий свойствами массовости, конечности, определенности, детерминированности. Основные особенности вычислительных алгоритмов состоят в следующем.

Массовость – это возможность применять многократно один и тот же алгоритм к различным вариантам его исходных данных. Алгоритм служит, как правило, для решения не одной конкретной задачи, а некоторого класса задач. Так алгоритм сложения столбиком применим к любому конечному набору натуральных чисел. Для каждого алгоритма есть некоторое множество объектов, допустимых в качестве исходных данных. Например, в алгоритме деления вещественных чисел делимое может быть любым, а делитель – тоже любым, но за исключением нуля.

Конечность – это обязательное наличие искомого результата после завершения алгоритма либо четкая фиксация причины, по которой результат не мог быть получен. Следует отметить, что на практике встречаются примеры формально бесконечных алгоритмов, например алгоритм обработки данных системы абонентских пунктов банка. Этот алгоритм состоит в непрерывном повторении цепочки действий («идентифицировать клиента», «зафиксировать запрос клиента на транзакцию», «проверить допустимость транзакции»,

«выполнить транзакцию» и т.д.), выполняемых с определенной частотой (через каждую секунду, минуту, час) во все время функционирования данной банковской системы.

Определенность – это наличие на каждом шаге алгоритма у исполнителя достаточной информации для того, чтобы его можно было выполнить. Исполнителю также нужно четко знать, каким именно образом этот шаг выполняется. Сами шаги инструкции должны быть достаточно простыми, а исполнитель должен однозначно понимать смысл каждого из действий, составляющих алгоритм (например, при вычислении корней квадратного уравнения он должен понимать смысл вычисления дискриминанта).

Детерминированность – это отсутствие элементов случайности при выполнении алгоритма. При многократном применении алгоритма к одним и тем же исходным данным должен получаться всегда один и тот же результат. Поэтому, например, процесс преобразования информации, в котором участвует бросание монеты, не может быть назван алгоритмом.

10.2 Формы представления алгоритмов

Для записи алгоритма могут использоваться различные формы его представления.

Вербальная форма представления алгоритма предполагает запись алгоритма на русском языке (или любом другом естественном языке) в виде последовательности пронумерованных инструкций. Как правило, эта форма записи алгоритма громоздка, неудобна и недостаточно наглядна.

Например, в ней описание алгоритма нахождения НОД (наибольшего общего делителя) двух целых положительных чисел m и n может быть представлено в виде последовательности следующих четырех шагов:

Шаг 1: Сравнить m и n .

Шаг 2: Если m равно n , то m и есть исходный НОД, расчет окончен. Иначе перейти к шагу 3.

Шаг 3: Если m больше n , то уменьшить значение m на величину n и вернуться к шагу 1. Иначе перейти к шагу 4.

Шаг 4: Уменьшить значение n на величину m и вернуться к шагу 1.

7.04 Запись алгоритма

Эта современная запись алгоритма нахождения НОД – весьма упрощенная. Запись, данная первоначально Евклидом, заполняет целую страницу текста, причем последовательность элементарных действий там значительно сложнее.

Представление алгоритма в форме блок-схемы реализуется в виде набора геометрических элементов (блоков), соединенных стрелками. Каждый блок – это «шаг» алгоритма, его отдельное действие. Направление стрелок между блоками задает последовательность действий. В табл. 4 представлены основные стандартные элементы блоксхем.

Табл. 4

Название элемента	Обозначение	Пояснение
Пуск-останов		Начало или конец вычислений. Внутри фигуры пишут: «начало» или «конец» соответственно
Ввод-вывод		Операция ввода-вывода данных
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условия, указанного внутри фигуры
Модификатор		Заголовок цикла с параметром
Предопределенный процесс		Вычисления по подпрограмме
Документ		Вывод данных на печать
Линия перехода		Соединяет между собой блоки, указывая очередность их выполнения

Элементы блок-схемы различаются по своему внешнему виду и назначению. Так, элементы, содержащие инструкции по каким-либо преобразованиям величин, обозначаются прямоугольниками, а элементы, содержащие проверку условий, – ромбами. Операции ввода-вывода данных обозначаются параллелограммами. Начало и конец алгоритма обозначаются прямоугольниками с двумя скругленными противоположными сторонами, внутри которых пишут: «начало» или «конец» соответственно.

Из прямоугольника выходит единственная стрелка, входить в него может несколько стрелок. Из обеих острых вершин ромба выходят стрелки: одна из них помечается словом «да», другая – словом «нет», они задают дальнейшее направление вычислений в случае, соответственно, выполнения или невыполнения записанного внутри ромба условия.

Назначение и правила использования всех типов блоков приведены в табл. 4 в графах Название элемента и Пояснение. С использованием приведенных в табл. 4 стандартных элементов вышеуказанный алгоритм нахождения НОД двух положительных целых чисел m и n примет вид блок-схемы на рис. 6. Внутри блоков в ней знаком $:=$ обозначена операция присваивания переменной величине, указанной слева от знака, ее значения, вычисляемого по формуле, стоящей справа от знака.

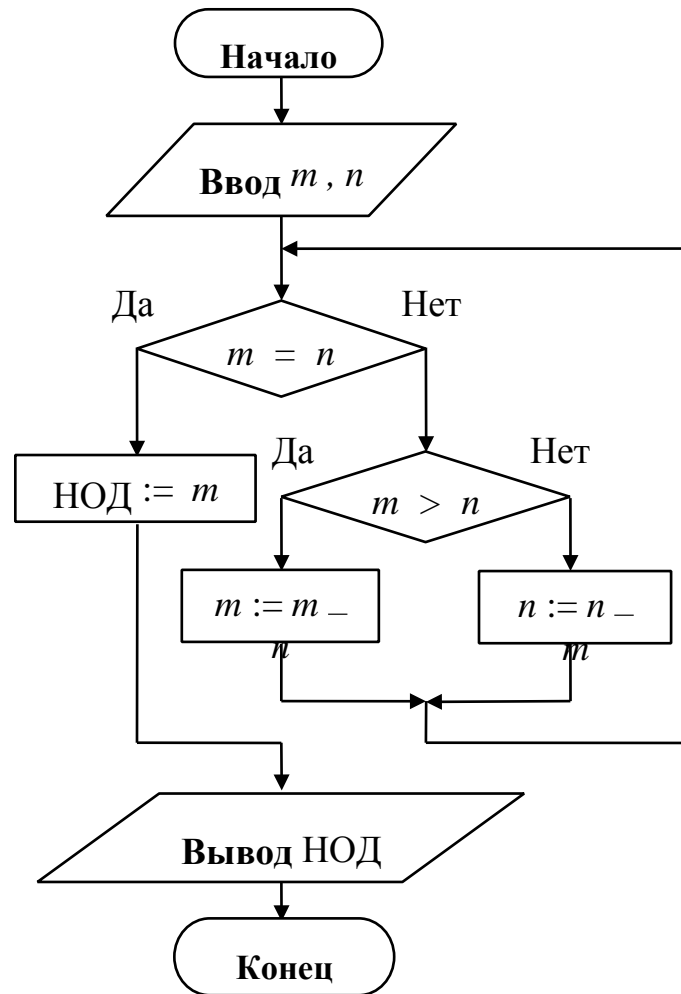


Рис. 6. Блок-схема алгоритма нахождения НОД

Представление алгоритма в форме псевдокода основано на описании этапов решения задачи с помощью ограниченного набора стандартных синтаксических конструкций. В псевдокоде, в частности, могут использоваться отдельные инструкции формальных алгоритмических языков программирования. Например, возведение X в степень A обозначается как $X^{**}A$, извлечение квадратного корня из X обозначается как **Sqrt (X)**.

Так же как в формальных языках программирования, в псевдокоде присутствуют ключевые слова, смысл которых заранее оговорен и фиксирован. В табл. 5 приведен базовый перечень ключевых слов.

В качестве ключевых могут использоваться также соответствующие английские слова-аналоги: **else** вместо **иначе**, **then** вместо **то**.

В самом общем виде запись алгоритма в форме псевдокода выглядит следующим образом:

алг название алгоритма (аргументы – входные параметры и параметры – результаты работы алгоритма) **дано** | условия применимости алгоритма **надо** | цель выполнения алгоритма
нач описание промежуточных (внутренних) величин алгоритма
команда 1;
команда 2;
...
команда N

кон

Здесь часть записи от слова **алг** до слова **нач** называется заголовком алгоритма, а часть, заключенная между словами **нач** и **кон**, – его телом. Внутри тела также могут встречаться слова **нач** и **кон**, группа команд между ними образует свой отдельный блок команд. Команды отделяются друг от друга символом «;».

Сразу после названия алгоритма в круглых скобках указываются характеристики (**арг** или **рез**) и тип значения (**цел**, **вещ**, **сим**, **лог** или **таб**) всех входных (**арг**) и выходных (**рез**) величин. При описании массивов служебное слово **таб** дополняется граничными парами значений по каждому индексу элементов массива.

В конце любой строки в текст псевдокода после знака «|» можно вставлять комментарий, облегчающий понимание алгоритма.

Разделы «дано» и «надо» в записи алгоритма не являются обязательными.

К числу основных действий, составляющих тело алгоритма, относятся команды ввода-вывода, присваивания, перехода, ветвления и циклов. Для обозначения этих команд используются соответствующие ключевые слова.

Команда ввода: ключевое слово **ВВОД**, за которым указываются имена переменных, для которых вводятся значения. Например, команда **ВВОД a,b,c** означает ввод значений соответственно для переменных a,b,c.

Команда вывода: ключевое слово **ВЫВОД**, за которым следуют имена выводимых переменных, выводимые выражения и тексты (тексты помещаются в кавычки). Например, команда **ВЫВОД "S = "**, S означает вывод имени переменной S, за которым после знака равенства = следует вывод текущего значения этой переменной.

Команды присваивания используются для вычисления выражений и присваивания их значений переменным. Общий вид команды: "переменная" := "выражение", где знак " := " означает команду замены прежнего значения переменной, стоящей в левой части, на вычисленное значение выражения, стоящего в правой части.

Табл. 5

Ключевое слово	Пояснение
Алг	Начало описания алгоритма
Арг	Входные данные (аргументы) алгоритма
Вещ	Вещественный тип данных
Все	Конец команды ветвления вычислений
Да	Логическая константа «TRUE»
Дано	Описание условий применимости алгоритма
До	Конечное значение параметра цикла
И	Логическая связка «И»
Или	Логическая связка «ИЛИ»
Иначе	Часть конструкции если ... то ... иначе ...
Кон	Конец всего алгоритма или блока команд
Кц	Конец цикла
Лит	Символьный (литерный) тип данных
Лог	Логический (булевский) тип данных
Надо	Описание цели алгоритма
Нач	Начало тела алгоритма или блока команд
нет	Логическая константа «FALSE»
Нц	Начало цикла
От	Начальное значение параметра цикла
Рез	Выходные данные (результаты) алгоритма
Таб	Табличный тип данных (массив данных)
То	Часть конструкции если ... то ... иначе

Цел	Целый тип данных
Шаг	Шаг изменения параметра цикла

Например, команда $x := y + z$ означает присваивание переменной x значения суммы величин y и z , а команда $k := k+1$ означает увеличение текущего значения переменной k на единицу.

Команда перехода: ключевые слова **идти к**, после которых указывается номер той строки, команда которой должна выполняться следующей. Таким образом, команда перехода изменяет естественную последовательность выполнения команд по возрастанию номеров строк. Например, записанная в 5-й строке команда **идти к 10** означает, что далее должна выполняться команда, записанная в 10-й строке, а не команда, записанная в следующей 6-й строке.

Для организации ветвлений вычислительного процесса применяются команды, начинающиеся с ключевых слов **если** и **выбор**, циклические вычисления организуются с помощью команд, начинающихся с ключевых слов **пока** и **для**. Подробно эти команды рассмотрены далее при описании базовых структур алгоритмов.

С использованием псевдокода вышеуказанный алгоритм нахождения НОД двух положительных целых чисел m и n примет следующий вид:

1. **алг** Наибольший Общий Делитель (**арг цел** m, n , **рез цел** nod)
2. **дано** | положительные целые числа m, n
3. **надо** | nod – наибольший общий делитель чисел m, n
4. **нач**
5. **ввод** m, n ;
6. **если** $m = n$ **то идти к 9 все**;
7. **если** $m > n$ **то** $m := m - n$ **иначе** $n := n - m$ **все**;
8. **идти к 6**;
9. $nod := m$;
10. **вывод** “Значение НОД равно”, nod
11. **кон**

Эта форма представления облегчает запись алгоритма на стадии его проектирования и дает возможность в дальнейшем легко перевести алгоритм в более широкий набор команд формального языка программирования.

10.3 Базовые структуры алгоритмов

Алгоритм решения любой вычислительной задачи можно описать, используя комбинации из следующих трех стандартных базовых конструкций алгоритмов:

- последовательного алгоритма – алгоритма линейной структуры; □
- ветвящегося алгоритма – алгоритма разветвляющейся структуры; □
- циклического алгоритма – алгоритма циклической структуры.

Алгоритм линейной структуры – это объединение всех действий в единую цепь, в которой каждое последующее действие строго и однозначно следует за предыдущим действием (рис.7). На языке псевдокода соответствующая последовательность команд запишется в виде:

команда 1; команда 2; . . . ; команда N

Алгоритм разветвляющейся структуры содержит проверку одного либо нескольких условий, по результатам которых происходит переключение на один из возможных двух либо один из возможных нескольких вариантов дальнейшего развития процесса. Различают четыре вида ветвящегося алгоритма.

Ветвление «если-то-иначе» (рис. 8) встречается на практике наиболее часто. В нем присутствует проверка одного условия, по результатам которой происходит выбор между двумя возможными цепочками дальнейших действий.

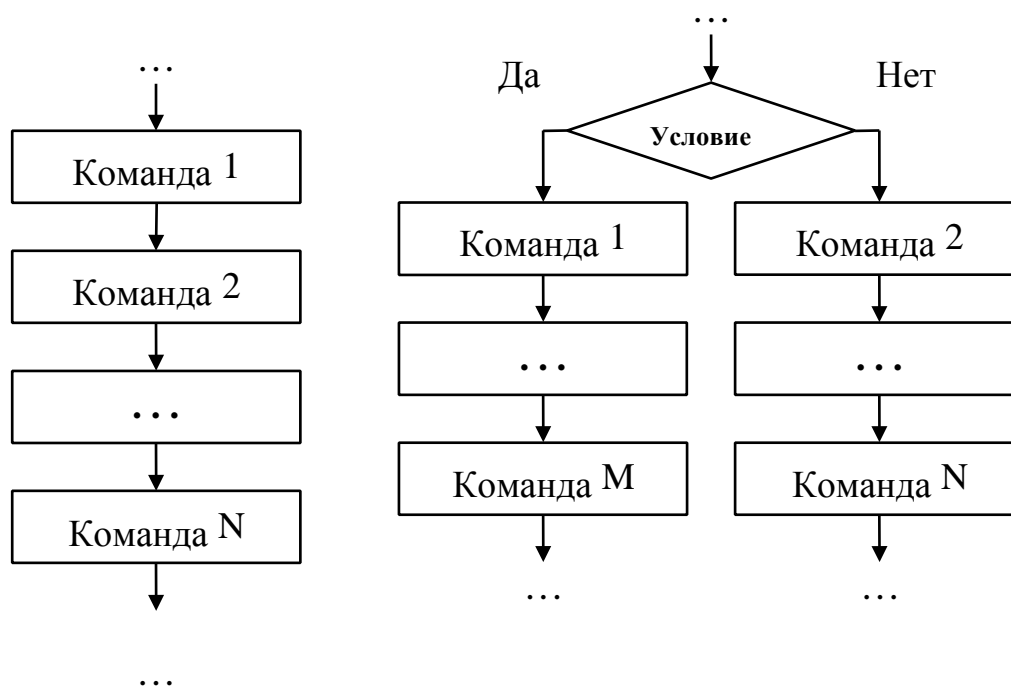


Рис. 7. Алгоритм линейной структуры
 Рис. 8. Алгоритм разветвляющейся структуры
 (вариант «если-то-иначе»)

На языке псевдокода соответствующая последовательность команд запишется в виде:

```

если Условие
    то команда 1; . . . ; команда M
    иначе команда 2; . . . ; команда N
все
    
```

Ветвление «**если-то**» (рис. 9) позволяет при соблюдении проверяемого в нем условия выполнить заданную цепочку действий: *команды 1...N*. При несоблюдении указанного условия *команды 1...N* пропускаются и сразу начинает выполняться следующая за ветвлением *команда N+1*. На языке псевдокода вариант «**если-то**» запишется в виде:

```

если Условие
    то команда 1; . . . ; команда N
все; команда N+1;
    
```

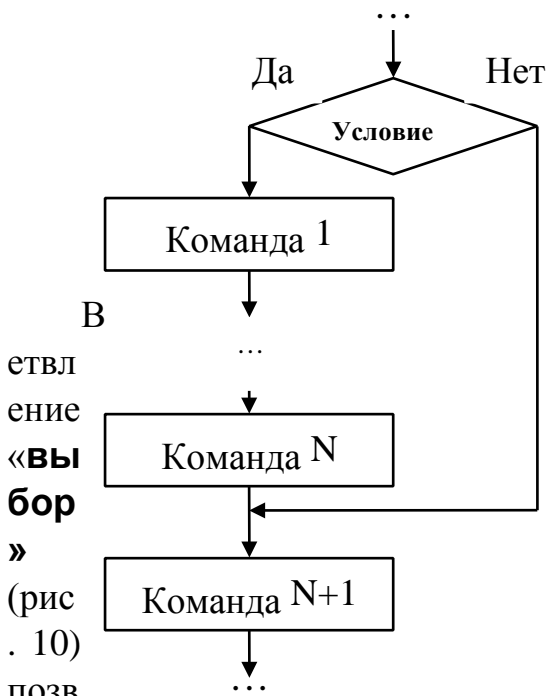


Рис. 9. Алгоритм разветвляющейся структуры (вариант «если-то»)

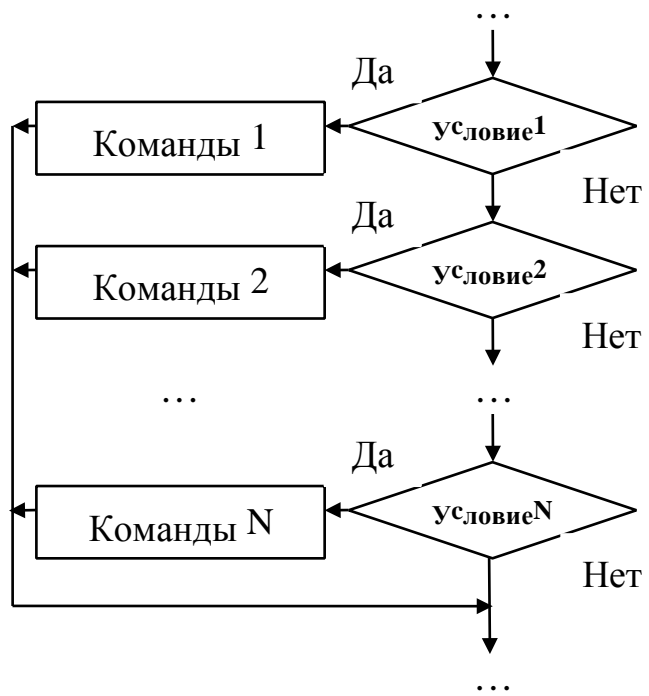


Рис. 10. Алгоритм разветвляющейся структуры (вариант «выбор»)

имеющихся альтернатив – цепочек команд. Для каждой альтернативы сначала проверяется соответствующее ей условие, срабатывает та первая альтернатива, у которой оно выполнится. Если не выполнится ни одно из условий выбора – ни одна из N цепочек команд не сработает, а управление перейдет к следующей после ветвления команде.

На языке псевдокода вариант «**выбор**» запишется в виде: **выбор**
при условии 1: команды 1
при условии 2: команды 2

при условии N: команды N
все

Ветвление «**выбор-иначе**» (рис. 11) предусматривает проверку условий только у первых $N-1$ альтернатив, а у последней N -й цепочки команд условие отсутствует. Если не сработает ни одна из первых $N-1$ альтернатив, то тогда автоматически выполнится N -я цепочка команд. Таким образом, в отличие от ветвления «**выбор**», здесь обязательно произойдет срабатывание одной из имеющихся N цепочек команд.

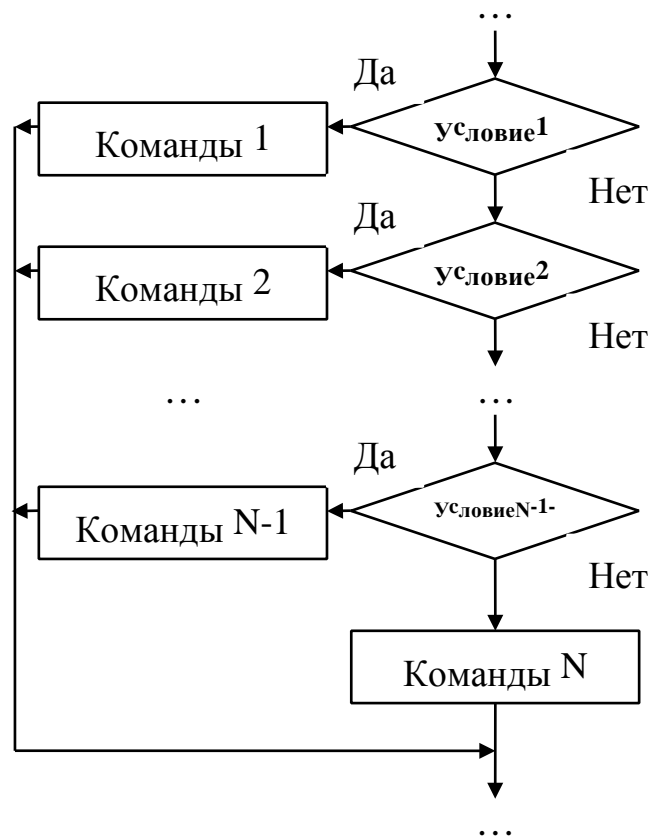


Рис. 11. Алгоритм разветвляющейся структуры (вариант «выбор-иначе»)

На языке псевдокода вариант «**выбор-иначе**» запишется в виде:

выбор

при условии 1: команды 1

при условии 2: команды 2

.....

при условии N-1: команды N-

1 иначе команды N все

Алгоритм циклической структуры обеспечивает повторение операции или группы операций при выполнении некоторого условия, называемого условием цикла. Такое повторение может быть многократным, но не должно быть бесконечным. Если повторение продолжается сколь угодно много раз, то говорят о заиклиивании алгоритма. При реализации на компьютере заиклиивание приводит к необходимости прервать цикл, не дожидаясь его формального завершения.

На рис. 12 представлены цикл с предусловием (а) и цикл с постусловием (б). Повторяющееся тело цикла составляют команды $1 \dots N$, команда $N+1$ в цикл не входит. В цикле с предусловием тело цикла не выполнится ни разу, если первая проверка условия цикла покажет его несоблюдение. В цикле с постусловием тело цикла обязательно выполнится хотя бы один раз.

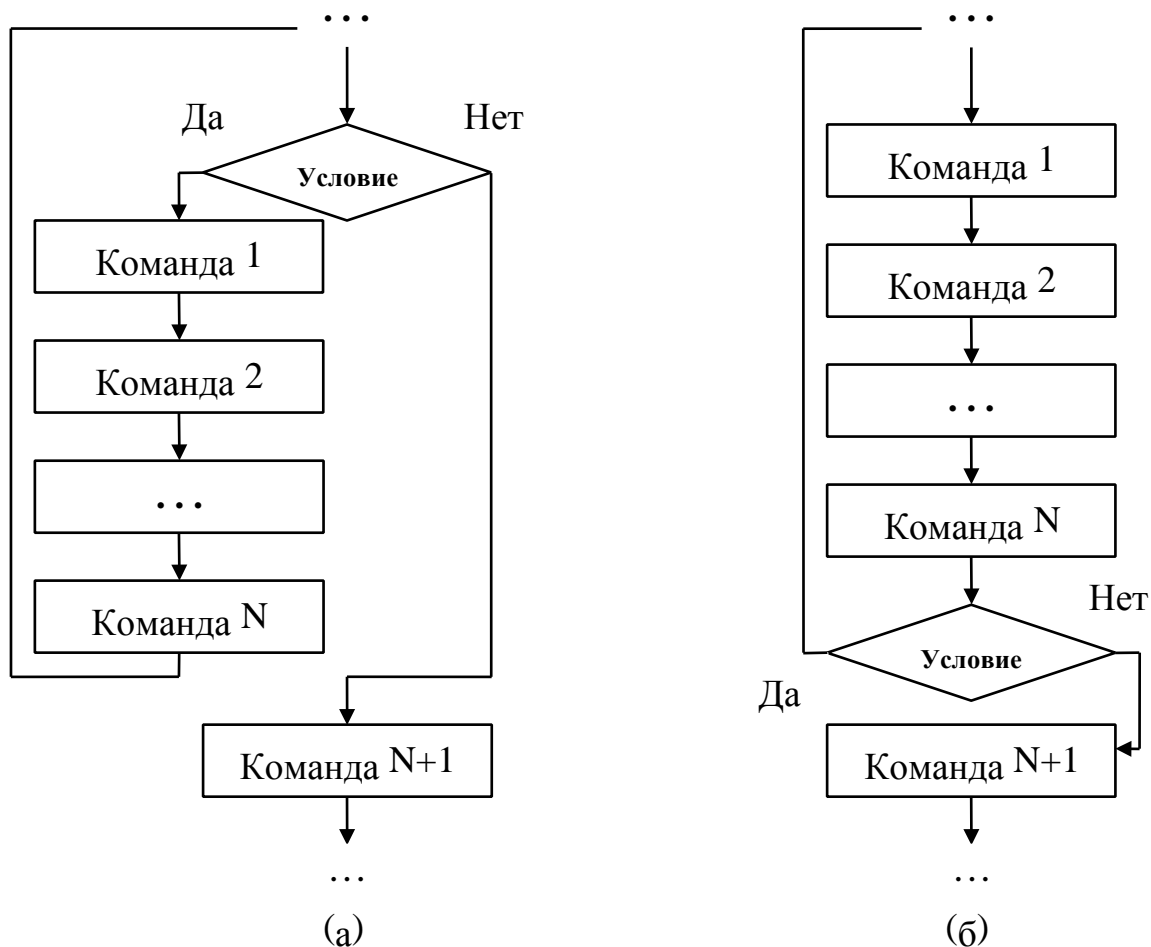


Рис. 12. Алгоритмы циклической структуры

На языке псевдокода циклы с предусловием или с постусловием отображаются соответственно следующими вариантами команды **пока**:

(а)

нц пока Условие*команда 1; команда 2; . . . ; команда N***кц;***команда N+1;*

(б)

нц команда 1; команда 2; . . . ; команда**N пока Условие кц;***команда N+1;*

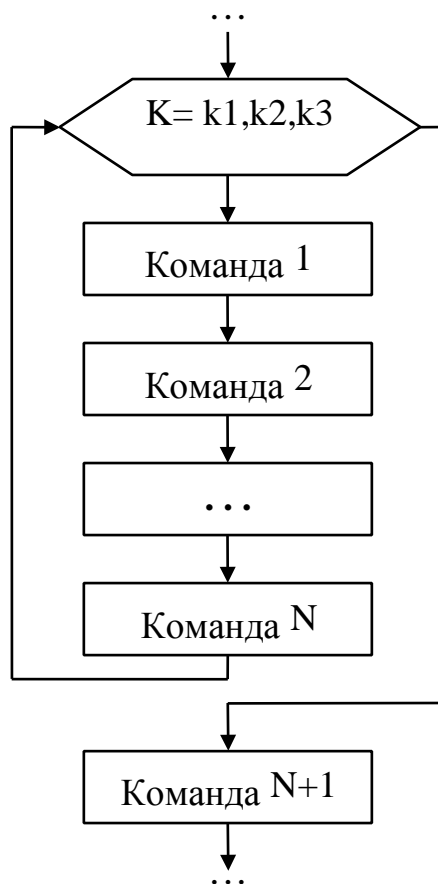


Рис. 13. Цикл с параметром

В тех случаях, когда число повторов выполнения тела цикла заранее известно, для отображения цикла удобно использовать блок модификатор (табл. 4), в котором размещается заголовок цикла (рис. 13).

На языке псевдокода цикл с параметром отображается с помощью команды **для**:

нц

для K **от** k1 **до** k2 **шаг** k3

команда 1; команда 2; . . . ; команда

N **кц**; *команда N+1*

На практике базовые структуры разрабатываемого сложного алгоритма сначала описываются словами на уровне идеи, затем, по мере приближения к программной реализации, алгоритм получает всё более формальные очертания и в итоге формулируется в виде блок-схем и фрагментов псевдокода.

10.4 Контрольные вопросы

1. Какой цели служит формализация понятия «алгоритм»?
2. В чем состоит свойство детерминированности вычислительного алгоритма?
3. Каковы формы представления вычислительного алгоритма?
4. В чем состоят преимущества и недостатки представления алгоритма в виде блок-схемы?
5. На какой стадии разработки алгоритма эффективно использование псевдокода?
6. Каковы базовые структуры вычислительных алгоритмов?
7. В каких случаях результатом ветвления становится пропуск команды или блока команд?
8. В каком из базовых типов цикла возможна ситуация, когда ни разу не выполнится тело цикла?
9. В каком из базовых типов цикла число повторов выполнения тела цикла точно задается заранее?
10. В чем выражается заикливание алгоритма?

10.5 Тест по материалам лекции

- 1. Под вычислительным алгоритмом принято понимать:** Любую конечную последовательность действий.
 - Особый способ представления информации в компьютере.
 - Полное, понятное, точное и результативное предписание совершить набор вычислительных действий, направленных на решение задачи.
 - Отображение предметного мира в виде системы команд.
- 2. Обязательным свойством вычислительного алгоритма является:** Присутствие в его описании кодов компьютерных команд.
 - Высокая скорость его реализации на компьютере. Повторяемость результата работы алгоритма при повторных запусках с теми же самыми исходными данными.
 - Невозможность пропуска какого-либо из его действий в процессе выполнения.
- 3. К обязательным свойствам алгоритма не относится:**

- Массовость.
- Детерминированность.
- Цикличность. □ Конечность.

4. Свойство алгоритма, обеспечивающее гарантированное получение результата расчетов, – это: □ Возможность запуска алгоритма для любых допустимых исходных данных расчета.

- Отсутствие в алгоритме элементов случайного выбора.
- Свойство конечности алгоритма.
- Наличие в алгоритме операции вывода результата расчетов.

5. Свойство алгоритма «массовость» означает, что:

- Это – широко распространенный алгоритм.
- Алгоритм может быть реализован на массовых портативных вычислительных устройствах.
- Он применим к любым допустимым исходным данным задачи. □ Это – алгоритм решения часто встречающейся задачи.

6. Представление алгоритма в форме блок-схемы наиболее эффективно:

- На начальной стадии проектирования алгоритма.
- При очень большом числе шагов алгоритма.
- Для его перевода на язык машинных команд.
- Для зрительной интерпретации алгоритма.

7. Представление алгоритма в форме псевдокода обязательно предусматривает:

- Присутствие в его описании кодов машинных команд. □ Использование синтаксиса конкретного алгоритмического языка высокого уровня.
- Использование стандартных синтаксических конструкций.
- Использование математических формул.

8. Алгоритм линейной структуры обязательно предусматривает: □ Присутствие в нем операций линейной алгебры.

- Его интерпретацию в виде геометрических фигур.
- Исполнение каждого последующее шага строго и однозначно за предыдущим. □ Его запись на языке псевдокода в одну линию.

9. Любой алгоритм циклической структуры обязательно предусматривает:

- Обязательное выполнение тела цикла.
- Отсутствие в нем проверок каких-либо условий.

- Проверку условий завершения цикла. □ Предварительную фиксацию числа повторов тела цикла.

10. При выполнении алгоритма следующего базового типа возможно заикливание:

- Алгоритм разветвляющейся структуры.
- Цикл с параметром.
- Цикл с постусловием.
- Алгоритм линейной структуры.

Тема 11 Программирование

11.1 РАБОТА В ИНТЕГРИРОВАННОЙ СРЕДЕ РАЗРАБОТКИ ПРИЛОЖЕНИЙ НА ЯЗЫКЕ VBA

Для вызова интегрированной среды разработки приложений (IDE) необходимо, выбрать в меню следующие команды: **Сервис** ® **Макрос** ® **Редактор Visual Basic**. Общий вид IDE приведен на рис. 1.1. IDE состоит из нескольких компонентов: главного меню, панели инструментов, окна проекта, окна свойств, панели элементов, конструктора форм, окна контрольных значений и нескольких других вспомогательных окон.

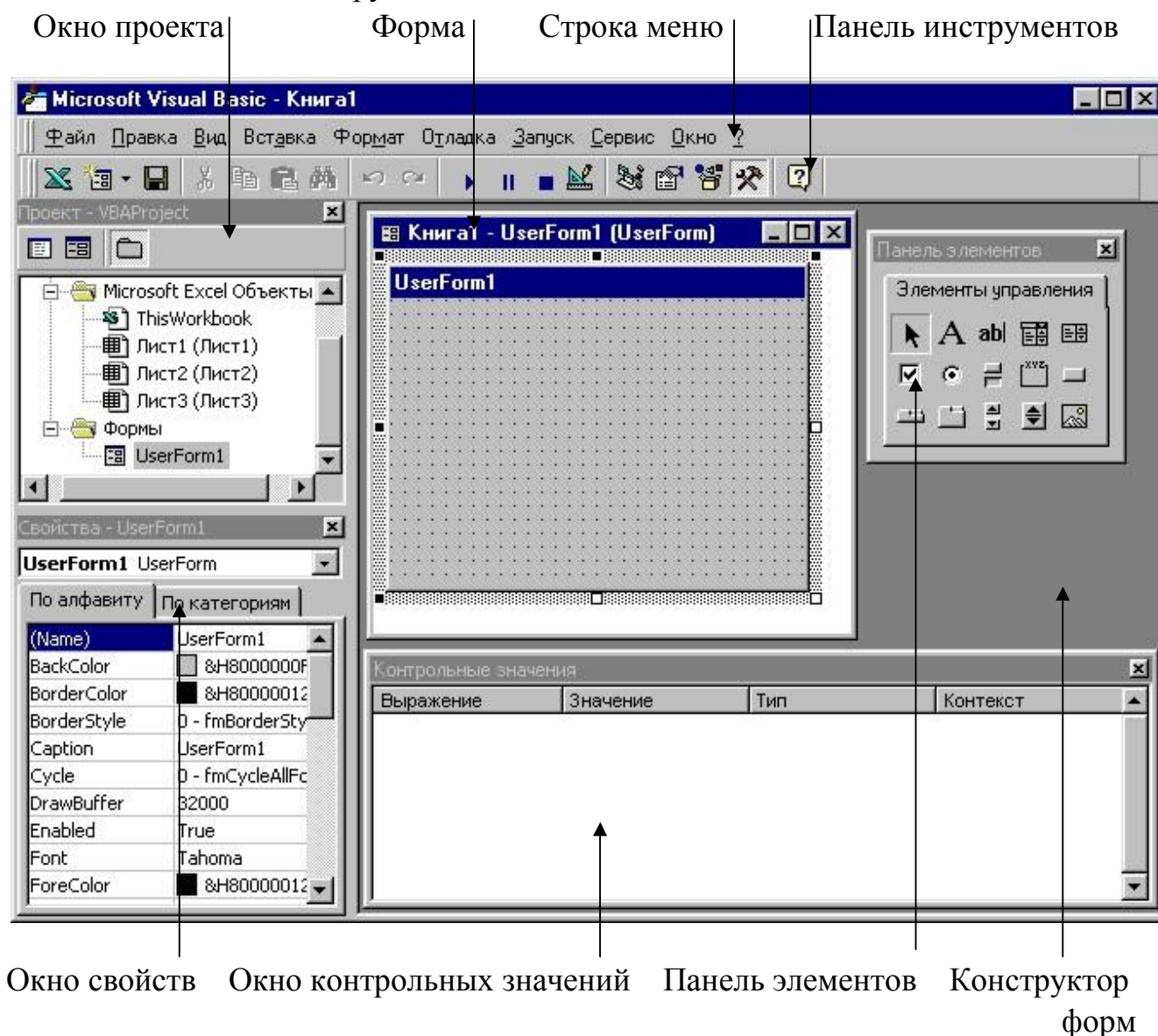


Рис. 1.1. Общий вид IDE VBA

Главное меню – это строка текста, расположенная в верхней части окна Visual Basic, и состоит из нескольких пунктов.

Меню **Файл** предназначено для работы с файлами, из которых образуются приложения. В нем можно создавать, сохранять и печатать проекты. Меню **Правка** выполняет стандартные операции с буфером обмена – вырезание,

копирование и вставка. Они применяются не только к фрагментам программы, но и к управляющим элементам. В меню **Вид** включаются режимы просмотра различных компонентов и инструментов. Просматривать можно формы и программные модули. Меню **Вставка** позволяет добавлять процедуры, формы, модули и модули класса. Команды меню **Формат** определяют расположение и размеры элементов и форм. При помощи команд меню **Отладка** можно запустить и остановить приложение, расставить точки прерывания и выбрать просматриваемые объекты, а также выполнить другие операции, помогающие следить за работой приложения. Команды меню **Запуск** запускают и останавливают приложение, прерывают и возобновляют выполнение программы, что особенно удобно в процессе отладки. Меню **Сервис** позволяет включить дополнительные элементы, запустить макросы и настроить параметры редактора. Меню **Окно** позволяет выстроить окна IDE (каскадное или мозаичное расположение), упорядочить значки свернутых форм, а также создает список, позволяющий быстро перейти к одному из открытых окон IDE. Меню **?** – помощь пользователю. Для быстрого вызова главного меню необходимо нажать клавишу F10.

Панель инструментов находится под главным меню (рис. 1.2). Если она отсутствует, необходимо выполнить команду **Вид** ® **Панели инструментов** ® **Стандарт**.

Окно проекта напоминает собой окно Проводника Windows, и предназначено для быстрого просмотра составляющих проекта, который объединяет в себе все объекты, составляющие приложение (рис. 1.3). Это стандартные объекты открытого приложения Microsoft Office (документ редактора MS Word, книга и её листы MS Excel), формы, модули и классы.

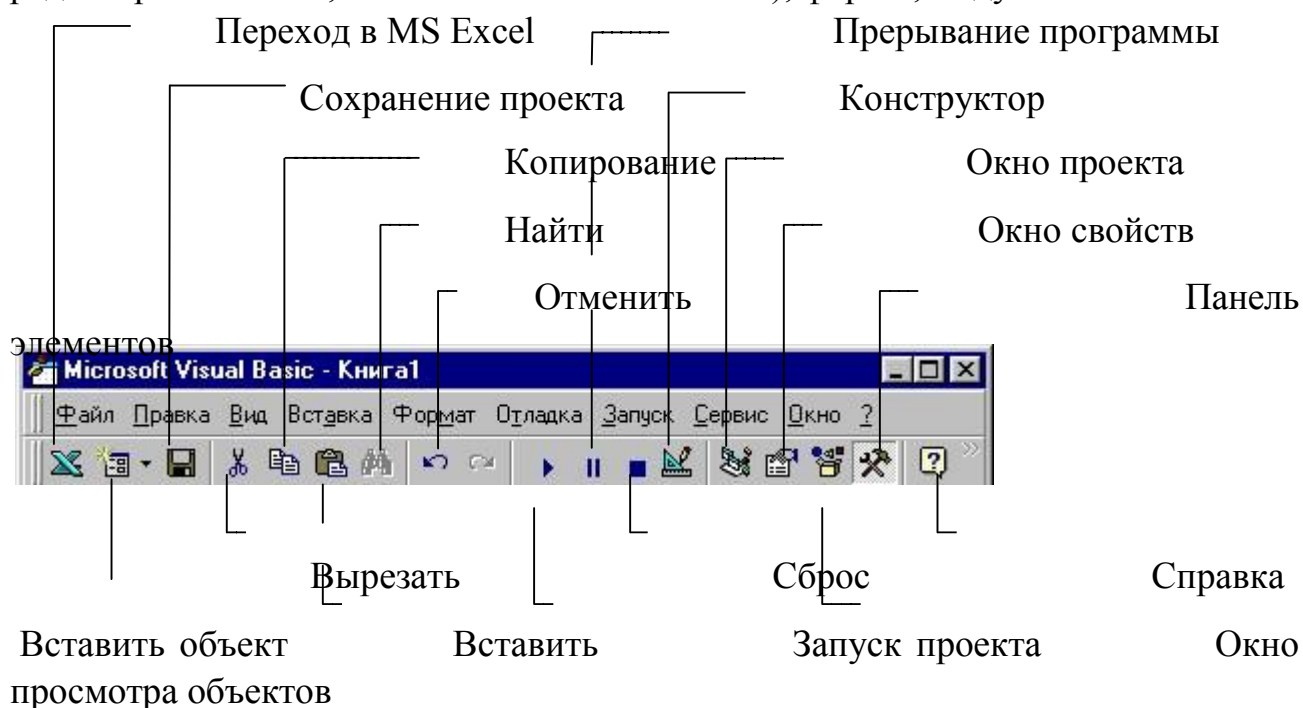


Рис. 1.2. Панель инструментов VBA

Окно свойств отображает различные атрибуты выделенного объекта (рис. 1.4). Все объекты (формы, управляющие элементы и т.д.) имеют атрибуты, которые изменяют не только внешний вид объекта, но и его поведение. Все эти атрибуты называются свойствами. Следовательно, каждый объект обладает набором свойств.



Рис. 1.3. Окно проекта

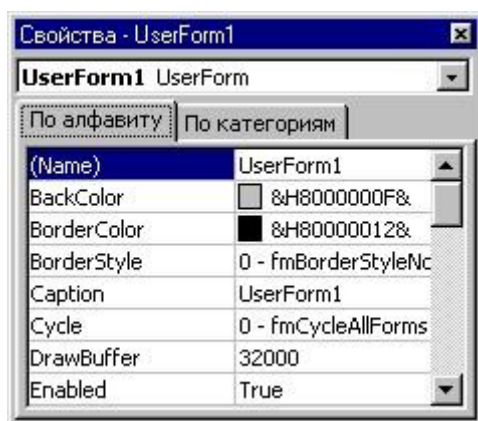


Рис. 1.4. Окно свойств

Окно контрольных значений позволяет просматривать значения контрольных переменных в процессе проверки правильности работы (отладки) проекта, что позволяет находить ошибки в логике работы программ.

Конструктор форм расположен в центре экрана редактора VBA. Здесь выводится либо изображение формы, что позволяет производить визуальное конструирование макета формы и расположенных на ней элементов, либо окно программы (более подробно об этом будет рассказано ниже).

11.2 2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ VBA

Процесс разработки программы на языке VBA – **проекта**, может состоять из нескольких этапов, в зависимости от конечного результата. Если необходимо

получить программу, которая будет производить определенные вычисления или действия, расширяющие математические возможности стандартного приложения Microsoft Office, то достаточно создать **программный модуль**. Для применения этой программы можно поместить в рабочей области приложения кнопку, нажатие которой будет вызывать выполнение программы. Для этого в приложении необходимо включить панель инструментов с помощью команды **Вид** ® **Панели инструментов** ® **Элементы управления**, а затем создать кнопку с соответствующим программным кодом. Либо выполнять программу с помощью команды **Сервис** ® **Макрос** ® **Макросы**.

Разработка “полноценной” программы (для выполнения которой требуется отдельное окно, с различными элементами управления) будет включать два этапа. Первый этап – этап визуального программирования, на котором создается окно (**форма**) программы, где располагаются необходимые элементы управления. Второй – этап программирования, на котором создаются части программы (**процедуры**), выполняющиеся в ответ на определенные события. Событием является, например, щелчок левой кнопкой мыши на командной кнопке (событие Click), нажатие клавиши на клавиатуре (событие KeyPress) и т.д. Использовать такое приложение можно нажатием кнопки – «Запуск проекта».

2.1. Объекты, свойства и методы VBA



Одним из основных понятий VBA является объект. **Объект** – это то, чем вы управляете с помощью программы на языке VBA, например, форма, кнопка, рабочий лист или диапазон ячеек MS Excel. Каждый объект обладает некоторыми **свойствами**. Например, форма может быть видимой или невидимой в данный момент на экране. Другой пример свойства объекта – шрифт для отображения информации в ячейке (объекте) рабочего листа.

Объект содержит также список методов, которые к нему применимы. **Методы** – это то, что вы можете делать с объектом. Например, показать форму на экране или убрать её можно с помощью методов Show и Hide.

Таким образом, **объект** – это программный элемент, который имеет свое отображение на экране, содержит некоторые переменные, определяющие его **свойства**, и некоторые **методы** для управления объектом. Например, в MS Excel имеется много встроенных объектов:

- Range(“Адрес”)** - диапазон ячеек (может включать только одну ячейку).
- Cells(i, j)** - ячейка, находящаяся на пересечении i-й строки и j-го столбца рабочего листа MS Excel (i и j – целые числа).
- Rows(№ строки)** - строка с заданным номером.

Columns(№ столбца) - столбец с заданным номером

Sheets(“Имя”) - лист с указанным именем.

Sheets(№ листа) - лист с указанным номером.

WorkSheet - рабочий лист.

Установка значений свойств – это один из способов управления объектами. Синтаксис установки значения свойства объекта следующий:

Объект. Свойство = Выражение

Основным свойством объектов **Cells** и **Range**, является **Value** (значение), которое, однако, можно не указывать. Например:

Range(“A5:A10”). **Value** = 0 или **Range**(“A5:A10”) = 0 – в диапазон ячеек A5:A10 заносится значение 0.

Cells(2, 4). **Value** = n или **Cells**(2, 4) = n – в ячейку, находящуюся на пересечении 2-й строки и 4-го столбца (ячейка с адресом “D2”), заносится значение переменной n.

Синтаксис чтения свойств объекта следующий:

Переменная = Объект. Свойство

Например:

Xn = **Cells**(1, 2).**Value** или **Xn** = **Range**(“B1”).**Value** – переменной Xn присваивается значение из ячейки B1 текущего рабочего листа.

Синтаксис применения методов к объекту:

Объект. Метод

Например:

Sheets(2).**Activate** – сделать активным лист с №2.

Sheets(“Диаграмма”).**Delete** – удалить лист с именем “Диаграмма”.

Range(“A5:A10”).**Clear** – очистить диапазон ячеек A5:A10.

Range(“A2:B10”).**Select** – выделить диапазон ячеек A2:B10.

В MS Excel имеются объекты, которые содержат другие объекты. Например, рабочая книга содержит рабочие листы, рабочий лист содержит диапазон ячеек и т.д. Объектом самого высокого уровня является **Application** (приложение). Если вы изменяете его свойства или вызываете его методы, то результат применяется к текущей работе MS Excel. Например:

Application.Quit - завершение работы с Excel.

Отметим, что точка после имени объекта может использоваться для перехода от одного объекта к другому. Например, следующее выражение очищает вторую строку рабочего листа **Май** в рабочей книге **Отчет**:

Application.Workbooks("Отчет").Worksheets("Май").Rows(2).Delete

Нужно отметить следующее:

- Можно не писать имя объекта **Application**, так как это подразумевается по умолчанию.
- При работе с подобъектом уже активизированного объекта нет необходимости указывать содержащий его объект.
- VBA использует некоторые свойства и методы, которые возвращают объект к которому они относятся (это позволяет быстро указывать нужный объект). Примеры таких свойств: `ActiveCell` (активная ячейка), `ActiveSheet` (активный лист), `ActiveWorkBook` (активная рабочая книга). Так, установить значение активной ячейки можно следующим образом:

ActiveCell.Value = " Да".

2.2. Описание данных

Все объекты, которыми оперирует язык программирования VBA, относятся к определенному типу.

Тип данных определяет:

- область возможных значений переменной;
- структуру организации данных;
- операции, определенные над данными этого типа.

Типы данных подразделяются на простые (скалярные) и сложные (структурированные). У простых типов данных возможные значения данных едины и неделимы. Сложные же типы имеют структуру, в которую входят различные простые типы данных. Скалярные типы данных представлены в таблице 2.1.

Таблица 2.1. Скалярные типы VBA

Имя типа	Русское название типа	Возможные значения
Boolean	Логический	True, False
Byte	Байтовый	0...255
Integer	Целое	-32768...+32767
Long	Длинное целое	-2147483648...+2147483647
Single	Число с плавающей точкой	-3,4E38...-1,4E-45 для отрицательных значений. 1,4E-45...3,4E38 для положительных значений.

Double	Число с плавающей точкой двойной точности	-1,7E308...-4,9E-324 для отрицательных значений. 4,9E-324...1,7E308 для положительных значений.
Currency	Денежный	Десятичные числа с фиксированной позицией запятой. Возможны 15 цифр до запятой и 4 после.
String	Строковый	Есть два вида строк: строки фиксированной длины (до 2^{16} символов) и строки переменной длины (до 2^{31} символов). Данные записываются в кавычках.
Date	Дата	Даты изменяются в диапазоне от 1.01.100г. до 31.12.9999г.
Object	Объект	Ссылка на объект (указатель)
Variant	Вариант	Универсальный тип, значением которого могут быть данные любого из перечисленных выше типов, объекты, значения NULL и значения ошибок ERROR.

Переменные в программе можно описывать или не описывать. В последнем случае ей будет присвоен тип **Variant**. Явно описывать переменную можно как в начале блока, так и в любом месте, где возникла необходимость использовать новую переменную. Лучше все переменные описывать явно и, как правило, в начале блока. Для запрета использования переменных, которые не были описаны явно, в начало программы необходимо вставить оператор **Option Explicit**.

2.2.1. Описание простых переменных

Описание простых переменных имеет следующий синтаксис:

Dim ИМЯ_ПЕРЕМЕННОЙ **As** ИМЯ_ТИПА

Одним оператором **Dim** можно описать произвольное число переменных, но конструкция **As** должна быть указана для каждой из них, иначе переменным без **As** будет присвоен тип **Variant**.

Например.

Dim X As Byte, Z As Integer, C, Слово As String

Здесь переменная *X* - это переменная байтового типа, переменная *Z* - целого типа, переменная *C* - типа вариант (по умолчанию), переменная *Слово* строкового типа.

2.2.2. Описание констант

Данные, не изменяющиеся внутри программы можно считать константами. Их можно описать следующим образом:

Const ИМЯ_КОНСТАНТЫ As ИМЯ_ТИПА=ПОСТОЯННОЕ_ВЫРАЖЕНИЕ

Например.

Const Pi As Double = 3.141593

2.2.3. Описание массивов

Для хранения векторов, матриц и т.д. можно использовать массивы.

Массив - это структурированный тип данных, который представляет собою последовательность ячеек памяти, имеющих общее имя и хранящих данные одного типа. Каждый элемент массива определяется индексом (номером). Количество элементов в массиве называется **размерностью массива**. Массив описывается следующей конструкцией:

Dim ИМЯ_МАССИВА(СПИСОК_РАЗМЕРНОСТЕЙ) As ИМЯ_ТИПА

В списке размерностей массива каждое измерение отделяется запятой и определяется заданием нижней и верхней границ изменения индексов.

Например.

Dim X(1 TO 5) As Integer, Y(1 To 10, 1 To 20) As Double

Здесь *X* - одномерный массив, состоящий из 5 элементов целого типа, *Y* - двумерный массив, у которого 10 строк и 20 столбцов с элементами числового типа двойной точности.

2.3. Выражения

Выражения устанавливают порядок выполнения действий над элементами данных. Выражения состоят из операндов и знаков операций. Операндами являются константы, переменные, указатели функций, выражения, взятые в скобки.

2.3.1. Виды операций

Операции бывают арифметические, отношения и логические:

– **арифметические операции:**

^ возведение в степень,

* умножение,

- / деление,
- \ деление нацело,
- mod остаток от деления,
- + плюс,
- минус;
- **операции отношения:**
 - < меньше,
 - > больше,
 - <= меньше или равно,
 - >= больше или равно,
 - = равно,
 - <> не равно;
- **логические операции:**
 - Not логическое отрицание,
 - And логическое "И",
 - Or логическое "ИЛИ".

Результатом логической операции может быть одно из двух значений: True ("истина") или False ("ложь").

2.3.2. Приоритет выполнения операций

Если выражение содержит несколько операций, то приоритет их выполнения следующий:

1. Сначала выполняются арифметические операции в таком порядке, как они представлены в таблице 2.2.

Таблица 2.2. Приоритет арифметических операций

Описание операции	Обозначение в VBA
Унарный минус (изменение знака)	- (в начале выражения)
Возведение в степень	^
Умножение и деление	*, /
Деление нацело и остаток от деления	\, Mod
Сложение и вычитание	+, -

2. Далее операции отношения (они имеют одинаковый приоритет).
3. Последними выполняются логические операции в таком порядке, как они представлены в таблице 2.3.

Таблица 2.3. Приоритет логических операций

Описание операции	Обозначение в VBA
Логическое отрицание	Not
Логическое "И"	And
Логическое "ИЛИ"	Or

Если выражение содержит несколько операций одинакового приоритета, то порядок их выполнения слева направо. Чтобы изменить порядок действий в выражении используются круглые скобки.

Выражения бывают арифметические, отношения и логические.

Арифметические выражения записываются с помощью операндов числовых типов и арифметических операций, а результатом является числовое значение. В арифметическом выражении можно использовать стандартные математические функции, которые приведены в таблице 2.4.

Таблица 2.4. Стандартные математические функции VBA

Математическая запись	Имя функции в VBA	Описание
$\frac{1}{2}X^{\frac{1}{2}}$	Abs(число)	Возвращает значение, тип которого совпадает с типом переданного аргумента, равное абсолютному значению указанного числа.
arctg X	Atn(число)	Возвращает значение типа Double, содержащее арктангенс числа.
cos X	Cos(число)	Возвращает значение типа Double, содержащее косинус угла.
]X[Int(число)	Возвращает значение типа, совпадающего с типом аргумента, которое содержит целую часть числа.
ln X	Log(число)	Возвращает значение типа Double, содержащее натуральный логарифм числа.
ex	Exp(число)	Возвращает значение типа Double, содержащее результат возведения числа <i>e</i> (основание натуральных логарифмов) в указанную степень.
Sign X	Sgn(число)	Возвращает значение типа Variant (Integer), соответствующее знаку указанного числа.
sin X	Sin(число)	Возвращает значение типа Double, содержащее синус угла.

\sqrt{X}	Sqr(число)	Возвращает значение типа Double, содержащее квадратный корень указанного числа.
tg X	Tan(число)	Возвращает значение типа Double, содержащее тангенс угла.

Выражения отношения определяют истинность или ложность результата <http://www.abbyy.com/buy> при сравнении двух операндов. Сравнить можно данные любого одинакового типа. Результат операции отношения только логический: True - "истина" или False - "ложь".

Логические выражения. Результатом логического выражения является логическое значение True или False. Простейшими видами логических выражений являются: логическая константа, логическая переменная, логическая функция, выражение отношения. Логические операции выполняются только над операндами логического типа.

Пример. Записать $1 \leq X \leq 5$ и определить значение выражения при $X=3.1$. Выражение в VBA будет выглядеть так:

$X \geq 1 \text{ And } X \leq 5$

Результатом выражения будет *True*.

Чтобы получить перечень всех математических функций, достаточно набрать имя любой известной математической функции (например, *SIN*), а затем нажать клавишу **F1** и ниже описания выбранной функции выбрать ссылку на **Математические функции**. В полученном перечне можно получить справку о назначении любой из встроенных математических функций и ее аргументе.

Чтобы получить перечень всех производных математических функций и правила их формирования, достаточно набрать имя любой известной математической функции (например, *SIN*), а затем нажать клавишу **F1** и ниже описания выбранной функции выбрать ссылку на **Производные математические функции**.

Ниже, в таблице 2.5, приведен список функций, которые могут быть получены с помощью встроенных математических функций.

Таблица 2.5. Производные математические функции

Математическая запись	Название функции	Комбинация встроенных функций
sc X	Секанс	1/Cos(X)
csc X	Косеканс	1/Sin(X)
ctg X	Котангенс	1/Tan(X)

$\arcsin X$	Арксинус	$\text{Atn}(X/\text{Sqr}(-X*X+1))$
$\arccos X$	Арккосинус	$\text{Atn}(-X/\text{Sqr}(-X*X+1))+2*\text{Atn}(1)$
$\text{arcsc } X$	Арксеканс	$\text{Atn}(X/\text{Sqr}(X*X-1))+\text{Sgn}(X-1)*2*\text{Atn}(1)$
$\text{arccsc } X$	Арккосеканс	$\text{Atn}(X/\text{Sqr}(X*X-1))+(\text{Sgn}(X)-1)*2*\text{Atn}(1)$
$\text{arctg } X$	Арккотангенс	$\text{Atn}(X)+2*\text{Atn}(1)$
$\text{sh } X$	Гиперболический синус	$(\text{Exp}(X)-\text{Exp}(-X))/2$
$\text{ch } X$	Гиперболический косинус	$(\text{Exp}(X)+\text{Exp}(-X))/2$
$\text{th } X$	Гиперболический тангенс	$(\text{Exp}(X)-\text{Exp}(-X))/(\text{Exp}(X)+\text{Exp}(-X))$
$\text{sch } X$	Гиперболический секанс	$2/(\text{Exp}(X)+\text{Exp}(-X))$
$\text{csch } X$	Гиперболический косеканс	$2/(\text{Exp}(X)-\text{Exp}(-X))$
$\text{cth } X$	Гиперболический котангенс	$(\text{Exp}(X)+\text{Exp}(-X))/(\text{Exp}(X)-\text{Exp}(-X))$
$\text{arsh } X$	Гиперболический арксинус	$\text{Log}(X+\text{Sqr}(X*X+1))$
$\text{arch } X$	Гиперболический арккосинус	$\text{Log}(X+\text{Sqr}(X*X-1))$
$\text{arth } X$	Гиперболический арктангенс	$\text{Log}((1+X)/(1-X))/2$
$\text{arsch } X$	Гиперболический арксеканс	$\text{Log}((\text{Sqr}(-X*X+1)+1)/X)$
$\text{arcsch } X$	Гиперболический арккосеканс	$\text{Log}((\text{Sgn}(X)*\text{Sqr}(X*X+1)+1)/X)$
$\text{arth } X$	Гиперболический арккотангенс	$\text{Log}((X+1)/(X-1))/2$
$\log_n X$	Логарифм по основанию N	$\text{Log}(X)/\text{Log}(N)$

11.3 3. ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА VBA

3.1. Правила записи операторов

При записи операторов необходимо придерживаться следующих правил:

- Каждый новый оператор записывается с новой строки.
- Чтобы записать несколько операторов на одной строке, их разделяют между собой двоеточием (:).

- Если оператор не помещается в одной строке, то необходимо поставить в конце строки пробел и знак подчеркивания (_), а затем продолжить не поместившуюся часть на следующей строке.

3.2. Оператор присваивания

Оператор присваивания используется, если какой-то переменной нужно присвоить новое значение. Он имеет следующий синтаксис:

ИМЯ_ПЕРЕМЕННОЙ = ВЫРАЖЕНИЕ

Сначала вычисляется выражение в правой части, а затем результат присваивается переменной, стоящей в левой части.

Например. Записать с помощью оператора присваивания следующее ма-

тематическое выражение:

$$y = \sqrt[3]{a} + \frac{a^2 + e^{-b}}{\sin^2 a - \ln b}$$

На VBA это выражение можно записать в виде следующего оператора:

$Y = a^{(1/3)} + (a^2 + \text{Exp}(-b)) / (\text{Sin}(a)^2 - \text{Log}(b))$

3.3. Операторы ввода-вывода

3.3.1. Оператор и функция MsgBox

Оператор **MsgBox** осуществляет вывод информации в диалоговом окне и устанавливает режим ожидания нажатия кнопки пользователем.

Он имеет следующий синтаксис:

MsgBox *СООБЩЕНИЕ*[,*КНОПКИ*][, *ЗАГОЛОВОК*]

Аргументы:

СООБЩЕНИЕ - обязательный аргумент, задающий в окне выводимое информационное сообщение. Может состоять из нескольких текстовых строк, объединенных знаком **&**. Использование в этом аргументе **Chr(13)** приводит к переходу на новую строку при выводе информации.


КНОПКИ - значение этого аргумента определяет категории появляющихся в окне кнопок. От значения аргумента кнопки зависит также, появляется ли в

окне какой-либо значок. Если не указано, какие кнопки необходимо отображать в окне сообщений, то используется значение по умолчанию, соответствующее кнопке ОК. В табл. 3.1 приведены возможные комбинации кнопок и значков в окне сообщений.

ЗАГОЛОВОК - задает заголовок окна.

Функция **MsgBox** возвращает значение типа Integer, указывающее, какая кнопка была нажата в диалоговом окне.

Таблица 3.1. Допустимые значения переменной кнопки

Отображение	Аргумент
Кнопка ОК	VbOKOnly
Кнопки ОК и Отмена	VbOKCancel
Кнопки Да и Нет	VbYesNo
Кнопки Да, Нет и Отмена	VbYesNoCancel
Кнопки Прекратить, Повторить и Игнорировать	VbAbortRetryIgnore
Кнопки Повторить и Отмена.	VbRetryCancel
Информационный знак	VbInformation
Знак 	VbCritical
Знак вопроса	VbQuestion
Знак восклицания	VbExclamation

Например. Вывести сообщение о текущей дате.

Для этого можно использовать следующий оператор:

MsgBox "Сегодня на календаре" & Date , , "Внимание"

В результате будет выведено следующее окно (рис.3.1).

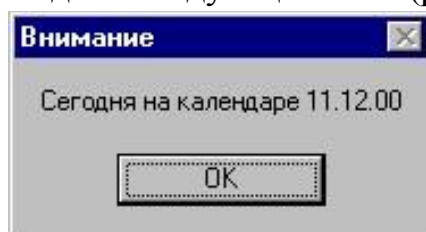


Рис. 3.1

После щелчка по кнопке ОК окно сообщения закроется, и выполнение программы возобновится с оператора, стоящего непосредственно за вызовом MsgBox.

3.3.2. Функция InputBox

Функция **InputBox** осуществляет ввод значений переменных с помощью окна ввода и имеет следующий синтаксис:

ИМЯ_ПЕРЕМЕННОЙ = **InputBox**(*СООБЩЕНИЕ*[, *ЗАГОЛОВОК*])

Аргументы:

СООБЩЕНИЕ - обязательный аргумент. Задает в окне информационное сообщение, обычно поясняющее смысл вводимой величины

ЗАГОЛОВОК - задает заголовок окна.

Например, Ввести значение переменной N с клавиатуры, предусмотрев значение по умолчанию равное 10.

Для этого можно использовать следующий оператор:

$N = \text{InputBox}(\text{"Введите N"}, \text{"Ввод исходных данных"}, 10)$

В результате будет выведено следующее окно для ввода значения переменной N (рис.3.2).

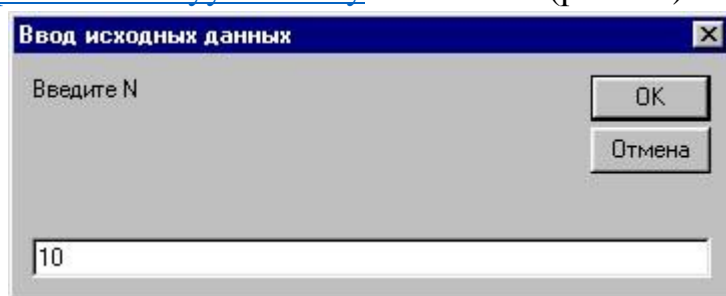


Рис. 3.2

Если значение по умолчанию подходит пользователю, то после щелчка кнопки ОК окно ввода закроется, переменной N присвоится значение 10 и выполнение программы возобновится с оператора, стоящего непосредственно за вызовом **InputBox**.

Если же значение по умолчанию не подходит пользователю, то перед щелчком по кнопке ОК необходимо ввести нужное значение переменной N.

3.4. Условный оператор IF

Для реализации разветвляющегося вычислительного процесса в VBA используется оператор **If...Then...Else**, который представляет собой простейшую форму проверки условий. Он имеет следующий синтаксис:

If УСЛОВИЕ Then ОПЕРАТОР_1 Else ОПЕРАТОР_2

ОПЕРАТОР_1 выполняется, если УСЛОВИЕ истинно, в противном случае выполняется ОПЕРАТОР_2. При этом оператор If...Then...Else записывается в одну строку.

УСЛОВИЕ – это выражение логического типа. Результат выражения всегда имеет булевский тип. Выражение может быть простым и сложным. При записи простых условий могут использоваться все возможные операции отношения, указанные в табл. 3.2.

Таблица 3.2. Логические отношения

Операция	Название	Выражение	Результат
=	Равно	$A = B$	True, если A равно B
<>	Не равно	$A \neq B$	True, если A не равно B
>	Больше	$A > B$	True, если A больше B
<	Меньше	$A < B$	True, если A меньше B
>=	Больше или равно	$A \geq B$	True, если A больше или равно B
<=	Меньше или равно	$A \leq B$	True, если A меньше или равно B

Сложные условия образуются из простых путем применения логических операций и круглых скобок. Список логических операций приведен в табл. 3.3.

Таблица 3.3. Логические операции

	Название	Выражение	A	B	Результат
Not	Логическое отрицание	Not A	False		True
			True		False
And	Логическое И	A And B	True	True	True
			True	False	False
			False	True	False
			False	False	False
Or	Логическое ИЛИ	A Or B	True	True	True
			True	False	True
			False	True	True
			False	False	False

В условном операторе допустимо использование блока операторов вместо любого из операторов. В этом случае условный оператор имеет вид:

If УСЛОВИЕ Then

 БЛОК_ОПЕРАТОРОВ_1

Else

 БЛОК_ОПЕРАТОРОВ_2

End If

В условном операторе может проверяться несколько условий. В этом случае условный оператор имеет вид:

If УСЛОВИЕ_1 Then

 БЛОК_ОПЕРАТОРОВ_1

ElseIf УСЛОВИЕ_2 Then

 БЛОК_ОПЕРАТОРОВ_2

Else

End If

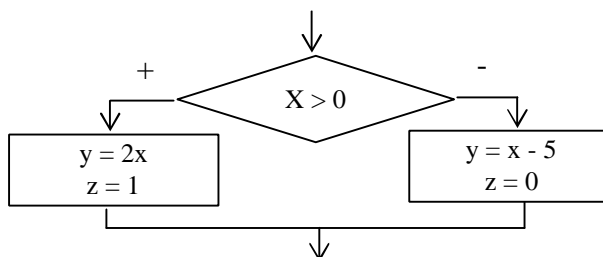
Пример 1. Написать часть программы для алгоритма на рис. 3.3.

If $x > 0$ *Then* $y = 2 * x$ $z = 1$ *Else*

$y = x - 5$

$z = 0$ *Рис. 3.3*

End If



Пример 2. Написать часть программы для алгоритма на рис. 3.4.

If $x > 0$ *Then* $y = 2 * x$

x

Else

If $x = 0$ *Then*

$y = x - 5$ *Else*

$y = x^2$

End If

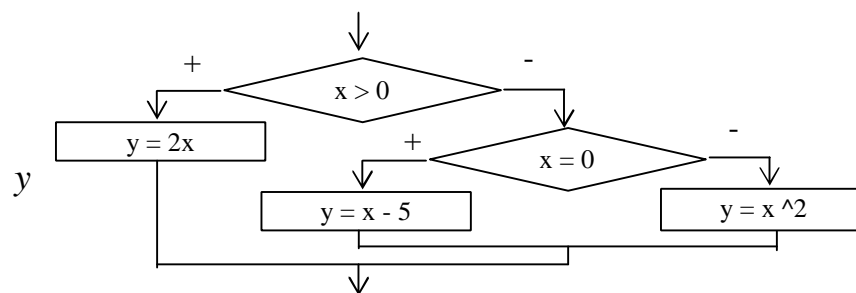


Рис. 3.4

End If

3.5. Оператор выбора **Select Case**

Оператор **Select Case** удобно использовать, когда в зависимости от значения некоторого выражения, имеющего конечное множество допустимых значений, необходимо выполнить разные действия. Он также относится к условным операторам, но имеет другой вид:

Select Case *ПРОВЕРЯЕМОЕ_ВЫРАЖЕНИЕ*

Case *ЗНАЧЕНИЯ_1*

ОПЕРАТОРЫ_1

Case *ЗНАЧЕНИЯ_2*

ОПЕРАТОРЫ_2

...

Case *ЗНАЧЕНИЯ_N*

ОПЕРАТОРЫ_N

[**Case Else**

ИНАЧЕ_ОПЕРАТОРЫ]

End Select

ПРОВЕРЯЕМОЕ_ВЫРАЖЕНИЕ может иметь любой скалярный тип, кроме вещественного. *ЗНАЧЕНИЯ* состоят из произвольного количества значений или диапазонов, отделенных друг от друга запятыми.

Тип *ЗНАЧЕНИЙ* должен совпадать с типом

ПРОВЕРЯЕМОГО_ВЫРАЖЕНИЯ.

Сначала вычисляется *ПРОВЕРЯЕМОЕ_ВЫРАЖЕНИЕ*. Если его значение совпадает с одним из значений *ЗНАЧЕНИЯ_1*, то выполняются *ОПЕРАТОРЫ_1* и управление передается оператору, стоящему после **End Select**. Если его значение не совпадает ни с одним из значений *ЗНАЧЕНИЯ_1*, то выполняются *ИНАЧЕ_ОПЕРАТОРЫ* и управление передается оператору, стоящему после

End Select

Например. Написать часть программы для алгоритма на рис. 3.5, определяющего значение переменной S в зависимости от значения переменной n.

```
Dim n As Integer Dim s As String n=1997
```

```
Select Case n Case 1960 s = "Год рождения"
```

```
Case 1961 to 1966 s =
```

```
"Дошкольные годы" Case 1967 to 1977, 1981 to 1985
```

```
s = "Годы учёбы"
```

```
Case Else
```

```
s = "Годы работы"
```

```
End Select
```

```
' Результат s = "Годы работы"
```

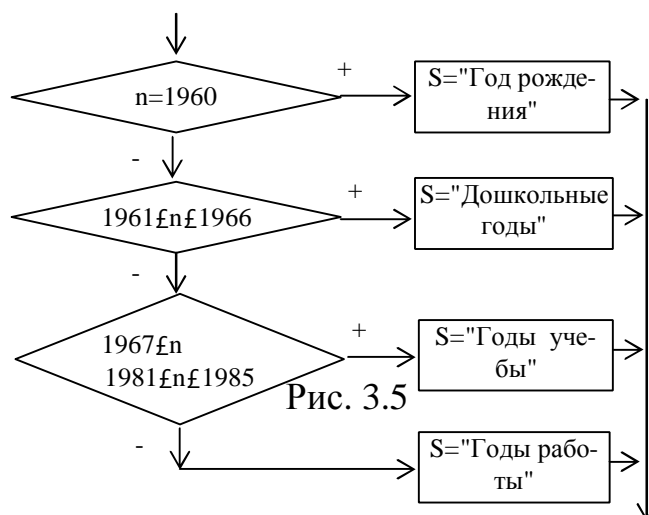


Рис. 3.5

3.6. Операторы цикла

Для реализации циклического вычислительного процесса, т. е. многократного выполнения одного или нескольких операторов, служит оператор цикла **For...Next**, который имеет следующий синтаксис:

```

For СЧЕТЧИК=НАЧ_ЗНАЧЕНИЕ To КОН_ЗНАЧЕНИЕ Step ШАГ
    БЛОК_ОПЕРАТОРОВ
[Exit For]
    БЛОК_ОПЕРАТОРОВ
Next СЧЕТЧИК
  
```

Цикл **For...Next** перебирает значения переменной *СЧЕТЧИК*, которая является параметром цикла, от начального до конечного значения с указанным шагом изменения. При этом обеспечивается выполнение блока операторов тела цикла при каждом новом значении счетчика. Если **Step ШАГ** в конструкции отсутствует, то по умолчанию считается, что шаг равен 1. По оператору **Exit For** можно выйти из оператора цикла до того, как *СЧЕТЧИК* достигнет последнего значения.¹

¹ **Примечание.** Не рекомендуется принудительно изменять значения параметра цикла, его начального и конечного значения в теле цикла **For...Next**.

Для перебора объектов из группы подобных объектов, например, ячеек из диапазона или элементов массива, удобно использовать оператор цикла **For... Each...Next**.

For Each ЭЛЕМЕНТ **In** ГРУППА

БЛОК_ОПЕРАТОРОВ

[Exit For]

БЛОК_ОПЕРАТОРОВ

Next ЭЛЕМЕНТ

В VBA для организации циклов с неизвестным заранее числом повторений используются и другие операторы цикла:

циклы с предусловием –	Do While ... Loop,
	Do Until ... Loop;
	Do ... Loop
циклы с постусловием –	While,
	Do ... Loop Until.

Ниже приведен синтаксис этих операторов цикла:

' Цикл с предусловием Do While ... Loop

Do While УСЛОВИЕ

БЛОК_ОПЕРАТОРОВ

[Exit Do]

БЛОК_ОПЕРАТОРОВ

Loop

' Цикл с предусловием Do Until ... Loop

Do Until УСЛОВИЕ

БЛОК_ОПЕРАТОРОВ

[Exit Do]

БЛОК_ОПЕРАТОРОВ

Loop

' Цикл с постусловием **Do ... Loop While**

Do

БЛОК_ОПЕРАТОРОВ

[Exit Do]

БЛОК_ОПЕРАТОРОВ

Loop While УСЛОВИЕ

' Цикл с постусловием **Do ... Loop Until**

Do

БЛОК_ОПЕРАТОРОВ

[Exit For]

БЛОК_ОПЕРАТОРОВ

Loop Until УСЛОВИЕ

Оператор **Do While...Loop** обеспечивает многократное повторение блока операторов до тех пор, пока *УСЛОВИЕ* соблюдается, а оператор **Do Until...Loop** пока *УСЛОВИЕ* не соблюдается. Операторы **Do...Loop While**, **Do...Loop Until** отличаются от перечисленных выше операторов тем, что сначала блок операторов выполняется по крайней мере один раз, а потом проверяется *УСЛОВИЕ*.

Для избежания закливания в теле цикла должен быть хотя бы один оператор, который изменяет значения переменных, стоящих в *УСЛОВИИ*.

Оператор **Exit Do** обеспечивает досрочный выход из оператора цикла.

Пример 1. Составить программу для алгоритма

фрагмент на рис. 3.6.

$S = 0$

$P = 1$

For $i = 1$ to 5

$S = S + 2$

$P = P * S$

Next i

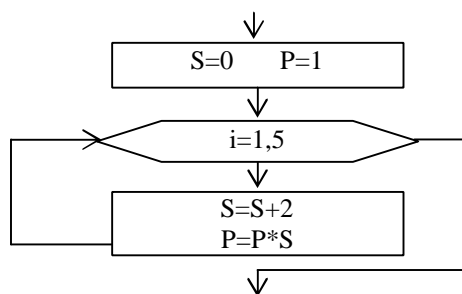


Рис. 3.6

Пример 2. Составить фрагмент программы для алгоритма на рис. 3.7.

$k = 0$ $f = 1$
 Do While $f \leq 15$ $k = k + 1$
 $f = f + 2$
 Loop

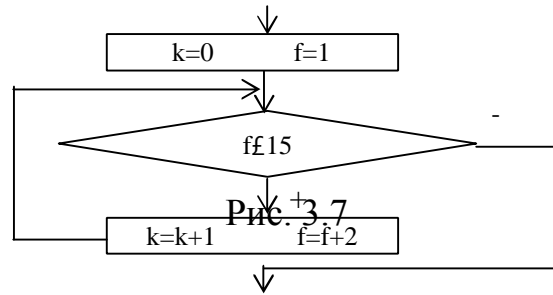


Рис. 3.7

Пример 3. Составить фрагмент программы, соответствующей алгоритму на рис.3.8. $D = 1$

$S = 0$
 Do
 $S = S + D$
 $D = D + 1$

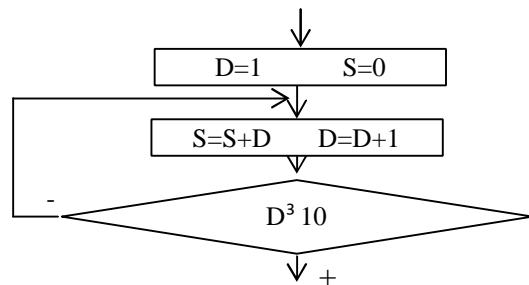


Рис. 3.8

Loop Until $D \geq 10$

4. СТРУКТУРА ПРОГРАММЫ.

11.4 МОДУЛИ, ПРОЦЕДУРЫ И ФУНКЦИИ

Модуль представляет собой текстовый ASCII-файл с программным кодом, содержащим подпрограммы, переменные и константы. Проект может состоять из множества программных модулей. Для их создания необходимо выполнить команду **Вставка** ® **Модуль**. Рабочее окно модуля представлено на рис. 4.1.

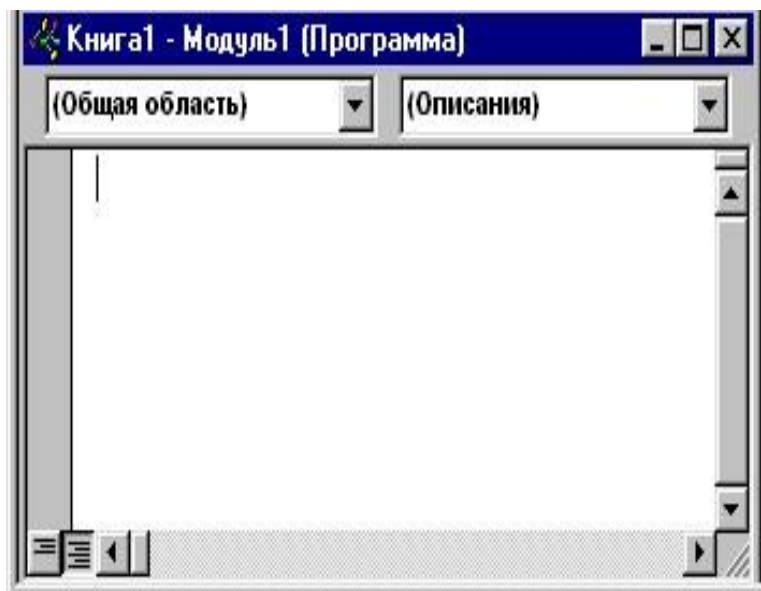


Рис. 4.1. Окно модуля

Основу программ в VBA составляют процедуры и функции.

Процедура Sub– это обособленная совокупность операторов VBA, выполняющая определенные действия. В общем случае процедура принимает некоторые параметры (переменные, которые передаются процедуре в качестве исходных данных), выполняет программу и может возвращать результирующие значения, которые присваиваются параметрам внутри процедуры. Однако чаще используются процедуры без параметров. Например, процедуры, выполняющиеся при возникновении определенных событий. Вложенность процедур в другие процедуры не допускается. Структура процедуры следующая: *[ДОСТУП] Sub ИМЯ_ПРОЦЕДУРЫ ([СПИСОК_ПАРАМЕТРОВ]) ТЕЛО_ПРОЦЕДУРЫ*
End Sub

Ключевое слово *ДОСТУП* является необязательным и определяет область видимости процедуры. **Public** указывает, что процедура доступна для всех других процедур во всех модулях (глобальная). **Private** указывает, что процедура доступна для других процедур только того модуля, в котором она описана (локальная). *СПИСОК_ПАРАМЕТРОВ* также является необязательным элементом и позволяет передавать процедуре различные исходные данные при вызове, которые называются формальными параметрами. При этом ключевое слово **Dim** не указывается. *ТЕЛО_ПРОЦЕДУРЫ* состоит из описательной части и блока операторов, выполняющихся один за другим. Если необходимо прекратить выполнение процедуры в некотором конкретном месте, это можно сделать с помощью оператора **Exit Sub**. *ИМЯ_ПРОЦЕДУРЫ* – это любой идентификатор, определенный пользователем. **Идентификатор** – это последовательность букв, цифр и символа подчеркивания, начинающаяся с буквы (пробелы внутри идентификатора недопустимы). Имя процедуры всегда определяется на уровне модуля. Для использования процедуры в тексте программы (т.е. для её вызова), необходимо указать имя процедуры и список фактических параметров, которые должны по типу и порядку следования совпадать с формальными параметрами.

Функция Function во многом похожа на процедуру, но в отличие от неё при вызове всегда возвращает значение. Функция получает параметры, называемые **аргументами**, и выполняет с ними некоторые действия, результат которых возвращается функцией. Структура функции следующая:

[ДОСТУП] Function ИМЯ_ФУНКЦИИ(СПИСОК_АРГУМЕНТОВ) As ТИП
ТЕЛО_ФУНКЦИИ
ИМЯ_ФУНКЦИИ = ВЫРАЖЕНИЕ

End Function

ТИП определяет тип данных возвращаемого результата. В теле функции обязательно должен присутствовать, по крайней мере, один оператор, присваивающий имени функции значение вычисляемого выражения. Досрочное завершение функции возможно с помощью оператора **Exit Function**. В программе вызов функции осуществляется с помощью оператора присваивания, в правой части которого указывается имя функции с перечнем фактических параметров, как и любой другой встроенной функции, например, Sqr, Cos или Chr.

Процедуры и функции, не описанные явно с помощью ключевых слов Public или Private, по умолчанию являются общими.

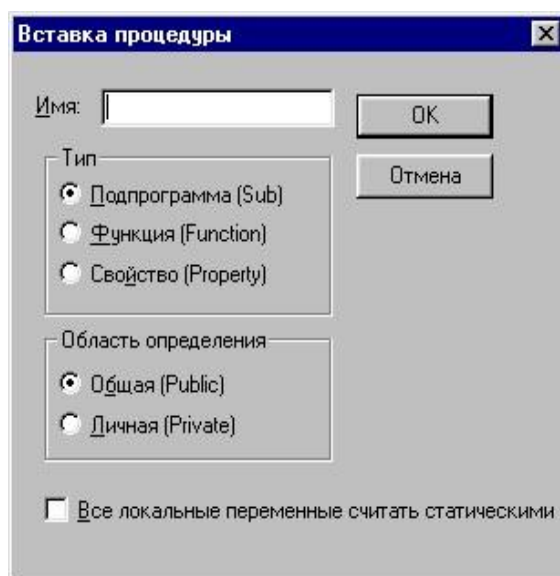
Для быстрого добавления в модуль подпрограмм удобно воспользоваться командой **Вставка** ® **Процедура**. В появившемся окне (рис. 4.2) нужно выбрать необходимые опции.

В MS Excel с функциями, созданными пользователем, можно работать с помощью Мастера функций точно так же, как и со встроенными функциями рабочего листа.

Пример. В MS Excel создать функцию пользователя, математически определенную следующим образом:

$$y = \sin(x) \cdot e^{-5x}$$

Создадим модуль, как указано выше (рис 4.2), и введем в него текст следующей программы:



**Рис. 4.2. Диалоговое окно “Вставка процедуры”
Public Function Y(x As Single) As Single**

$$Y = \text{Sin}(x) * \text{Exp}(- 5 * x)$$

End Function

Для использования созданной функции на рабочем листе MS Excel введем в ячейку A2 число 0.1. В ячейке B2 вычислим значение функции Y при $x = 0.1$. Для этого в ячейку B2 достаточно ввести формулу =Y(A2). Это можно сделать и с помощью Мастера функций, который будет содержать функцию Y наряду с другими встроенными функциями MS Excel (рис 4.3).

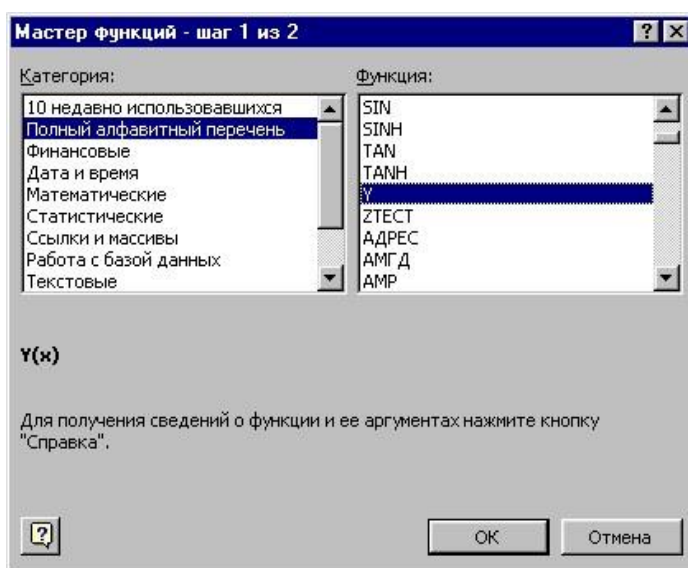


Рис. 4.3. Вызов функции Y()

Общие принципы организации программ VBA в модуле следующие. Обычно текст программы начинается с опций, которые управляют описанием переменных, способом сравнения строк и т. д.

Затем следует объявление **глобальных** для данного модуля переменных и констант, т.е. таких, которые используются во всех процедурах модуля.

Далее располагают непосредственно текст функций и процедур, составляющих саму программу.

Разделителем операторов в одной строке при записи программы является символ “:”.

Для переноса оператора на другую строку используется символ “_” (знак подчеркивания).

Иногда внутри программы удобно помещать **комментарии** – пояснительный текст, который игнорируется компилятором и может быть записан в любом месте программы. Комментарии удобно также использовать

при отладке программы для временного отключения операторов. Каждая строка комментариев начинается со знака апострофа.

Пример организации модуля:

Option Base 1

Option Explicit

' PI – глобальная константа

Const PI As Double = 3.14159

' x - глобальная переменная

Dim x As Double

' Функция Disc вычисляет площадь круга

Public Function Disc(R As Double) As Double

x = 2

Disc = PI * R ^ 2

End Function

' Функция Rec вычисляет площадь треугольника

Public Function Rec(a As Double, b As Double, c As Double) As Double

' p – локальная переменная **Dim p As Double**

p = (a + b + c) / 2

Rec = Sqr (p * (p - a) * (p - b) * (p - c))

End Function

' Процедура Result вызывает функции Disc и Rec и выводит результаты

Public Sub Result ()

' R_1, R_2, a, b, c - локальные переменные

Dim R_1 As Double

Dim R_2 As Double

Dim a As Double

Dim b As Double

Dim c As Double

R_1 = Disc(2. 5)

x = x + 2

```
MsgBox "Площадь круга = " & CStr(R_1) & ", x = " & CStr(x)
a = 1   b = 1   c = Sqr(2)
R_2 = Rec(a, b, c)
MsgBox "Площадь треугольника = " & CStr(R_2)
End Sub
```

Здесь инструкция **Option Explicit** указывает на необходимость описания типов переменных, используемых на данном листе модуля.

Инструкция **Option Base 1** указывает, что индексация элементов массива будет начинаться с 1.

Функция **Sqr** вычисляет квадратный корень аргумента.

Функция **CStr** переводит числовой формат в строковый.

11.5 5. СОЗДАНИЕ ФОРМ В VBA И ВКЛЮЧЕНИЕ ИХ В ПРОЕКТЫ

5.1. Создание форм. Свойства, события и методы форм

Форма – это главный объект, образующий визуальную основу приложения. По своей сути форма представляет собой окно, в котором можно размещать различные управляющие элементы при создании приложений. Для создания формы необходимо выполнить команду **Вставка** ® **UserForm**. В окне Конструктора форм появится форма, имеющая стандартный вид для ОС Windows (рис. 5.1).

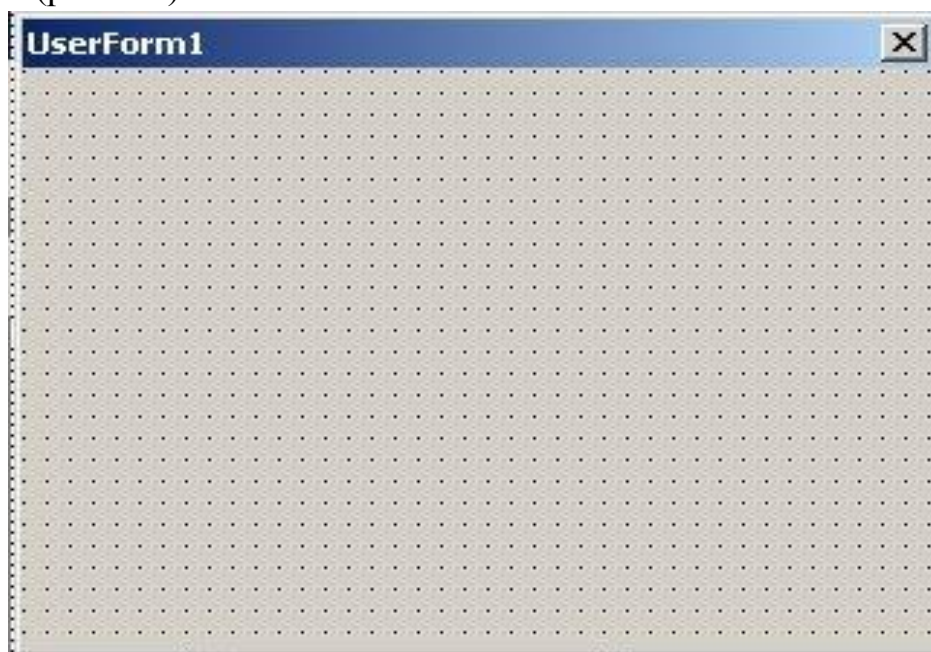


Рис. 5.1. Объект форма

Как и любой другой объект VBA форма имеет набор **свойств**, основные из которых приведены в таблице 5.1. Для получения справки по любому свойству достаточно выделить его в окне свойств и нажать F1.

Таблица 5.1. Основные свойства формы

Свойство	Описание
<i>BackColor</i>	Цвет фона для формы.
<i>BorderStyle</i>	Определяет тип границы, окружающей форму
<i>Caption</i>	Текст, который выводится в заголовке формы.
<i>Font</i>	Определяет тип и вид шрифта в форме.
<i>Height</i>	Определяет высоту формы в твипах.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Width</i>	Определяет ширину формы в твипах.

Свойства можно изменять в режиме конструирования в окне свойств, либо программно в режиме выполнения. Например, в ходе выполнения программы можно изменить заголовок формы командой:

```
frmForm1.Caption = "Привет"
```

Программы в ОС Windows управляются **событиями**. Каждый раз, когда нажимается кнопка, перемещается мышь, изменяются размеры формы и т.д., ОС генерирует сообщение. Сообщение доставляется соответствующему объекту, например форме, а та генерирует соответствующее событие. Следовательно, можно составить фрагмент программы, в котором объект будет реагировать на событие определенным образом, т.е. любому стандартному событию соответствует определенная процедура. Чтобы просмотреть события связанные с формой, необходимо в режиме конструирования дважды щелкнуть на ней – появится окно программы, в котором щелкнуть на списке Процедура. В табл. 5.2 приведены наиболее часто используемые события.

Таблица 5.2. Основные события форм

Событие	Описание
<i>Initialize</i>	Происходит во время конфигурации и до загрузки формы в память.
<i>Activate</i>	Происходит после загрузки формы в память.
<i>Deactivate</i>	Происходит, если форма перестает быть активной.
<i>Click</i>	Происходит при нажатии левой кнопки мыши на форме.

Следующий пример изменяет заголовок формы при активизации, и уменьшает размер формы после щелчка левой кнопкой мыши на форме.

```
Private Sub UserForm_Activate()
```

```
    frmForm1.Caption = "Щелчок на форме уменьшает её размеры"  
End Sub
```



```

Private Sub UserForm_Click()
frmForm1.Width = frmForm1.Width / 2
frmForm1.Height = frmForm1.Height / 2
frmForm1.Caption = "Сделай это еще раз!"
End Sub

```

Также форма обладает набором методов и инструкций. **Метод** определяет действие, которое может быть выполнено с объектом. **Инструкция** инициирует действие. Она может выполнить метод или функцию, В табл. 5.3 и 5.4 приведены наиболее часто используемые методы и инструкции для работы формами.

Таблица 5.3. Основные методы форм

Метод	Описание
<i>Hide</i>	Скрывает объект UserForm, но не выгружает его.
<i>Show</i>	Выводит на экран объект UserForm.

Таблица 5.4. Основные инструкции форм

Инструкция	Описание
<i>Load</i>	Загружает объект UserForm, но не отображает его на экране.
<i>Unload</i>	Удаляет объект UserForm из памяти.

В следующем примере предполагается, что в проекте созданы две формы frmForms. При запуске проекта происходит событие Initialize для формы frmForm1, форма frmForm2 загружается и выводится на экран. Когда при помощи мыши выбирается frmForm2, она делается невидимой, и появляется форма frmForm1. Если же выбирается frmForm1, frmForm2 появляется вновь. 'Событие Initialize формы frmForm1.

```

Private Sub UserForm_Initialize()

```

```

    Load frmForm2

```

```

    frmForm2.Show

```

```

End Sub

```

```

' Событие Click для формы frmForm2

```

```

Private Sub UserForm_Click()

```

```

    frmForm2.Hide

```

```

End Sub

```

```

' Событие Click для формы frmForm1

```

```

Private Sub UserForm_Click()

```

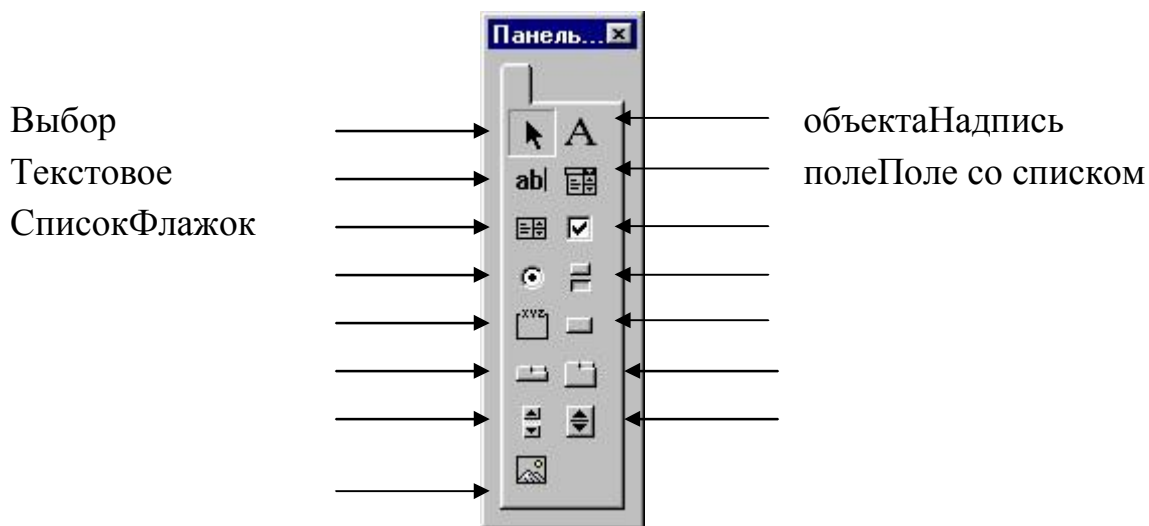
```

    frmForm2.Show End Sub

```

5.2. Выбор и использование управляющих элементов

Создание управляющих элементов на форме выполняется с помощью **Панели инструментов**, которая выводится на экран командой **Вид ® Панель элементов** (рис. 5.2).



Переключатель

Выключатель

Рамка

Командная кнопка

Набор вкладок

Набор страниц

Полоса прокрутки

Счетчик

Рисунок

Рис. 5.2. Панель элементов VBA

С помощью кнопок этой панели можно поместить в форму необходимый элемент управления. Для этого нужно щелкнуть на значке элемента управления, далее при нажатой левой кнопке мыши определить размер и место расположения элемента в форме. Когда элемент на форме выделен (рамка объекта содержит маленькие прямоугольники) можно изменять его размеры и перемещать с помощью мыши, а также просматривать и изменять его свойства в окне свойств.

Каждый управляющий элемент (объект) характеризуется набором свойств (которые можно изменять в режимах конструирования или выполнения), событий и методов.

Для каждого объекта проекта необходимо определить его имя. В соответствии с общепринятыми соглашениями об именах объектов первые три символа имени должны отражать вид элемента, а остальные символы - назначение. В табл. 5.5. представлены сочетания первых трех символов для наиболее часто используемых элементов.

Таблица 5.5. Рекомендуемые сочетания первых трех символов имен

Объект	Первые 3 символа имени	Пример имени
Форма	frm	frmMyForm
Надпись	lbl	lblInfo
Текстовое поле	txt	txtInput
Командная кнопка	cmd	cmdExit
Флажок	chk	chkSound
Переключатель	opt	optLevel
Список	lsb	lsbTypes
Рамка	fra	fraChoices
Полоса прокрутки	vcb	vcbSpeed
Рисунок	pic	picChema

Командная кнопка является самым распространенным элементом управления, и может использоваться для организации выполнения вычислений и других действий, вызова процедур и функций пользователя, открытия форм и т.д. Основные свойства командной кнопки представлены в табл. 5.6. В свойстве `Caption` можно ставить символ `&` перед буквой, которая будет использоваться в сочетании с клавишей `Alt` для ускоренного доступа к кнопке. Также можно перейти к кнопке клавишей `Tab`, а затем нажать `Enter`.

Таблица 5.6. Свойства командных кнопок

Свойство	Описание
<i>BackColor</i>	Цвет фона кнопки.
<i>Caption</i>	Текст, который выводится на кнопке.
<i>Enabled</i>	Значение <code>False</code> делает кнопку недоступной.
<i>Font</i>	Определяет тип и вид шрифта на кнопке.
<i>ForeColor</i>	Определяет цвет шрифта на кнопке.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Picture</i>	Добавляет рисунок на кнопку.
<i>PicturePosition</i>	Определяет расположение текста и рисунка на кнопке.
<i>Visible</i>	Значение <code>False</code> делает кнопку невидимой.

Основным **событием** кнопки является **Click**. Для написания программного кода, который будет выполняться при нажатии командной кнопки, достаточно два раза щелкнуть на ней левой кнопкой мыши в режиме конструирования проекта.

Наиболее полезным **методом** командной кнопки является **SetFocus**, позволяющий вернуться к кнопке (передать ей фокус). Например, следующая команда позволяет вернуться к кнопке по умолчанию после ввода данных в текстовое поле: **cmdMyButtum.SetFocus**

Текстовое поле применяется для ввода или вывода информации. Основные свойства текстового поля представлены в табл. 5.7.

Таблица 5.7. Свойства текстового поля

Свойство	Описание
<i>Enabled</i>	Значение False делает поле недоступным.
<i>Font</i>	Определяет тип и вид шрифта в текстовом поле.
<i>ForeColor</i>	Определяет цвет шрифта в текстовом поле.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>MaxLength</i>	Определяет количество вводимых символов в текстовое поле.
<i>PasswordChar</i>	Определяет символ, отображаемый при вводе в текстовое поле.
<i>Text</i>	Определяет содержимое текстового поля.

Например, для очистки содержимого текстового поля в ходе выполнения программы необходимо ввести в требуемом месте программного кода команду: **txtResult.Text=""**

Основным **событием** текстового поля является **Change**, происходящее при вводе или удалении символов. Например, команду **cmdMyButtum.SetFocus** можно поместить в процедуру события **Change** текстового поля.

Надпись применяется как самостоятельно для вывода справочной информации, так и в виде "подсказок" для текстового поля, списка или другого элемента. Главное её отличие от текстового поля в том, пользователь не может изменить текст надписи (хотя его можно изменить как свойство во время выполнения программы). Основные свойства надписи представлены в табл. 5.8.

Таблица 5.8. Свойства надписи

Свойство	Описание
<i>Caption</i>	Определяет текст, содержащийся в надписи.
<i>Font</i>	Определяет тип и вид шрифта надписи.
<i>ForeColor</i>	Определяет цвет шрифта надписи.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Picture</i>	Добавляет рисунок в надпись.

<i>PicturePosition</i>	Определяет расположение текста и рисунка надписи.
-------------------------------	---

Список позволяет работать с перечнем из нескольких вариантов. Пользователь может просмотреть содержимое списка и выбрать один из вариантов для последующей обработки. Прямое редактирование содержимого списка невозможно. Если в списке помещаются не все строки, то автоматически добавляется вертикальная полоса прокрутки. Основные свойства списка представлены в табл. 5.9.

Таблица 5.9. Свойства списка

Свойство	Описание
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>ListIndex</i>	Возвращает номер текущей выделенной строки списка -1.
<i>Text</i>	Содержимое текущей выделенной строки списка.

Для списка чаще всего используются **события Click** и **DblClick** (двойной щелчок левой кнопкой мыши на одной из строк списка). Во втором случае пользователь одновременно выделяет строку и начинает ее обработку.

Работа со списком начинается с его заполнения **методом AddItem**, который может вызываться несколько раз подряд. Часто метод AddItem помещается в процедуру **UserForm_Initialize()**, чтобы список заполнялся при загрузке формы. Метод **RemoveItem** удаляет строки из списка. Метод **Clear** очищает сразу весь список. Следующий пример показывает, как работают списки, при этом предполагается, что в проекте создана форма с двумя списками (List1 и List2). Двойной щелчок на любой строке одного списка перемещает её в другой список. Строка включается в другой список до того, как она будет удалена из текущего.

```
Private Sub UserForm_Initialize()
```

```
    List1.AddItem "Стол"
```

```
    List1.AddItem "Стул"
```

```
    List1.AddItem "Диван"
```

```
    List1.AddItem "Кресло"
```

```
    List1.AddItem "Кровать"
```

```
End Sub
```

```
Private Sub List1_DblClick()
```

```
    List2.AddItem List1.Text
```

```
    List1.RemoveItem
```

```
    List1.ListIndex
```

```
End Sub
```

```

Private Sub List2_dblClick()
  List1.AddItem List2.Text
  List2.RemoveItem
  List2.ListIndex
End Sub

```

Переключатели позволяют выбрать один вариант из группы. Обычно они группируются в рамках (см. далее), однако их можно располагать прямо на форме, если используется только одна группа переключателей. Основные свойства переключателя представлены в табл. 5.10.

Таблица 5.10. Свойства переключателя

Свойство	Описание
<i>Caption</i>	Задаёт текст, определяющий назначение переключателя.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Value</i>	Значение True указывает, что переключатель выбран.

Наиболее важным является свойство Value значение True (переключатель находится в установленном состоянии), которого в режиме конструирования задается только у одного переключателя в группе. В режиме выполнения это свойство чаще всего проверяется в процедуре события Click кнопки, нажатой после установки нужного переключателя, что позволяет проверить перед вызовом следующей процедуры некоторое условие. Однако определенные действия можно выполнять сразу же после выбора переключателя в процедуре его события **Click**.

Флажок частично аналогичен переключателю, но в отличие от него может использоваться как отдельный самостоятельный элемент. Даже объединенные в группу флажки работают независимо друг от друга. Основные свойства флажков такие же, как и у переключателя (см. табл. 5.10). Однако свойство Value может принимать три значения (флажок находится в установленном состоянии, снятом или неопределенном).

Наиболее часто используемым событием флажков является **Click**, в процедуре которого можно проверять состояние флажка по свойству Value. Следующий пример иллюстрирует работу флажков, при этом предполагается, что в проекте создана форма с двумя флажками (ChkBold и ChkItalic) и текстовым полем TxtExam (рис.5.3). После ввода символов в текстовое поле, с помощью флажков можно делать текст полужирным или курсивом. Свойства **FontBold** и **FontItalic** текстового поля устанавливаются способами начертания текста.

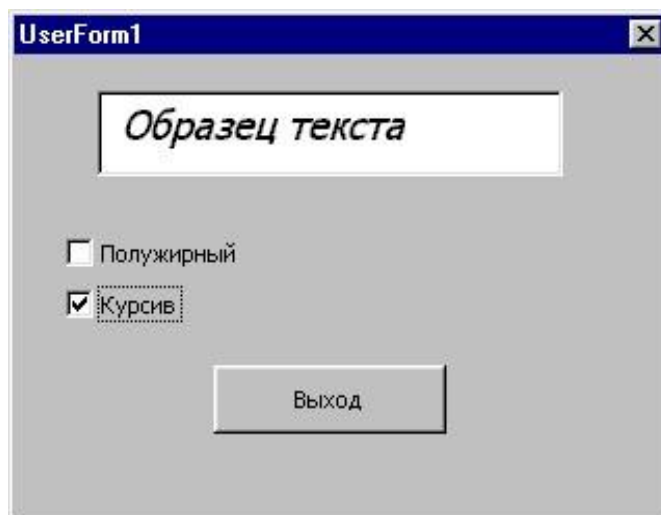


Рис. 5.3. Использование флажков

```
Private Sub Chkbold_Click()  
    If ChkBold.Value = True Then  
        TxtExam.FontBold = True  
    Else  
        TxtExam.FontBold = False  
    End If  
End Sub  
Private Sub ChkItalic_Click()  
    If ChkBold.Value = True Then  
        TxtExam.FontItalic = True  
    Else  
        TxtExam.FontItalic = False  
    End If  
End Sub
```

Рамка используется для группировки переключателей или флажков, и помещается на форму раньше элементов, находящихся внутри неё. Переключатели находящиеся внутри рамки, работают как самостоятельная группа и не влияют на состояние переключателей в других рамках. Основным свойством рамки является **Caption**, которое задает текст, определяющий назначение элементов в рамке.

Рисунок используется для простейшего вывода изображения на форме. Он может отображать растровые файлы (.BMP), значки (.ICO), метафайлы (WMF),

а также файлы в формате JPEG (.JPG) и GIF (.GIF). Основные свойства рисунка представлены в табл. 5.11.

Таблица 5.11. Свойства рисунка

Свойство	Описание
<i>Autosize</i>	Значение True подгоняет размер элемента под размер содержимого.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Picture</i>	Задаёт файл для рисунка.

События и методы рамок и рисунков практически не используются.

