ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
“КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ
ФИНАНСОВО-ЭКОНОМИЧЕСКИЙ ИНСТИТУТ”

Кафедра математики и экономической информатики

ТЕОРЕТИЧЕСКИЕ РАЗДЕЛЫ КУРСА “ИНФОРМАТИКА”

Учебное пособие для экономических специальностей вузов

Казань 2010

Учебное пособие подготовили преподаватели Казанского государственного финансово-экономического института: И.А. Кодолова – к.э.н., доцент, Н.З. Тартаковская – к.э.н., доцент (раздел 1), В.А. Дьяченко – к.ф-м.н., доцент (раздел 2), Ю.В. Степанова – к.с.н., доцент, Р.Р. Батаршина – ст. преп., Л.Г. Фатыхова (раздел 3). Под редакцией доцента И.А. Кодоловой.

Рецензенты: проф. Г.И. Кирилова, доцент С.Г. Свалова.

Батаршина Р.Р., Дьяченко В.А., Кодолова И.А., Степанова Ю.В., Тартаковская Н.З., Фатыхова Л.Г. Теоретические разделы курса “Инфоматика” / Под ред. И.А. Кодоловой: Учебное пособие для экономических специальностей вузов.– Казань: КГФЭИ, 2010.

В учебном пособии в соответствии с требованиями государственного образовательного стандарта высшего профессионального образования подробно рассматриваются отдельные теоретические разделы курса «Информатика»: основные понятия курса, основы программирования, проблемы информационной безопасности.

Главная цель учебного пособия – дать будущим специалистам-экономистам теоретические знания по курсу “Информатика”, рассмотреть проблемы информационной безопасности, овладеть основами программирования. Учебник может быть использован при самостоятельной работе студентов

Учебное пособие предназначено для студентов высших учебных заведений, бакалавров, специалистов, магистров и преподавателей экономических специальностей вузов.

ВВЕДЕНИЕ

Современный этап развития человеческого общества характеризуется переходом к всеобщей информатизации, внедрению компьютеризации и информационных технологий во все сферы деятельности и отрасли экономики. За прошедшие годы существенным образом изменилась техническая база, методология подготовки и решения задач с помощью компьютерных средств, но осталась актуальной задача подготовки высококвалифицированных экономических кадров, способных творчески решать управленческие задачи в новых экономических условиях.

Экономисты - профессионалы новой формации должны хорошо владеть базовыми теоретическими знаниями в области информатики, иметь практические навыки применения современной персональной вычислительной техники, знать перспективы развития информационных технологий, уметь оценивать информационные ресурсы для принятия оптимальных управленческих решений, понимать проблемы информационной безопасности, владеть основами программирования.

Учебное пособие предназначен для студентов высших учебных заведений, обучающихся по направлениям «Экономика» и “Менеджмент”. Учебное пособие охватывает основные разделы курса «Информатика» как комплексной научной дисциплины, занимающейся изучением информационных процессов, разработкой на этой основе информационной технологии, а также решением проблем эффективного использования информационных систем во всех сферах общественной деятельности.

Содержание учебника составляют три раздела.

В *первом разделе* “Базовые понятия курса “Информатика” изложены теоретические основы курса: даны основные определения, рассмотрены структура и свойства экономической информации, приведена классификация экономической информации, даны подходы к оценке экономической информации. В разделе рассмотрены программные и технические средства

реализации информационных процессов, способы представления информации в компьютерах, логические основы построения персональных компьютеров.

Во *втором разделе* “Основы алгоритмизации и программирования” изложены вопросы, связанные с теорией и практикой программирования. В разделе даны основные понятия алгоритмизации: свойства и формы представления алгоритма, базовые алгоритмические структуры, этапы развития программирования. В разделе изложены вопросы, связанные с основными понятиями языка программирования VBA (Visual Basic Application), рассмотрены основные элементы языка и приведен набор текстов программ.

Третий раздел “Основы информационной безопасности” посвящен вопросам информационной безопасности. В разделе изложены: объекты и элементы защиты информации в компьютерных системах обработки данных, угрозы безопасности информации. В разделе рассмотрены методы и средства защиты информации: криптографические методы защиты информации, системы электронной цифровой подписи. Второй раздел также содержит вопросы защиты от компьютерных вирусов и спама, основные вопросы защиты информации в корпоративных системах.

Материал учебного пособия имеет четкую структуризацию. Выделены основные определения, главы учебника иллюстрированы схемами и таблицами, в которых сгруппированы основные изучаемые теоретические положения курса, что организует процесс эффективного восприятия основных вопросов изучаемого предмета.

Для каждой главы приведены вопросы для самоконтроля по теме изучаемого предмета. В конце каждой главы приведены контрольные тесты, что позволяет студентам систематизировать проработанный материал и проверить, в целом, глубину его усвоения.

В учебном пособии представлен глоссарий основных понятий, использованных в курсе, обращение к которому будет способствовать более лучшему пониманию материала.

Раздел 1. БАЗОВЫЕ ПОНЯТИЯ КУРСА “ИНФОРМАТИКА”

Глава 1. Введение в экономическую информатику

1.1. Информационные процессы в экономике. Основные понятия информатики и информатизации

Бурное развитие компьютерной техники и информационных технологий послужило толчком к развитию общества, построенного на использовании различной информации и получившего название информационного общества. В информационном обществе процесс компьютеризации дает людям доступ к надежным источникам информации, обеспечивает высокий уровень автоматизации обработки информации в производственной и социальной сферах. Движущей силой развития становится производство информационного, а не материального продукта. Материальный продукт становится более информационноемким.

По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе производятся информация и знания. Материальной и технологической базой информационного общества являются различного рода системы на базе компьютерной техники и компьютерных сетей, информационной технологии, телекоммуникационной связи.

Информационное общество – общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формы – знаний.

Внедрение современных компьютеров, средств переработки и передачи информации в различные сферы деятельности послужило началом нового эволюционного процесса, называемого информатизацией.

Информатизация общества — это организованный, социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав

граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.[1]

Информатизация общества является одной из закономерностей современного социального прогресса. Этот термин все настойчивее вытесняет широко используемый до недавнего времени термин «компьютеризации общества». Эти понятия имеют существенные различия.

При *компьютеризации общества* основное внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение результатов переработки информации и ее накопление.

При *информатизации общества* основное внимание уделяется комплексу мер, направленных на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех видах человеческой деятельности.

Таким образом, «информатизация общества» является более широким понятием, чем «компьютеризация общества», и направлена на скорейшее овладение информацией для удовлетворения всех потребностей.

Одним из ключевых понятий при информатизации общества стало понятие «информационные ресурсы». Информационные ресурсы страны, региона относятся к стратегическим ресурсам, аналогичны по значимости материальным, природным и трудовым ресурсам.

Информационные ресурсы – отдельные документы и отдельные массивы документов, а так же документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных).

Таким образом, информационные ресурсы – это знания, подготовленные для использования и зафиксированные на материальном носителе.

Информационные ресурсы являются базой для создания информационных продуктов. Любой информационный продукт отражает информаци-

[1] Федеральный закон “Об информации, информатизации и защите информации” от 20 февраля 1995 г. № 24-ФЗ.

онную модель его производителя и воплощает его собственное представление о конкретной предметной области, для которой он создан. Информационный продукт, являясь результатом интеллектуальной деятельности человека должен быть зафиксирован на материальном носителе в виде документов, статей, программ, книг и т.д.

Информационный продукт – совокупность данных, сформированная производителем для распространения в вещественной или невещественной форме.

Научным фундаментом процесса информатизации общества является информатика, призванная создавать новые информационные технологии и системы для решения задач информатизации.

Информатика – комплексная научно-техническая дисциплина, занимающаяся изучением структуры, общих свойств информации и информационных процессов, разработкой на этой основе информационной техники и технологии, а также решением научных и инженерных проблем создания, внедрения и эффективного использования компьютерной техники и технологии во всех сферах человеческой деятельности.

Бурное развитие информатики в настоящее время обусловлено появлением микропроцессорной техники, созданием интегрированных сетей передачи данных, способных осуществлять в полном объеме все процедуры по преобразованию информации. В обществе ведется разработка концепций информационной политики и информационного обслуживания различных категорий пользователей; решаются проблемы, связанные с предоставлением информационных услуг в самых разных сферах научной, производственной, образовательной и других видов деятельности. Быстрыми темпами развивается глобальная информационная сеть Интернет, обеспечивающая широкие слои населения и группы пользователей разнообразной информацией.

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработ-

кой информации главным образом с помощью компьютеров и телекоммуникационных средств связи во всех сферах человеческой деятельности.

Информатику можно представить как науку, состоящую из трех взаимосвязанных частей – технических средств, программных средств, алгоритмических средств. В свою очередь, информатику, как в целом, так и каждую ее часть обычно рассматривают с разных позиций: как отрасль экономики, фундаментальную науку, прикладную дисциплину.

Информатика как *отрасль экономики* состоит из однородной совокупности предприятий, занимающихся производством компьютерной техники, программных продуктов и разработкой современной технологии переработки информации. Значение информатики как производства в том, что от нее во многом зависит рост производительности труда в других отраслях экономики. Более того, для нормального развития этих отраслей производительность труда в самой информатике должна возрастать более высокими темпами.

Информатика как *фундаментальная наука* занимается разработкой методологии создания информационного обеспечения процессов управления экономическими объектами на базе компьютерных информационных систем. Одна из главных задач информатики как фундаментальной науки – получение обобщенных знаний об информационных системах, выявление общих закономерностей их построения и функционирования.

Информатика как *прикладная дисциплина* занимается разработкой методов и средств преобразования информации. Главная функция информатики как прикладной науки заключается в проектировании и разработке информационных систем и технологий в конкретных областях.

Информатика является комплексной научно-технической дисциплиной, призванной создавать новые информационные технологии. Комплекс индустрии информатики становится ведущим в информационном обществе. Тенденция к все большей информированности в обществе в существенной степени зависит от прогресса информатики.

1.2. Информация и данные

Неизбежность информатизации общества обусловлена резким возрастанием роли и значения информации. В современном обществе для нормального функционирования экономики не достаточно традиционных ресурсов – материальных, трудовых, природных, финансовых и других. Важнейшим стратегическим ресурсом общества становится информация.

Термин ***«информация»*** произошел от латинского слова "*information"*, что означает разъяснение какого-либо факта, события, явления.

В широком смысле, ***информация*** определяется как обмен сведениями между людьми. С точки зрения кибернетики, ***информация*** – это мера устранения неопределенности.

Информатика рассматривает информацию как концептуально связанные между собой сведения, данные, понятия, изменяющие наши представления о явлении или объекте окружающего мира.

Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают неполноту знаний о них, степень неопределенности.

В строго научном плане, понятие "***информация***" связывается с *вероятностью* осуществления того или другого события. Чем выше вероятность исхода (результата) конкретного события, тем меньшее количество информации появляется после его осуществления, и, наоборот.

Следовательно, ***информация*** - это мера устранения неопределенности в отношении исхода интересующего нас события. Причем характерным является то обстоятельство, что информативность сообщения (количество информации в нем) не всегда пропорциональна объему (длине) этого сообщения.

Информация не существует сама по себе, она подразумевает наличие объекта (источника), отражающего (воспроизводящего) информацию, и субъекта (приемника, потребителя), воспринимающего ее.

Информация может существовать в самых разнообразных формах, например, в виде текстов, рисунков, фотографий, электрических импульсов, магнитных записей и т.д.

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются ***информационными объектами***.

Процессы, связанные с определенными операциями над информационными объектами, называются ***информационными процессами***.

Наряду с информацией в информатике часто употребляется понятие ***данные***. Термин ***«данные»*** произошел от латинского слова "*data"*, что означает факт.

Данные – это факты, которые выступают в качестве средства представления информации и обеспечивают возможность хранения, передачи и обработки информации.

Другими словами, ***данные*** служат исходным "сырьем" для получения информации, т.е. одни и те же данные могут нести различную информацию для разных потребителей.

Данные могут фиксироваться на конкретном физическом носителе и могут обрабатываться различными техническими средствами. Человек извлекает информацию из данных и оценивает ее, а затем принимает управленческое решение.

Следовательно, данные представляют собой факты или идеи, выраженные средствами формальной знаковой системы, обеспечивающей возможность их хранения, передачи и обработки. Такую формальную знаковую систему называют *языком представления данных*. Синтаксис этого языка характеризует способ представления информации, а его семантика — саму информацию.

1.3. Экономическая информация и ее свойства

Каждая область человеческой деятельности связана со "своей" информацией. Экономическая наука, деятельность общества в экономической сфере оперируют информацией, которая называется *экономической*.

Как категория экономическая информация, с одной стороны, соответствует общему понятию "информация", с другой — неразрывно связана с экономикой. Экономическая информация представляет собой лишь одну из разновидностей информации.

Экономическая информация - это информация, отражающая и обслуживающая процессы производства, распределения, обмена и потребления материальных благ. Она включает в себя сведения о материальных, трудовых и стоимостных аспектах процессов, воспроизводимых в экономике и устраняющих неопределенность в отношении исходов этих процессов.

Экономическая информация служит инструментом управления. Поэтому ее необходимо рассматривать как одну из разновидностей *управленческой информации*, которая обеспечивает решение задач организационно-экономического управления народным хозяйством.

Экономическая информация представляет собой совокупность сведений (данных), отражающих состояние народного хозяйства и определяющих направление развития его и отдельных звеньев.

В информационных процессах, осуществляемых в управлении, экономическая информация играет роль предмета труда (исходная, "сырая" информация) и продукта труда (результатная, "обработанная" информация). Говоря о понятии "***экономическая информация***" с кибернетических позиций, информационный процесс управления можно охарактеризовать как превращение сведений (исходных данных) в экономическую информацию, необходимую для принятия решений, направленных на обеспечение заданного состояния народного хозяйства и его оптимального развития.

Экономической информации свойственны некоторые особенности, обусловленные ее сущностью. Принципиальное значение для создания систем обработки экономической информации и формирования информационных технологий имеют следующие ее ***свойства***:

- преобладание алфавитно-цифровых знаков;
- необходимость оформления результатов обработки данных в форме, удобной для восприятия человеком;
- широкое распространение документов как носителей исходных данных и результатов их обработки;
- значительный объем переменных и постоянных (условно-постоянных) данных;
- дискретность - экономическая информация характеризует состояние объекта или процесса либо на определенный момент времени, либо за определенный интервал времени;
- организованность - экономическая информация отражает результат интеллектуальной деятельности человека;
- неоднородность - необходимость различать элементы и свойства отражаемых процессов;
- рассредоточенность источников и принципиальная невозможность концентрации и централизации процессов сбора данных;
- сохраняемость (неиссякаемость) при ее использовании (потреблении);
- возможность многократного применения одних и тех же данных, в том числе разными потребителями одновременно;
- возможность сохранения передаваемой информации у отправителя;
- возможность длительного хранения с воспроизведением и обновлением;

- способность к преобразованию, агрегированию по определенным признакам, детализации (разукрупнению) и сжатию (укрупнению);
- определенная самостоятельность данных по отношению к своему носителю.

В условиях выполнения функций управления теми или иными объектами, экономическая информация должна отвечать определенным ***требованиям***. Наиболее существенные из них:

- достоверность и полнота;
- ценность и актуальность;
- ясность и понятность;
- документальность: юридически подтвержденная в документах подписями (визами) соответствующих должностных лиц.

Информация *достоверна*, если она не искажает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильного решения. Информация *полна*, если ее достаточно для понимания и принятия решений. Неполнота информации сдерживает принятие решений или может повлечь за собой ошибки.

Ценность информации зависит от того, какие задачи решаются с ее помощью. *Актуальную* информацию, т.е. соответствующую современному моменту, важно иметь при работе в постоянно изменяющихся условиях.

Если ценная и актуальная информация выражена непонятными словами, она может стать бесполезной. Информация становится *ясной и понятной*, если она передана языком, на котором говорят те, кому предназначена эта информация.

1.4. Классификация экономической информации

Экономическая информация насчитывает много разновидностей (типов), которые выделяются на основе *классификационных признаков*.

Экономическая информация подразделяется по ***принадлежности к сфере материального производства и непроизводственной сфере***, а внутри - по отраслям и подотраслям народного хозяйства.

Подразделяют экономическую информацию по ***стадиям воспроизводства и элементам производственного процесса***. В силу этого различается экономическая информация, отражающая снабжение, производство, распределение и потребление, а также материальные, трудовые и финансовые ресурсы.

По ***функциям управления*** экономическая информация может подразделяться на следующие группы: плановую, нормативно-справочную, учетную, оперативную (текущую).

Плановая информация – информация о параметрах объекта управления на будущий период. На эту информацию ориентируется вся деятельность предприятия.

Нормативно-справочная информация содержит различные нормативы и справочные данные, ее обновление происходит достаточно редко.

Учетная информация – это информация, которая характеризует деятельность предприятия за определенный прошлый период времени. На практике, в качестве учетной информации может выступать информация бухгалтерского учета, статистическая информация и информация оперативного учета.

Оперативная (текущая) информация – это информация, используемая в оперативном управлении и характеризующая производственные процессы в текущий (данный) период времени.

По ***месту возникновения*** экономическую информацию можно разделить на входную, выходную, внутреннюю и внешнюю.

Входная информация – это информация, поступающая на предприятие или ее подразделение из другой организации.

Выходная информация – это информация, поступающая с предприятия в другую организацию (подразделение).

Внутренняя информация возникает внутри объекта, а *внешняя*– за пределами объекта.

По полноте отражения событий экономическая информация бывает — достаточная (полная), недостаточная и избыточная. Для решения задач экономического управления необходима конкретная минимальная информация, т.е. *достаточная*. *Избыточная* информация содержит излишние данные, которые либо вообще не используются для решения задач, либо выполняют контрольно-дублирующие функции;

По ***стадиям обработки*** экономическая информация может быть первичной, вторичной, промежуточной, результатной.

Первичная информация – это информация, возникает непосредственно в процессе деятельности объекта и регистрируется на начальной стадии.

Вторичная информация – это информация, которая получается в результате обработки первичной и может быть промежуточной или результатной.

Промежуточная информация используется в качестве исходных данных для последующих расчетов.

Результатная информация получается в процессе обработки первичной и промежуточной информации и используется для выработки управленческих решений.

По ***способу отображения*** экономическая информация подразделяется на текстовую и графическую.

Текстовая информация – это совокупность алфавитных, цифровых и специальных символов, с помощью которых информация представляется на физическом носителе.

Графическая информация – это различного рода графики, диаграммы, схемы, рисунки и т.д.

По ***стабильности*** информация может быть переменной и постоянной (условно-постоянной).

Переменная информация отражает фактические количественные и качественные характеристики производственно-хозяйственной деятельности предприятия. Она может меняться для каждого случая как по назначению, так и по количеству.

Постоянная (условно-постоянная) информация – это неизменная и многократно используемая в течение длительного периода времени информация.

Постоянная информация может быть справочной, нормативной, плановой:

– постоянная справочная информация включает описание постоянных свойств объекта в виде устойчивых длительное время признаков;

– постоянная нормативная информация содержит местные, отраслевые и общегосударственные нормативы;

– постоянная плановая информация содержит многократно используемые на предприятии плановые показатели.

1.5. Структура экономической информации

Важной характеристикой экономической информации является ее *структура*. В структуре информации различают два взаимосвязанных аспекта:

– состав элементов, образующих структуру информации;

– взаимосвязь элементов структуры.

Рассматривая с этих позиций структуру информации, выделяют простые и сложные единицы информации.

В ***логическом подходе*** к структуре экономической информации выделяют следующие единицы измерения:

– реквизит;

– показатель;

– экономический документ;

– информационный массив;

– информационный поток;

– информационная система.

Простой, элементарной составляющей единицей информации является реквизит.

Реквизит – это минимальная структурная единица информации, описывающая определенное свойство объекта, процесса, явления. Реквизит нельзя разделить на более мелкие составные элементы. Синонимом слова «реквизит» является слово «атрибут». Каждый реквизит характеризуется именем (названием), типом и значением.

В зависимости от характера отображаемого свойства реквизиты делят на реквизиты-признаки и реквизиты-основания.

Реквизиты-признаки отражают качественные свойства экономического объекта. Реквизиты-признаки служат для логической обработки составных единиц, т.е. для поиска, сортировки, группировки, выборки и т.д.

Реквизиты-основания характеризуют количественную сторону процесса или явления. Реквизиты-основания выражаются в цифровой форме и используются для выполнения арифметических операций.

Отдельно взятый реквизит не может полностью характеризовать экономический процесс. Для исчерпывающей его характеристики необходима определенная совокупность реквизитов, описывающих качественные и количественные свойства отображаемого объекта. Совокупность реквизитов-признаков и реквизитов-оснований представляет собой *сообщение* об объекте. Каждое сообщение имеет определенную форму.

Показатель представляет собой составную единицу экономической информации, включает один реквизит-основание и группу взаимосвязанных с ним реквизитов-признаков.

Экономический показатель – основная единица экономической информации, так как она имеет экономический смысл. Структурно экономическую информацию можно рассматривать как совокупность показателей. Вместе с тем в целях обработки информации и реализации функций управления показатели могут образовывать более сложные составные структурные единицы информации: документы, массивы, информационные потоки, информационную базу.

Экономический документ – это организованная совокупность взаимосвязанных по смыслу экономических показателей. Экономический документ является основной и наиболее удобной формой представления информации, отличается наглядностью и обеспечивает юридический статус информации. Наиболее распространенной формой представления экономических документов является табличная форма.

Информационный массив представляет собой определенным образом организованную совокупность взаимосвязанных по смыслу экономических документов.

Информационный массив с позиции логической структуры представляет собой набор данных (документов) одной формы (одного названия), относящихся к одной задаче. В системах обработки информации массив (файл) является основной структурной единицей, предназначенной для хранения, передачи и обработки информации.

Информационные массивы могут объединяться в более крупные структурные единицы – информационный поток и информационную систему.

Информационный поток – это совокупность информационных массивов, относящихся к конкретной управленческой деятельности, имеющих динамический характер.

Информационная система – вся совокупность информационных потоков, относящихся к одному экономическому объекту и характеризующая управленческую работу в целом.

Рассмотренные структурные единицы экономической информации отражают их логическое построение без учета особенностей представления данных на технических носителях.

При организации автоматизированной обработки экономической информации понятие структуры данных связано с представлением ее на различных носителях, таким образом структурные единицы информации выделяются в зависимости от носителя и способов фиксации данных на нем. Это составляет основу физического подхода к рассмотрению структур информации.

Физический подход к структуре обусловлен автоматизированной обработкой экономической информации. Все структурные единицы информации обрабатываются с помощью технических средств. Обрабатываемая информация измеряется в технических единицах: байт и Кбайт.

Ведущей структурной единицей информации при автоматизированной обработке информации является файл.

Файл – это именованная область внешней памяти, выделенная для хранения массива данных.

Во внутренней структуре файла выделяют более простые единицы: запись, поле, символ.

Символ – это минимальный элемент в структуре файла (буква, цифра, знак). Символ не несет смысловой нагрузки.

Символы объединяются в ***поля***, образующие минимальные смысловые элементы.

Запись – это совокупность полей, обеспечивающих характеристику отдельных объектов.

1.6. Оценка экономической информации

При оценке экономической информации используются различные подходы: *синтаксический, семантический, прагматический и структурный*. Для оценки информации используют различные параметры: количество информации, объем данных и качество информации, рис.1.1.

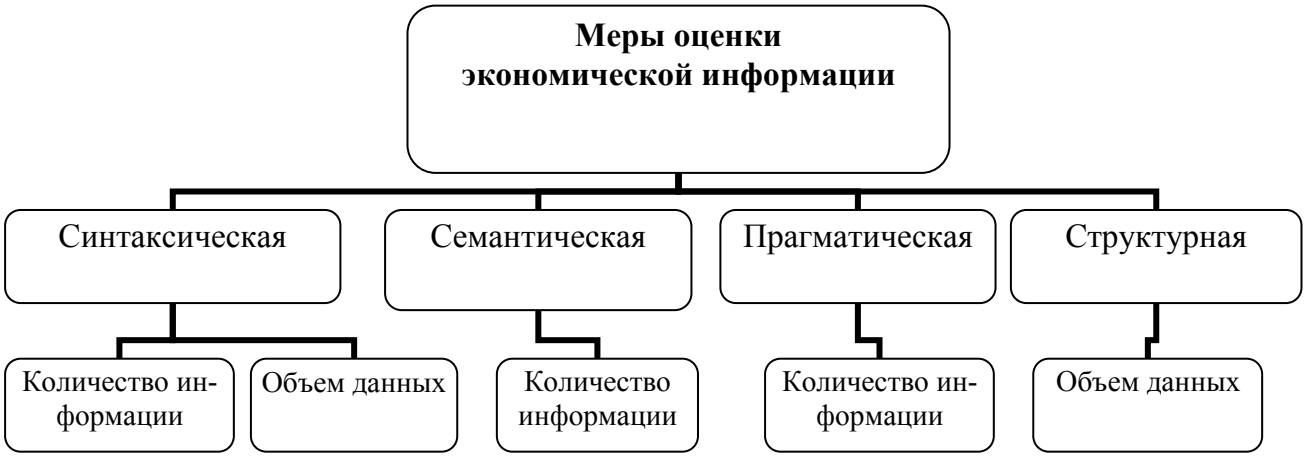


Рис. 1.1. **Меры оценки экономической информации**

Синтаксический подход к оценке экономической информации связан со способом представления информации, способом передачи и хранения. При синтаксическом подходе не рассматривается смысловое содержание информации.

Экономическую информацию, рассматриваемую с точки зрения синтаксического подхода называют ***данными***, так как при этом смысловая сторона информации не имеет значения.

Синтаксический подход к оценке информации нередко называется статистическим, как его назвал Клод Шеннон, опубликовавший в 1948 г. книгу по математической теории связи.

Для измерения количества информации в синтаксическом подходе используют ***энтропийный подход***.

К. Шеннон ввел понятие "количество информации" как меры неопределенности состояния системы, снимаемой при получении информации.

Энтропия – это количественно выраженная неопределенность состояния системы, которая уменьшается при получении информации. Очевидно, что чем больше информации получает наблюдатель, тем больше снижается неопределенность и энтропия системы сокращается.

Таким образом, *количество информации при синтаксическом подходе* измеряется уменьшением (изменением) неопределенности состояния системы.

Тогда информацию можно трактовать как меру уменьшения неопределенности при совершении какого-либо события, и чем менее вероятно сообщение, тем больше информации оно содержит.

Количество информации, поступающей от источника к получателю оценивается как разность энтропий:

$$I(x)=H_1(x)-H_2(x), \qquad (1.1)$$

где $I(x)$ – количество полученной информации после получения некоторого сообщения;

$H_1(x)$ – энтропия системы X до получения сообщения;

$H_2(x)$ – энтропия системы X после получения сообщения.

Если система X обладает дискретным состоянием, то есть переходит из состояния в состояние скачком, то количество возможных сообщений равно $\boldsymbol{n}$, а вероятность нахождения системы в каждом из состояний равна p_i , где $i = 1, 2, \ldots, n$, то согласно теореме Шеннона энтропия системы $H(x)$ равна:

$$H(x) = - \sum p_i \ log_2 p_i \qquad (1.2)$$

Поскольку обработка и обмен информацией в вычислительных машинах осуществляется в двоичной системе счисления, то за основание логарифма принято 2, и количество информации измеряется в двоичных единицах или битах (двоичный разряд).

Единица количества информации – ***бит*** – это такое количество информации, которое содержит сообщение, уменьшающее неопределенность знаний в 2 раза. Бит является наименьшей единицей измерения информации.

Семантический подход к оценке экономической информации предполагает учет *смыслового содержания информации*.

При семантическом подходе к измерению смыслового содержания информации используется *тезаурусная мера*, предложенная российским ученым Ю. Шнедером.

Тезаурус – это систематизированная совокупность сведений и знаний, с указанием смысловых связей между ними, которыми располагает пользователь или система.

Тезаурусная мера связывает семантические свойства информации со способностью пользователя принимать и понимать поступившее сообщение. Так, для понимания и использования полученной информации получатель должен обладать определенным запасом знаний. Если получатель информации не понимает принятое сообщение, то количество воспринимаемой им информации равняется нулю, и наоборот, если пользователь информации знает абсолютно все о предмете, то сообщение не даст ему ничего нового и количество информации также будет равно нулю.

Следовательно, под *семантической (смысловой) ценностью* информации понимается *мера расширения, развития тезариуса* воспринимаемой стороной при приеме и интерпретации сообщения.

Тезаурусный метод подтверждает факт, что информация обладает свойством относительности. Количество семантической информации, то есть количество новых знаний, получаемых пользователем является величиной относительной.

Прагматический подход к оценке экономической информации связан с определением *ценности, полезности* использования информации при выработке потребителем решения для достижения своей цели.

Прагматическая ценность информации определяет ее полезность для достижения поставленной цели. Эта характеристика информации достаточно условна поскольку определяется способностями использования данных в конкретной системе. При этом рассматриваются такие свойства информации, как достаточность, актуальность, доступность, своевременность, достоверность, точность и др.

Прагматический подход анализирует потребительские свойства информации, соответствие информации цели управления. При оценке *количества информации в прагматическом аспекте* учитывают временную зависимость информации от момента принятия решения. Так как в экономических системах управления ценность информации со временем может настолько понизиться, что информация будет совершенно бесполезной для принятия решения.

В экономических системах управления прагматический подход к оценке экономической информации является наиболее важным, так как при этом анализируется полезность информации с точки зрения реализации процессов управления. Семантический и синтаксический подходы к оценке информации имеют подчиненное значение.

При ***структурном подходе*** происходит абстрагирование от содержательности и ценности информации, основной упор делается на количественные характеристики ее составляющих *информационных единиц*.

Существует несколько подходов к структуризации информации. Так, с позиции экономического содержания ведущее место занимает иерархический подход, который предполагает многоуровневое построение информационных единиц исходя из организации управленческого процесса. Из простых информационных единиц образуются сложные, составные.

С точки зрения этого подхода структурными единицами экономической информации являются: реквизиты, показатели, массивы, информационная база. Элементарными неделимыми единицами экономической инфор-

мации являются реквизиты, отражающие определенные свойства объекта или процесса.

Таким образом, *количество информации при структурном подходе* зависит от правильно выбранной структуры информации.

Важной, стороной *оценки информации* является определение ее качества.

Качество информации - это совокупность свойств экономической информации, характеризующих степень ее соответствия потребностям пользователей.

Качество информации, особенно при принятии управленческих решений, обусловливается целым рядом свойств, таких как содержательность, репрезентативность, доступность, достаточность, актуальность, своевременность, точность, достоверность и др.

Содержательность информации отражает ее семантическую емкость, выражаемую в отношении количества семантической информации в сообщении к объему обрабатываемых данных. Чем выше это отношение, тем больше пропускная способность информационной системы, т.е. для получения одних и тех же сведений нужно обрабатывать меньший объем данных.

Репрезентативность информации заключается в выборе механизма ее отбора и формирования в целях адекватного отражения свойств объекта.

Доступность информации проявляется в понимании ее пользователем.

Достаточность информации означает, что она содержит тот набор показателей, который достаточен для принятия управленческого решения.

Актуальность информации оценивается степенью сохранения ее ценности для управления в момент использования.

Своевременность информации означает ее поступление не позже назначенного времени совершения события (решения задачи).

Точность информации определяется степенью близости получаемой информации к реальному состоянию объекта или протекающего процесса.

Достоверность - это свойство отражения реально существующих объектов в пределах необходимой точности.

Сочетание вышеперечисленных основных параметров позволяет говорить о ценности, надежности и эффективности информационных систем экономических и прочих объектов управления.

Вопросы для самоконтроля

1. Что является объектом изучения информатики как научного направления?
2. Чем вызвано появление и развитие информатики?
3. Назовите основные черты информационного общества.
4. В чем состоят принципиальные различия между информацией и данными?
5. Какие особенности присущи экономической информации?
6. Назовите основные признаки классификации экономической информации.
7. В каких аспектах рассматривается экономическая информация?
8. Что понимается под логической структурой экономической ин формации?
9. В чем состоит принципиальное различие между реквизитом-признаком и реквизитом-основанием?
10. Чем измеряется количество информации при синтаксическом подходе к ее оценке?
11. Какие свойства информации оценивают ее качество?

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	Информацию, существенную и важную в настоящий момент, называют…	• объективной • актуальной • полезной • достоверной
2.	Информация достоверна, если она…	• достаточна для принятия решений • отражает истинное положение дел

		• полезна • используется в современной системе обработки информации
3.	Семантический аспект информации определяет…	• определяет синтаксическое соотношение ее элементов • информацию с точки зрения ее актуальности для получателя • информацию с точки зрения ее практической полезности для получателя • смысловое соотношение ее элементов
4.	Прагматический аспект – это характеристика информации с точки зрения ее…	• структуры • количества • полезности • смысла
5.	В теории информации под информацией понимают	• сигналы от органов чувств человека • повтор ранее принятых сообщений • сведения, устраняющие или уменьшающие неопределенность • характеристику объекта, выраженную в числовых величинах
6.	Такое свойство информации, как _____ характеризует возможность ее получения.	• объективность • полезность • актуальность • доступность
7.	Свойство информации, которое характеризует степень ее соответствия реальности,–это…	• важность • адекватность • содержательность • надежность
8.	К свойствам информации относятся: а) полнота б) цикличность в) выразительность г) достоверность д) актуальность е) направленность	• б), в), е) • в), д), е) • а), г), д) • а), б), в)
9.	Энтропия как мера информации максимальна, если …	• события неравновероятны • события равновероятны • события детерминированы • информация точна
10.	Выберите правильное определение понятия «Информация»	• Информация – изложение, разъяснение какого – либо факта, события. • Информация – совокупность всех сведений об объекте. • Информация – мера устранения неопределенности об объекте или процессе.
11.	Выберите правильное определение понятия «экономическая информация» (ЭИ):	• ЭИ – разновидность управленческой информации. • ЭИ – совокупность сведений, используемая для управления народным хозяйством и его отраслями. • ЭИ – совокупность всех сведений об объекте или процессе.

12.	ЭИ по функциям управления распадается на:	• Достаточную • Аналитическую • Прогнозную • О средствах труда • Плановую • Учетную
13.	По признаку полноты в ЭИ выделяют:	• Достаточную • Долгосрочную • Недостаточную ЭИ • Первичную • Избыточную
14.	По месту возникновения ЭИ выделяют:	• Первичную • Внутреннюю • Входную • Внешнюю • Внутренняя
15.	По способу отображения данных выделяют:	• Переменную • Алфавитно – цифровую • Графическую • Достоверную • Числовую • Текстовую
16.	По признаку стабильности выделяют:	• Учетная • Постоянную • Условно – постоянную • Недостоверную • Переменную
17.	Какая единица ЭИ является наименьшей при логическом подходе к структуре ЭИ?	• Запись • Массив • Реквизит • Показатель
18.	Данные – это …	• Мера устранения неопределенности в отношении исхода некоторого события • Вероятность выбора • Отрицание энтропии • Информация представленная в формализованном виде
19.	Атрибутивные свойства информации – это…	• свойства, характеризующие стабильность • информации во времени • свойства, характеризующие полезность • информации • свойства, без которых информация не существует • свойства, характеризующие изменение информации во времени
20.	Семантическая мера количества информации определяется	• ценностью использования информации • тезаурусом • степенью изменения неопределенности • состояния системы • степенью изменения определенности • состояния системы

Глава 2. Программные средства реализации информационных процессов

2.1. Назначение и классификация программного обеспечения

Под ***программным обеспечением*** (*Software*) понимается совокупность программ, выполняемых вычислительной системой и необходимых для эксплуатации технических средств.

Программное обеспечение можно классифицировать по различным признакам. По сфере использования программное обеспечение подразделяется на три класса программных продуктов, представленных на рис. 2.1:

- системное программное обеспечение;
- пакеты прикладных программ;
- инструментарий технологии программирования.

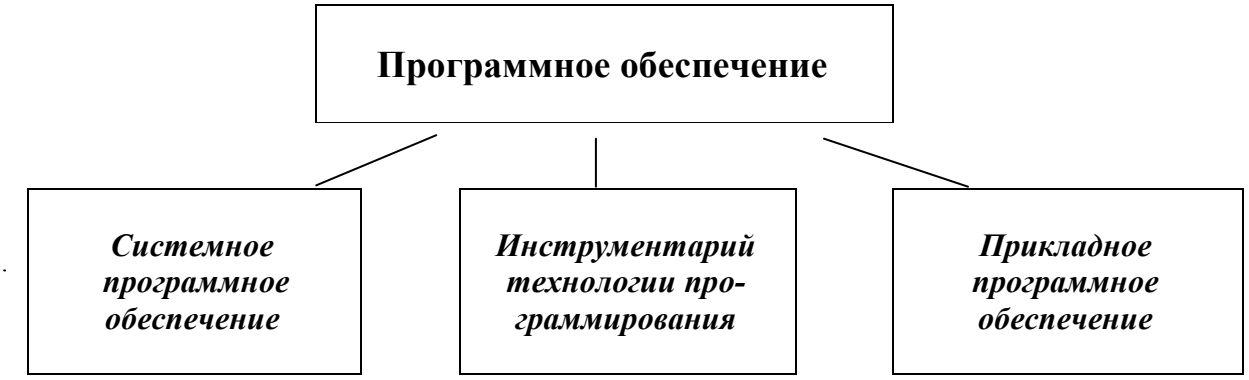


Рис. 2.1. **Классификация программного обеспечения**

Системное программное обеспечение (*System Sortware*) – это совокупность программ и программных комплексов для обеспечения работы компьютеров и вычислительной системы в целом.

Системное программное обеспечение направлено:

- на создание операционной среды функционирования других программ;

- на обеспечение надежной и эффективной работы самого компьютера и вычислительной сети;
- на проведение диагностики и профилактики аппаратуры компьютера и вычислительных сетей;
- на выполнение вспомогательных технологических процедур (копирование, архивирование, восстановление файлов программ и баз данных и т.д.).

Данный класс программных продуктов тесно связан с типом компьютера и является его неотъемлемой частью. Программные продукты данного класса носят общий характер применения, независимо от специфики предметной области. К ним предъявляются высокие требования по надежности и технологичности работы, удобству и эффективности использования.

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

Инструментарий технологии программирования обеспечивает процесс разработки программ и включает специализированные программные продукты, которые являются инструментальными средствами разработчика. Программные продукты данного класса поддерживают все технологические этапы процесса проектирования, программирования (кодирования), отладки и тестирования создаваемых программ. Пользователями технологии программирования являются системные и прикладные программисты.

Прикладное программное обеспечение (*Application Sortware*) предназначено для решения определенного класса задач конкретной предметной области.

Прикладное программное обеспечение служит программным инструментарием решения функциональных задач и являются самым многочисленным классом программных продуктов. В данный класс входят программные продукты, выполняющие обработку информации различных предметных областей.

2.2. Состав и назначение системного программного обеспечения

Системное программное обеспечение предназначено для обеспечения работоспособности компьютера, организации процесса поиска и обработки информации и предоставления пользователю удобных способов диалога с компьютером. К этому классу относятся: операционные системы, драйверы, операционные оболочки, утилиты.

Системное программное обеспечение можно классифицировать следующим образом – *базовое программное обеспечение*, которое, как правило, поставляется вместе с компьютером, и *сервисное программное обеспечение*, которое может быть приобретено дополнительно, рис. 2.2.

Рис. 2.2. **Классификация системного программного обеспечения компьютера**

Базовое программное обеспечение (*Base Software*) – это минимальный набор программных средств, обеспечивающих работу компьютера.

Сервисное программное обеспечение - это программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

2.2.1. Базовое программное обеспечение

В состав базового программного обеспечения входят:

- операционная система;
- операционные оболочки;
- сетевая операционная система.

В наборе базового программного обеспечения главное место занимает операционная система (*Operating system*).

Операционная система (ОС) – это совокупность программ, предназначенных для управления работой всех устройств персонального компьютера и управления процессом выполнения прикладных программ пользователя.

Операционная система выполняет такие *функции*, как:

- контроль работоспособности оборудования ПК;
- реализация процедуры начальной загрузки;
- управление работой устройств ПК;
- управление файловой системой;
- диалоговое взаимодействие пользователя с ПК;
- загрузка и выполнение прикладных программ;
- распределение ресурсов ПК (оперативной памяти, процессорного времени и периферийных устройств между прикладными программами) и др.

Операционные системы делятся на:

- одно- и многозадачные – в зависимости от числа выполняемых прикладных программ;

- одно- и многопользовательские – в зависимости от числа пользователей, работающих с операционной системой;

- несетевые и сетевые, обеспечивающие работу в локальной вычислительной сети.

К числу современных операционных систем прежде всего относятся: *Windows XP, Windows NT, Vista, LINUX, Unix, OS/2* и др.

Операционная система содержит специальные программы (драйверы), которые управляют работой стандартных внешних устройств компьютера.

Драйвер – это программа, расширяющая возможности операционной системы. Драйвер служит для управления работой периферийных устройств компьютера: дисководов, дисплеем, клавиатурой, принтером, манипулятором “мышь” и др.

Драйвер устройства должен учитывать специфику работы внешнего устройства, поэтому каждому устройству соответствует свой драйвер.

Драйверами также считаются программы, обеспечивающие управление расширенной памятью, а также создание и обслуживание виртуальных устройств.

Драйверы могут быть стандартными, либо загружаемыми.

Стандартные (внутренние) драйверы – это драйверы, которые входят в стандартный комплект поставки персонального компьютера, они подключаются к операционной системе автоматически.

Загружаемые (внешние) драйверы – это программы, которые хранятся на диске и предназначены для управления внешними устройствами, которые отличаются от стандартных устройств по своим техническим параметрам. Возможность использования внешних загружаемых драйверов облегчает адаптацию операционной системы к новым внешним устройствам.

Операционные оболочки – специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы.

Операционные оболочки существенно упрощают задание управляющей информации для выполнения команд операционной системы, уменьшают напряженность и сложность работы конечного пользователя.

Наиболее популярны следующие виды оболочек операционной системы Windows: *Total Commander, Norton Navigator* и др.

Сетевые операционные системы – это комплекс программ, обеспечивающий обработку, передачу и хранение данных в сети.

Сетевые операционные системы обеспечивают поддержку *сетевых функции*, таких как:

- совместное использование файлов и принтеров при высокой производительности;
- эффективное выполнения прикладных программ, ориентированных на архитектуру "*клиент-сервер*";
- дистанционный доступ к сети;
- работа на различных платформах и с различным сетевым оборудованием;
- интеграция с Интернетом, то есть поддержка соответствующих протоколов и программного обеспечения Web-сервера;
- организация внутренней электронной почты и телеконференций.

К числу сетевых операционных систем относятся: *Windows NT* (NT – *New Technology* – “новая технология”), *Windows XP, OS/2,* операционные системы семейства *Unix.*

2.2.2. Классификация операционных систем

Операционная система управляет всеми ресурсами компьютера и обеспечивает максимальную эффективность его функционирования. В соответствии с этим главной функцией операционной системы является распределение процессоров, памяти, других устройств и данных между вычислительными процессами.

Управление ресурсами включает решение следующих задач, не зависящих от вида ресурса: планирование ресурса, то есть выявление, кому, когда и в каком количестве необходимо выделить данный ресурс; контроль за состоянием ресурса, то есть поддержание оперативной информации о том, занят или не занят ресурс, какое количество ресурса уже распределено, а какое свободно.

Операционные системы различаются особенностями реализации алгоритмов управления ресурсами компьютера, областями использования и по другим признакам. Так, в зависимости от особенностей алгоритма управления процессором операционные системы подразделяются на одно- и многозадачные, одно- и многопользовательские, на одно- и многопроцессорные, а также на локальные и сетевые.

Одно- и многозадачные операционные системы. По числу одновременно выполняемых задач операционные системы делятся на два класса:

- однозадачные (например, *MS DOS, MSX*);
- многозадачные (*ОС ЕС, OS/2, Unix, Windows*) и др.

Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая интерфейс пользователя с компьютером более простым и удобным; включают средства управления периферийными устройствами, диалогового управления файлами, средства диалогового общения с пользователем.

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

Вытесняющая и невытесняющая многозадачность. Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одповременно протекающими в системе вычислительными процессами во многом определяет особенность ОС. Среди множества способов реализации многозадачности можно выделить две группы алгоритмов:

- вытесняющая многозадачность (*Windows NT/XP, OS/2, Unix*);
- невытесняющая многозадачность (*NetWare, Windows 3.x*).

Основным различием между вытесняющим и невытесняющим алгоритмами многозадачности является степень централизации планирования вычислительных процессов. В первом случае планирование этих процессов целиком возлагается на операционную систему, а во втором - распределено между операционной системой и прикладными программами.

При *невытесняющей* многозадачности активный вычислительный процесс выполняется до тех пор, пока прикладная программа по собственной инициативе не отдаст указание операционной системе выбрать из очереди другой процесс, готовый к выполнению.

При *вытесняющей* многозадачности решение о переключении процессора с одного активного вычислительного процесса на другой принимается ОС, а не прикладной программой.

В зависимости от областей использования многозадачные ОС подразделяются на три типа:

- пакетной обработки (например, *ОС ЕС*);
- с разделением времени (*Unix, VMS, Windows, Linux*);
- режима реального времени (*QNX, RT/11*).

Некоторые операционные системы могут совмещать 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 называют *фоновым режимом*.

Много- и однопользовательский режимы. По числу одновременно работающих пользователей ОС подразделяются на однопользовательские (*MS DOS, Windows 98*) и многопользовательские (*Unix, Windows NT, Windows XP*).

Главным отличием *многопользовательских* систем от однопользовательских является наличие средств защиты информации каждого пользова-

теля от несанкционированного доступа других пользователей. Следует заметить, что не каждая многозадачная система является многопользовательской и не каждая однопользовательская ОС - однозадачной.

Много- и однопроцессорные системы. Важным свойством ОС является отсутствие или наличие в ней средств поддержки многопроцессорной обработки. В наши дни становится общепринятым введение в ОС функций поддержки *многопроцессорной* обработки данных. Такие функции имеются в операционных системах *Solaris 2.x* фирмы Sun, *Open Server 3.x* компании Santa Crus Operations, *OS/2* фирмы IBM, *Windows NT/XP* фирмы Microsoft и *NetWare 4.1* фирмы Novell.

В системе с многопроцессорной обработкой данных ОС подразделяются по способу организации вычислительного процесса на асимметричные и симметричные. *Асимметричная* ОС целиком работает только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. *Симметричная* ОС полностью децентрализована и использует все количество процессоров, разделяя их между системными и прикладными задачами.

Сетевые и локальные системы. Одним из важных признаков классификации ОС является деление их на сетевые и локальные.

Сетевые ОС предназначены для управления ресурсами компьютеров, объединенных в сеть с целью совместного использования данных. Они имеют мощные средства разграничения доступа к информации, ее целостности и сохранности, а также использования сетевых ресурсов.

Сетевая операционная система составляет основу любой компьютерной сети. Каждый компьютер в сети в некоторой степени автономен, поэтому под сетевой операционной системой, с одной стороны, понимается вся совокупность операционных систем отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам - протоколам.

С другой стороны, сетевая ОС - это операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети. В большинстве случаев сетевые ОС устанавливаются на одном достаточно мощном компьютере-сервере, предназначенном исключительно для обслуживания сети и совместно используемых ресурсов. Все остальные ОС будут считаться *локальными* и могут применяться на любом ПК, подключенном к сети в качестве рабочей станции. На каждой рабочей станции действует своя собственная локальная сетевая операционная система, отличающаяся от ОС автономного компьютера наличием дополнительных средств, позволяющих компьютеру работать в сети.

Локальная сетевая ОС такого типа не имеет фундаментальных отличий от ОС автономного компьютера, но она обязательно содержит программную поддержку для сетевых интерфейсных устройств (драйвер сетевого адаптера), а также средства для удаленного входа в другие компьютеры сети и средства доступа к удаленным файлам, однако эти дополнения существенно не меняют структуру самой операционной системы.

2.2.3. Сервисное программное обеспечение

Сервисное программное обеспечение является расширением базового программного обеспечения компьютера, которое можно классифицировать по функциональному признаку следующим образом, рис. 2.2.:

- программы диагностики работоспособности компьютера;
- антивирусные программы;
- программы обслуживания дисков;
- программы архивирования данных;
- программы обслуживания сети;
- программы обеспечения компьютерной безопасности.

Эти программы часто называют *утилитами*.

Утилиты – это программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров.

Утилиты расширяют и дополняют соответствующие возможности операционной системы, либо решают самостоятельные важные задачи.

Наибольшее распространение сегодня имеют комплекты утилит: *Norton Utilities, Checket Pro Deliuxe 2.0* и др.

Программы диагностики работоспособности компьютера – это совокупность программно-аппаратных средств ПК для обнаружения сбоев в работе компьютера. Они предназначены для проверки работоспособности отдельных узлов, блоков и всей машины в целом, являясь инструментом специалистов по эксплуатации и ремонту технических средств компьютера. Эти средства можно подразделить на средства диагностики ПК, тестового контроля, аппаратного и программно-аппаратного контроля.

Средства диагностики обеспечивают автоматический поиск ошибок и выявление неисправностей с определенной локализацией их в ПК и его отдельных модулях.

Программно-логический контроль основан на использовании избыточного кода исходных и промежуточных данных ПК, что позволяет находить ошибки при изменении значения отдельных битов данных.

Тестовый контроль осуществляется с помощью специальных тестов для проверки правильности работы ПК или его отдельных устройств.

Аппаратный контроль ведется автоматически с помощью встроенного в ПК оборудования.

Программно-аппаратный контроль включает программный и аппаратный контроль.

Приведем названия программ, которые используются для диагностики работоспособности компьютера: *PC Wizard, SiSoftware, ASTRA32* и др.

Антивирусные программы – предназначены для предотвращения заражения компьютера вирусами и ликвидации последствий заражения.

Антивирусные программы оцениваются по ряду критериев:

- точность обнаружения (идентификация) вируса;
- возможность защиты данных от инфицирования (восстановление файлов);
- эффективное устранение обнаруженных вирусов (восстановление файлов);
- простота использования;
- возможность работы в локальных сетях и др.

Современные антивирусные программы являются интегрированными средствами для выявления и устранения компьютерных вирусов. Одним из наиболее перспективных направлений развития антивирусных средств является создание сетевых версий этих продуктов. Сетевой антивирусный программный продукт устанавливается на сервер и при обнаружении вируса блокирует дальнейшую работу с пораженными ресурсами.

Наиболее известные антивирусные программы: *Norton Antivirus, Dr.Web, Virex* и др.

К ***программам обслуживания дисков*** относятся дисковые компрессоры; дисковые дефрагментаторы; программы резервного копирования данных; архиваторы; программы, оптимизирующие использование оперативной памяти; программы защиты и восстановления данных; антивирусные программы и др.

Приведем примеры наиболее известных программ обслуживания дисков: программа дефрагментации диска (*DEFRAG*), программа проверки диска (*Scan Disk*), программа уплотнения диска (*DrvSpace*), программа резервирования (копирования) данных на диске и др.

Программы архивирования данных позволяют сжимать информацию на дисках и создавать архивы данных. Архивирование данных упрощает их хранение за счет того, что большие группы файлов и каталогов сводятся в один архивный файл. При этом повышается и эффективность использования носителя за счет того, что архивные файлы обычно имеют повышенную

плотность записи информации. Архиваторы часто применяют для создания резервных копий важных данных.

В настоящее время применяется несколько десятков программ-архиваторов, которые различаются перечнем функций и параметрами работы. Из числа наиболее популярных программ можно выделить: *ARJ, PKPAK, ZIP, HYPER, RAR* и др.

Программы обслуживания сети предназначены для создания и функционирования компьютерных сетей. Они синхронизируют работу абонентов сети и распределяют информационные, программные и технические ресурсы сети между абонентами. Их основной задачей является передача информации в сети, обеспечение совместимости данных независимо от технических, программных и информационных особенностей абонентов.

К ***программам обеспечения компьютерной безопасности*** относятся средства пассивной и активной защиты данных от повреждения, а также от несанкционированного доступа, просмотра и изменения данных. В качестве средств пассивной защиты используют служебные программы, предназначенные для резервного копирования. Нередко они обладают и базовыми свойствами диспетчеров архивов (архиваторов). В качестве средств активной защиты применяют антивирусное программное обеспечение. Для защиты данных от несанкционированного доступа, их просмотра и изменения служат специальные системы, основанные на криптографии.

2.3. Инструментарий технологии программирования

Инструментарий технологии программирования – это программные продукты поддержки (обеспечения) технологии программирования.

В рамках инструментария технологии программирования сформировались следующие группы программных продуктов, рис. 2.3.:

- средства для создания приложений, включающие:

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

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

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

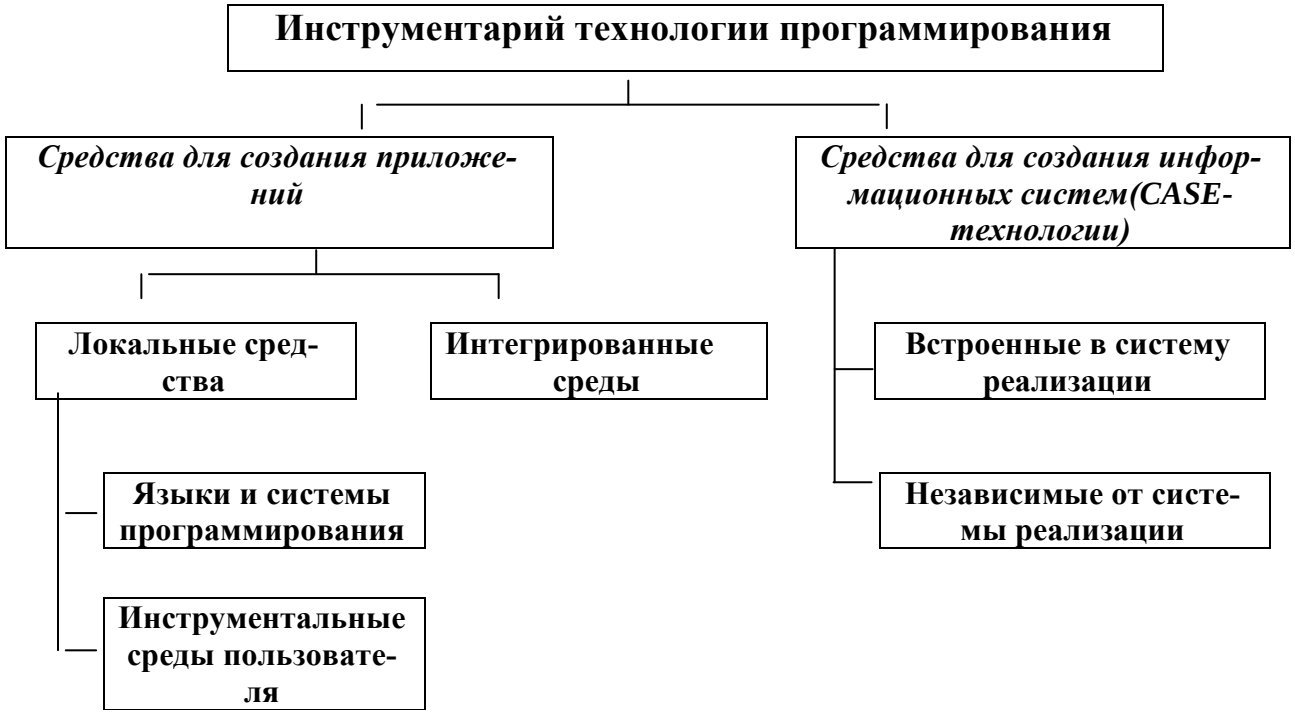


Рис. 2.3. **Классификация инструментария технологии программирования**

Локальные средства разработки программ включают языки программирования и системы программирования, а также инструментальную среду пользователя.

Язык программирования – формализованный язык для описания алгоритма решения задачи на компьютере.

Синтаксис языка – совокупность правил, определяющих допустимые конструкции языка.

Средства для создания приложений – совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.

Языки программирования можно условно разделить на классы:

- *машинные языки* – языки программирования, воспринимаемые аппаратной частью компьютера (*машинные коды*);
- *машинно-ориенированные языки* – языки программирования, которые отражают структуру конкретного типа компьютера (*Ассемблер*);
- *алгоритмические языки* - независящие от архитектуры компьютера языки программирования для отражения структуры алгоритма (*Паскаль, Фортран, Бейсик* и др.);
- *процедурно-ориентированные языки* – языки программирования, где имеется возможность описания программы как совокупности процедур (подпрограмм) (*Фортран, Бейсик, Паскал, Си* и др.);
- *объектно-ориентированные* – языки программирования, базирующиеся на объектной декомпозиции предметной области программы (*Delphi*, *Visual Си++, Visual Basic* и др.);
- *проблемно-ориентированные языки* – языки программирования, ориентированные на решение задач определенного класса, например искусственного интеллекта (*Пролог, Лисп, Симула* и др.).

Пролог – язык логического программирования, предназначенный для рещения логических задач, моделирования логического умозаключения человека.

Лисп - язык функционального программирования, разработанный для обработки символьной информации и исследований по проблематике искусственного интеллекта.

В представленной классификации машинные и машинно-ориентированные языки относятся к языкам программирования *низкого уровня*, остальные считаются языками программирования *высокого уровня*.

Любая программа, подготовленная на языке программирования высокого уровня, должна быть преобразована в машинную программу, состоящую из машинных команд. Для этих целей служит специальные программы - трансляторы. Программы – трансляторы производят преобразование *исходного кода* программы в объектный код, рис. 2.4.

Рис. 2.4. **Схема процесса создания загрузочного модуля программы**

Трансляторы реализуются в виде компиляторов и интерпретаторов.

Компиляторы формирует полный текст программы в машинных кодах, лишь после этого она может быть выполнена.

Интерпретаторы последовательно преобразуют каждый отдельный оператор входной программы в машинный код и сразу его выполняет.

Интегрированные среды разработки программ – предназначены для комплексного применения на всех технологических этапах создания программ. Основное назначения инструментария данного вида – повышение производительности труда программистов, автоматизация создания кодов программ, обеспечивающих интерфейс пользователя графического типа, разработка приложений для архитектуры клиент-сервер, запросов и отчетов.

CASE-технологии создания информационных систем – это специальный программный комплекс для проектирования, анализа программного обеспечения и сопровождения сложных программных систем.

Основное достоинство CASE-технологии – поддержка коллективной работы над проектом за счет возможности работы в локальной сети разработчиков, экспорта и импорта любых фрагментов проекта, организацию управления проектом создания информационной системы.

2.4. Состав и назначение прикладного программного обеспечения

Программное обеспечение, предназначенное для решения определенных классов задач пользователя, называют *прикладным*.

Прикладное программное обеспечение состоит из пакетов прикладных программ (ППП) и прикладных программ пользователя, рис.2.5.

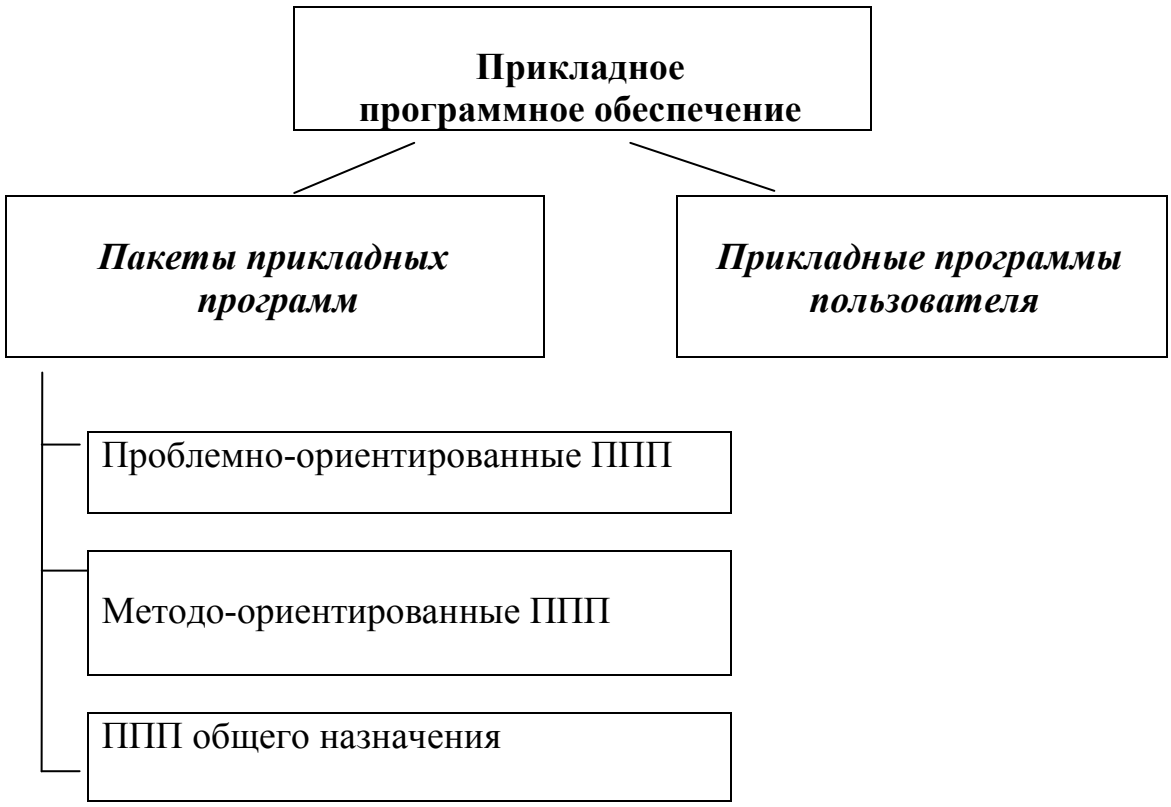


Рис. 2.5. **Классификация прикладного программного обеспечения**

В настоящее время значительное место в прикладном программном обеспечении занимают пакеты прикладных программ.

Пакет прикладных программ – комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области.

Пакеты прикладных программ по сфере применения можно разделить на проблемно-ориентированные, общего назначения и интегрированные пакеты.

Отличительными чертами ***проблемно-ориентированных*** пакетов прикладных программ являются их сравнительно узкая направленность на определенный круг решаемых задач и большое разнообразие.

Пакеты общего назначения (методоориентированные пакеты) предназначены для решения типовых задач обработки данных. К пакетам прикладных программ общего назначения можно отнести наиболее распространенные программные продукты: текстовые процессоры, табличные процессоры, графические редакторы, системы управления базами данных.

Интегрированные пакеты прикладных программ – это набор нескольких программных продуктов, объединенных удобным пользовательским инструментом.

2.4.1. Проблемно-ориентированные пакеты прикладных программ

Проблемно-ориентированные ППП – это самый представительный класс программных продуктов, внутри которого проводится классификация по разным признакам: по типам предметных областей, по информационным системам, по функциям и комплексам задач и др.

Класс проблемно-ориентированных ППП состоит из следующих подклассов пакетов:

– ППП автоматизированного бухгалтерского учета;

– ППП финансовой деятельности;

– ППП управления персоналом;

– ППП управления материальными запасами;

– ППП управления производством и др.

Основные тенденции в области развития проблемно-ориентированных программных средств заключаются в создании программных комплексов в виде автоматизированных рабочих мест (АРМ) управленческого персонала; создании интегрированных систем управления на базе компьютерных сетей; организации данных больших информационных систем.

В настоящее время широко используются проблемно-ориентированные ППП в комплексной автоматизации финансово-бухгалтерской деятельности в банках, на промышленных предприятиях и в сфере торговли, справочно-правовых операций и документооборота.

ППП по автоматизации бухгалтерского учета. Пакеты прикладных программ по бухгалтерскому учету прошли в своем развитии несколько этапов. Первый этап (*1985-1990 гг.*) характеризовался функциональной примитивностью и сложностью адаптации к быстро меняющимся правилам ведения бухгалтерского учета. Второй этап (*1991-1995 гг.*) отличался большей функциональной полнотой и более легкой приспособляемостью к правилам ведения бухгалтерского учета. Пакеты прикладных программ, созданные на втором этапе, функционировали не только автономно, но и в локальных сетях предприятий. Наряду с универсальными пакетами стали появляться пакеты, ориентированные на более узкий круг заказчиков, например для автоматизации расчетов торговых предприятий. Третий этап (*1995-2000 гг.*) отличает комплексный подход к ППП и их узкая специализация. В данном случае эти ППП уже являются интегрированными и служат для полной автоматизации деятельности предприятий.

Четвертый этап (с начала XXI в.) это современные интегрированные ППП по бухгалтерскому учету, которые предполагают поставку вместе с программными средствами методики организации производства и консалтинговых услуг.

В настоящее время широко используются такие бухгалтерские программы, как *“1С:Бухгалтерия”, “Инфобухгалтер”, “Парус”, “Бэст”* и др.

ППП по автоматизации расчетов в розничной и оптовой торговле. Пакеты прикладных программ для розничной и оптовой торговли занимают отдельное направление в автоматизации управления предприятий. Они обеспечивают не только автоматизацию бухгалтерского учета, но и помогают осуществить оперативное управление предприятием; используя раз-

нообразное торговое оборудование, обеспечить учет и контроль па удаленных складах и филиалах больших магазинов и торговых баз.

Пакеты прикладных программ для автоматизации торговли отличаются от бухгалтерских ППП. Так, в бухгалтерском учете операции фиксируются обычно не в момент совершения действия, а с некоторым опозданием. Пакеты прикладных программ для бухгалтерского учета ориентированы чаще всего на определенные отчетные периоды (день, месяц, квартал, год). Пакеты прикладных программ для торговли настроены в основном на оперативный учет. Время и последовательность обработки документов имеют приоритетное значение, так как без этого сложно контролировать текущее состояние товаров, взаиморасчеты и др.

Среди российских фирм-разработчиков, лидирующих на рынке прикладного программного обеспечения в сфере финансово-хозяйственной и управленческой деятельности торговых предприятий и корпораций, можно выделить такие компании, как "*1С", "Интеллект-Сервис", "Инфософт", "Клиент-Серверные технологии", "Алеф Консалтинг энд Софт*" и др.

ППП автоматизации проектирования. Успешное функционирование промышленных предприятий во многом определяется наличием информационных технологий на базе комплексной автоматизации основных технологических и производственных процессов. Для автоматизации конструкторских работ все шире используют системы автоматизированного проектирования (САПР).

Первые ППП в этой области автоматизации появились на российском рынке в конце 1980-х гг. (наиболее известным является *AutoCad*). В настоящее время рынок САПР представлен программными продуктами, предназначенными как для автоматизации отдельных проектных и конструкторских решений, так и в виде интегрированных ППП, способных охватить весь технологический цикл подготовки производства.

Существующие САПР можно разделить на три больших класса.

К первому классу (САПР легкие) относят программные продукты, предназначенные для работы на ПК в автономном режиме или в рамках корпоративной сети. Такая САПР позволяет существенно облегчить подготовку конструкторской документации.

Второй класс (САПР среднего уровня) представлен программными продуктами в основном зарубежного производства, позволяющими осуществить двух- и трехмерное проектирование сложных объектов.

Третий класс (САПР полного цикла) представлен продуктами, которые обеспечивают автоматизацию всех процессов, начиная от конструкторской документации и заканчивая готовым продуктом. Такие системы ориентированы на использование новейшего производственного оборудования, и прежде всего станков с числовым программным управлением.

Системы принятия решений. В условиях рыночной экономики руководитель должен иметь возможность оперативно анализировать текущее состояния предприятия по целому ряду показателей для принятия правильных управленческих решений. Актуальность такой проблемы нашла отражение в ряде программных продуктов, в которых встроены специальные аналитические инструменты.

Быстрый рост потребности общества в системах принятия решений (СПР) привел к созданию корпоративных стандартов и появлению систем данного класса.

Рынок аналитических систем быстро растет и это относится в первую очередь к системам, ориентированным на работу с нечетко структурированными аналитическими задачами. Важным элементом СПР является специальная "семантическая прослойка", позволяющая применять привычные профессиональные термины. Получаемые отчеты могут быть просмотрены в табличном и графическом виде. При большом объеме данных эти программы предоставляют пользователю различные средства навигации по данным и их анализ с различной степенью детализации с использованием OLAP-технологий.

Справочно-правовые системы. В последнее время на предприятиях отмечается активное внедрение компьютерных справочно-правовых систем (СПС), поскольку они предоставляют пользователю удобный и эффективный инструмент работы с огромным массивом законодательной информации.

Развитие и распространение справочно-правовых систем началось в 1989 г. с появлением СПС - ЮСИС, а в 1990 г. была разработана система “Гарант". В настоящее время насчитываются уже десятки компьютерных СПС, наиболее известные и распространенные из которых "*Консультант-Плюс"* и *"Гарант"*. Каждую СПС отличает не только содержимое баз данных, но и возможность организации диалога, порядок и условия обработки запросов, достоверность списка документов, отобранных системой по запросу пользователя. При сравнении нескольких СПС важнейшим критерием их оценки является степень интеллектуальности системы при обработке запросов пользователя.

Системы электронного бизнеса. Понятие "электронная коммерция", или "электронный бизнес", включает продажи, маркетинг, финансовый анализ, платежи, поиск сотрудников, поддержку пользователей и деловых партнерских отношений.

Идея и концепция электронного бизнеса появились раньше, чем персональный компьютер. Применявшиеся еще в 1970-х гг. приложения для электронного обмена данными и электронного перевода средств были первым опытом электронного бизнеса. Главным недостатком автоматизации в этой сфере на ранних этапах была ее высокая стоимость, обусловленная стоимостью нестандартных программных и аппаратных средств. Решения такого рода были возможны только для очень крупных предприятий.

По мере совершенствования потенциала ПК и сети Интернет появилась возможность использовать их для повышения эффективности ведения бизнеса, а также для формирования деловых отношений нового типа. Необ-

ходимые условия для расширения электронного бизнеса - это недорогие мультимедийные ПК и мощные серверы.

2.4.2. Методо-ориентированные пакеты прикладных программ

Данный класс включает в себя программные продукты, обеспечивающие независимо от предметной области и функций информационных систем математические, статистические и другие методы решения задач.

Наиболее распространены методы математического программирования, решения дифференциальных уравнений, имитационного моделирования, исследования операций.

Методы статистической обработки и анализа данных (описательная статистика, регрессионный анализ, прогнозирование значений технико-экономических показателей и т.п.) имеют всевозрастающее применение. Так, современные табличные процессоры значительно расширили набор встроенных функций, реализующих статистическую обработку, предлагают информационные технологии статистического анализа. Вместе с тем необходимость в использовании специализированных программных средств статистической обработки, обеспечивающих высокую точность и многообразие статистических методов, также растет.

Среди методо-ориентированных программ выделяют:

- математические программы – *Mathematiccs, MathCAD, Maple, Mathlab* и др.;
- статистические программы – *Statgrafhics, Statictica, Forecast PRO* и др.

На базе методов сетевого планирования с экономическими показателями проекта, формированием отчетов различного рода оформилось новое направление методо-ориентированных программных средств – *управление проектами*, пользователями этих программ являются менеджеры проектов.

Среди программ, предназначенных для управления проектами, широко используются такие системы: *Project Expert, Audit Expert* и др.

2.4.3. Пакеты прикладных программ общего назначения

К пакетам прикладных программ ***общего назначения*** можно отнести наиболее распространенные программные продукты:

– текстовые процессоры;

– табличные процессоры;

– графические редакторы;

- программы подготовки презентаций;

– системы управления базами данных;

– интегрированные пакеты и др.

Текстовый процессор – программа, предназначенная специально для подготовки, редактирования и печати текстовых данных. Наиболее известные текстовые процессоры: *Microsoft Word, Wordstar, Multi-Edit, Chiwriter* и др.

Табличный процессор – комплекс взаимосвязанных программ, предназначенный для автоматизированной обработки данных, представленных в табличном виде.

Табличные процессоры представляют собой удобное средство для проведения бухгалтерских и статистических расчетов. В каждом пакете имеются сотни встроенных математических функций и алгоритмов статистической обработки данных. Кроме того, имеются мощные средства для связи таблиц между собой, создания и редактирования электронных баз данных.

Самые популярные табличные процессоры – *Microsoft Excel, Multiplan, Visicalc, Lotus 1-2-3* и др.

Графические редакторы - это обширный класс программ, предназначенных для создания и обработки графических изображений. В данном классе различают следующие категории: растровые редакторы, векторные редакторы и программные средства для создания и обработки трехмерной графики (3D-редакторы).

Графический редактор предоставляет возможности рисования линий, кривых, раскраски областей экрана, создания надписей различными шрифтами и т.д. Широко применяются графические редакторы: *Corel DRAW, Adobe Photoshop, Adobe Illustrator* и др.

Графические редакторы широко применяются при решении сложных инженерных задач, на их базе создаются ***системы автоматизированного проектирования*** чертежей: *Autocad*, *Microcad*, *Cadkey, Drawing Processor* и др.

Программы подготовки презентаций – это специализированные программы, предназначенные для создания изображений и их показа на экране, подготовки слайд-фильмов, видеофильмов, их редактирования, определения порядка следования изображений.

Самые популярные программы подготовки презентаций – *Microsoft Power Point, Freelance Graphics, Harvard Graphics* и др.

Система управления базами данных (СУБД) – система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающих от прикладных программ конечных пользователей.

Системы управления базами данных дают возможность объединять большие объемы информации и обрабатывать их, сортировать, делать выборку по определенным условиям и т.п.

Наибольшей популярностью пользуются СУБД: *Microsoft Access, Dbase, Rbase, FoxPro, Clipper, Paradox* и др.

Сетевые СУБД ориентированы на хранение и ведение единого информационного фонда сети на серверах баз данных. К ним относятся: *Oracle, Informix, Ingress, Progress* и др.

Интегрированные пакеты – это набор нескольких программных продуктов, объединенных в удобный инструмент.

К интегрированным пакетам относятся мощные программные пакеты, которые объединяют в себе все или некоторые классы перечисленных пакетов общего назначения. Наиболее развитые из них состоят из текстового ре-

дактора, электронной таблицы, СУБД, средств поддержки электронной почты, программ создания презентационной графики, органайзера.

Примером интегрированных пакетов могут служить *Works, Framework, Microsoft Office.*

Наиболее распространенным интегрированным пакетом является *Microsoft Office.* В этот мощный профессиональный пакет входят такие необходимые программы, как текстовый процессор *Microsoft Word*, электронная таблица *Microsoft Excel*, СУБД *Microsoft Access*, программ подготовки презентаций *Power Point*. А также специальные программы для организации работы офисов, среди этих программ *Microsoft Outlook* - средство для коллективной обработки данных, *Microsoft Front Page* - приложение для создания Web-страниц и ряд др. При этом, все составные части интегрированного пакета *Microsoft Office* составляют единое целое, и даже внешне все программы выглядят единообразно, что облегчает их освоение.

Редакторы HTML (Web-редакторы). Это особый класс редакторов, объединяющих свойства текстовых и графических редакторов. Они предназначены для создания и редактирования Web-документов. Web-документы - это электронные документы, при подготовке которых следует учитывать ряд особенностей, связанных с приемом/передачей информации в Интернете.

Теоретически для создания Web-документов можно использовать обычные текстовые редакторы и процессоры, а также некоторые из графических редакторов векторной графики, но Web-редакторы обладают рядом полезных функций, повышающих производительность труда Web-дизайнеров. Программы этого класса можно также эффективно применять для подготовки электронных документов и мультимедийных изданий.

Наиболее известные Web-редакторы: *ADITOR, ARACHNOPHILIA, Dreams Weaver* и др.

***Браузеры (обозреватели, средства просмотра Web-документов*).** К этой категории относятся программные средства, предназначенные для просмотра электронных документов, выполненных в формате HTML (докумен-

ты этого формата используются в качестве Web-документов). Современные браузеры воспроизводят не только текст и графику, но и музыку, человеческую речь, обеспечивают прослушивание радиопередач в Интернете, просмотр видеофильмов, работу со службами электронной почты, с системой телеконференций (групп новостей) и др.

К браузерам относятся следующие программы: *Widows Internet Explorer, Google chrome, Mozila Firefox, Opera* и др.

Экспертные системы. Предназначены для анализа данных, содержащихся в базах знаний, и выдачи рекомендаций по запросу пользователя. Такие системы применяют в случаях, когда исходные данные хорошо формализуются, но для принятия решения требуются обширные специальные знания. Характерной особенностью экспертных систем является их способность к саморазвитию. Исходные данные хранятся в базе знаний в виде *фактов*, между которыми с помощью специалистов-экспертов выявляется определенная система отношений. Если на этапе тестирования экспертной системы выявляется, что она дает некорректные рекомендации и заключения по конкретным вопросам или не может дать их вообще, это означает либо отсутствие важных фактов в ее базе, либо нарушения в логической системе *отношений*. И в том и в другом случае экспертная система сама может сгенерировать достаточный набор запросов к эксперту и автоматически повысить свое качество. С использованием экспертных систем связана особая область научно-технической деятельности, называемая инженерией знаний.

Отдельные категории прикладных программных средств представляют *обучающие, развивающие, справочные, развлекательные* системы и программы. Характерной особенностью этих классов программного обеспечения являются повышенные требования к их мультимедийной составляющей (использованию музыкальных композиций, средств графической анимации и видеоматериалов).

Вопросы для самоконтроля

1. Для чего необходимо классифицировать программное обеспечение ПК?
2. В чем различие между операционной системой и операционной оболочкой?
3. Какие программные средства называются утилитами? Каковы их разновидности?
4. Расскажите о назначении и видах ПО технического обслуживания ЭВМ.
5. Раскройте понятие "многозадачность операционных систем".
6. Каковы особенности интегрированных ПШ автоматизации бухгалтерского учета?
7. В чем существенные различия между сетевыми и локальными операционными системами?
8. Зачем необходима совместимость операционных систем?
9. Перечислите требования, предъявляемые к современным операционным системам.

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	В функции операционной системы **не входит**…	• управление основной памятью компьютера • выполнение арифметических операций • организация и поддержка файловой системы • поддержка работы периферии компьютера
2.	По реализации пользовательского интерфейса операционные системы разделяются на…	• локальные и глобальные • программные и аппаратные • графические и неграфические • общие и частные
3.	Операционной системой является…	• UNIX • API • Adobe • IBM PC
4.	Драйвер относится к _____ программному обеспечению.	• системному • инструментальному • сервисному • прикладному

5.	В основные функции операционной системы **не входит**…	• планирование и диспетчеризация задач • распределение памяти • трансляция программ для ЭВМ • обслуживание всех операций ввода/вывода
6.	Антивирусные программы, драйверы и архиваторы относятся к _____ программному обеспечению.	• прикладному • системному • предметному • служебному (сервисному)
7.	Системным программным обеспечением является…	• «1С: Предприятие» • ORACLE • TCP/IP • OS/2
8.	В состав операционной системы **не входят** …	• планировщики заданий • программы-архиваторы • обрабатывающие программы • управляющие программы
9.	К основным функциям операционных систем **не относится…**	• обмен информацией между различными внутренними устройствами • ведение файловой системы • распределение оперативной памяти персонального компьютера • проверка почтового ящика администратора персонального компьютера • обработка прерываний
10.	Программы, которые осуществляют упаковку и распаковку совокупности информации называются …	• драйверами • трансляторами • архиваторами • редакторами
11.	Приложение для просмотра гипертекстовых страниц называется…	• клиент • сервер • браузер • редактор
12.	Драйвер – это программа, которая позволяет…	• выполнять вспомогательные работы с устройствами ввода/вывода, носителями данных и т.п. • осуществлять диалог пользователя с компьютером • распределять оперативную память персонального компьютера • обеспечивать связь между операционной системой и внешними устройствами
13.	Хронологическая последовательность появления операционных систем: а) MS DOS б) Windows XP в) Windows'98 г) Windows Vista	• а), г), б), в) • а), г), в), б) • г), а), в), б) • а), в), б), г)
14.	К основным функциям операционных систем **не относится**…	• обмен информацией между различными внутренними устройствами • ведение файловой системы • распределение оперативной памяти персо-

		нального компьютера • проверка почтового ящика администратора персонального компьютера • обработка прерываний
15.	В основные функции «операционной системы» **не входит…**	• Распределение и организация виртуальной памяти • Обслуживание всех операций ввода/вывода • Разработка программ для ЭВМ • Загрузка в оперативную память подлежащих исполнению программ
16.	Драйвера – это…	• Программы для ознакомления пользователя с принципами устройства компьютера • Системы автоматизированного проектирования • Программы для согласования работы внешних и внутренних устройств компьютера • Компоненты компилятора
17.	Программа, обеспечивающая взаимодействие операционной системы с периферийным устройством (принтером, дисководом, дисплеем и т.п.), - это…	• Контролер • Драйвер • Транслятор • Компилятор
18.	Функцией утилит **не является…**	• Форматирование диска • Разработка программ для компьютера • Работа с архивами • Обслуживание жесткого диска
19.	Антивирусные программы относятся к_____ программному обеспечению.	• системному • служебному (сервисному) • инструментальному • прикладному
20.	Для решения задач из различных предметных областей предназначено_____ программное обеспечение	• прикладное • служебное (сервисное) • системное • специальное
21.	Компилятор служит для…	• редактирование текста исходной программы • тестирования программного обеспечения • редактирование текста результирующей программы на языке машинных команд • перевода исходной программы в эквивалентную ей результирующую программу на языке машинных команд или ассемблера
22.	Прикладным программным обеспечением являются….	• графический редактор • драйвер видеокарты • ядро операционной системы • программа форматирования диска
23.	Совокупность ЭВМ и программного обеспечения называется….	• интегрированной системой • вычислительной системой • построителем кода • встроенной системой

24.	Библиотеки прикладных программ содержат…	• часто используемые подпрограммы в виде готовых модулей • текстовые редакторы для создания текстов программ • отладчики для поиска ошибок • трансляторы с одного языка программирования на другой
25.	Ядро операционной системы – это…	• Программа или совокупность связанных программ, использующих аппаратные особенности компьютера • Программы, созданные пользователем • Программа для поиска неисправностей оборудования компьютера • Пакеты прикладных программ

Глава 3. Технические средства реализации информационных процессов

3.1. Техническая основа реализации информационных процессов

Электронно-вычислительная машина (ЭВМ), компьютер - комплекс технических средств, предназначенных для автоматизированной обработки информации в процессе решения вычислительных и информационных задач.

Первые компьютеры (электронно-вычислительные машины с программным управлением) были созданы в конце 40-х годов XX века и представляли собой гигантские вычислительные комплексы, использовавшиеся только для вычислительной обработки информации. По мере развития компьютеры существенно уменьшились в размерах, но приобрели дополнительное оборудование, необходимое для их эффективного использования.

В 70-е годы компьютеры из вычислительных машин сначала превратились в *вычислительные системы*, а затем в *информационно-вычислительные системы*.

Вычислительная система – это совокупность одного или нескольких компьютеров или процессоров, программного обеспечения и периферийного оборудования, организованная для совместного выполнения информационно-вычислительных процессов.

Первые вычислительные системы создавались с целью увеличить быстродействие путем параллельного выполнения вычислительных операций. Для современных вычислительных систем критерии их использования несколько иные – важно само информационное обслуживание пользователей, сервис и качество обслуживания.

В таблице 3.1 показана эволюция компьютерных информационных технологий. Как видно из таблицы, в настоящее время основные цели использования компьютеров – информационное обслуживание и управление, сейчас вычислительные машины и системы по существу выполняют функции информационно-вычислительных систем.

Таблица 3.1

Эволюция компьютерных информационных технологий

Параметр	**Этапы развития технологии**				
	50-е годы	***60-е годы***	***70-е годы***	***80-е годы***	***Настоящее время***
Цель использования компьютера	Научно-технические расчеты	Технические и экономические расчеты	Управление и экономические расчеты	Управление, предоставление информации	Телекоммуникации, информационное обслуживание и управление
Режимы работы компьютера	Однопрограммный	Пакетная обработка	Разделение времени	Персональная работа	Сетевая обработка
Тип пользователя	Инженеры-программисты	Профессиональные программисты	Программисты	Пользователи с общей компьютерной подготовкой	Малообученные пользователи
Тип диалога	Работа за пультом компьютера	Обмен перфоносителями и машинограммами	Интерактивный (через клавиатуру и экран)	Интерактивный с жестким меню	Интерактивный экранный тип “вопрос-ответ”

Рассмотрим укрупненную блок-схему классического компьютера, рис. 3.1.

1. ***Процессор*** (*центральный процессор*) – основной вычислительный блок компьютера, содержит важнейшие функциональные устройства:

- *устройство управления* с интерфейсом процессора (системой сопряжения и связи процессора с другими узлами машины);
- арифметико-логическое устройство;
- процессорную память.

Процессор, по существу, является устройством, выполняющим все функции элементарной вычислительной машины.

2. ***Оперативная память*** – запоминающее устройство, используемое для оперативного хранения и обмена информацией с другими узлами компьютера.

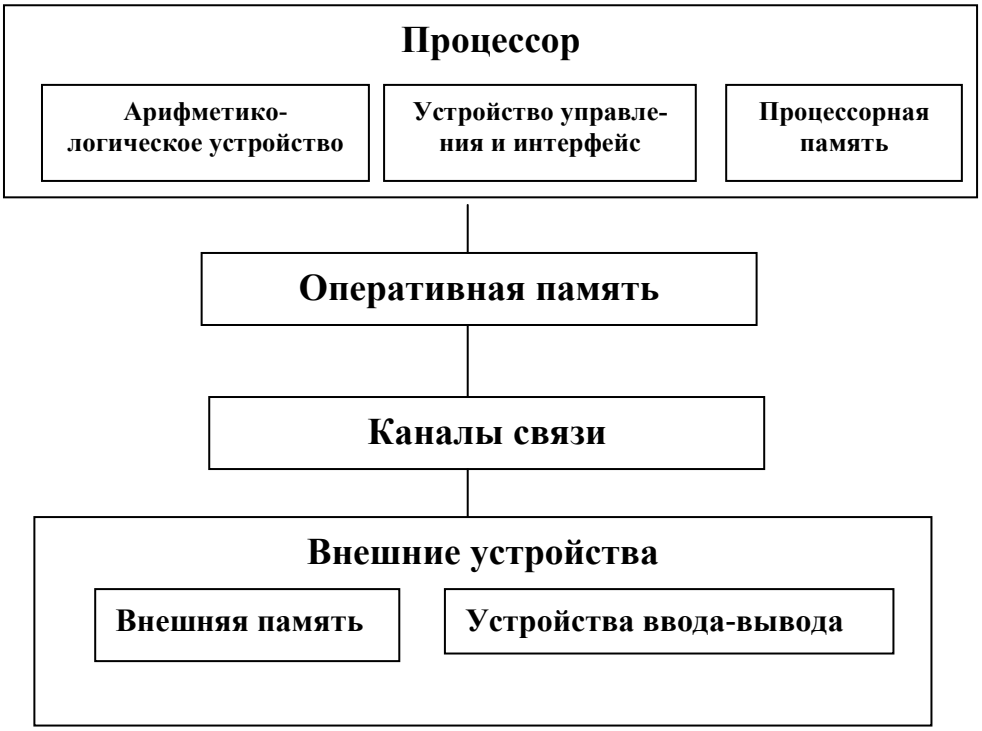


Рис. 3.1. **Блок-схема компьютера**

3. ***Каналы связи*** (внутримашинный интерфейс) – служат для сопряжения центральных узлов машины с ее внешними устройствами.

4. ***Внешние устройства*** обеспечивают эффективное взаимодействие компьютера с пользователями, объектами управления, другими машинами. В состав внешних устройств входят внешняя память и устройства ввода-вывода.

Вычислительная система может строится на основе целых компьютеров в этом случае вычислительная система называется ***многомашинной***, либо отдельных процессоров – ***многопроцессорная*** вычислительная система.

3.2. Поколения электронных вычислительных машин

Развитие электронных вычислительных машин можно условно разбить на несколько этапов (поколений ЭВМ). Каждый этап отражает период со-

здания, элементную базу, архитектуру и развитость программного обеспечения.

Первое поколение (*1946 г. - середина 1950-х гг.*).

Точкой отсчета эры ЭВМ считают 1946 год, когда был создан первый электронный цифровой компьютер “Эниак” (*Electronic Numerical Integrator and Computer*).

Основные характеристики машин первого поколения: элементная база - ***электронные лампы***; программирование в машинных кодах; отсутствие операционной системы; быстродействие — до 20 тыс. оп/с.

Пользователями машин первого поколения были ученые, решающие наиболее актуальные научно-технические задачи, связанные с развитием авиации, ракетостроения и т.д.

Среди известных отечественных машин первого поколения необходимо отметить *БЭСМ-1* (большая электронно-счетная машина), *Стрела, Урал, М-20.* Отечественная ЭВМ *М-20* (20 тыс. оп./с.) была одной из самых быстродействующих машин первого поколения в мире.

Второе поколение (*середина 1950-х гг. - середина 1960-х гг.*).

Развитее электроники привело к изобретению в 1948 году нового полупроводникового устройства – транзистора, который заменил лампы.

Основные характеристики машин второго поколения: элементная база — ***полупроводники***; программирование с использованием алгоритмических языков и библиотек стандартных программ; быстродействие - до 1 млн. оп./с.

Появление ЭВМ, построенных на транзисторах привело к уменьшению их габаритов, массы, энергопотребления и стоимости, а также к увеличению их надежности и производительности.

Среди известных отечественных машин второго поколения необходимо отметить *БЭСМ-4, М-220, Наири, Мир, МИНСК, РАЗДАН, Днестр.* Наилучшей отечественной ЭВМ второго поколения считается БЭСМ-6, созданная в 1966 году. Она имела основную и промежуточную память (на маг-

нитных барабанах), быстродействие порядка 1 млн. оп./с. и довольно обширную периферию (магнитные ленты и диски, графопостроители, разнообразные устройства ввода-вывода).

Третье поколение (*середина 1960-х гг. — 1970-е гг.*).

Создание технологии производства интегральных схем, состоящих из десятков электронных элементов, образованных в прямоугольной пластине кремния с длиной стороны не более 1 см, позволило увеличить быстродействие и надежность ЭВМ на их основе, а также уменьшить габариты, потребляемую мощность и стоимость ЭВМ.

Основные характеристики машин третьего поколения: основа элементной базы — ***интегральные схемы*** среднего уровня интеграции (сотни, тысячи транзисторов в одном корпусе); полномасштабная операционная система; программная совместимость моделей серии.

Машины третьего поколения имеют развитые операционные системы, обладают возможностями мультипрограммирования, то есть одновременного выполнения нескольких программ. Многие задачи управления памятью, устройствами и ресурсами стала выполнять операционная система.

Примеры машин третьего поколения – семейство *IBM-360, IBM-370, PDP-8, PDP-11,* отечественные *ЕС ЭВМ* (единая система ЭВМ), *СМ ЭВМ* (семейство малых ЭВМ).

Четвертое поколение (*1970-е гг. - 1980-е гг.*).

Успехи в развитии электроники привели к созданию больших интегральных схем (БИС), где в одном кристалле размещалось несколько десятков тысяч электронных элементов.

Основные характеристики машин четвертого поколения: основа элементарной базы - ***большие (БИС) и сверхбольшие интегральные схемы (СБИС)*** (десятки, сотни, тысячи транзисторов в одном корпусе); многопроцессорность; производительность — десятки миллионов операций в секунду.

В 1971 году был изготовлен первый микропроцессор – большая интегральная схема, в которой полностью размещался процессор ЭВМ простой архитектуры. Первый *персональный компьютер* был создан в 1976 году.

Начиная с 1980 года практически все ЭВМ стали создаваться на основе микропроцессоров. Самым востребованным компьютером стал персональный.

Пятое поколение (*с 1990-х гг. по настоящее время*).

Основную концепцию компьютеров пятого поколения можно сформулировать следующим образом:

- компьютеры на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных инструкций программ;
- компьютеры со многими сотнями параллельно работающих процессоров, позволяющих строить системы обработки данных и знаний, эффективные сетевые компьютерные системы.

Основные характеристики машин пятого поколения: основа элементной базы — ***сверхбольшие интегральные схемы***; производительность - до нескольких триллионов операций в секунду; десятки параллельно работающих микропроцессоров; работа в режимах векторной, скалярной, матричной и другой обработки данных и команд.

Поколение будущего — это оптоэлектронные (квантовые) компьютеры с массовым параллелизмом обработки данных и команд, моделирующих архитектуру нейронных систем, использующих принципы искусственного интеллекта и логического вывода, обладающие сверхнадежностью.

3.3. Классификация технических средств обработки информации

Вычислительные машины могут быть классифицированы по ряду признаков, рис. 3.2.:

- по принципу действия;
- по поколениям;
- по степени универсальности;
- по степени производительности;
- по особенностям архитектуры;
- по способу использования.

ЭВМ (компьютеры)

Принцип действия	Аналоговые
	Цифровые
	Гибридные

Поколение	Первое
	Второе
	Третье
	Четвертое
	Пятое
	Будущее

Степень универсальности	Универсальные (общего назначения)
	Специализированные
	Проблемно-ориентированные

Степень производительности	Ординарной
	Высокой
	Сверхординарной

Особенности архитектуры	Суперкомпьютеры
	Большие компьютеры
	Малые компьютеры
	Микрокомпьютеры
	Персональные компьютеры
	Мобильные, портативные и карманные
	Встраиваемые микропроцессоры

Способ использования	Индивидуального
	Коллективного

Рис. 3.2. **Классификация ЭВМ**

По ***принципу действия*** вычислительные машины делятся на три больших класса: аналоговые, цифровые, гибридные.

Критерием деления вычислительных машин на эти три класса является форма представления информации, с которой они работают.

Цифровые вычислительные машины – это вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, то есть цифровой форме.

Аналоговые вычислительные машины – это вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).

Гибридные вычислительные машины – это вычислительные машины комбинированного действия, работают с информацией, представленной и в цифровой и в аналоговой форме.

По принципу ***степень универсальности*** вычислительные машины руктивной и программной направленностью компьютера и подразделяются на: универсальные (общего назначения), специализированные, проблемно-ориентированные.

Универсальные (общего назначения) компьютеры предназначены для решения разнообразных по реализуемым алгоритмам задач (экономических, информационно-поисковых, научно-технических и др.). Характерными особенностями машин являются высокая производительность, огромный объем оперативной и внешней памяти, большое разнообразие выполняемых арифметических, логических и специальных операций, развитая система ввода-вывода информации с многообразным видом внешних устройств.

Специализированные компьютеры предназначены для решения сравнительно узкого класса задач или реализации строго регламентированной группы функций. Для этих компьютеров характерны строгая специализация структуры и наличие специального программного обеспечения. Сфера применения машин: управление техническими устройствами; маршрутизация

потоков данных и согласование работы узлов компьютерных сетей и т.д. В последние годы такие компьютеры начинают встраиваться в устройства бытовой техники.

Проблемно-ориентированные компьютеры занимают промежуточное положение среди машин названных групп. Проблемно-ориентированные компьютеры предназначены для решения более узкого круга задач, связанных, как правило, с управлением технологическими процессами, обрабатывают относительно небольшие объемы данных по несложным алгоритмам. Они обладают ограниченными, по сравнению с универсальными компьютерами, аппаратными и программными ресурсами.

По показателю ***степени производительности*** компьютеры подразделяются на три класса: ординарной, высокой и сверхординарной производительности.

Деление машин по этому признаку довольно условное. Электронно-вычислительные машины, которые сейчас относятся к классу высокой производительности, через несколько лет вполне могут оказаться в классе ординарной производительности.

На сегодняшний момент к компьютерам ***ординарной производительности*** относятся массовые персональные компьютеры. Обладая тем не менее высокими техническими характеристиками быстродействия и объема памяти, они служат для решения несложных задач индивидуальных пользователей или работают в составе небольших компьютерных сетей.

Компьютеры ***высокой производительности*** - это одно- или многопроцессорные машины, предназначенные для индивидуального применения при решении задач повышенной сложности либо при обслуживании локальных или региональных компьютерных сетей.

К компьютерам ***сверхординарной (сверхвысокой) производительности*** относят многопроцессорные машины или многомашинные вычислительные комплексы, целью эксплуатации которых является решение задач большой сложности (метеорология, управление космическими объектами,

моделирование микро- и макроэкономических процессов, обслуживание больших компьютерных сетей и др.).

По особенностям архитектурны компьютеры можно подразделить на шесть групп машин, расположенных по производительности – суперкомпьютеры (суперЭВМ), большие компьютеры (мэйнфреймы), малые компьютеры (миниЭВМ), микрокомпьютеры, персональные компьютеры, мобильные компьютеры.

Суперкомпьютеры (***суперЭВМ)*** – это мощные многопроцессорные вычислительные машины с быстродействием сотни миллионов – десятки миллиардов операций в секунду. В суперЭВМ применяются идеи массового параллелизма, когда данные одновременно обрабатывают сотни или тысячи процессоров.

Областью применения суперкомпьютеров являются крупномасштабные задачи, требующие больших объемов вычислений и моделирования. Особенно эффективны суперкомпьютеры для решения задач проектирования и масштабного анализа экономических процессов.

Большие компьютеры (мэйнфреймы) – это многопользовательские машины с центральной обработкой, высокой или сверхординарной производительностью, обеспечивающие подключение нескольких сотен внешних устройств, с большими возможностями для работы с базами данных и различными формами удаленного доступа.

Емкость оперативной памяти мэйнфреймов составляет до нескольких сотен гигабайтов, емкость внешней памяти - до десятков терабайтов.

В настоящий момент основным назначением больших ЭВМ является решение корпоративных задач в системах управления крупными комплексами - фирмами, корпорациями, аэропортами, банками, а также в научно-исследовательских центрах, органах государственного управления и для обслуживания больших компьютерных сетей.

Малые компьютеры (миниЭВМ) – это машины высокой, или сверхординарной производительности с одним или несколькими высокопроизво-

дительными процессорами. Основные назначения машины - решение задач высокой сложности при индивидуальном использовании, а также управление крупными компьютерными сетями в виде серверов.

К основным характеристикам мини-ЭВМ относятся: многопроцессорность с большой интеграцией элементов; емкость памяти в несколько сотен гигабайтов; возможность подключения до нескольких сотен внешних устройств ввода-вывода.

Наряду с использованием миниЭВМ для управления технологическими процессами, они применяются для вычислений в многопользовательских вычислительных системах, в системах автоматизированного проектирования, в системах моделирования несложных объектов, в системах искусственного интеллекта.

Микрокомпьютеры очень многочисленны и разнообразны, среди них можно выделить несколько подклассов:

- ***многопользовательские*** микрокомпьютеры – это мощные микрокомпьютеры, оборудованные несколькими видеотерминалами и функционирующие в режиме разделения времени, что позволяет эффективно работать на них сразу нескольким пользователям;
- ***компьютеры*** – однопользовательские микрокомпьютеры, удовлетворяющие требованиям общедоступности и универсальности применения;
- ***рабочие станции*** – представляют собой однопользовательские микрокомпьютеры для работы в вычислительных сетях, часто специализированные для выполнения определенных видов работ;
- ***серверы*** – многопользовательские мощные микрокомпьютеры в вычислительных сетях, выделенные для обработки запросов от всех рабочих станций сети;
- ***сетевые компьютеры*** – упрощенные микрокомпьютеры обеспечивающие работу в сети и доступ к сетевым ресурсам.

По ***способу применения*** выделяют компьютеры: коллективного пользования и индивидуального пользования.

Компьютерами ***коллективного пользования*** считаются такие, которые могут одновременно обслуживать работу нескольких пользователей. Обычно они имеют высокую производительность, могут работать в режиме разделения времени. Примером таких машин являются серверы компьютерных сетей или многопроцессорные мэйнфреймы.

Компьютеры ***индивидуального пользования*** в каждый момент времени может эксплуатироваться лишь одним пользователем. Примером являются ноутбуки и "компьютеры на ладони".

3.4. Персональные компьютеры

Персональные компьютеры (***ПК***) - относятся к классу микрокомпьютеров, но в виду их массовой распространенности заслуживают особого внимания.

Персональный компьютер должен обладать такими качествами:

- малая стоимость ПК, находящаяся в пределах доступности для индивидуального пользователя;
- гибкость архитектуры, обеспечивающая ее адаптируемость к разнообразному применению в сфере управления, науки, образования, в быту;
- дружественность операционной системы и прочего программного обеспечения, обуславливающая возможность работы с ней пользователя без специальной профессиональной подготовки;
- высокая надежность работы.

Среди современных ПК в первую очередь необходимо отметить персональные компьютеры американской фирмы IBM (*International Business Machine Corporation*), первые модели которой появились в 1981 году.

В настоящее время мировой парк компьютеров составляет около миллиарда штук, из них около 90% - это персональные компьютеры. Самыми распространенными моделями компьютеров в настоящее время являются IBM PC с микропроцессором Pentium III и 4.

Обобщенные характеристики современных персональных компьютеров приведены в таблице 3.2.

Таблица 3.2

Основные характеристики современных персональных компьютеров

Параметр	**Тип микропроцессора**				
	Pentium	***Pentium Celeron***	***Pentium II***	***Pentium III***	***Pentium 4***
Тактовая частота, МГц	**75-200**	**330-800**	**220-500**	**500-900**	**1000-2000**
Разрядность, бит	**64**	**64**	**64**	**64**	**64**
Объем ОЗУ, Мбайт	**8, 16, 32**	**32, 64, 128**	**32, 64, 128**	**64, 128, 256**	**128, 256, 512**
Объем кэш-памяти, Кбайт	**256, 512**	**128, 512, 1024**	**256, 512, 1024**	**256, 512, 1024**	**512, 1024, 2048**
Емкость НМД, Гбайт	**1,0 – 10,0**	**10,0 – 50,0**	**10,0 – 20,0**	**10,0 – 50,0**	**20,0 –200,0**

Персональные компьютеры можно классифицировать по ряду признаков.

По поколениям персональные компьютеры делятся на:

- 1-го поколения используют 8-битовые микропроцессоры;
- 2-го поколения используют 16-битовые микропроцессоры;
- 3-го поколения используют 32-битовые микропроцессоры;
- 4-го поколения используют 64-битовые микропроцессоры.

Персональные компьютеры классифицируются ***по конструктивным особенностям***: на стационарные и портативные.

Стационарные, или *настольные*, ПК используются для решения различных классов задач в условиях подключения к стационарной электрической сети.

Портативные ПК (ноутбуки, субноутбуки), обладая значительно меньшим размером и весом, могут использоваться как при стационарном, так и при автономном электрическом питании.

Мобильные компьютеры (КПК, "компьютеры на ладони", или карманные компьютеры) появились несколько лет назад, но благодаря малым габа-

ритам в сочетании с большим количеством функций, присущих персональным ПК, получили широкое распространение для ввода и редактирования данных, подготовки данных к обработке на стационарных и переносных ПК.

Мобильные компьютеры имеют архитектуру, отличную от персональных ПК, используют свои типы процессоров. Не имея электромеханических устройств для ввода-вывода данных, мобильные компьютеры обладают развитой операционной системой и большим объемом оперативной памяти. Основные используемые программы размещаются в постоянном запоминающем устройстве. При этом память КПК может быть расширена за счет флэш-карт. Возможность выхода в сеть Интернет делает КПК весьма перспективными для применения в сфере бизнес-услуг.

3.5. Структурная схема персонального компьютера

Аппаратные средства современных персональных компьютеров представляют собой совокупность электронных, электромеханических, электромагнитных и электронно-оптических устройств. Каждое устройство выполняет определенный набор функций, определяемых комбинацией входных управляющих электрических сигналов – команд.

Основное назначение компьютера – выполнять программы, представляющие собой набор команд.

Команда – это инструкция, предписывающая компьютеру выполнять ту или иную операцию (умножить два числа, записать данные на диск и т.д.) Все команды и все данные в компьютере представлены комбинациями битов (чисел).

Устройством, которое обрабатывает информацию, является процессор.

Процессор – электронное устройство, обрабатывающее различные виды информации в форме последовательности электрических импульсов. Такие последовательности можно записать в виде цепочки нулей и единиц (есть импульс – единица, нет импульса – нуль), которые называются машинным языком.

Последовательность этих команд называется ***программой.*** Устройство управления “переводит” команды программ на язык команд, понятных исполнителям, и синхронизирует их работу.

Исторически компьютер появился, как машина для вычислений и назывался электронной вычислительной машиной – ЭВМ. Структура такого устройства была описан знаменитым математиком Дж. Фон Нейманом в 1945 г., рис. 3.3.

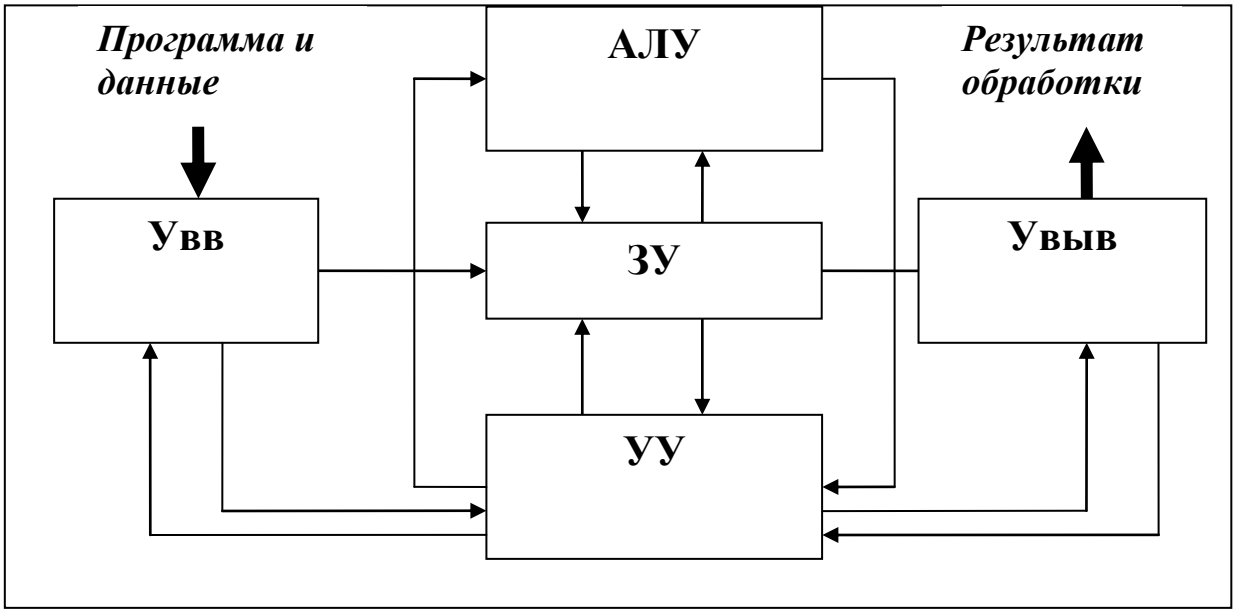


Рис. 3.3. **Классическая структура ЭВМ**

Структура ЭВМ — это модель, устанавливающая состав основных частей ЭВМ и способы установления связей между ними.

В классической структуре ЭВМ выделяют арифметико-логическое устройство, устройство управления, запоминающее устройство и внешние устройства ввода-вывода.

Охарактеризуем отдельные блоки вычислительных устройств.

Арифметико-логическое устройство (***АЛУ***) обеспечивает выполнение процедур преобразования данных. Преобразует информацию, выполняя сложение, вычитание и основные логические операции “И”, “ИЛИ”, “НЕ”.

Устройство управления (***УУ***) обеспечивает управление процессом обработки данных и организует весь процесс выполнения программ. Устройство управления выбирает команды программы из основной памяти, интерпретирует суть команды и запускает нужную схему арифметико-логического устройства.

Запоминающие устройства (***ЗУ***) обеспечивают промежуточное хранение обрабатываемых процессором данных. Основная память ЭВМ включает оперативную и постоянную память.

Оперативная память (***ОЗУ***) - устройство, обеспечивающее временное хранение команд и данных в процессе выполнения программы. Оперативные запоминающее устройство хранит данные, адреса и команды, обладает высокой скоростью записи и чтения чисел. Состоит из некоторого числа пронумерованных ячеек, в каждой из которых могут находиться обрабатываемые данные или программы. Все ячейки памяти одинаково доступны для других устройств компьютера.

Постоянная память (***ПЗУ***) - устройство, обеспечивающее постоянное хранение и возможность считывания важной информации для функционирования ЭВМ.

Устройства ввода-вывода (***Увв, Увыв***) – получают информацию извне, выводят ее получателю.

Структура современного персонального компьютера представлена рис. 3.4.

Достижения микроэлектроники позволили объединить на одной интегральной схеме, называемой ***микропроцессором*** (МП) или процессором, АЛУ и УУ.

Уменьшение габаритов ОЗУ позволило разместить МР и ОЗУ на одной электронной плате, называемой ***системной*** или ***материнской***.

Все связи между отдельными устройствами объединены в пучок параллельных проводов – ***системную шину***. В состав системной шины входят

шина данных, по которой передаются данные из ОЗУ в МП, шина адреса и шина управления.

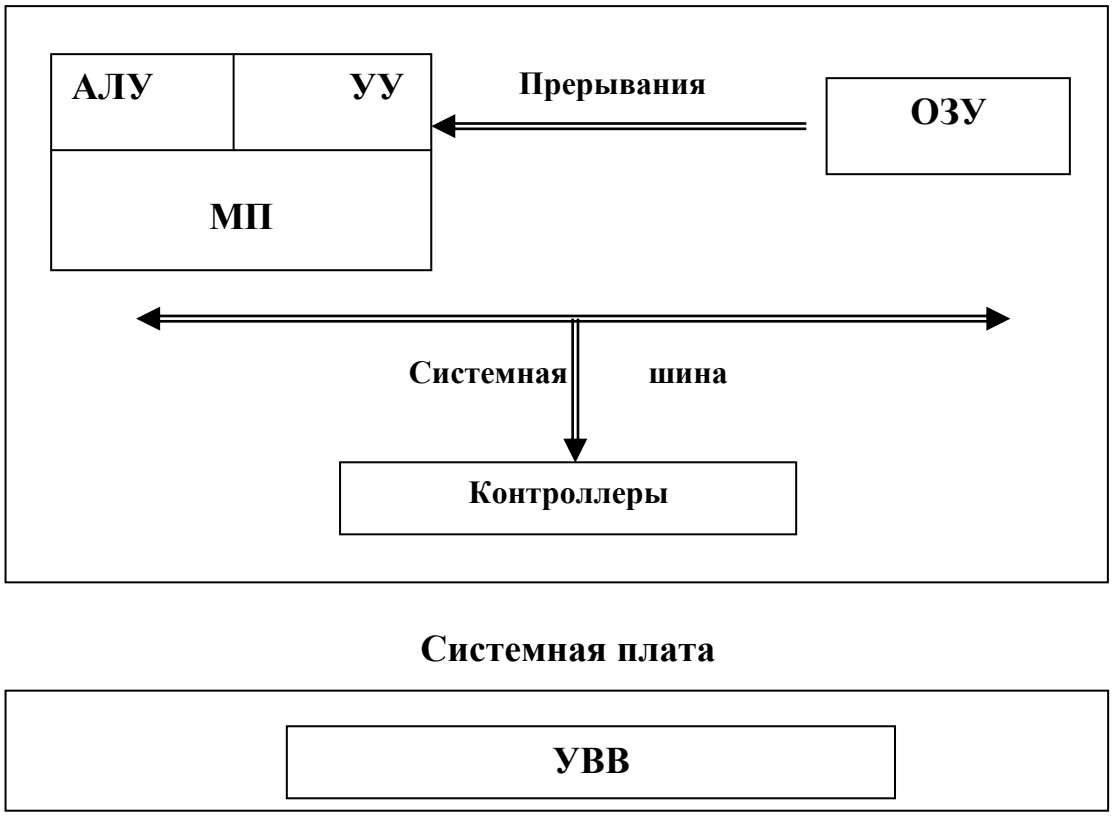


Рис. 3.4. **Структурная схема персонального компьютера**

Устройства ввода-вывода разделены на собственно устройства ввода-вывода и управляющие ими контролеры (карты), включаемые в системную плату или устанавливаемые прямо на ней.

Новым в структуре современного персонального компьютера и принципе его действия являются сигналы и понятие прерываний, рис. 3.4. Прерывания появились в связи с переходом от математических вычислений к обработке информации в реальном масштабе времени.

В современных компьютерах возможна параллельная работа нескольких процессоров. За счет распараллеливания выполнения одной задачи или параллельного выполнения многих задач достигается увеличение общей производительности компьютера. Для этого предусматриваются цепи, связывающие между собой отдельные процессоры.

3.6. Принципы функционирования персонального компьютера

Основные принципы организации электронных вычислительных машин были заложены Дж. Фон Нейманом:

1. *Принцип двойного кодирования.* Электронные машины должны работать не в десятичной, а в двоичной системе счисления.
2. *Принцип программного управления.* Электронная машина выполняет вычисления по программе. Программа состоит из набора команд, которые выполняются автоматически друг за другом в определенной последовательности.
3. *Принцип хранимой программы.* В процессе решения задачи программа должна размещаться в запоминающем устройстве машины, обладающем высокой скоростью выборки и записи.
4. *Принцип однотипности представления чисел и команд*. Программа, так же как и числа, с которыми оперирует машина, записывается в двоичном коде. Таким образом, по форме представления команды и числа однотипны.
5. *Принцип иерархичности памяти*. Сложность реализации единого емкого быстродействующего запоминающего устройства требует иерархического построения памяти. По меньшей мере, должно быть два уровня иерархии: основная память и внешняя.
6. *Принцип адресности основной памяти*. Основная память должна состоять из пронумерованных ячеек, каждая из которых доступна программе в любой момент времени по ее двоичному адресу.

Процессор и основная память являются ***центральными устройствами компьютера***, поскольку именно на их основе реализуется принцип программного управления. Все остальные устройства компьютера считаются *внешними*, или *периферийными*.

Внешние устройства компьютера - устройства, обеспечивающие ввод и вывод данных из основных устройств компьютера (устройства ввода-

вывода) и долговременное хранение информации, не обрабатываемой процессором в данный момент времени (внешние запоминающие устройства).

В одном компьютере может использоваться до нескольких сотен внешних устройств разного типа. Состав устройств ввода-вывода может изменяться в зависимости от классов задач, решаемых на компьютере.

Производительность и эффективность использования компьютера определяются не только составом и характеристиками ее устройств, но и способом организации их совместной работы. Связь между устройствами компьютера осуществляется с помощью сопряжений, которые называются интерфейсами.

Интерфейс представляет собой совокупность стандартизованных аппаратных и программных средств, обеспечивающих обмен информацией (сигналами) между устройствами. Наличие стандартных интерфейсов позволяет унифицировать передачу информации в виде сигналов между устройствами независимо от их особенностей.

Основной, центральной частью компьютера является процессор, объединяющий арифметико-логическое устройство и устройство управления в единое целое. У современных компьютеров значительно расширились номенклатура и число подключаемых устройств ввода-вывода; запоминающее устройство приняло иерархический вид за счет сверхоперативной кэш-памяти и разнообразных внешних накопителей.

Появился термин "***аппаратная платформа***" для классов и типов ЭВМ. Под этим термином стали понимать совокупность технических средств, определяющих среду функционирования конкретных программ обработки данных. В основу аппаратной платформы были положены совокупность интерфейсной системы передачи данных и тип используемого процессора.

Термин "архитектура ЭВМ" приобрел новое звучание применительно к современным компьютерам.

Архитектура компьютера - это совокупность основных устройств, узлов и блоков, а также структура основных информационных и управляющих связей между ними, обеспечивающая выполнение заданных функций; структура базового программного обеспечения, а также сочетание аппаратного и базового программного обеспечения, поддерживающее объединение компьютеров в сети.

Принципом построения и функционирования современных компьютеров различных классов является ***программное управление***, в основе которого находится представление алгоритма решения любой задачи в виде программы вычислений.

В общем случае *алгоритм* определяется как порядок выполнения операций над данными с целью получения конечного результата либо как конечный набор предписаний, определяющий решение задачи посредством конечного количества операций. Одна и та же задача может быть реализована по различным алгоритмам, в то же время для реализации одного и того же алгоритма могут использоваться различные программы, учитывающие особенности архитектуры компьютера.

При решении задачи вся информация должна быть доступна процессору и располагаться в оперативной памяти. Для современных ЭВМ принята байтовая структура памяти, т.е. все ее пространство условно разбивается на ячейки по 1 байту. Байты виртуально нумеруются, и к ним по адресам памяти обращается центральный процессор. Передача данных между внутренней памятью и центральным процессором осуществляется словами.

Исходя из предложенного Дж. Фон Нейманом иерархического принципа памяти, в современных компьютерах память по назначению, методам использования и параметрам подразделяется на оперативную, постоянную, внешнюю сменяемую, внешнюю несменяемую, процессорную.

Для четкого понимания ***принципа программного управления*** работой отметим, что это управление "внутри машины" реализуется благодаря взаимодействию двух блоков: центрального процессора и внутренней памяти.

Внутренняя память предназначается для кратковременного хранения программ и обрабатываемых данных. Она содержит данные (числа и символы), подлежащие обработке, промежуточные и окончательные результаты. Часть оперативной памяти может выступать как буфер для хранения отдельных параметров внешних устройств машины.

Устройство управления, являющееся составной частью центрального процессора, обеспечивает автоматическое выполнение программы путем принудительной координации работы всех остальных устройств ЭВМ. Устройство управления, считывая очередную команду, расшифровывает ее, определяет перечень необходимых компонентов для ее выполнения, загружает их из памяти и реализует. При этом каждая команда под воздействием сигналов устройства управления выполняется в цикле, рис. 3.5.

Программное управление осуществляется в несколько *этапов*:

1. Формирование адреса очередной команды. Адрес первой команды программы находится вне цикла специальным способом.
2. Нахождение и выборка из оперативной памяти команды, расшифровка ее содержания.
3. Поиск в оперативной памяти и чтение из нее необходимых данных.
4. Выполнение команды.

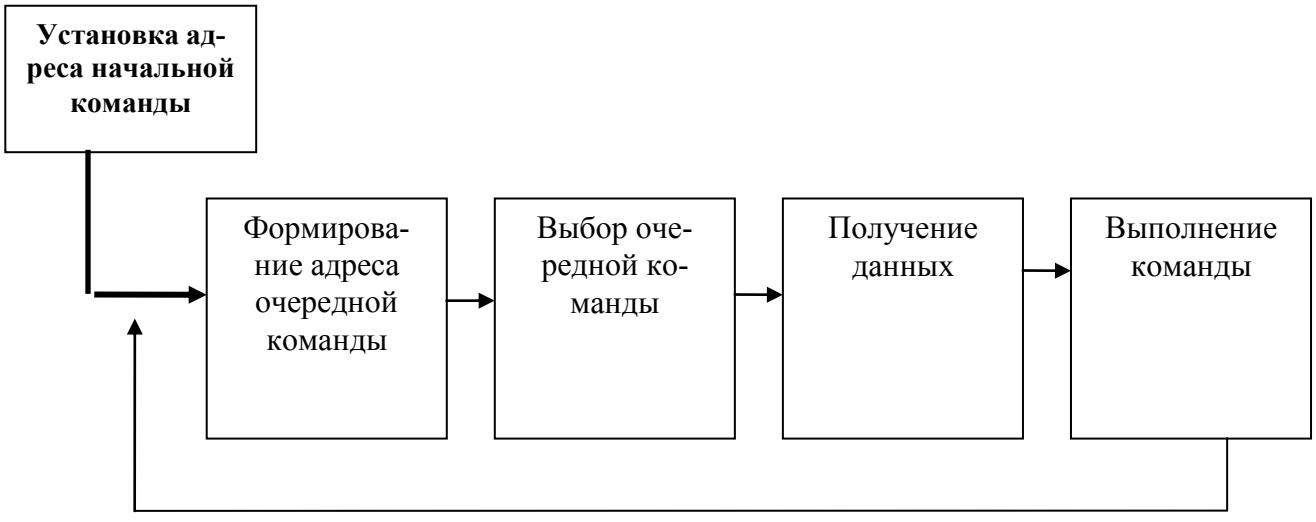


Рис. 3.5. **Схема принципа программного управления**

3.7. Основные архитектурные схемы вычислительных систем

Современные компьютеры, предназначенные для решения задач со сложными алгоритмами, управления другими объектами или компьютерными сетями, в своем составе имеют не один, а несколько процессоров, обеспечивая *многопрограммный* (мультипрограммный) режим работы всей системы.

Главная задача многопроцессорных систем - обеспечить достижение сверхбольших скоростей работы на основе распараллеливания вычислений. Классификация архитектур подобных систем, предложенная М. Флинном в 1960-х гг., остается актуальной до сих пор.

В ее основе находятся *два подхода*: независимость потоков заданий (команд), существующих в вычислительной системе и независимость данных, обрабатываемых в каждом потоке.

Согласно этой классификации, существуют четыре ***основные архитектуры вычислительных систем:***

1. *Архитектура ОКОД* (одиночный поток команд — одиночный поток данных), или *SISD* (*Singly Instruction stream - Singly Date stream*), соответствует однопроцессорной ЭВМ с невозможностью распараллеливания вычислений, рис. 3.6.

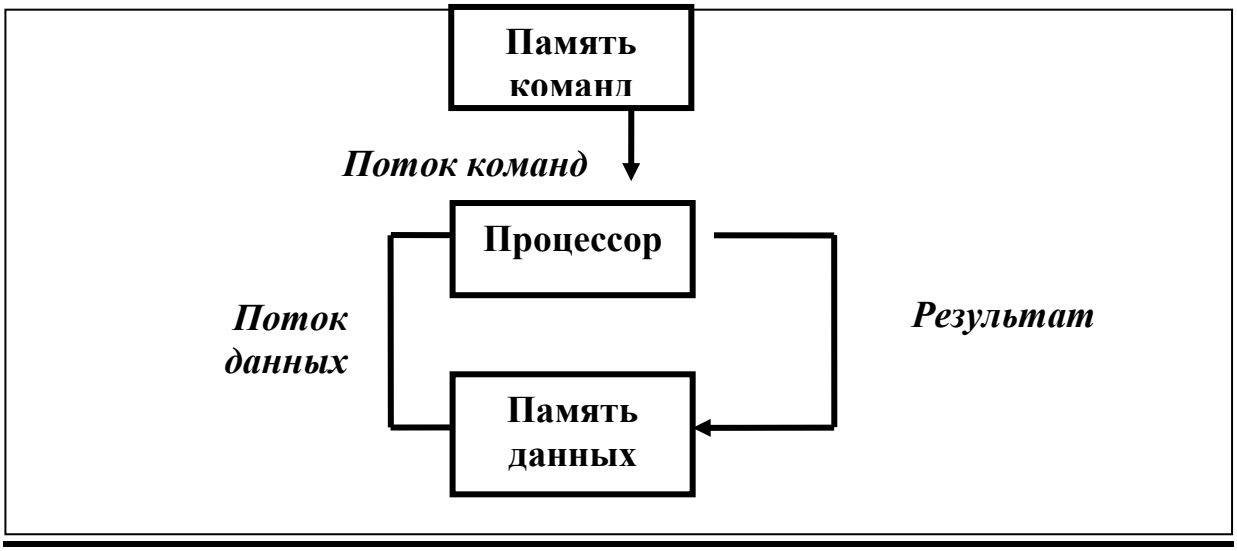


Рис. 3.6. **Архитектура ОКОД**

2. *Архитектура ОКМД* (одиночный поток команд - множество потоков данных), или *SIMD* (*Singly Instruction stream - Multiple Date stream*), соответствует матричной многопроцессорной системе обработки данных, рис. 3.7.

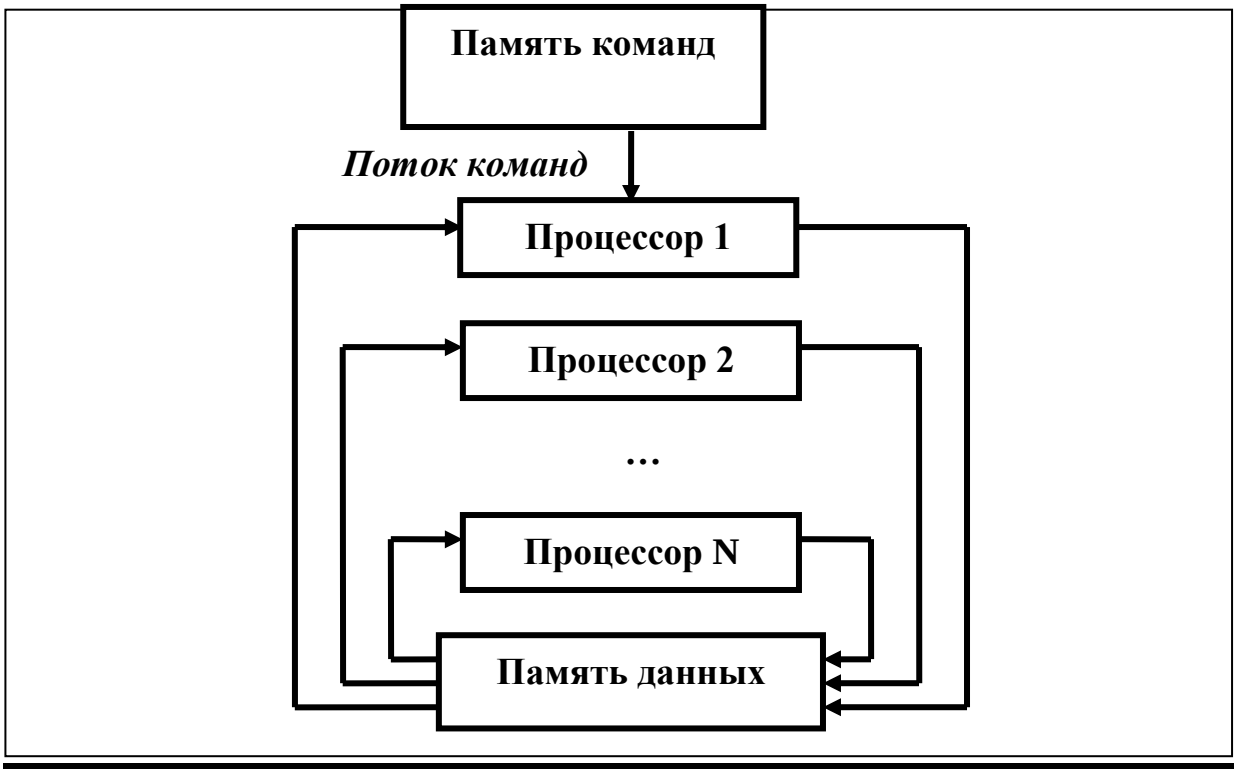


Рис. 3.7. **Архитектура ОКМД**

В такой системе используется несколько однородных, сравнительно простых быстродействующих процессоров, управляющих одной и той же последовательностью команд, но каждый процессор обрабатывает только свой поток данных. По такой схеме были построены такие суперкомпьютеры, как *Illiac-IV, Cyber-205, Gray I* и др.

3. *Архитектура МКОД* (множество потоков команд - одиночный поток данных), или *MISD* (*Multiple Instruction stream - Single Date stream*), рис. 3.8.

Данный тип архитектуры предполагает построение своеобразного процессорного конвейера, в котором результаты вычислений словно по цепочке передаются от одного процессора к другому. В современных ЭВМ по этому принципу реализована схема совмещения операций, где параллельно рабо-

тают различные блоки и каждый из них выполняет свою часть в общем цикле обработки команды.

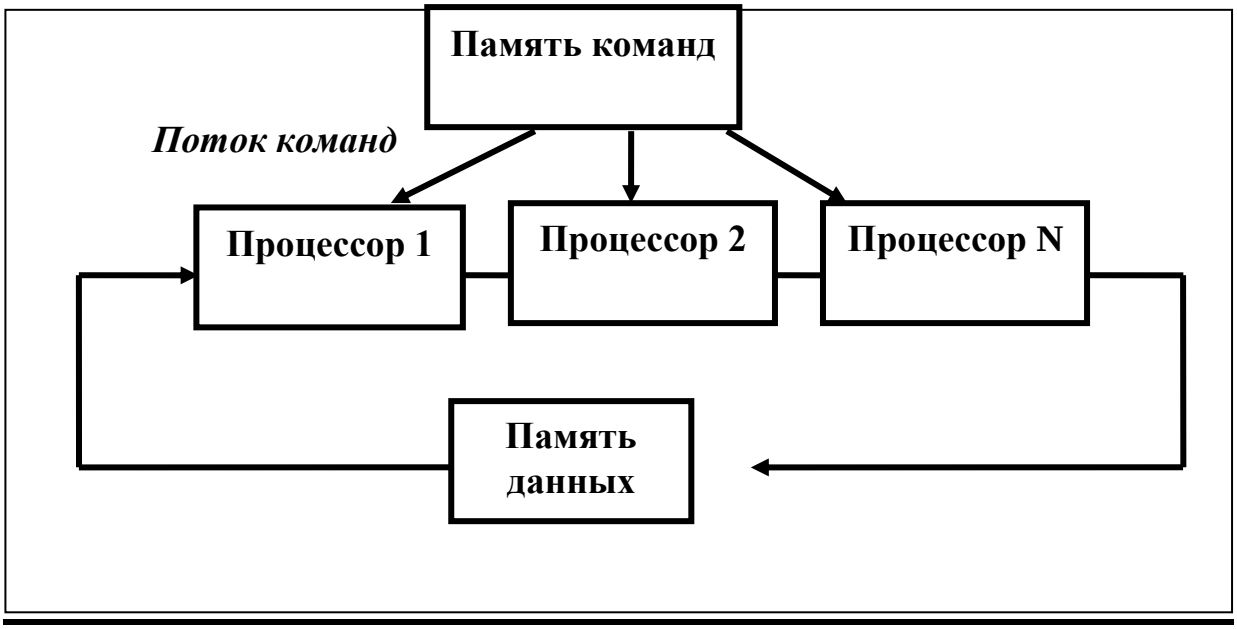


Рис. 3.8. **Архитектура МКОД**

4. *Архитектура МКМД* (множество потоков команд - множество потоков данных), или *MIMD* (*Multiple Instruction stream - Multiple Date stream*), рис. 3.9.

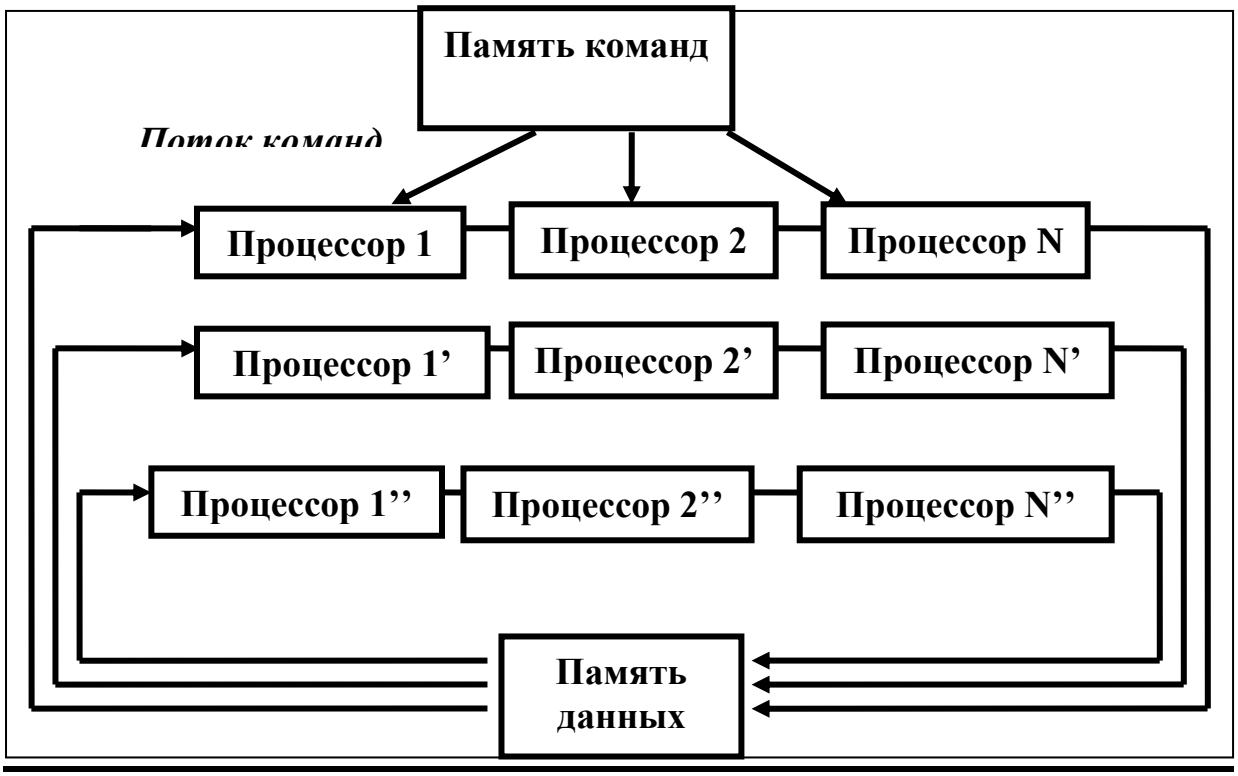


Рис. 3.9. **Архитектура МКМД**

Эта архитектура предполагает, что все процессоры системы работают по своим собственным программам с собственными потоками данных, причем процессоры могут быть независимыми друг от друга.

3.8. Режимы работы компьютеров

Под *режимом работы* компьютера понимают способ функционирования системы. Режим работы прежде всего определяется тем, как пользователь может участвовать в процессе обработки данных на компьютере: иметь непосредственный доступ к системе, либо принимать решение после завершения машиной задания.

При *непосредственном* доступе (характерен для персонального компьютера) пользователь может вмешиваться в процесс решения задачи через индивидуальное устройство (клавиатуру, пульт), вводя необходимые коррективы в данные и выполняемые машиной команды.

При доступе на уровне заданий пользователем предварительно определяется конкретный объем работы (задание) для системы, которое помещается в устройство ввода компьютера. Может быть подготовлено несколько автономных заданий, которые образуют так называемый *пакет заданий*. При этом каждое задание имеет свои исходные данные и управляющие операторы. Из сформированного пакета каждое задание поступает на обработку последовательно без участия человека, и пользователь не имеет возможности вмешиваться в процесс обработки заданий до полной обработки пакета.

При *многопрограммной* (мультипрограммной) организации обработки данных возможно несколько режимов работы компьютера: пакетный, режим разделения времени, диалоговый, режим "запрос — ответ", режим реального времени.

В отличие от машин с однопрограммной организацией решения задач компьютеры, работающие в многопрограммном режиме, располагают значительно более развитыми аппаратно-программными средствами. В числе ап-

паратных средств присутствуют таймеры, системы прерываний, сегментации и защиты памяти машины и др.; программное обеспечение дополняется специальными управляющими и контролирующими программами.

Пакетный режим. Суть пакетного режима рассмотрена выше. При многопрограммной обработке каждое задание идентифицируется в зависимости от важности задачи, срочности решения, информационной взаимосвязи задач. Параллельность выполнения заданий, выбор их очередности и выделение для них аппаратных ресурсов обеспечивает управляющая программа-супервизор.

Режим разделения времени. При этом режиме процессорное время последовательно поделено между группой обрабатываемых заданий (решаемых задач). По окончании выделенного задаче кванта времени задача возвращается в очередь ожидания обслуживания. При большом числе независимых пользователей каждый из них имеет одновременный и непосредственный доступ к компьютеру со своих терминалов (устройств ввода-вывода). Специальные аппаратно-программные средства обеспечивают распределение ресурсов системы между пользователями и могут изменять величины квантов времени между задачами.

Такой режим создает видимость у пользователя, что ему отданы все ресурсы компьютера и что его задача моментально включается в обработку.

Диалоговый режим. Характеризуется обменом сообщениями между пользователем и выполняемой программой. Выполнив законченный цикл обработки данных, диалоговая программа выдает пользователю информацию (результатную либо вопросного характера) и приостанавливает работу, ожидая реакцию со стороны пользователя. Пользователь может в любое время вмешаться в работу диалоговой программы, послав ей новые инструкции или приостановить ее работу.

Режим "запрос — ответ". Может быть организован для повышения загрузки ресурсов компьютера. Учитывая, что внешние запросы от пользователя могут поступать через какие-то промежутки времени, компьютер ре-

шает рутинные фоновые задачи. Поскольку машины специализируются на обработке запросов заранее определенных типов, каждому типу запросов должна соответствовать своя программа, постоянно находящаяся в памяти компьютера. Получив запрос, машина прерывает решение фоновой задачи, вызывает нужную программу и, выдав результаты запроса пользователю, вновь переходит к решению фоновой задачи. Этот режим работы компьютера удобен для таких задач, как резервирование мест в гостиницах, продажа билетов на различные виды транспорта, выдача справочных сообщений и др.

Режим реального времени. Применим для объектов, нуждающихся в непосредственном и постоянном контроле параметров их состояния и управления (космические объекты, технологические установки непрерывного производства). Обработка данных в этом режиме обеспечивается взаимодействием компьютера с внешними по отношению к ней процессами в темпе реальности этих процессов. Для этого ввод информации необходимо осуществлять с различного рода датчиков, а программа должна быть высоконадежной и находиться в постоянной готовности к приему сигналов.

3.9. Информация в технических устройствах

Обмен информацией в технических устройствах и системах осуществляется с помощью сигналов, отражающих физические характеристики объектов и процессов.

Существующие в технических устройствах сигналы делятся на ***непрерывные*** (или аналоговые) и ***дискретные***.

Непрерывность сигнала означает возможность его изменения на любую малую величину в любой заданный малый промежуток времени.

Примером аналоговой передачи сигнала является передача речи по телефонным проводам: речевая информация преобразуется в аналоговые электрические сигналы, которые по проводам передаются абоненту, а затем обратно преобразуются в речевую информацию.

До 1970-х г.г. технические устройства работали только с аналоговыми сигналами. Аналоговыми являлись и способы их обработки.

С появлением микропроцессоров и микросхем с высокой степенью интеграции стали получать распространение дискретные и цифровые сигналы и соответствующие способы их обработки.

Дискретность сигнала означает возможность его измерения только на конечном отрезке, в строго определенные моменты времени, то есть сам сигнал представляет собой не непрерывную функцию, а последовательность дискретных значений.

Дискретный сигнал, значения которого выражены определенными конечными числами, называется ***цифровым.***

Для обработки, хранения, передачи цифровых сигналов также существуют специальные технические устройства: аудио- и видео—компакт-диски, модемы и факсимильные средства связи.

Однако остается достаточно много систем и устройств, в которых информация может передаваться только в виде аналогового сигнала. В связи с этим используются способы преобразования аналогового сигнала в цифровой и обратно.

Современные компьютеры обрабатывают числовую, текстовую, графическую информацию, причем как в черно-белом, так и в цветном изображении. Для удобства работы с разнообразной информацией и прежде всего для ее хранения в современных компьютерах принята *байтовая организация памяти*, т.е. для представления каждого символа ему отводится 8 двоичных разрядов (бит) памяти, или 1 байт, рис. 3.9.

Последовательность нескольких битов и байтов часто называют полем данных. Поля постоянной длины:

- Полуслово – 2 байта;
- Слово – 4 байта;
- Двойное слово – 8 байт.

<table>
<tr><td>Байт</td><td>Байт</td><td>Байт</td><td>Байт</td><td>Байт</td><td>Байт</td><td>Байт</td><td>Байт</td></tr>
<tr><td colspan="2">Полуслово</td><td colspan="2">Полуслово</td><td colspan="2">Полуслово</td><td colspan="2">Полуслово</td></tr>
<tr><td colspan="4">Слово</td><td colspan="4">Слово</td></tr>
<tr><td colspan="8">Двойное слово</td></tr>
</table>

Рис. 3.10. **Байтовая структура памяти**

Для удобства работы с информацией введены следующие термины для обозначения совокупностей двоичных разрядов, табл. 3.3.

Для решения любой вычислительной задачи цифровая информация должна быть введена и зафиксирована в памяти машины, а перед непосредственным выполнением над ней вычислительных операций размещена в специальных устройствах - *регистрах*. Длина регистров, т.е. количество разрядов обрабатываемого двоичного числа, для различных типов современных компьютеров различная (определяется принятой архитектурой машины), но, как правило, составляет 16, 32, 64 разряда.

Таблица 3.3.

Единицы измерения памяти

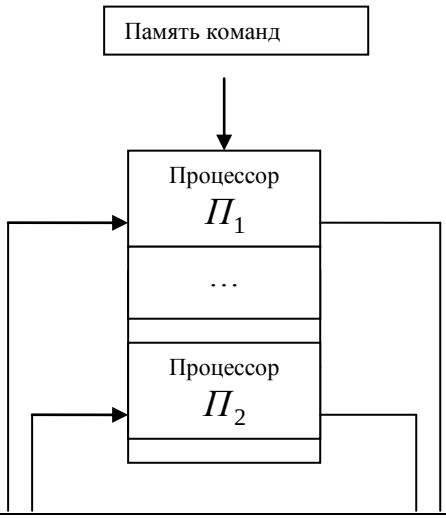
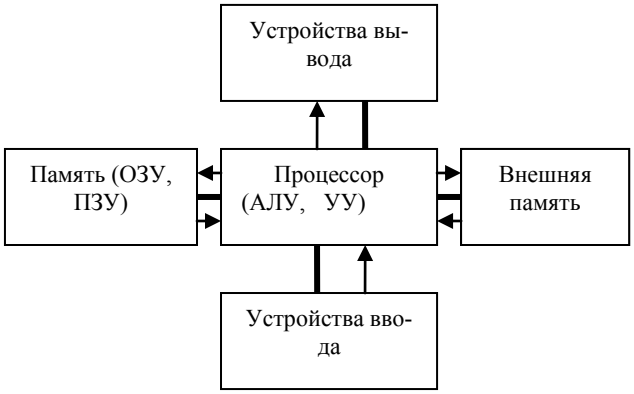
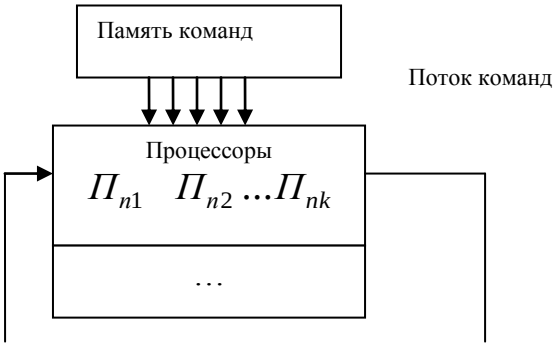
Количество двоичных разрядов в группе	**Наименование единицы измерения**
1	Бит
8	Байт
16	Полуслово
32	Слово
64	Двойное слово
8*1024	Кбайт (килобайт)
$8*1024^2$	1 Мбайт (мегабайт)
$8*1024^3$	1 Гбайт (гигабайт)
$8*1024^4$	1 Тбайт (терабайт)
$8*1024^5$	1 Пбайт (пентабайт)

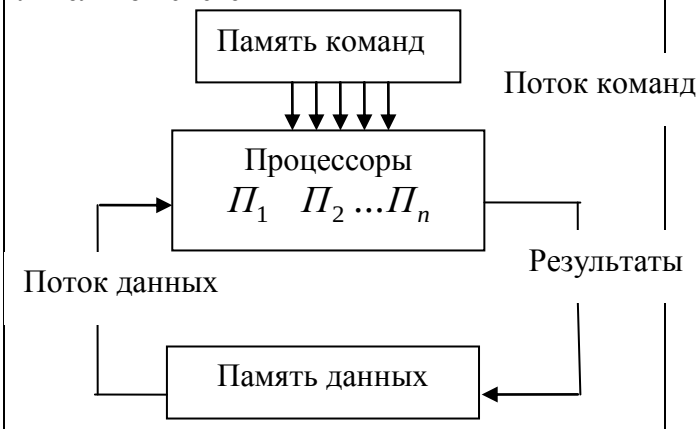
Вопросы для самоконтроля

1. В чем принципиальные различия между аналоговыми и цифровыми машинами?
2. Какие признаки характеризуют термин "поколение ЭВМ"?
3. Какие существуют подходы в современной классификации ЭВМ?
4. Что означает показатель "производительность ЭВМ"?
5. В чем принципиальное различие между центральными и внешними устройствами ЭВМ?
6. В чем заключается суть программного управления работой компьютеров?
7. Как формируется адрес очередной команды при программном управлении?
8. Назовите основные единицы хранения информации в памяти компьютера.
9. Что представляют собой четыре архитектуры построения вы числительных систем?
10. Какие режимы работы ЭВМ наиболее характерны для обработки данных?

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	На рисунке представлена структура вычислительной системы	• с одним потоком команд и одним потоком данных • с одним потоком команд и множеством потоком данных • с множеством потоком команд и множественным потоком данных • с множественным потоком команд и одним потоком данных

		
2.	На рисунке представлена функциональная схема ЭВМ, предложенная… 	• Р. Хартли • Биллом Гейтсом • С.А. Лебедевым • Дж. Фон Нейманом
2.	Правильный порядок значений по убыванию.	• 1 терабайт, 1 мегабайт, 1 гигабайт, 1 петабайт • 1 терабайт, 1 гигабайт, 1 петабайт, 1 мегабайт • 1 мегабайт, 1 гигабайт, 1 терабайт, 1 петабайт • 1 петабайт, 1 терабайт, 1 гигабайт, 1 мегабайт
3.	На рисунке представлена _____ схема вычислительной системы 	• многопроцессорная векторная • многопроцессорная магистральная • однопроцессорная • многопроцессорная матричная
4.	Совокупность ЭВМ и программного обеспечения называется…	• вычислительной системой • встроенной системой • интегрированной системой

		• построителем кода
5.	На рисунке представлена структура вычислительной системы  	• с множественным потоком команд и одним потоком данных с множественным потоком команд и множественным потоком данных • с одним потоком команд и множественным потоком данных с одним потоком команд и одним потоком данных
6.	Компьютеры, созданные для решения предельно сложных вычислительных задач, – это …	• суперкомпьютеры • серверы • карманные персональные компьютеры • персональные компьютеры
7.	Минимальной единицей адресуемой памяти в компьютере является…	• 1 герц • 1 байт • 1 бит • 1 килобайт
8.	1 гигабайт содержит…	• 1024 килобайт • 10000 мегабайт • 1000 килобайт • 1024 мегабайт
9.	Основные принципы построения цифровых вычислительных машин были разработаны…	• российским ученым академиком С.А. Лебедевым • Ч. Беббиджем в Англии • Адой Лавлейс • американским ученым Дж. фон Нейманом
10.	Для обработки в оперативной памяти компьютера числа преобразуются в…	• графические образы • символы латинского алфавита • числовые коды в восьмеричной форме • числовые коды в двоичной форме
11.	Количество двоичных разрядов, отводимых для машинной команды, определяет _____ процессора.	• емкость • частоту • разрядность • объем
12.	Последовательность смены элементной базы ЭВМ: а) дискретные полупроводниковые приборы б) электронно-вакуумные лампы в) интегральные микросхемы	• в), а), б) • б), а), в) • б), в), а) • а), б), в)
13.	Принципы функционирования компьютера фон Неймана включает:	• б, г • а, б

<table>
<tr><td></td><td>а) данные и программы, должны быть представлены в двоичной системе
б) ячейки памяти должны иметь адреса для доступа к ним
в) обязательное наличие внешней памяти (винчестера)
г) наличие операционной системы</td><td>• а, в
• б, в</td></tr>
<tr><td>14.</td><td>Персональные компьютеры относятся к…</td><td>• классу машин 3-го поколения
• особому классу машин
• классу машин 4-го поколения
• классу машин 2-го поколения</td></tr>
<tr><td>15.</td><td>Элементной базой первого поколения ЭВМ являлись…</td><td>• чипы
• полупроводниковые схемы
• транзисторы
• электронно-вакуумные лампы</td></tr>
<tr><td>16.</td><td>1024 килобайта равно…</td><td>• 1 мегабайту
• 1 гигабайту
• 1 мегаботу
• 1 мегабайту</td></tr>
<tr><td>17.</td><td>Для информационной техники предпочтительнее _____ вид сигнала.</td><td>• синхронизированный
• зашумленный
• цифровой
• непрерывный</td></tr>
<tr><td>18.</td><td>Выберите вариант, в котором объемы памяти расположены в порядке возрастания.</td><td>• 15 бит, 20 бит, 2 байта, 1010 байт, 1 Кбайт
• 15 бит, 20 бит, 2 байта, 1 Кбайт, 1010 байт
• 15 бит 2 байта, 20 бит, 1 Кбайт, 1010 байт
• 15 бит, 2 байта, 20 бит, 1010 байт, 1 Кбайт</td></tr>
<tr><td>19.</td><td>Функциональной частью компьютера, предназначенной для приема, хранения и выдачи данных, является…</td><td>• процессор
• оперативная память (ОЗУ)
• графопостроитель
• монитор</td></tr>
<tr><td>20.</td><td>Верным(и) является(ются) утверждение(я):
a) При выключении компьютера содержимое внешней памяти исчезает.
b) Сетевая плата является устройством приема-передачи данных.
c) Флоппи-диск является носителем информации.
d) Джойстик не является устройством ввода данных.</td><td>• b и c
• b и c и d
• d
• b и d</td></tr>
<tr><td>21.</td><td>Установите правильное соответствие между названиями и описаниями.
<table>
<tr><td>A. CMOS</td><td>1. Является энергозависимой.</td></tr>
<tr><td>B. ОЗУ</td><td>2. Запоминающее устройство носителем.</td></tr>
<tr><td>C. Дисковод</td><td>3. Является энергонезависим.</td></tr>
</table></td><td>• A-1, B-2, C-3
• A-2, B-1, C-3
• A-3, B-1, C-2
• A-2, B-3, C-1</td></tr>
</table>

	компакт-дисков минимальным энергопотреб.	
22.	Устройством, в котором хранение данных возможно только при выключенном питании компьютера, является…	• постоянная память (ПЗУ) • жесткий диск • оперативная память (ОЗУ) • гибкий магнитный диск
23.	Аппаратная кэш-память компьютера используется для…	• увеличения объема энергонезащищенной памяти • обмена информацией компьютер с периферийным устройством • уменьшения сбоев в работе компьютера • увеличения производительности процессора
24.	.Внешняя память компьютера предназначена для…	• долговременного хранения только данных, но не программ • долговременного хранения данных и программ • долговременного хранения только программ, но не данных • кратковременного хранения обрабатываемой в данный момент информации
25.	Совокупность ЭВМ и программного обеспечения называется…	• вычислительной системой • встроенной системой • интегрированной системой • построителем кода
26.	Устройствами ввода данных являются: а) жесткий диск б) джойстик в) мышь г) регистры д) привод CD-ROM	• в, г, д • б, д • б, в • б, в, г
27.	Основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой, называется …	• системой ввода/вывода • системой мультиплексирования • системной шиной • шиной питания
28.	Последовательность смены элементной базы ЭВМ: а) дискретные полупроводниковые приборы б) электронно-вакуумные лампы в) интегральные микросхемы	• в), а), б) • б), а), в) • б), в), а) • а), б), в)
29.	В состав внутренней памяти ЭВМ входят…	• постоянная память, оперативная память и кэш-память • накопители на гибких магнитных дисках • накопители на компакт-дисках

		• накопители на жестких магнитных дисках
30.	Процессор выполняет…	• представление данных в доступной человеческому восприятию форме • постоянное хранение данных и программ их обработки • обработку всех видов информации • генератор импульсов
31.	Устройством, в котором хранение данных возможно только при выключенном питании компьютера, является…	• постоянная память (ПЗУ) • гибкий магнитный диск • жесткий диск • оперативная память (ОЗУ)
32.	Устройством вывода является…	• модем • дисплей • дисковод • мышь
33.	Арифметико-логическое устройство в составе микропроцессора персонального компьютера предназначено для…	• распознавания и декодирования команд • установление логической последовательности выполнения элементарных операций • выборки данных из оперативной памяти • выполнения арифметических операций
34.	Устройством персонального компьютера, связывающим его с телефонной линией, является …	• мультиплексор • шлюз • факс • модем
35.	Устройством для резервного копирования больших объемов информации является …	• архиватор • сканер • плоттер • стример
36.	Разрядностью микропроцессора является …	• физический объем регистров микропроцессора • ширина шины адреса микропроцессора • размер кэш-памяти • количество бит, обрабатываемых микропроцессором за один такт работы
37.	Аббревиатура ROM расшифровывается как …	• внешняя память • память с последовательным доступом • память с произвольным доступом • память только для чтения
38.	ПЗУ является _____ памятью.	• энергонезависимой • динамической • энергозависимой

		• оперативной с произвольным доступом
39.	Внешними запоминающими устройствами являются: 1) жесткий диск 2 оперативная память (ОЗУ) 3) стример 4) кэш-память	• 2 и 4 • 3 и 4 • 1 и 3 • 1 и 2
40.	Принципы функционирования компьютера фон Неймана включает: а) данные и программы, должны быть представлены в двоичной системе б) ячейки памяти должны иметь адреса для доступа к ним в) обязательное наличие внешней памяти (винчестера) г) наличие операционной системы	• б, г • а, б • а, в • б, в
41.	Верным(и) является(ются0 утверждение(я): a) Сетевая плата не является устройством приема-передачи данных. b) Микропроцессор не имеет элементов памяти. c) Флэш-память является долговременной памятью. d) В мониторах на жидких кристаллах отсутствует электромагнитное излучение.	• b и d • c и d • a • b, c, d
42.	Энергозависимыми устройствами памяти является …	• Flash USB Drive • регистры микропроцессора • ОЗУ • кэш-память
43.	Укажите, какие из следующих высказываний являются истинными. а) Появление второго поколения ЭВМ было обусловлено переходом от электронных ламп к транзисторам. б) В ЭВМ первого поколения отсутствовало устройство управления. в) В ЭВМ первого поколения отсутствовала оперативная память. г) Машины третьего поколения – это семейства машин с единой архитектурой, то есть программно совместимых. д) Компьютер с процессором Intel Pentium III относится к четвертому поколению ЭВМ.	• а, г, д • а, б, г • б, в, д • б, в, г
44.	Центральный процессор персонального компьютера выполняет	• генерацию импульсов • систематизацию данных • постоянное хранение данных и программ после их обработки • обработку всех видов информации
45.	На производительность микропроцессорной	• частота тактового генератора

	системы **не влияет**…	• количество внешних устройств • разрядность системной шины • организация интерфейса памяти

Глав 4. Способы представления информации в компьютерах

4.1. Системы счисления

4.1.1. Позиционные системы счисления

Системы счисления – это совокупность приемов и правил наименования и обозначения чисел, позволяющих установить взаимнооднозначное соответствие между любым числом и его представлением в виде конечного числа символов.

Все системы счисления можно разделить на позиционные и непозиционные.

Непозиционная система счисления – это система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от местоположения (позиции) в изображении числа.

Непозиционной системой счисления является самая простая система с одним символом (палочкой). Для изображения какого-либо числа в этой системе надо записать количество палочек, равное данному числу.

Например, запись числа 12 в этой системе будет иметь вид: 111111111111, где каждая палочка обозначена символом 1.

В общем случае непозиционные системы счисления характеризуются сложным способом записи чисел и правилами выполнения арифметических операций. В настоящее время все наиболее распространенные системы счисления относятся к разряду позиционных.

Систему счисления, в которой значение цифры определяется ее местоположением (позицией) в изображении числа, называют ***позиционной***.

Упорядоченный набор символов (цифр) $\{a_0, a_1, \ldots, a_n\}$, используемый для представления любых чисел в заданной позиционной системе счисления, называют ее *алфавитом*, число символов (цифр) алфавита $p=n+1$ – ее *основанием*, а саму систему счисления называют *p-ичной*.

Основание позиционной системы счисления – количество различных символов, используемых для изображения чисел в данной системе счисления.

Самой привычной является десятичная система счисления. Ее алфавит – {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, а основание p=10, то есть в этой системе для записи любых чисел используется только десять разных символов (цифр).

Десятичная система счисления основана на том, что десять единиц каждого разряда объединяются в одну единицу соседнего старшего разряда, поэтому каждый разряд имеет вес, равный степени 10. Следовательно, значение одной и той же цифры определяется ее местоположением в изображении числа, характеризуемым степенью числа 10.

Например, в изображении числа 222.22 цифра 2 повторяется пять раз, при этом первая слева цифра 2 означает количество сотен (ее вес равен 10^2); вторая – количество десятков (ее вес равен 10^1), третья - количество единиц (ее вес равен 10^0), четвертая количество десятых долей единицы (ее вес равен 10^{-1}) и пятая цифра – количество сотых долей единицы (ее вес равен 10^{-2}). То есть число 222.22 может быть разложено по степеням числа 10:

$$222.22 = 2 \times 10^2 + 2 \times 10^1 + 2 \times 10^0 + 2 \times 10^{-1} + 2 \times 10^{-2}$$

Аналогично любое число в десятичной системе счисления можно представить следующим образом:

$$1304.5 = 1 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

Таким образом, любое число A можно представить в виде полинома путем разложения его по степеням числа 10:

$$A_{10} = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \ldots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + \ldots$$
$$+ a_{-m} \times 10^{-m} + \ldots,$$

Последовательность из коэффициентов которого представляет собой десятичную запись числа A_{10}:

$$A_{10} = a_n\, a_{n-1} \ldots a_1\, a_0 \,.\, a_{-1} \ldots a_{-m},$$

Точка, отделяющая целую часть числа от дробной, служит для фиксации конкретных значений каждой позиции в этой последовательности цифр и является началом отсчета.

В общем случае для задания *p*-ичной системы счисления необходимо определить основание *p* и алфавит, состоящий из *p* различных символов (цифр) a_i, $i=1,\ldots, p$.

За основание системы можно принять любое натуральное число – два, три, четыре и т.д. Обычно в качестве алфавита берутся последовательные целые числа от 0 до (*p* - 1) включительно.

Для записи произвольного числа в двоичной системе счисления используются цифры 0, 1, троичной – 0, 1, 2, пятеричной – 0, 1, 2, 3 и т.д. В тех случаях, когда общепринятых (арабских) цифр не хватает для обозначения всех символов алфавита системы счисления с основанием $p>10$, используют буквенное обозначение цифр a, b, c, d, e, f.

В таблице 4.1 приведены алфавиты некоторых систем счисления.

Таблица 4.1.

Системы счисления

Основание	Система счисления	Алфавит системы счисления
2	Двоичная	0, 1
3	Троичная	0, 1, 2
4	Четвертичная	0, 1, 2, 3
5	Пятеричная	0, 1, 2, 3, 4
8	Восьмеричная	0, 1, 2, 3, 4, 5, 6, 7
10	Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
12	Двенадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
16	Шестнадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Таким образом, возможно бесчисленное множество позиционных систем счисления: двоичная, троичная, четверичная и т.д.

Запись чисел в одной из систем счисления с основанием *p* означает сокращенную запись выражения:

$$A_p = a_n x\, p^n + a_{n-1} x\, p^{n-1} + \ldots + a_1 x\, p^1 + a_0 x\, p^0 + a_{-1} x\, p^{-1} + \ldots + a_{-m} x\, p^{-m},$$

где a_i – цифры системы счисления; n и m – число целых и дробных разрядов, соответственно, A_p – запись числа A в *p*-ичной системе счисления.

Изображением числа A в *p*-ичной системе счисления является последовательность цифр a_k.

4.1.2. Перевод чисел из одной системы счисления в другую

Рассмотрим задачу перевода чисел из одной системы счисления в другую. Пусть известна запись числа A в системе счисления с основанием *p*:

$$A_p = a_з x\, p^n + a_{n-1} x\, p^{n-1} + \ldots + a_1 x\, p^1 + a_0 x\, p^0 + a_{-1} x\, p^{-1} + \ldots + a_{-m} x\, p^{-m},$$

где a_i – цифры *p*-ичной системы счисления.

Требуется найти запись этого числа A в системе счисления с основанием *d*:

$$A_d = b_n x\, d^n + b_{n-1} x\, d^{n-1} + \ldots + b_1 x\, d^1 + b_0 x\, d^0 + b_{-1} x\, d^{-1} + \ldots + b_{-m} x\, d^{-m},$$

где b_i – цифры *d*-ичной системы счисления.

При переводе чисел из *p*-ичной системы счисления в *d*-ичную ($A_p \rightarrow A_d$) нужно учитывать, средствами какой арифметики должен быть осуществлен перевод, то есть в какой системе счисления (*p*-ичной или *d*-ичной) должны быть выполнены все действия.

Пусть перевод $A_p \rightarrow A_d$ должен осуществляться средствами d-ичной арифметики. В этом случае перевод произвольного числа A, заданного в системе счисления с основанием *p*, в систему счисления с основанием *d* выполняется по *правилу замещения*.

Правило замещения чаще всего используется для преобразования чисел из любой системы счисления в десятичную.

Перевод в десятичную систему числа A, записанного в *p*-ичной системе счисления в виде $A_p = (a_n\, a_{n-1} \ldots\, a_1\, a_0 \,.\, a_{-1} \ldots\, a_{-m})_p$ сводится к вычислению мно-

гочлена $A_{10}= a_n x\ p^n + a_{n-1} x\ p^{n-1} + \ldots + a_1 x\ p^1 + a_0 x\ p^0 + a_{-1} x\ p^{-1} + \ldots + a_{-m} x\ p^{-m}$ средствами десятичной арифметики.

Пример 1. Переведем число $A_2 = 1011{,}1$ в десятичную систему счисления.

Разряды 3 2 1 0 -1

Число $1\ 0\ 1\ 1,1_2 = 1 x\ 2^3 + 1 x\ 2^2 + 1 x\ 2^1 + 1 x\ 2^0 + 1 x\ 2^{-1} = 11{,}5_{10}$

$2\ 7\ 6,5_8 = 2 x\ 8^2 + 7 x\ 8^1 + 6 x\ 8^0 + 5 x\ 8^{-1} = 190{,}625_{10}$

$1\ F\ 3_{16} = 1 x\ 16^2 + F x\ 16^1 + 3 x\ 16^0 = 499_{10}$

Пусть теперь перевод $A_p \rightarrow A_d$ должен осуществляться средствами p-ичной арифметики. В этом случае для перевода любого числа используется *правило деления* – для перевода целой части числа и *правило умножения* – для перевода дробной части.

Для перевода целого числа A_p из *p*-ичной системы счисления в систему счисления с основанием *d* необходимо A_p разделить с остатком “нацело” на число *d*, записанное в той же *p*-ичной системе счисления. Затем неполное частное, полученное от такого деления, нужно снова разделить с остатком на *d* и т.д., пока последнее полученное частное не станет равным нолю.

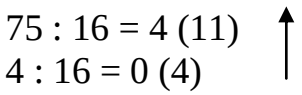
Представлением числа A_p в новой системе счисления будет последовательность остатков деления, изображенных *d*-ичной цифрой и записанных в порядке, обратном их получения.

Пример 2. Переведем число $A_{10} = 47$ в двоичную систему счисления с использованием десятичной арифметики, при *d*=2, имеем:

47 : 2 = 23 (1)
23 : 2 = 11 (1)
11 : 2 = 5 (1)
5 : 2 = 2 (1)
2 : 2 = 1 (0)
1 : 2 = 0 (1) ↑

В процессе деления получим двоичное изображения искомых цифр A_2 = 101111.

Пример 3. Переведем число A_{10} = 75 в шестнадцатеричную систему счисления с использованием десятичной арифметики, при *d*=16, имеем:



Первый остаток 11_{10} в 16-ричной системе счисления обозначается шестнадцатиричной цифрой B_{16}, поэтому окончательно получим: A_{16} = 4B.

4.1.3. Двоичная, восьмеричная и шестнадцатеричная системы счисления

Показал полезность применения двоичной системы немецкий математик Г. Лейбниц в 1703 г. Однако лишь благодаря работам Дж. Фон Неймана, опубликованным в 1940-х гг., двоичная система получила практическое использование при создании компьютерных средств.

Применение двоичной системы в вычислительной технике было обусловлено такими обстоятельствами, как двухпозиционный характер работы электронных элементов, высокая экономичность двоичной системы счисления и простота выполнения операций с двоичными числами. Как отмечалось в отчете Дж. Фон Неймана (1946 г.): "основное же преимущество двоичной системы по сравнению с десятичной состоит в том, что основная часть машины по своему характеру является не арифметической, а логической. Новая логика, будучи системой типа "да - нет", в основном двоична. Поэтому двоичное построение арифметических устройств существенно содействует построению более однородной машины, которая может быть лучше скомпонована и более эффективна".

В современной вычислительной технике, в устройствах автоматики и связи используется в основном двоичная система счисления, что обусловлено рядом преимуществ перед другими системами. Так, для ее реализации нуж-

ны технические устройства лишь с двумя устойчивыми состояниями, например материал намагничен или размагничен. Это обеспечивает более надежное и помехоустойчивое представление информации, дает возможность применения аппарата булевой алгебры для выполнения логических преобразований информации. Кроме того, арифметические операции в двоичной системе счисления выполняются наиболее просто.

Недостаток двоичной системы – быстрый рост числа разрядов, необходимых для записи больших чисел. Этот недостаток имеет существенное значение. Если возникает необходимость кодировать информацию “вручную”, например при составлении программы на машинном языке, используют восьмеричную или шестнадцатеричную системы счисления.

Примеры изображения чисел в данных системах счисления представлены в таблице 4.2.

Таблица 4.2.

Представление чисел в двоичной, восьмеричной и шестнадцатеричной системах счисления

10-ичная	2-ичная	8-ичная	16-ичная
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему счисления осуществляется путем замены каждой цифры эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

Пример 4 . Переведем число $537,1_8$ в двоичную систему счисления.

$$537,1_8 = \underset{5}{\downarrow}101\ \underset{3}{\downarrow}011\ \underset{7}{\downarrow}111,\ \underset{1}{\downarrow}001_2$$

Пример 5 . Переведем число $1A3,F_{16}$ в двоичную систему счисления.

$$1A3,F_{16} = \underset{1}{\downarrow}1\ \underset{A}{\downarrow}1010\ \underset{3}{\downarrow}0011,\ \underset{F}{\downarrow}1111_2$$

Чтобы *перевести число из двоичной системы* счисления в восьмеричную или шестнадцатеричную, его нужно разбить влево или вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной или шестнадцатеричной цифрой.

Пример 6 . Переведем число $10101001,10111_2$ в восьмеричную систему счисления.

$$10101001,10111_2 = \underset{2}{\downarrow}10\ \underset{5}{\downarrow}101\ \underset{1}{\downarrow}001,\underset{5}{\downarrow}101\ \underset{6}{\downarrow}110_2 = 251,56_8$$

Пример 7. Переведем число $10101001,10111_2$ в шестнадцатеричную систему счисления.

$$10101001,10111_2 = \underset{A}{\downarrow}1010\ \underset{9}{\downarrow}1001,\underset{B}{\downarrow}1011\ \underset{8}{\downarrow}1000_2 = A9,B8_{16}$$

4.1.4. Выполнение арифметических операций в двоичной, восьмеричной и шестнадцатеричной системах счисления

Правила выполнения арифметических операций сложения, вычитания, умножения и деления в 2-, 8- и 16-ичной системах счисления будут такими же, как и в десятичной системе, только надо пользоваться особыми для каждой системы таблицами сложения и умножения.

Таблицы сложения для 2-ичной, 8-ичной и 16-ичной систем счисления представлены в таблицах 4.3-4.5.

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

Таблица 4.3.

Сложение в двоичной системе

+	**0**	**1**
0	0	1
1	1	10

Таблица 4.4.

Сложение в восьмеричной системе

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица 4.5.

Сложение в шестнадцатеричной системе

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Пример 8. Сложим десятичные числа 15 и 6 в 2-, 8- и 16-ичной системах счисления.

Двоичная система: $1111_2 + 110_2 = 10101_2$

$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 1111 \\ + \quad 0110 \\ \hline 10101 \end{array}$$

Восьмеричная система: $17_8 + 6_8 = 25_8$

1

$$\begin{array}{r} 17 \\ + \quad 6 \\ \hline 25 \end{array}$$

Шестнадцатеричная система: $F_{16} + 6_{16} = 15_{16}$

$$\begin{array}{r} {\scriptstyle 1} \\ F \\ + \quad 6 \\ \hline 15 \end{array}$$

Ответ: $15 + 6 = 21_{10} = 10101_2 = 25_8 = 15_{16}$

Проверка. Преобразуем полученные суммы к десятичному виду:

$$10101_2 = 2^4 + 2^2 + 2^0 = 16 + 4 + 1 = 21$$

$$25_8 = 2*8^1 + 5*8^0 = 16 + 5 = 21$$

$$15_{16} = 1*16^1 + 5*16^0 = 16 + 5 = 21$$

4.2. Представление числовой информации. Прямой, обратный и дополнительный коды числа

Информация в памяти компьютера записывается в виде цифрового *двоичного кода.* С этой целью компьютер содержит большое количество ячеек памяти и регистров для хранения двоичной информации. Большинство этих ячеек имеет одинаковую длину *n*, и используются для хранения *n* бит двоичной информации (бит – один двоичный разряд).

В вычислительных машинах применяются две формы представления двоичных чисел:

- естественная форма, или форма с фиксированной запятой (точкой);
- нормальная форма, или форма с плавающей запятой (точкой).

С *фиксированной запятой* все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой (точкой), отделяющей целую часть от дробной.

С *плавающей запятой* каждое число изображается в виде двух групп цифр. Первая группа цифр называется *мантиссой*, вторая – *порядком*, при-

чем абсолютная величина мантиссы должна быть меньше 1, а порядок – целым числом. В общем виде число в форме с плавающей запятой может быть представлено в виде:

$$X = M * q^{p},$$

где M – мантисса, p – порядок числа, q – основание системы счисления.

Нормальная форма представления чисел является основной в современных персональных компьютерах.

Целые числа в памяти компьютера могут представляться без знака или со знаком.

Целые числа без знака. Обычно занимают в памяти компьютера один или два байта, табл. 4.6. В однобайтовом формате принимают значения от 00000000_2 до 11111111_2. В двухбайтовом формате принимают значения от 00000000 00000000_2 до 11111111 11111111_2.

Таблица 4.6.

Диапазон значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$0 \dots 2^8 - 1$	0 … 255
2	$0 \dots 2^{16} - 1$	0 … 65535

Пример 9. Представим число $72_{10} = 1001000_2$ в однобайтовом формате.

Номер разрядов	7	6	5	4	3	2	1	0
Биты числа	0	1	0	0	1	0	0	0

Пример 10. Представим число $72_{10} = 1001000_2$ в двухбайтовом формате.

Номер разрядов	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**

Пример 11. Представим число 65535 в двухбайтовом формате.

Номер разрядов	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

числа																	

Целые числа со знаком. Целые числа со знаком обычно занимают в памяти компьютера один, два или четыре байта, табл. 4.7

Таблица 4.7.

Диапазон значений целых чисел со знаком

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$-2^7 \dots 2^7 - 1$	-128 … 127
2	$-2^{15} \dots 2^{15} - 1$	-32768 … 32768
4	$-2^{31} \dots 2^{31} - 1$	-2147483648 … 2147483647

В компьютерах применяются три формы записи (кодирования) целых чисел со знаком: *прямой код, дополнительный код, обратный код*.

Прямой код. Прямой *n*-разрядный двоичный код отличается от двоичного тем, что в нем отводится один, как правило, самый старший разряд для знака, а оставшиеся *n-1* разрядов – для значащих цифр.

Значение знакового разряда равно 0 для положительных чисел, и 1 – для отрицательных.

Пример 12. Представим число $1_{10} = 1_2$ в прямом коде.

Номер разрядов	7	6	5	4	3	2	1	0
Биты числа	0	0	0	0	0	0	0	1

↑ ——Знак числа “+”

Пример 13. Представим число -1_{10} в прямом коде.

Номер разрядов	7	6	5	4	3	2	1	0
Биты числа	1	0	0	0	0	0	0	1

↑ ——Знак числа “-”

Дополнительный код. Использование чисел со знаком (прямого кода представления чисел) усложняет структуру компьютера. Поэтому в современных компьютерах, как правило, отрицательные числа представляются в виде дополнительного или обратного кода, что при суммировании двух чисел

с разными знаками позволяет заменить вычитание на обычное сложение и упростить тем самым конструкцию арифметико-логического устройства компьютера.

Можно дать простое правило для получения дополнительного кода двоичных чисел:

1. Получить инверсию заданного числа (все его 0 заменить на 1, а все 1 – на 0):

0 000 0010 1100 0101	число
1 111 1101 0011 1010	инверсия числа

2. Образовать дополнительный код заданного числа путем дополнения 1 к инверсии этого числа.

1 111 1101 0011 1010	инверсия числа
+ 1	слагаемое 1
1 111 1101 0011 1011	дополнительный код числа

Проверим правильность перевода:

0 000 0010 1100 0101	число
+ 1 111 1101 0011 1011	дополнительный код числа
10 000 0000 0000 0000	0

Так как перенос старшего разряда не учитывается, то результат суммирования равен 0, что подтверждает правильность преобразования.

Старший бит дополнительного кода двоичных чисел выполняет функцию знака числа, то есть равен 0 для положительных чисел и 1 – для их дополнений (отрицательных чисел). При этом положительные числа в дополнительном коде изображаются так же, как и в прямом – двоичными кодами с цифрой 0 в знаковом разряде.

Обратный код. Для представления отрицательных чисел используется также обратный код, который получается инвертированием всех цифр двоичного кода абсолютной величины числа: ноли заменяются единицами, а единицы – нолями. При этом, необходимо помнить, что все операции с отрицательными числами выполняются в формате машинного слова. Это значит,

что к двоичному числу слева дописываются ноли до нужного количества разрядов.

Пример 14. Представим число -1_{10} в обратном коде.

Число: -1

Код модуля числа: 0000 0001

Обратный код числа: 1111 1110

Пример 15. Представим число -127_{10} в обратном коде.

Число: -127

Код модуля числа: 0111 1111

Обратный код числа: 1000 0000

Таким образом, положительные числа в прямом, обратном и дополнительных кодах изображаются одинаково – двоичными кодами с цифрой 0 в знаковом разряде.

Отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся. При выводе таких чисел из машины происходит обратное преобразование в отрицательные десятичные числа.

4.3. Представление символьной информации

Символьная информация хранится и обрабатывается в компьютере в форме цифрового кода, то есть каждому символу ставится в соответствие отдельное бинарное слово-код. Так как многие типы информации содержат в значительном объеме цифровую информацию, то применяются две системы кодирования: символьной информации и десятичных чисел.

Необходимый набор символов, предусмотренный в конкретном компьютере, обычно включает в себя:

- буквенно-цифровые знаки алфавита;
- специальные знаки (пробел, скобки, знаки препинания и др.);
- знаки операций.

Среди наборов символов наибольшее распространение получили знаки кода *ASCII* (*ASCII – American Standard Code for Information Interchange*) - американский стандартный код обмена информацией.

ASCII - это семиразрядный код, обеспечивающий 128 различных битовых комбинаций. Стандартный знакогенератор современного персонального компьютера IBM PC имеет 8-битовую кодировку символов, состоящую из двух таблиц кодирования: базовой и расширенной. Базовая таблица построена по стандарту *ASCII* и одинакова для всех IBM-совместимых компьютеров. Расширенная таблица относится к символам с номерами от 128 до 255 и может отличаться на компьютерах разного типа.

Для представления букв русского алфавита в рамках *ASCII* первоначально был разработан вариант кодировки – КОИ-7 (код обмена информацией 7-битный). Расположение символов во второй половине таблицы этой кодировки резко отличается от принятого фирмой IBM, что затрудняет использование зарубежного программного обеспечения на отечественных машинах.

В настоящее время находят широкое применение и другие виды кодировки. Так, в связи с массовым использованием операционных систем и других продуктов компании Microsoft в нашей стране нашла применение кодировка символов русского языка, известная как кодировка *Windows-1251*. Эта кодировка используется на большинстве персональных компьютеров, работающих на платформе Windows.

Другая распространенная кодировка носит название КОИ-8 (код обмена информацией восьмизначный). Сегодня кодировка КОИ-8 имеет широкое распространение в компьютерных сетях на территории России и в некоторых службах российского сектора Интернета. В частности, в нашей стране она является стандартом в сообщениях электронной почты и телеконференций.

В последнее время все большее распространение получает универсальная система кодирования текстовых данных – *UNICODE*. В данной системе символы кодируются не восьмиразрядными двоичными числами, а 16-

разрядными числами. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65536 различных символов – этого достаточно для размещения в одной таблице всех широко употребляемых языков.

4.4. Представление графической информации

Современные компьютерные системы способны обрабатывать не только текстовые и цифровые данные. Они позволяют работать также с изображениями и с аудио- и видеоинформацией. В отличие от методов представления символьной и числовой информации, для представления изображений, аудио- и видеоинформации пока не существует общепринятых стандартов.

Наиболее распространенные из существующих методов представления изображений можно разделить на две большие категории: *растровые методы и векторные методы.*

При ***растровом методе*** изображение представляется как совокупность точек, называемых *пикселями* (элемент изображения). Поскольку линейные координаты и индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных.

Для кодирования цветных графических изображений применяется *принцип декомпозиции* произвольного цвета на основные составляющие.

В качестве таких составляющих используют три основных цвета: красный (*Red, R*), зеленый (*Green, G*) и синий (*Blue, B*). Такая система кодирования называется ***RGB*** (по первым буквам названий основных цветов).

Если для кодирования яркости каждой из основных составляющих использовать по 256 значений (восемь двоичных разрядов), то на кодирование цвета одной точки надо затратить 24 разряда, рис. 4.1.

Красный	Зеленый	Синий

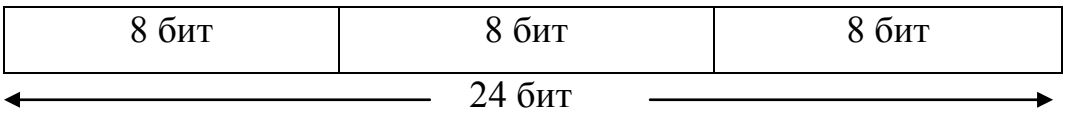


Рис. 4.1. **Кодирование цветного изображения**

При этом система кодирования обеспечивает однозначное определение 16,5 млн. различных цветов, что близко к чувствительности человеческого глаза.

Режим представления цветной графики с использованием 24 двоичных разрядов называют *полноцветным* (*True Color*).

Кроме *RGB*, другими популярными системами кодирования цветных изображений являются *CMY* и *HSV*.

CMY *(Cyan-Magenta-Yellow* – голубой-пурпурный-желтый) – цветовая система, применяемая для получения цветных изображений на белой поверхности. Эта система используется в большинстве устройств вывода, таких как лазерные и струйные принтеры. Новые цвета в системе *CMY* получают вычитанием цветовых составляющих из белого цвета.

Существует другой вариант системы *CMY* - система ***CMYK***, в которой символ *K* означает *черный цвет*. Систему *CMYK* часто называют четырехцветной, а результат ее применения четырехцветной печатью.

Система ***HSV*** – одна из многих цветовых систем, в которых для представления новых цветов не смешивают основные цвета, а изменяют их свойства. Насыщенность определяется количеством белого цвета в оттенке. Так, красный оттенок с 50%-ной насыщенностью соответствует розовому.

Одним из недостатков растровых методов является трудность пропорционального изменения размеров изображения до произвольно выбранного значения. Единственный способ увеличить изображение – это увеличить сами пиксели. Однако это приводит к появлению зернистости – пикселизации.

Векторные методы позволяют избежать проблем масштабирования, характерных для растровых методов. В этом случае изображение представляется в виде совокупности линий и кривых. С помощью подобной технологии

описываются различные шрифты, поддерживаемые современными принтерами и мониторами. Они позволяют изменять размер символов в широких приделах и по этой причине получили название масштабируемых шрифтов. Однако векторная технология не позволяет достичь фотографического качества изображений объектов как при использовании растровых методов.

Вопросы для самоконтроля

1. Что такое система счисления?
2. В чем отличие позиционной системы счисления от непозиционной?
3. Что называется основанием системы счисления?
4. Что понимают под алфавитом системы счисления?
5. Сформулируйте правила выполнения арифметических действий в позиционных системах счисления?
6. Какие системы счисления являются наиболее эффективными для использования в цифровых автоматах?
7. Какие способы перевода чисел из одной системы счисления в другую вы знаете?
8. В чем заключается преимущество использования восьмеричной и шестнадцатеричной систем счисления?
9. Что такое прямой, обратный и дополнительный коды?
10. Сформулируйте правила, определяющие правильность выполнения операций сложения чисел со знаком и без знака в ЭВМ.
11. Какие системы кодирования символьной информации Вы знаете?
12. Какие системы кодирования графической информации Вы знаете?

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	Дано целое десятичное число X= -50_{10}. Его 8-битный дополнительный код …	• 11001111 • 11001110 • 10110001

		• 1001110
2.	Последняя цифра чисел 57_8 и 56_8 в восьмеричной системе счисления равна …	• 6 • 5 • c • 3
3.	Максимальное шестнадцатеричное число, кодируемое одним байтом равно …	• FF • AA • 1515 • 15F
4.	Последняя цифра суммы чисел 55_{16} и 56_{16} в шестнадцатеричной системе счисления равна ….	• 3 • 6 • 1 • B
5.	Результат вычисления выражения 16*8 + 4*4 +1 имеет в двоичной системе счисления вид …	• 122001 • 112001 • 10010001 • 10011001
6.	Число 33_{10} в двоичной системе счисления имеет вид …	• 001100 • 100000 • 100111 • 100001
7.	Дано целое число X=-5. Его запись в 8-битном дополнительном коде …	• 11111011 • 11111010 • 1011 • 1101
	Последняя цифра суммы числе 54_8 и 56_8 в восьмеричной системе счисления равна …	• 6 • 9 • 2 • 4
8.	Дополнительный код чисел 1_{10} в однобайтовом формате имеет вид …	• 00000001 • 10000001 • 01111110 • 01111111
9.	Системой счисления, в которой вычисляются равенства 3 + 3 = 6, является	• любая с основанием большим 5 • любая с основанием большим 6 • только десятичная • любая
10.	Результат вычисления выражения 2^7 + 2^4 +1 имеет в двоичной системе счисления вид …	• 10010001 • 70040001 • 20020001 • 10010100
11.	Максимальное восьмеричное число, кодируемое одним байтом равно …	• 378_8 • 400_8 • 256_8 • 377_8
12.	Для кодирования 32 различных состояний достаточно _____ двоичных разрядов.	• 32 • 8 • 20 • 5
13.	Дополнительный код числа X= -35_{10} в однобайтовом формате …	• 11011101 • 01011101 • 10100100

		• 11011100
14.	Если числа в двоичной системе счисления имеют вид 11001_2 и 1010_2, то их сумма в двоичной системе счисления равна …	• 101010_2 • 101111_2 • 11100_2 • 100011_2
15.	Дополнительный код числа 2_{10} в однобайтовом формате имеет вид …	• 00000010 • 10000010 • 01111110 • 01111101
16.	Дополнительный код числа 8_{10} в однобайтовом формате имеет вид …	• 01111000 • 00001000 • 01110111 • 10001000
17.	Для хранения на диске фразы «ПРОСТОЙ_РАСЧЕТ» в системе кодирования UNICODE (16 бит на символ) необходимо _____ байт.	• 26 • 14 • 13 • 28
18.	$37_8 + 1AC2_{16} =$ Результатом выполнения указанной операции будет…	• 1101011100101_2 • 16341_8 • 6880_{10} • $1AE1_{16}$
19.	Число 11100001_2 в десятичной системе счисления – это…	• 132 • 227 • 225 • 262
20.	$24_6 + 304_5 - 5_{10} =$ В десятичной системе счисления результат выполнения операции запишется как…	• 323 • 230 • 90 • 330
21.	Число 129 в двоичной системе счисления – это…	• 10000001 • 10000010 • 10000000 • 11000000
22.	В восьмеричной системе счисления **НЕПРАВИЛЬНОЙ** записью числа является…	• 17770 • 1020304 • 165481 • 10101010
23.	14 + 3 = 22 Данное равенство будет истинным в системе счисления с основанием …	• 7 • 5 • 10 • 2
24.	CMYK является…	• системой представления цвета • типом монитора • графическим редактором • форматом графических файлов
25.	С помощью одного **байта** можно запомнить _____ различных состояний.	• 8 • 256 • 1024 • 1
26.	Количество байт для кодирования слова	• 8

	ТЕСТ в кодовой таблице UNICODE (два байта на символ)…	• 6 • 4 • 64
27.	RGB является…	• графическим редактором • системой представления цвета • форматом графических файлов • видом монитора
28.	В студенческой группе 16 студентов, из них 4 девушки. В сообщении о том, что староста группы – девушка, содержится _____ информации.	• 1 бит • 4 бита • 2 бита • 16 бит
29.	При кодировании 16 битами в Unicode информационный объем пушкинской фразы ***Я помню чудное мгновение*** составляет	• 24 бита • 24 байта • 384 бита • 384 байта
30.	Аббревиатура RGB, обозначающая цветовую модель, расшифровывается как…	• Red Green Blue • Red Grey Blue • Ready Go Back • Red Green Black
31.	Вещественное число X с плавающей точкой представляется в виде (M- мантисса, p – порядок, q – основание системы счисления) …	• $X = M + q^p$ • $X = M * q^p$ • $X = q^p * M$ • $X = M * E^p$

Глава 5. Логические основы построения персональных компьютеров

5.1. Аппарат алгебры логики

Основу любого дискретного вычислительного устройства, в том числе персонального компьютера, составляют элементарные логические схемы. Работа этих схем основана на законах и правилах алгебры логики, которая оперирует двумя понятиями: истинности и ложности высказывания.

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

Высказывание – некоторое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Аппарат алгебры логики (*булевой* алгебры) создан в 1854 году Дж. Булем как попытка изучения логики мышления математическими методами. Впервые практическое применение булевой алгебры было сделано К. Шенноном в 1938 году для анализа и разработки релейных переключательных сетей. Результатом чего явилась разработка метода представления любой сети, состоящей из совокупности переключателей и реле, математическими выражениями и принципов их преобразования на основе правил булевой алгебры.

Аппарат булевой алгебры, как и любая другая формальная математическая система состоит из трех множеств: *элементов, операций* над ними и а*ксиом*.

Элементы. *Наименьшим* элементом алгебры логики является ***0,*** *наибольшим* элементом - ***1***.

Поэтому множество элементов булевой алгебры выбирается бинарным B = {0,1}, а сама алгебра называется *бинарной*, или *переключательной*.

Ее элементы называются константами, или логическими 0 и 1, которым в ряде случаев соответствуют бинарные цифры, а в других случаях – логические значения *истина* (*True*) и *ложь* (*False*).

Операции. Основными, или базовыми, операциями булевой алгебры служат: И (AND), ИЛИ (OR) и НЕ (NOT), табл. 5.1.

Операция ***И*** называется ***логическим умножением*** или ***конъюнкцией*** и обозначается знаком умножения { ×, ^}.

Операция ***ИЛИ*** называется ***логическим сложением*** или ***дизъюнкцией*** и обозначается знаком сложения { +, v}.

Операция ***НЕ*** называется ***логическим отрицанием*** или ***инверсией*** (дополнением) и обозначается знаком { -, ¬}.

Таблица 5.1.

Базовые логические операции

Операция	Название операции	Обозначение операции
И (AND)	Логическое умножение - конъюнкция	***** ^
ИЛИ (OR)	Логическое сложение - дизъюнкция	+ v
НЕ (NOT)	Логическое отрицание - инверсия	- ¬

При выполнении операций применяются отношение эквивалентности “=” и скобки “()”, которые определяют порядок выполнения операций. Если скобок нет, то операции выполняются в следующей последовательности: логическое отрицание, логическое умножение и логическое сложение.

5.2. Основные аксиомы и законы алгебры логики

Аксиомы (постулаты) алгебры логики:

1. ***Дизъюнкция*** двух переменных равна 1, если хотя бы одна из них равна 1:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 1

2. ***Конъюнкция*** двух переменных равна 0, если хотя бы одна из них равна 0:

0 * 0 = 0

0 * 1 = 0

1 * 0 = 0

1 * 1 = 1

3. ***Инверсия*** одного значения переменой совпадает с ее другим значением:

$\bar{1} = 0$

$\bar{0} = 1$

Законы алгебры логики:

1. Сочетательный:

$$(a + b) + c = a + (b + c)$$

$$(a * b) * c = a * (b * c)$$

2. Переместительный:

$$a + b = b + a$$

$$a * b = b * a$$

3. Распределительный:

$$a * (b + c) = a * b + a * c$$

$$a + b * c = a * b + a * c$$

5.3. Логические элементы персональных компьютеров

К основным логическим элементам современных персональных компьютеров относятся электронные схемы, реализующие операции И, ИЛИ, НЕ И– НЕ, ИЛИ – НЕ и другие, а также триггер.

Логический элемент - это часть электронной логической схемы, которая реализует электронную логическую функцию.

С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно в схемах бывают от двух до восьми входов и один или два выхода.

Входные и выходные сигналы, соответствующие двум логическим состояниям в логических элементах – 1 и 0 - имеют один из двух установленных уровней напряжения. Высокий уровень обычно соответствует значению “истина” (“1”), а низкий – значению “ложь” (“0”).

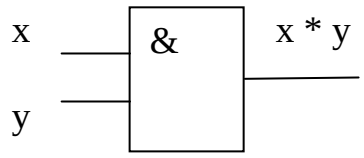
Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию.

Работу логических элементов описывают с помощью таблиц истинности.

Таблица истинности – это табличное представление вычислительной (логической) схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

***Логическая схема* И** - эта схема реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах логической схемы И с двумя входами представлено на рис.5.1.

Единица на входе схемы И будет тогда и только тогда, когда на всех входах будут единицы. Когда на одном входе будет ноль, на выходе также будет ноль.



x	***y***	***x * y***
0	0	0
0	1	0
1	0	0
1	1	1

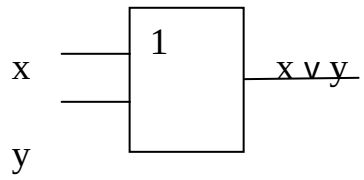
Рис. 5.1. **Условное обозначение и таблица истинности схемы И**

Связь между выходом ***z*** этой схемы и входами ***x*** и ***y*** описывается соотношением: ***z = x * y*** (читается как ***x*** и ***y***).

Операция конъюнкции на структурных схемах обозначается знаком & (читается как амперсэнд), являющимся сокращенной записью английского слова ***and***.

Логическая схема **ИЛИ** - эта схема реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы ИЛИ будет единица, на ее выходе также будет единица.

Условное обозначение на структурных схемах логической схемы ИЛИ с двумя входами представлено на рис.5.2.



x	***y***	***x v y***
0	0	0
0	1	1
1	0	1
1	1	1

Рис. 5.2. **Условное обозначение и таблица истинности схемы ИЛИ**

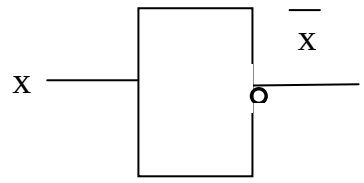
Знак 1 на схеме соответствует обозначению, то есть значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1.

Связь между выходом ***z*** этой схемы и входами ***x*** и ***y*** описывается соотношением: ***z = x*** ∨ ***y*** (читается как ***x*** или ***y***).

Логическая схема **НЕ** - эта схема реализует операцию отрицания (инвертор).

Связь между входом *x* этой схемы и выходом *z* можно записать соотношением: $z = \overline{x}$, ***x,*** где $\overline{x}$ читается как "***не x***" или "***инверсия x***".

Условное обозначение на структурных схемах логической схемы НЕ представлено на рис.5.3.



x	$\overline{x}$
0	1
1	0

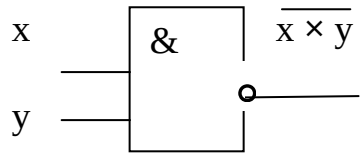
Рис. 5.3. **Условное обозначение и таблица истинности схемы НЕ**

Если на входе схемы 0, то на выходе 1. Когда на входе 1, на выходе 0.

***Логическая схема* И–НЕ** - эта схема состоит из элемента И и инвертора и осуществляет отрицание результата схемы И.

Связь между выходом z этой схемы и входами x и y описывается следующим образом: $z = \overline{x \times y}$, где $\overline{x \times y}$ читается как "**инверсия** x и y".

Условное обозначение на структурных схемах логической схемы И–НЕ с двумя входами представлено на рис. 5.4.



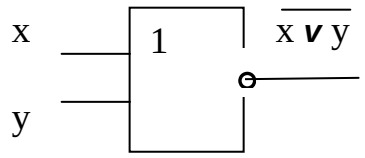
x	y	$\overline{x \times y}$
0	0	1
0	1	1
1	0	1
1	1	0

Рис. 5.4. **Условное обозначение и таблица истинности схемы И–НЕ**

***Логическая схема* ИЛИ–НЕ** - эта схема состоит из элемента ИЛИ и инвертора и осуществляет отрицание результата схемы ИЛИ.

Связь между выходом z этой схемы и входами x и y описывается следующим образом: $z = \overline{x \vee y}$, где $\overline{x \vee y}$ читается как "**инверсия** x или y".

Условное обозначение на структурных схемах логической схемы ИЛИ–НЕ с двумя входами представлено на рис.5.5.



x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

Рис. 5.5. **Условное обозначение и таблица истинности схемы ИЛИ–НЕ**

5.4. Логические устройства с памятью

В персональных компьютерах, помимо рассмотренных выше логических схем, используются логические устройства с памятью – триггеры. Выходные сигналы триггера зависят не только от входных сигналов, действующих в настоящий момент, но и от сигналов, действующих на входы до этого.

Триггер **-** (от англ. *trigger* – защелка, спусковой крючок) электронное устройство с двумя устройствами состояния равновесия, соответствующим логической “1” и логическому “0”, способное многократно переходить из одного состояния в другое под воздействием внешних сигналов, предназначенных для записи и хранения 1 бита данных.

Функциональная схема триггера представлена на рис. 5.6.

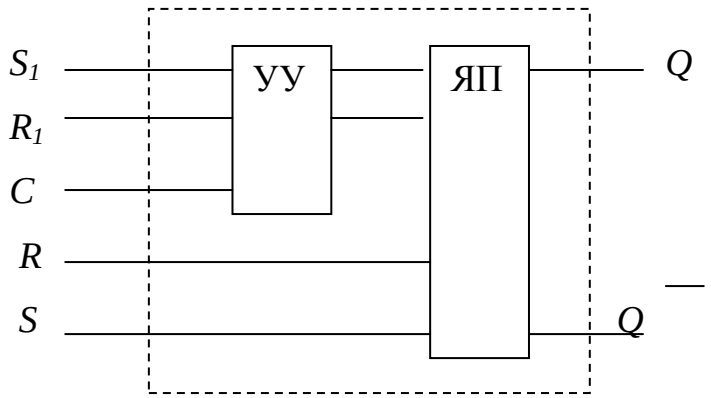


Рис. 5.6. **Функциональная схема триггера**

Триггер функционирует следующим образом. Устройство управления (УУ) преобразует сигналы так, что ячейка памяти (ЯП) принимает одно из двух устойчивых состояний, в зависимости от входных сигналов. Входные сигналы поступают на входы S_1 и R_1 преобразуются устройствами управления (УУ) и поступают на внутренние входы ячейки памяти (ЯП). В общем случае устройство управления может отсутствовать. Тогда сигналы подаются непосредственно на входы *R* и *S* ячейки памяти.

В триггерах также может осуществляться синхронизация приема информации с помощью входа *C*. При наличии входа С триггер называют синхронным, в противном случае – асинхронным. В асинхронных триггерах информация может записываться непрерывно. В этом случае она определяется информационными сигналами, действующими на входах в данный момент времени, то есть изменение состояния ячейки памяти происходит мгновенно после изменения состояний информационных входов.

В синхронном триггере информация заносится только в момент действия так называемого синхронизирующего сигнала.

В схемотехнике приняты следующие обозначения входов и выходов триггеров:

Q – прямой выход триггера;

$\overline{Q}$ - инверсный выход триггера;

S – раздельный вход установки в единичное состояние (напряжение высокого уровня на прямом выходе *Q);*

R – раздельный вход установки в нулевое состояние (напряжение низкого уровня на прямом выходе *Q);*

C – вход сихронизации.

В основе любого триггера лежит кольцо из двух инверторов, рис.5.7.

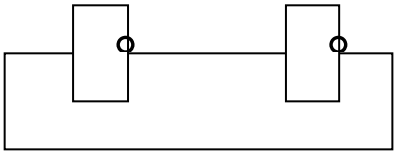


Рис. 5.7. **Кольцо инверторов**

Это кольцо принято обозначать в виде так называемой защелки, рис. 5.8.

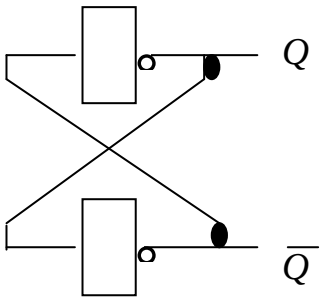


Рис. 5.8. **Элемент-защелка**

Самый распространенный тип тригерра – *RS*-триггер. Функциональная схема *RS*-триггера содержит защелку (два элемента И-НЕ или ИЛИ-НЕ), а также два разделительных статистических входа управления, рис. 5.9.

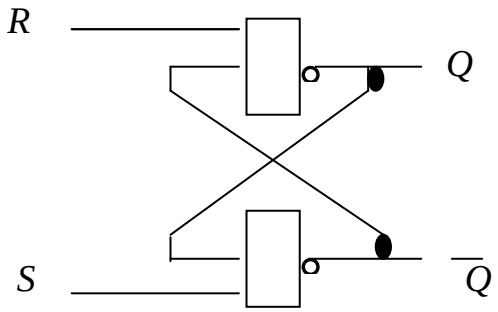


Рис. 5.9. **Функциональная схема *RS*-триггера**

В зависимости от используемых элементов можно получить триггеры с прямыми входами или с инверсными входами. Эти входы управления называют *R* (*reset* – сброс) и *S* (*set* – установка).

Триггер, изображенный на рис. 5.9, это триггер без устройства управления. Таблица истинности для данного триггера с прямыми входами имеет вид, рис. 5.10.

Входы	**Выходы**

R	S	Q	$\overline{Q}$
0	0	Без изменений	
0	1	1	0
1	0	0	1
1	1	Не определено	

Рис. 5.10. **Таблица истинности для триггера с прямыми входами**

Для триггера с инверсными входами таблица истинности имеет вид, рис. 5.11.

Входы		**Выходы**	
$\overline{R}$	$\overline{S}$	Q	$\overline{Q}$
0	0	Не определено	
0	1	0	1
1	0	1	0
1	1	Без изменений	

Рис. 5.11. **Таблица истинности для триггера с инверсными входами**

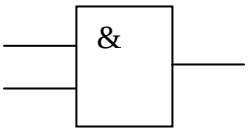
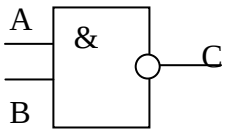
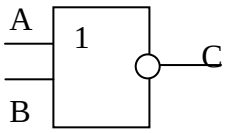
Поскольку триггер может запомнить только один разряд двоичного кода, то для запоминания байта нужно 8 триггеров, для запоминания кило-байта соответственно $8 * 2^{10} = 8192$ триггеров. Современные микросхемы памяти персональных компьютеров содержат миллионы триггеров.

Вопросы для самоконтроля

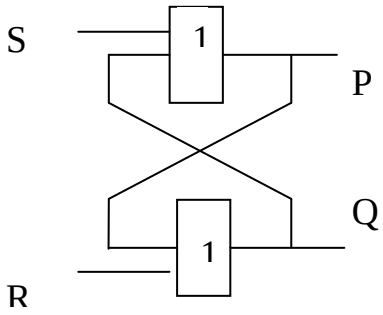
1. Что такое алгебра логики?
2. Назовите области применения булевой алгебры.
3. Какие элементы булевой алгебры Вы знаете?
4. Назовите базовые операции булевой алгебры?

5. Какие основные законы алгебры логики Вы знаете?
6. Что такое таблица истинности?
7. Что такое логический элемент компьютера?
8. Какие базовые логические элементы современных персональных компьютеров Вы знаете?
9. Что такое триггер?
10. Определите функциональную схему RS – триггера.

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	На рисунке  представлено условное обозначение логического элемента …	•И •ИЛИ •ИЛИ - НЕ •НЕ
2.	На рисунке  представлена логическая схема выражения…	• C=НЕ (А И В) • C=НЕ (А ИЛИ В) • C=А ИЛИ В • C=А И В
3.	На рисунке  представлена логическая схема выражения…	• C=А ИЛИ В • C=НЕ (А И В) • C=А И В • C=НЕ (А ИЛИ В)
4.	Логическое высказывание «неверно, что Аня – отличница, но плохая спортсменка» является ложным, в случае, когда	• Аня – отличница и плохая спортсменка • Аня – не отличница и плохая спортсменка • Аня – не отличница и хорошая спортсменка • Аня – отличница и хорошая спортсменка
5.	Логическому высказыванию «Неверно, что А» соответствует таблица истинности …	• 4 • 3 • 1

	1 A B 0 0 1 0 1 1 1 0 0 1 1 0 2 A B 0 0 0 0 1 1 1 0 1 1 1 0 3 A B 0 0 1 0 1 1 1 0 0 1 1 1	• 2
6.	Таблица истинности A B ? 0 0 0 0 **1** 1 1 0 1 1 1 0 соответствует логической операции…	• Отрицание • И • ИЛИ • Исключающее ИЛИ
7.	Заданы логические выражения: a. $(x + y)$ mod 2 = 0 b. *x* mod 2 = 0 **and** *y* mod 2 = 0 c. *x* mod 2 = 1 **and** *y* mod 2 = 1 Если *x* и *y* нечетные числа, то значение ИСТИНА принимают выражения	• a, c • a, b, c • a, b • b, c
8.	Заданы логические выражения: a. mod (*x*, 3) = 0 b. mod (*x*, 3) = 1 **or** mod (*x*, 3) = 2 c. **not** (mod (*x*, 3) = 1 **or** mod (*x*, 3) = 2) Если *x* кратно 3, то значение ИСТИНА принимают выражения	• b, c • a, c • a, b • b
9.	Логическому высказыванию «A или B, но не оба» соответствует таблица истинности с номером … 1 A B 0 0 1 0 1 1 1 0 0 1 1 0 2 A B 0 0 0 0 1 1 1 0 1 1 1 0 3 A B 0 0 0 1 1 0 1 1	• 1 • 2 • 4 • 3
10.	Таблица истинности A B ? 0 0 0 0 **1** 0 1 0 0 1 1 1 соответствует логической операции…	• ИЛИ • И • Исключающее ИЛИ • Отрицание
11.	Правильным результатом выполнения логической операции дизъюнкции (ИЛИ) является…	• ЛОЖЬ ИЛИ ЛОЖЬ=ИСТИНА • ИСТИНА ИЛИ ИСТИНА=ЛОЖЬ • ИСТИНА ИЛИ ЛОЖЬ=ЛОЖЬ • ЛОЖЬ ИЛИ ИСТИНА=ИСТИНА
12.	Результатом выполнения логической	• A – ИСТИНА, B – ЛОЖЬ, C –

	операции $(A \vee B) \wedge C$ будет ИСТИНА, если…	ЛОЖЬ • А – ИСТИНА, В – ИСТИНА, С – ЛОЖЬ • А–ИСТИНА, В–ЛОЖЬ, С–ИСТИНА • А – ЛОЖЬ, В – ЛОЖЬ, С – ЛОЖЬ
13.	Электронная схема, представленная на рисунке, называется…  	• Сумматор • Триггер • Транзистор • Реле
14.	Результатом выполнения логической операции $A \vee B \wedge C$ будет ЛОЖЬ, если…	• А – ИСТИНА, В – ЛОЖЬ, С – ИСТИНА • А – ИСТИНА, В – ЛОЖЬ, С – ЛОЖЬ • А – ЛОЖЬ, В – ЛОЖЬ, С – ЛОЖЬ • А – ИСТИНА, В – ИСТИНА, С – ЛОЖЬ
15.	Равенство (**A** OR **B**) AND **B** = **C** (здесь OR – Логическое ИЛИ, AND – логическое И) Выполняется при значениях …	• A=0, B=1, C=1 • A=0, B=0, C=1 • A=1, B=1, C=0 • A=1, B=0, C=1
16.	Для того, чтобы логическое выражение (a ^ a)? (¬b ^ ¬b) При любых значениях логических переменных a и b всегда значение «истина», вместо знака вопроса …	• Нельзя поставить ни знак дизъюнкции (v), ни знак конъюнкции (^) • Можно поставить как знак дизъюнкции (v), так и знак конъюнкции (^) • Можно поставить знак дизъюнкции (v), но не знак конъюнкции (^) • Можно поставить конъюнкции (^), но не знак дизъюнкции (v)
17.	В случае истинности логического выражения (A <=X AND X <=D) можно утверждать, что …	• X принадлежит отрезку [C; D] и не принадлежит отрезку [A; B] • X обоим отрезкам: [A; B] и [C; D] • X принадлежит одному из отрезков [A; B], [C; D] • X не принадлежит ни одному из отрезков [A; B], [C; D]
18.	Равенство NOT(**A** AND **B**)=**B** OR **C** (здесь OR – логическое ИЛИ, AND –	• A=1, B=0, C=0 • A=1, B=1, C=1

<table>
<tr><td></td><td>логическое И, NOT – отрицание) Выполняется при значениях …</td><td>• A=0, B=0, C=1
• A=0, B=0, C=0</td></tr>
<tr><td>19.</td><td>Равенство NOT A AND NOT B =C (здесь AND – логическое И, NOT – отрицание) выполняется при значениях …</td><td>• A=0, B=0, C=0
• A=0, B=0, C=1
• A=1, B=1, C=1
• A=1, B=0, C=1</td></tr>
<tr><td>20.</td><td>Таблица истинности, приведенная на рисунке, отражает выражение …<table><tr><td>A</td><td>B</td><td>C</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>• C= NOT A OR NOT B
• C=NOR A XOR NOT B
• C=NOT A AND NOT B
• C= A AND NOT B</td></tr>
<tr><td>21.</td><td>Применяя побитовую операцию AND к числам 11111_2 и 10101_2, получим двоичный код десятичного числа …</td><td>• 32
• 31
• 21
• 0</td></tr>
<tr><td>22.</td><td>Для запоминания 1 байта информации достаточно ___ триггера (ов).</td><td>• 8
• 2
• 1
• 16</td></tr>
<tr><td>23.</td><td>Операция объединения высказываний в логике называется</td><td>• Дизъюнкция
• Импликация
• Конъюнкция
• Инверсия</td></tr>
<tr><td>24.</td><td>Логическая операция A↔B называется…</td><td>• импликация
• инверсия
• дизъюнкция
• эквиваленция</td></tr>
<tr><td>25.</td><td>Приоритеты выполнения операций в логическом выражении в порядке убывания:</td><td>• дизъюнкция, инверсия, конъюнкция, импликация
• импликация, инверсия, конъюнкция, дизъюнкция
• инверсия, конъюнкция, дизъюнкция, импликация
• конъюнкция, инверсия, дизъюнкция, импликация</td></tr>
<tr><td>26.</td><td>Импликацией A=>B называется высказывание, которое…</td><td>• ложно тогда, когда ложны оба высказывания A и B
• ложно тогда и только тогда, когда A истинно и B ложно
• ложно тогда и только тогда, когда A ложно и B истинно
• ложно тогда, когда истинны оба высказывания A и B</td></tr>
<tr><td>27.</td><td>Студент сдал экзамены на оценки A и B. Студент является ударником, если истинно логическое выражение…</td><td>• (A>3) AND NOT (4<=B)
• (A>3) AND (4<=B)
• (A>3) OR (4<=B)
• NOT ((A>3) AND (4<=B))</td></tr>
<tr><td>28.</td><td>Логический элемент, выполняющий логическое сложение, называется…</td><td>• конъюнктор
• сумматор
• инвертор</td></tr>
</table>

		• дизъюнктор

РАЗДЕЛ 2. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Глава 6. Понятие алгоритма и его основные формы

6.1. Алгоритм и его свойства

Слово "АЛГОРИТМ", как известно, происходит от видоизменённого имени древнего гениального арабского учёного Аль Хорезми. Поначалу дадим нестрогое, простейшее определение понятия "алгоритм":

АЛГОРИТМ – ***это некая система предписаний (инструкций), обязательно ведущих к достижению некоторого желаемого результата.***

Если как следует призадуматься над этой фразой, то мы ***всю жизнь*** от рождения и до последнего часа живём в мире алгоритмов.

Иногда алгоритм предельно прост и выполняется рефлекторно. Например, человек прикоснулся к раскалённому предмету. Он сразу отдёргивает руку и дует на неё или подставляет под холодную воду. Рука спасена.

Ещё один простой пример. Если горит красный сигнал светофора, то переходить улицу нельзя, и мы стоим. Цвет сменился на жёлтый, а затем на зелёный – мы начинаем и совершаем переход.

В течение жизни алгоритмы сменяют друг друга и всё усложняются. Примером тут может служить уже целый **комплекс** алгоритмов, называемый "Правила дорожного движения".

Приведём несколько "академичный" пример алгоритма. Допустим, проводится тестирование. Вместе с вопросом испытуемому предлагается ряд ответов, из которых только один правильный. Алгоритм поведения человека в данном случае сильно зависит от степени его подготовки. Если он хорошо представляет себе проблему, то сразу указывает верный ответ. Если – не очень, то обычно применяется следующий алгоритм: исходя от противного, отбрасываются заведомо негодные ответы, в сузившемся круге поиска легче догадаться, что правильно, а что – нет. Наконец, крайний случай – не готов! Тогда алгоритма просто нет – наугад!

И таких примеров из обычной повседневной жизни можно привести великое множество.

Теперь уточним определение алгоритма:

АЛГОРИТМ – *это* **конечная последовательность** *простейших формул и логических правил*, **чётко** *и* **недвусмысленно** *определяющих* **весь ход решения** *какой-либо* **задачи**, *который состоит в* **упорядоченном** *выполнении различных операций над данными с целью получения искомого ответа (результата) за* **к о н е ч н о е число** *шагов*.

Алгоритм должен обладать следующими обязательными ***свойствами*** (характеристиками). К основным свойствам алгоритма относятся:

1. ***Понятность***. Элементы алгоритма (формулы, логические предписания) должны однозначно распознаваться (быть понятыми) "исполнителем" (человеком или устройством).

2. ***Дискретность***. Описываемый алгоритмом процесс и сам алгоритм могут быть разбиты на отдельные самостоятельные или взаимосвязанные элементарные этапы, возможность выполнения которых на ЭВМ у пользователя не вызывает сомнений.

3. ***Определённость.*** Предписания, входящие в алгоритм, должны быть **точными** и **понятными**. Они должны обеспечивать **однозначность** результата вычислительного процесса при заданных исходных данных и не допускать произвольных действий.

Именно благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче. То есть, исполнитель (человек или устройство) может выполнять его *слепо*, не имея о решаемой задаче никакого представления и всё равно приходя к нужному результату.

4. ***Результативность***. Это свойство означает, что алгоритм должен приводить к получению результата за **конечное** число шагов, иначе работа с

ним теряет всякий практический смысл. Само *количество шагов* может определяться какими-либо *ограничениями*, например, количеством повторяемых действий или заданной точностью расчёта (величиной разницы между результатами, полученными на предыдущем и текущем шаге алгоритма).

5. ***Массовость.*** Предполагается, что алгоритм разрабатывается в общем виде, т. е. он может быть пригоден для решения не одной, а некоторого класса задач определённого типа. При этом они могут различаться лишь значениями исходных данных. Например, алгоритм расчёта стоимости партии товаров (в отличие от величины получаемого при этом результата!) с учётом назначенной для него скидки не зависит от значений цены единицы товара, его количества и процента скидки. Аналогично, алгоритм расчёта заработной платы для бригады работников одинаков для любого их количества и сумм, начисленных каждому из них.

Отсутствие или неполнота любого из этих свойств лишает алгоритм совершенства и возможности его успешной автоматической реализации.

6.2. Формы представления алгоритма

Алгоритм как набор инструкций может быть представлен в разных формах:

а) *словесной,*

б) *словесно-формульной,*

в) *в виде псевдокода,*

г) *графической,*

д) *программной (с помощью операторов или команд).*

Словесная форма алгоритма предполагает описание порядка выполнения каких-либо действий на *естественном* языке (припомните – в незнакомом городе вам объясняют, как добраться, например, до вокзала).

Словесно-формульная запись сочетает в себе применение конструкций естественного языка и понятных математических обозначений. Например, что-

бы описать алгоритм вычисления выражения $\mathbf{Y = x^2 + x}$ при **x=3,5**, нужно

- придать переменной x значение 3,5;
- вычислить выражение x^2;
- вычислить сумму $x^2 + x$
- записать вычисленное значение вместо Y.

Такая форма записи понятна, но непригодна для описания больших и сложных алгоритмов.

Псевдокод (ПСК) – это способ записи алгоритма на условном подмножестве естественного языка с элементами **языка программирования** и общепринятыми математическими обозначениями. ПСК занимает промежуточное место между естественным и формальным (алгоритмическим, специально приспособленным для записи алгоритма) языками. ПСК, помимо формул обязательно включает в себя небольшое количество служебных (ключевых) слов, имеющих всегда одно и то же значение при описании любого алгоритма: ***начало, дано, конец, ввод, вывод, если, перейти*** и т.д. Как правило, для облегчения восприятия, они выделяются начертанием шрифта, например, полужирным.

Примером псевдокода может служить описание алгоритма вычисления и последующей печати суммы **Y** десяти произвольных чисел **x.**

1. **Начало**
2. **Дано** Сч=1, Y=*0 (Сч – переменная для подсчёта количества чисел x, Y – переменная-накопитель суммы значений x)*
3. **Ввод** значения **x**
4. **Если** Сч > 10 **перейти к** пункту **7**
   Иначе выполнить
   Y = Y + x *(увеличить значение Y на величину x)*
5. **Выполнить** Сч = Сч + 1 *(увеличить значение Сч на 1)*
6. **Перейти к** пункту 4
7. **Вывод** на печать значения **Y**
8. **Конец**

Фразы *в скобках курсивом* – это пояснения к отдельным переменным или действиям. Единого или формального определения ПСК не существует, поэтому возможны различные версии ПСК, отличающиеся друг от друга.

Псевдокод легко воспринимается человеком, но абсолютно непригоден для написания реальной программы, "понятной" для ЭВМ. Кроме того, если алгоритм достаточно сложен, то ПСК теряет свою наглядность.

Графическая форма представления алгоритма или блок-схема является более наглядной и компактной. В ней каждое элементарное действие обозначается определенной геометрической фигурой. Все фигуры соединяются линиями перехода, которые определяют очерёдность и порядок выполнения действий.

Кроме того, (что очень важно!) блок-схема не зависит от языка программирования, на котором будет реализован алгоритм. Поэтому она доступна для понимания любым человеком, знакомым с применяемыми в ней обозначениями, но вовсе не владеющим программированием. Вот как выглядит алгоритм нахождения суммы десяти (10) произвольных чисел в виде блок-схемы:

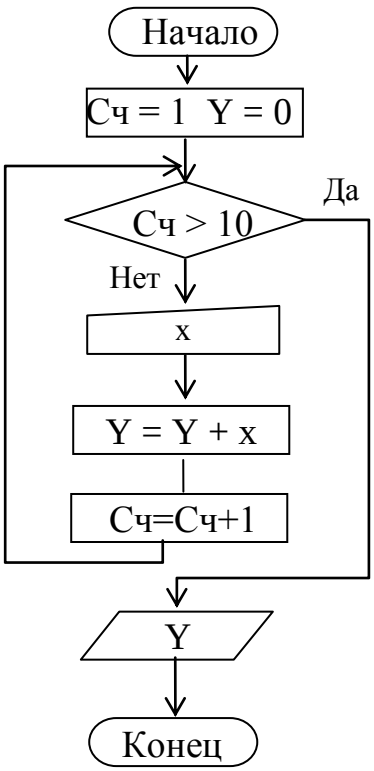


Рис. 6.1. Графическая форма циклического алгоритма

Условные обозначения, применяемые при составлении блок-схем алгоритмов, и правила их выполнения определены в ***ГОСТ 19.701-90 (ИСО 5807-85) «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения»***.

ПРИМЕЧАНИЕ 1: подробности о назначении отдельных фигур в блок схеме и приёмах их отображения см. в Приложении 1 "Графическая схема алгоритмов (ГСА)".

Операторный (или программный) способ записи алгоритма уже предполагает применение конкретного алгоязыка высокого уровня и знание его основных составляющих – ***операторов***. В отличие от ПСК, алгоязык, подобно любому языку, имеет свой синтаксис и пунктуацию, которые нужно неукоснительно соблюдать.

Алгоритм, описываемый в операторах, может понять только специалист в области программирования, но компьютеру он по-прежнему «непонятен». Для этого он должен быть "переведён" на язык ***машинных команд*** специальной программой-транслятором. Алгоритм в командах компьютер уже умеет «распознавать» и выполнять. Здесь, наконец, можно уже говорить о **программе** для компьютера.

ПРОГРАММА – *это последовательность* ***недвусмысленных*** *инструкций (операторов или команд), которую компьютер* ***чётко выполняет одну за другой*** *до тех пор, пока не дойдёт до оператора «конец».*

6.3. Базовые алгоритмические структуры

Таким образом, программирование – это наука о том, как заставить компьютер делать то, что нам нужно, и так, как нам нужно.

Всё разнообразие алгоритмов можно создать, комбинируя в разных сочетаниях следующие базовые алгоритмические структуры.

6.3.1. *Последовательная (линейная) алгоритмическая структура*

Эта структура – самая простая. Её элементы образуют простую после-

довательность (цепочку), в которой они и выполняются друг за другом, никогда не нарушая и не прерывая эту цепочку - линейно - от начала к концу (рис.10.2).

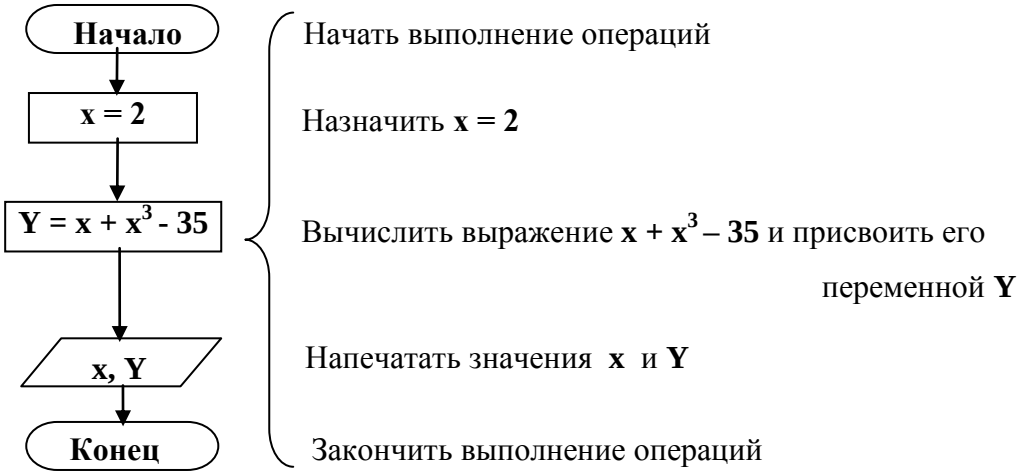


Рис. 6.2. Графическая форма алгоритма, пояснённая псевдокодом

6.3.2. *Ветвящаяся (разветвлённая) структура*

Иногда её называют "структура с условием (условиями)". Сердцевиной такой структуры действительно является операция проверки некоего условия, в результате которой дальнейшее выполнение алгоритма может идти по одному из предусмотренных путей. Например, такие ветвящиеся структуры:

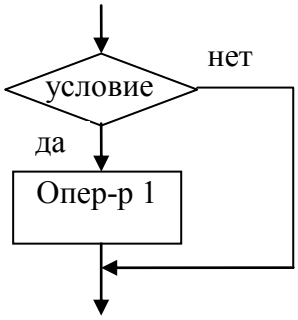


Рис. 6.3. С ***одной*** ветвью

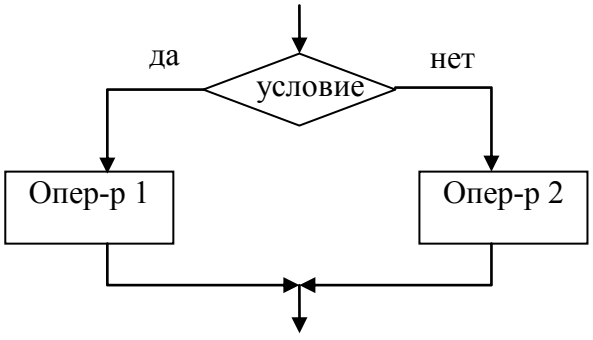


Рис. 6.4. С ***двумя*** ветвями

Ветвящиеся структуры

Этой структуре соответствует такой псевдокод:

*Если результат проверки <условия> **"да"**, то **выполнить** <оператор1>, если **"нет"** – то **пропустить** выполнение <оператора1>, после чего продолжить работу*

В алгоязыке этой структуре соответствует оператор:

If <усл-е> **THEN** <опер1> **End If**

Этой структуре соответствует такой псевдокод:

*Если результат проверки <условия> **"да"**, то **выполнить** <оператор1>, если **"нет"** – то **выполнить** <оператор2>, после чего продолжить работу*

В алгоязыке этой структуре соответствует оператор:

If <усл-е> **THEN** <опер1> **ELSE** <опер2> **End If**

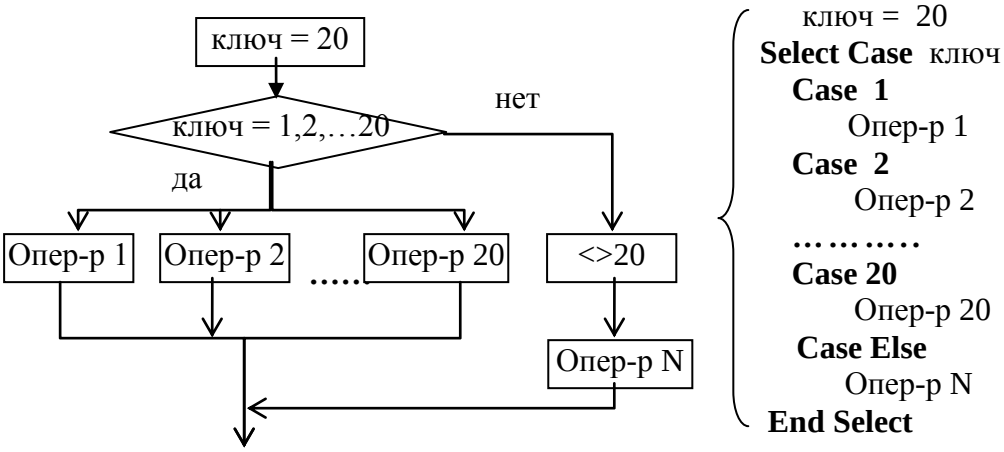


Рис. 6.5. С ***несколькими*** (N) ***альтернативными*** ветвями (справа - соответствующий оператор алгоязыка)

В псевдокоде такая варианту разветвлённой структуры соответствует запись:

Пусть ключ = 20 (количество вариантов выбора пути решения)
Если ключ = 1, то выполнить <оператор1>,
если ключ = 2, то выполнить <оператор2>,
… *дальнейшие проверки значения ключа* **…**
если ключ = 20, то выполнить <оператор20>,
если ключ НЕ равен ни одному из значений списка 1,2, … 20,
то выполнить <операторN>

6.3.3. ***Циклические структуры*** *(от греч.* ***kiklos*** *– круг)*

Договоримся, что "тело цикла" – это некоторый набор операций, которые должны повторно выполняться раз за разом, пока не наступит момент завершения повторов (циклов). Причём, тело цикла может представлять собой любую алгоритмическую структуру, в том числе, и циклическую. Выделяют 3 циклических структуры: цикл с предусловием (условие для выполнения цикла проверяется перед его началом), цикл с постусловием (условие проверяется после первого выполнения цикла) и цикл с заданным (вычисляемым) числом повторов.

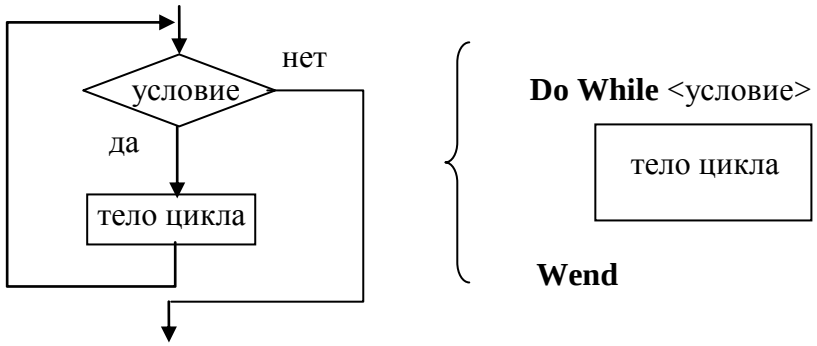


Рис. 6.6. Блок-схема и оператор выполнения цикла с ***пред***условием

В псевдокоде такому варианту циклической структуры соответствует запись:

Проверить <условие>

Если оно несправедливо – не выполняется - ("нет"), то выйти из цикла.

Если оно справедливо ("да"), то выполнить тело цикла,

Вернуться к новой проверке условия

Заметим: для того, чтобы алгоритм не "зацикливался" до бесконечности, обычно в теле цикла предусматривают изменение значений, входящих в проверяемое <условие>. И в какой-то из проверок это условие, наконец, нарушается, приводя к выходу из цикла.

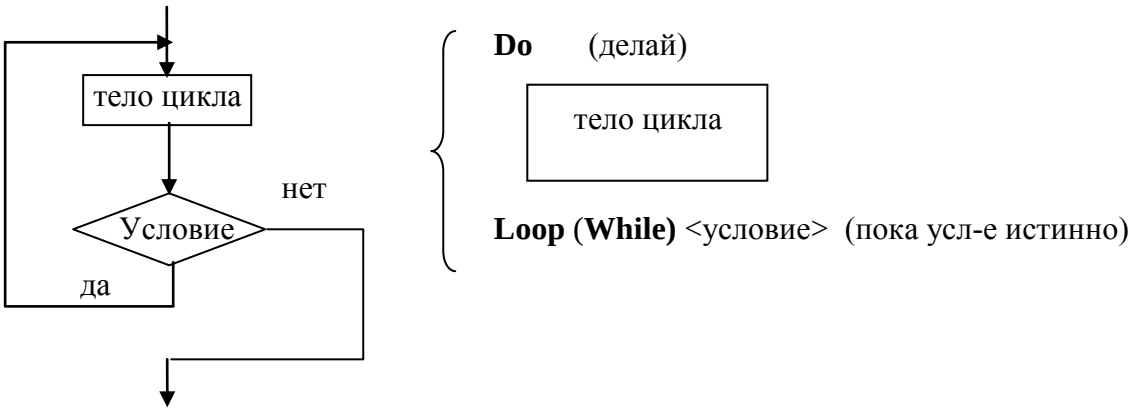
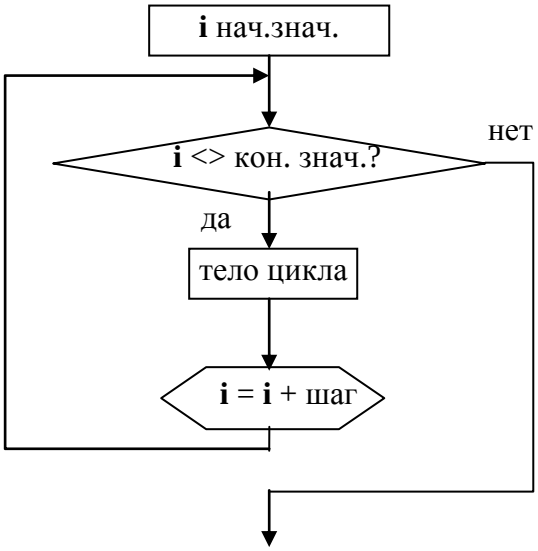


Рис. 6.7. Блок-схема и оператор выполнения цикла с ***пост***условием

Этот вариант циклической структуры, вроде бы, мало отличается от предыдущего - условие проверяется не ***до*** входа в цикл, а ***после*** этого.

Но эта разница, небольшая, на первый взгляд, приводит к разнице принципиальной: в цикле с ***пред***условием вполне может оказаться, что условие ***сразу будет нарушено*** и цикл не состоится, а в цикле с ***пост***условием тело цикла, ***хоть один раз, но всё равно будет выполнено***, независимо от результата проверки условия!

Третий вариант организации цикла отличается от первых двух своей изначальной определённостью. Если в тех вариантах трудно заранее определить, когда именно закончится выполнение циклов, то это - цикл, в котором число повторов тела цикла определено (явно или неявно). Он так и называется "цикл с ***заданным числом повторов***" (см. рис. 6.8) :



В алгоязыке этому соответствует:

For I = <нач.знач.> **to** <кон.знач> **step** <шаг>

Next(модификация знач-я переменной i)

Рис. 6.8. Блок-схема и оператор выполнения цикла с ***заданным числом повторов*** (как правило – он подчиняется закону арифметической прогрессии)

В Н И М А Н И Е ! Шаг может быть и отрицательным!

В псевдокоде такому варианту циклической структуры соответствует запись:

Задать: начальное значение счётчика **i**,

Проверить условие **i** <> (не равно) <конечное значение>

Если оно несправедливо ("нет"), то выйти из цикла.

Если - справедливо ("да"), то перейти к выполнению тела цикла.

После этого увеличить **i** на величину <шага> и вернуться к новой проверке условия

6.4. Этапы развития программирования

Традиционно программированием всегда занимались программисты-профессионалы. В самом развитии программирования можно выделить следующие исторически сложившиеся этапы.

1. ***Программирование в машинных кодах.*** Чтобы посмотреть, как выглядит текст программы, записанный в машинных кодах, щелкнем по кнопке **Пуск** (находясь в какой-либо из операционных систем семейства Windows) и выберем в **Главном меню** пункт **Выполнить. В** диалоговом окне **Запуск программы** (рис. 6.9**)** введём имя программы ***debug.exe,*** нажмём кнопку ОК и в командной строке появившегося окна наберём команду просмотра содержимого ячеек памяти: ***d 9876:5432***.

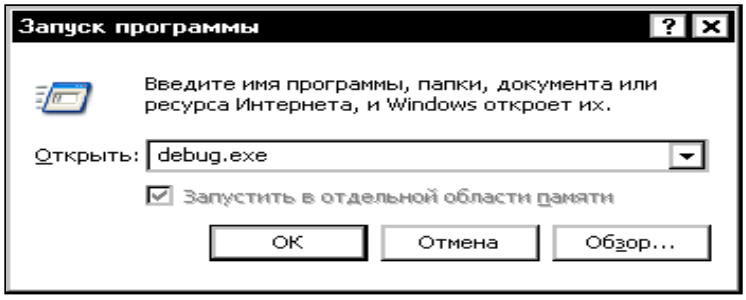


Рис. 6.9. Окно "Запуск программы"

После нажатия клавиши Enter нам будет предъявлено содержимое оперативной памяти по указанным адресам, записанное в машинных кодах (рис. 6.10).

Машинные коды записаны в шестнадцатеричной системе счисления. Так выглядят программы, когда они загружены в компьютер.

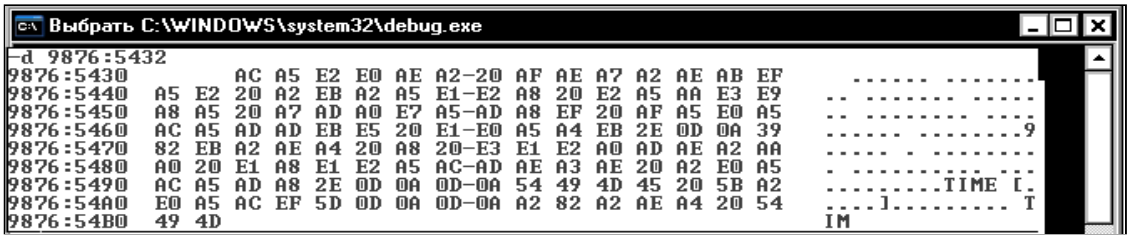


Рис. 6.10. Фрагмент программы в машинных кодах

Человеку очень трудно воспринимать и уж, тем более, исполнять такое

представление программы. В настоящее время машинные коды создают и работают с ними не программисты, а специальные программы-компиляторы.

2. ***Программирование на Ассемблере*** *(языке низкого уровня).* Чтобы посмотреть, как выглядит программа на Ассемблере, проделаем те же действия, что и в предыдущем примере, только в окне программы ***debug.exe*** введём другую команду: ***u 9876:5432***.

Вы увидите результат, в котором команды процессора записаны не цифрами, а буквами. Например, команда **Переместить** будет записана как **MOV** (см. строку, выделенную рамкой на рис. 6.11)**:**

```
9876:5432 AC            LODSB
9876:5433 A5            MOVSW
9876:5434 E2E0          LOOP    5416
9876:5436 AE            SCASB
9876:5437 A220AF        MOV     [AF20],AL
9876:543A AE            SCASB
9876:543B A7            CMPSW
9876:543C A2AEAB        MOV     [ABAE],AL
9876:543F EF            OUT     DX,AX
9876:5440 A5            MOVSW
9876:5441 E220          LOOP    5463
9876:5443 A2EBA2        MOV     [A2EB],AL
9876:5446 A5            MOVSW
9876:5447 E1E2          LOOPZ   542B
9876:5449 A820          TEST    AL,20
9876:544B E2A5          LOOP    53F2
9876:544D AA            STOSB
9876:544E E3E9          JCXZ    5439
9876:5450 A8A5          TEST    AL,A5
```

Рис. 6.11. Фрагмент программы на Ассемблере

Такая форма записи более близка человеческому восприятию, чем машинные коды (особенно, если он знает английский!). Однако язык ассемблер "привязан" ***к устройству*** конкретного процессора, а ***не к алгоритму*** программируемой задачи. Сколько разновидностей устройств, конструкций процессоров, столько и ассемблеров. То есть это - машинно-ориентированный алгоязык. Такие языки не могли получить большого распространения в среде широкого круга пользователей вычислительной техники. Но, тем не менее, они продолжают использоваться профессионалами для написания специальных программ, обеспечивающих доступ к отдельным регистрам и областям памяти и работу с ними.

3. Алгоритмическое программирование. В 1950-х годах появились

языки, в которых для написания программ использовались уже общеупотребительные слова и простые правила синтаксиса. Перевод таких программ на машинные коды производился специальной программой-компилятором, поэтому для программиста стало уже неважным, где, в каких ячейках хранятся результаты расчета и какие регистры процессора используются для выполнения той или иной операции. Программист теперь сосредоточился на записи разработанного им ***алгоритма*** в среде алгоритмического ***языка программирования***. Таков, например, язык **ФОРТРАН** (Fortran — Formula Translator - переводчик формул), разработанный для инженерных расчетов, или **КОБОЛ** - для экономических расчетов. Во главу угла было поставлено понятие класса алгоритмов.

4. ***Процедурное программирование.*** С течением времени было замечено, что в программах встречаются *одинаковые* группы операторов, отличающиеся только значениями входящих в них параметров.

Повторяющиеся блоки операторов стали стандартизировать, то есть выделять из общей программы в *отдельные подпрограммы, процедуры* и *функции*. В результате возникли такие **процедурные** языки программирования, как **Паскаль, С**. Это — универсальные языки. Они не были ориентированы на решение конкретного типа задач, т.е., не являлись проблемно-ориентированными, как, например, Фортран.

5. Объектно-ориентированное программирование. Наконец в середине 1980-х годов был введен принцип *многократного* применения кода (текста) *ранее написанных* программ. Такие готовые программные блоки назвали ***объектами***. Их стали использовать, как "полуфабрикаты" для создания вполне конкретных конструкций, обладающих заданными характеристиками. Так, при пошиве костюма используют шаблоны (выкройки) "костюма вообще", а потом уже подгоняют его по фигуре и требованиям заказчика.

Таким образом, при создании *новой* программы *объекты просто* ***перенастраиваются*** в соответствии с требованиями решаемой программистом задачи. Это могут быть встречающиеся в ***разных*** программах ***одинаковые по***

форме *окна,* ***командные кнопки,*** *списки,* похожие *меню,* одинаковые *шрифты* и т. д. Все они являются ***настраиваемыми*** объектами. Примерами объектно-ориентированных языков являются, например, языки **C++** и **Object Pasca1**.

6. Визуальное программирование. С разработкой в середине 1990-х годов операционной системы Windows человек приобрёл возможность ***графического управления*** компьютером. На экране монитора размещаются уже не просто картинки, а *графические* элементы управления, *реагирующие* на определенные *действия*, в частности, на действия со стороны пользователя (на *события*). Щелчком мыши по графическому элементу можно запустить на выполнение целую программу, не пользуясь при этом клавиатурой.

До перехода к системам визуального программирования создавались весьма объёмные программы на языках высокого уровня Pascal, Fortran, PL/1 и других. Все они описывали и на их основе исполнялись самые разнообразные алгоритмы. Однако, в основном, это были алгоритмы обработки числовой и текстовой информации, не приспособленные к имитации или использованию интерфейса Windows.

Появление графических (визуальных) систем управления сделало и программирование также ***визуальным***. Теперь можно с помощью мыши *выбирать из библиотек* нужные *компоненты*, размещать их в рабочем окне будущей программы (**делая**, таким образом, из компонентов **объекты**), и *настраивать* объекты с помощью *свойств*, затем уже добавляя процедуры для обработки событий (создавая таким образом из объектов графические элементы управления).

Так происходит, например, когда строится дом из стандартных блоков, но по индивидуальному проекту. Их назначение фиксировано, но количество и компоновка могут быть различными. Однако, блок-туалет нельзя использовать как блок-спальню. Зато внутри каждого такого блока можно заменить или добавить отдельные части с их собственными характеристиками (свойствами).

На рис. 6.12 приведена упрощённая схема «превращения» компонента через объект в элемент управления. Можно сказать, что компонент сродни чер-

тежу, по которому изготавливают множество совершенно одинаковых объектов, например, одинаковых квартир. Люди, поселившись в квартире, обустраивают ее в соответствии со своими потребностями и вкусом, превращая квартиру в уникальный элемент своей жизни.

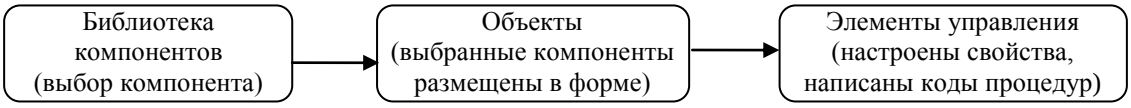


Рис. 6.12. От компонента библиотеки - к элементу управления

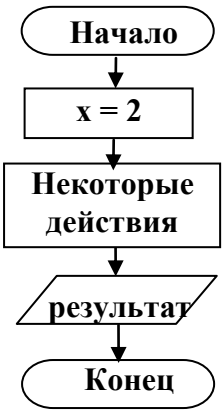
Первой средой визуального программирования стала широко теперь распространённая среда ***объектно-ориентированного визуального*** программирования ***Visual Basic for Application*** (Visual Basic для приложений).

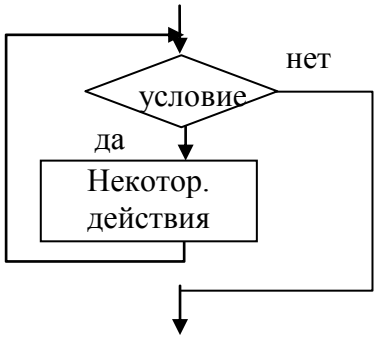
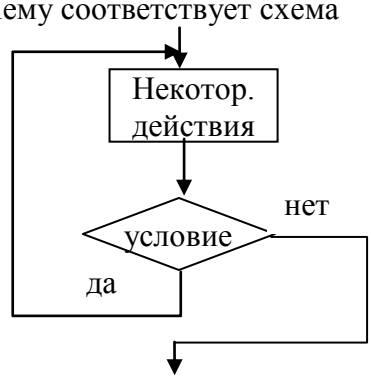
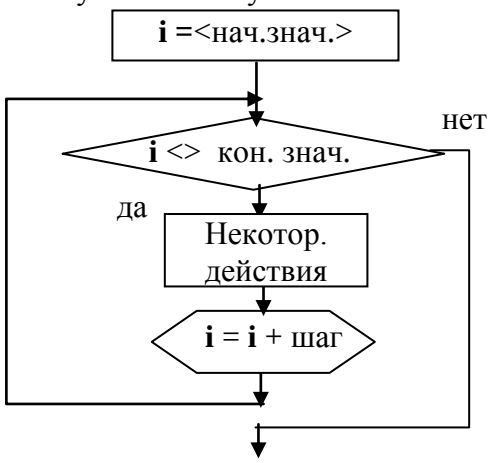
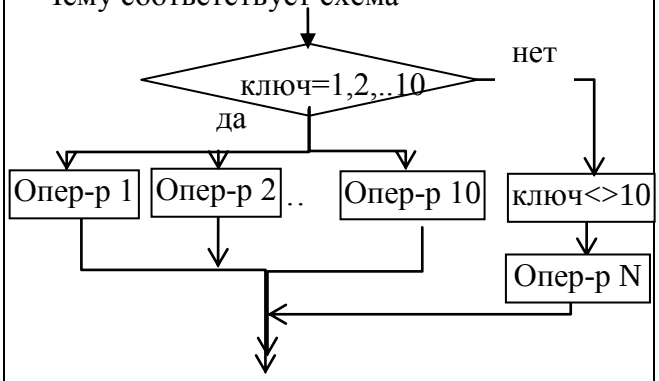
Тестовые задания

Ниже приведено несколько тестовых заданий для закрепления изложенного материала. В каждом задании предлагается вопрос и несколько вариантов ответа на него, один (или несколько) из которых является правильным, а остальные — нет. Укажите правильный(е) ответ(ы).

№ вопр.	Вопросы	Предлагаемые ответы
1	Алгоритм – это…	а) четкая последовательность слов, б) чёткая инструкция действий для обязательного получения результата, в) выполнение действий в случайном порядке, г) произвольные действия пользователя, не всегда приводящие к результату.
2	Свойствами алгоритма не являются…	а) полновесность и уникальность, б) массовость и дискретность, в) результативность и определённость, г) дискретность и понятность.
3	Алгоритм наиболее нагляден и компактен в форме…	а) шестнадцатиричных кодов, б) словесного описания, в) графических фигур (блок-схемы), г) псевдокодовой записи.
4	Дискретность алгоритма означает, что можно…	а) выполнять его по частям в любом порядке, б) разбивать его на любые произвольные части, в) решать задачи, связанные только с натуральным дискретным рядом чисел, г) дробить и его на отдельные самостоятельные части, решающие более мелкие конкретные

		задачи.
5	Массовость алгоритма состоит в том, что…	а) его можно тиражировать сколько угодно раз, б) он популярен и часто используется, в) он пригоден для решения не одно, а многих задач, имеющих общий смысл и порядок действий, г) он пригоден для решения любых задач.
6	Алгоритм может восприниматься человеком и компьютером в форме	а) словесного описания, б) псевдокода, в) программы, г) блок-схемы.
7	Описание алгоритма связано с устройством процессора, если он записан…	а) в машинных командах, б) на ассемблере, в) в операторах языка высокого уровня, г) в псевдокодах.
8	В блок-схеме каждому действию соответствует…	а) порядковый номер, б) фигура заданной формы, в) поясняющий текст, г) любое произвольное обозначение с формулой в нём.
9	Транслятор переводит …	а) блок-схемы в программы, б) машинные команды в операторы языка высокого уровня, в) программу с алгоязыка на другой алгоязык, г) операторы алгоязыка на язык машинных команд.
10	Цикл с постусловием может выполняться…	а) один раз, б) ни одного раза, в) бесконечно, г) заданное число раз.
11	Цикл с предусловием может выполняться…	а) один раз, б) ни одного раза, в) бесконечно, г) заданное число раз.
12	Цикл со счетчиком может выполняться…	а) один раз, б) ни одного раза, в) бесконечно, г) заданное число раз.
13	Наиболее приближенной к реальности является…	а) разветвленная структура алгоритма, б) линейная структура алгоритма, в) циклическая структура алгоритма, г) циклическая структура с разветвлениями.
14	Оператор **If** <условие> **then** <опер1> **endif** соответствует	а) циклическому алгоритму, б) разветвленному с одной ветвью, в) разветвленному с двумя ветвями, г) циклическому разветвленному алгоритму.
15	Оператор **If** <условие> **then** <опер1> **else** <опер2> **endif** соответствует	а) циклическому алгоритму, б) разветвленному алгоритму с одной ветвью, в) разветвленному алгоритму с двумя ветвями, г) циклическому разветвленному алгоритму.

16	Оператор **Select Case** ключ **Case** 1 <опер1> **Case** 2 <опер2> ….. **Case** N <операторN> **End Select** соответствует	а) алгоритму с несколькими ветвями, б) разветвленному алгоритму с одной ветвью, в) разветвленному алгоритму с двумя ветвями, г) циклическому разветвленному алгоритму.
17	Оператор **Do While** <условие> <тело цикла> **Wend** соответствует	а) циклу с предусловием, б) циклическому разветвленному алгоритму, в) циклу с постусловием, г) циклу с заданным числом повторов.
18	Оператор **For** I=<нач.знач.> **to**<конеч.знач.> **step** <шаг> <тело цикла> **Next** соответствует	а) циклу с предусловием, б) циклическому разветвленному алгоритму, в) циклу с постусловием, г) циклу с заданным числом повторов.
19	Оператор **Do** <тело цикла> **While** <условие> соответствует	а) циклу с предусловием, б) циклическому разветвленному алгоритму, в) циклу с постусловием, г) циклу с заданным числом повторов.
20	Современное программирование - это программирование…	а) на ассемблере, б) в машинных кодах, в) процедурное, г) объектно-ориентированное.
21	Какую структуру алгоритма отражает схема  	а) разветвленную, б) линейную, в) циклическую с постусловием, г) циклическую с предусловием.

22	Чему соответствует схема 	а) циклический алгоритм с постусловием б) разветвленный алгоритмм, в) линейный алгоритм, г) циклический алгоритм с предусловием.
23	Чему соответствует схема 	а) циклический алгоритм с постусловием, б) разветвленный алгоритм, в) циклический алгоритм с предусловием, г) линейный алгоритм.
24	Чему соответствует схема 	а) циклический алгоритм с постусловием, б) разветвленный алгоритм с несколькими альтернативными ветвями, в) циклический алгоритм с предусловием, г) цикл с заданным числом повторов.
25	Чему соответствует схема 	а) циклический алгоритм с постусловием, б) разветвленный алгоритм с несколькими альтернативными ветвями, в) циклический алгоритм с предусловием, г) цикл с заданным числом повторов.

Глава 7. Объектно-ориентированное программирование в среде VBA (Visual Basic for Application).

7.1. Что такое VBA?

VBA (*Visual Basic for Application - Visual Basic для приложений*) — это современный язык программирования, поддерживаемый всеми приложениями пакета версий Microsoft Office 2003 и выше, в состав которого входят такие популярные приложения, как Microsoft Access, Microsoft Exce1, Microsoft PowerPoint, Microsoft Word и др.

VBA — это относительно несложный язык программирования. Он очень прост и удобен в освоении и позволяет быстро получить довольно значимые и, главное, наглядные результаты — конструировать качественные приложения для решения многих задач в среде Microsoft Windows. В известном смысле, VBA - прямой наследник популярного языка программирования Basic, но, по сравнению с ним значительно более совершенный и обладающий совершенно новыми качествами.

Поэтому и возможностей, как внутренних (например, в части вывода на экран всевозможных форм), так и внешних (взаимодействие с другими приложениями), у него гораздо больше. Создавать собственные офисные приложения с его помощью гораздо проще и быстрее, чем с помощью других языков программирования.

Являясь развитым языком программирования, VBA также включает в себя полноценную ***интегрированную среду*** разработки с полным набором стандартных специализированных окон, упрощающих проектирование, отладку и тестирование программ. Интегрированная среда разработки VBA – это ***редактор*** Visual Basic.

Он имеет своё окно с панелью меню и набором панелей инструментов, которые позволяют получить доступ к целому ряду окон, предоставляющих инструментальные средства, необходимые для создания программ. Кроме того,

редактор VBA включает специализированные средства для быстрого создания пользовательского интерфейса, что превращает его в *визуальную* среду разработки приложений.

7.2. Основные понятия и элементы языка VBA: объекты, свойства, методы, события, классы объектов

Поскольку язык VBA относится к категории объектно-ориентированных, то основными понятиями в нём являются ***объекты, свойства, методы, события*** и ***классы***.

1. Объекты

В соответствующей литературе даются разные определения ***объектов.*** Из объектов состоит любое программное приложение. И каждый объект обладает набором своих специфических характеристик (параметров, свойств).

Можно дать упрощенное и понятное определение объекта.

Объект *– это готовая программная конструкция интерфейса "человек - компьютер", которая наделена совокупностью свойств (параметров) и методов их обработки.*

Иначе говоря, объекты - это те "блоки", из которых строится здание конкретного приложения, работающего под управлением операционной системы из семейства Windows.

Но можно дать и более строгое определение.

Объект *– это программная конструкция, которая позволяет* ***инкапсулировать*** *данные, описывающие некий компонент прикладной области, вместе с программами, предназначенными для обработки этих данных.*

Под термином "***инкапсуляция***" (сугубо внутренняя принадлежность) подразумевается что характеристики (свойства), приписанные объекту, недоступны воздействиям на них из внешней среды. Но вместе с тем сам объект

должен и может предоставить для этого свои, строго ограниченные, специальные функции – методы.

Объектами программной среды можно назвать окна (обычные или диалоговые), элементы интерфейса (кнопка), управляющие элементы (флажок, переключатель), меню, поля ввода и т.д. При этом в объекте часть данных может быть открыта для других программных конструкций, а другой частью могут оперировать только программные компоненты, относящиеся только к данному объекту. То же самое относится и к *процедурам* (небольшим подпрограммам), принадлежащим объекту, — некоторые из них могут быть вызваны извне, а другие, наоборот, используются только внутри объекта.

Все программные продукты MS Office обладают своими ***стандартными*** наборами объектов. Так, в Excel это - *ячейки, рабочие листы, встроенные функции и методы анализа*, в Access – *таблицы данных, формы, запросы, программа "Построитель выражений"* и т.д. Все они реализованы и могут быть прочитаны или изменены в среде VBA.

Вместе с тем, пользователь, владеющий программированием в VBA, способен создавать *свои*, уникальные и нужные ему объекты с их свойствами внутри этих оффисных приложений.

2. Свойства и **методы объекта. События**

Любой объект всегда имеет уникальное *имя*, с помощью которого всегда можно обратиться к объекту, не спутав его с другими.

Но, кроме того, при описании любого объекта указывают его размер, цвет, назначение и т.п. Иначе говоря, он обладает целым набором ***свойств*** или определённых характеристик. Каждая из них может принимать какие-либо значения. Свойства предназначены для хранения информации о текущем состоянии отдельных сторон, качеств объекта.

Свойства объекта – *это характеристики его текущего состояния в приложении, его параметры. Их значения определяют уникальность объек-*

та, его отличие или сходство по сравнению с другими объектами.

Если мы хотим программно придать объекту значение определенного свойства, то нужно строго соблюдать такой способ записи (синтаксис):

Имя_объекта . Имя_свойства = Значение_свойства

Правильные и строго определённые изменения свойств возможны только под воздействием соответствующих ***методов***, присущих данному объекту.

Метод – *команда или набор команд (подпрограмма), предназначенных для целенаправленных изменений свойства или свойств объекта*

Таким образом, доступ к свойствам объекта возможен только с помощью его собственных методов, т. е. специально подготовленных команд обработки этих свойств

Состоянием элемента управляют с помощью посылаемых ему *сообщений,* указывающих объекту на необходимость выполнить тот или иной метод для достижения требуемого результата.

Сообщения *выдаются самой системой в ответ на* ***действия*** *пользователя или других функционирующих в ней программ, которые в терминологии VBA называются* ***событиями.***

3. Классы объектов

Одно из важнейших понятий объектно-ориентированного программирования - это понятие ***класса,*** описывающее типовую структуру ***сходных по назначению*** элементов. В системе хранится *программное описание* каждого используемого *класса*, на основании которого при необходимости создаются *экземпляры* ***объектов***, представляющих конкретные элементы программной среды.

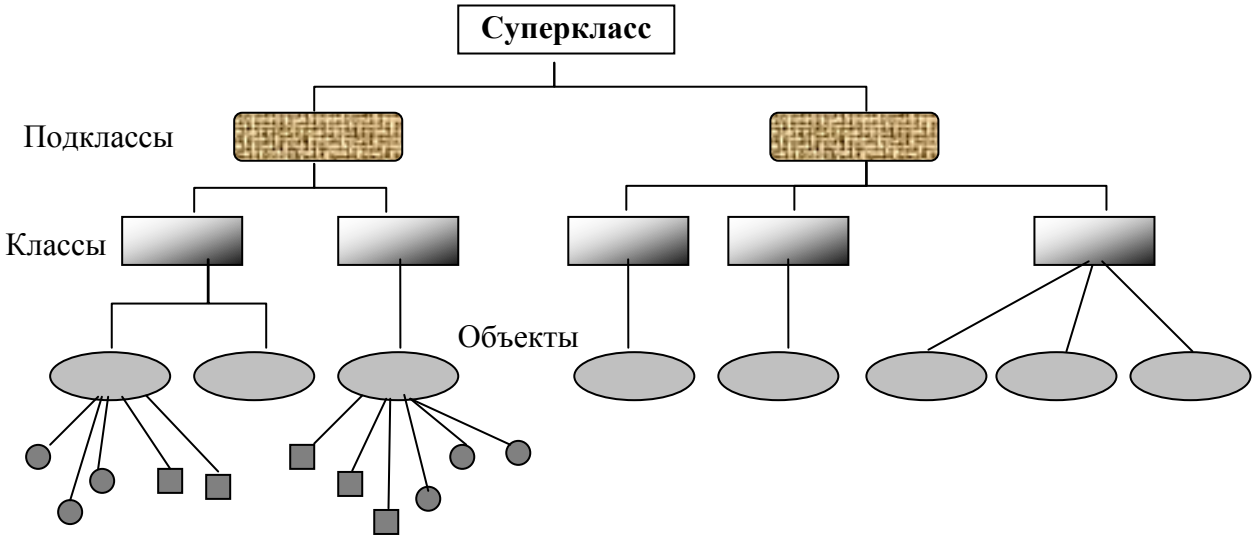
Например, в программе может быть описан ***класс*** "Кнопка", которая должна отображаться в окнах приложения. Каждая отдельная кнопка (экземпляр объекта этого класса) в любом окне приложения создается на основании этого описания, но отличается от других своими ***свойствами*** (размером, цветом, надписью, выполняемой после щелчка операцией и т.д.).

В языке VBA ***класс*** обычно описывается как определенный ***прототип***, на основе которого создается конкретный объект.

Таким образом, ***к л а с с*** *определяет для объекта* ***его назначение, свойства*** *и* ***те действия, которые могут быть выполнены над ним*** *и в этом понятии* ***объединяет*** *его с ему подобными***..**

Иначе говоря, класс – это семейство объектов, сходных друг с другом по перечисленным характеристикам. Поэтому с точки зрения программы интерес представляет не столько сам объект, сколько, то, *какими свойствами* он обладает и *какие действия* можно над ними совершить.

Схематично связь основных понятий объектно-ориентированного программирования можно проиллюстрировать так:



- это свойства объекта, - это методы обработки свойств

Рис. 7.1. Связь основных понятий объектно-ориентированного подхода

Однако, вспомним умное изречение: "Программированию можно научиться, только программируя!".

Легче всего постигать новое на примерах. Поэтому в следующем разделе перейдём к их рассмотрению. И мы начнём это с макросов.

Глава 8. Макросы в приложениях MS Office

Для начала познакомимся с макросами.

8.1. Понятие макроса

Любому пользователю, работающему с приложениями MS Office (Excel, Word, , Access и т. д.), известно, что при решении многих задач приходится выполнять один и тот же набор действий, причём, достаточно часто и всегда в одинаковом порядке.

В Microsoft Office есть очень удобное средство - вместо многократного повторения одной и той же последовательности операций можно создать *макрос*.

Макрос - *это* ***набор операций****, производимых пользователем и* ***автоматически*** *зафиксированных в виде программы.*

Такая программа при каждом её запуске будет выполнять за пользователя ту же нужную и выполненную им ранее последовательность действий.

Таким образом, макросы - это средство, с помощью которого можно описать произвольную последовательность действий пользователя б е з непосредственного программирования для её последующего многократного выполнения.

Для многих VBA - приложений (среди них - программы MS Office) макросы можно создавать как с помощью языка Visual Basic for Application, так и с помощью специальных функций MS Office, используемых для интерактивной записи макросов. Эти функции позволяют записывать все выполняемые пользователем действия при его работе с приложением до тех пор, пока не будет подана команда о прекращении записи.

После остановки процесса выполнения пользователем нужных действий записанная их последовательность *сохраняется в виде поименованного макроса*. Им можно пользоваться каждый раз, когда потребуется повторить выполнение записанной там цепочки операций.

Независимо от того, идёт ли работа в Word, Exce1, Access или PowerPoint, создание нового макроса осуществляется одним и тем же способом.

Общим для этого способа является то, что имя макросу можно давать, только чётко соблюдая следующие правила:

- имя макроса **всегда** начинается с **буквы** и *ни в коем случае* не может содержать *пробелы* или *знаки препинания*,
- в качестве прочих символов имени можно использовать любые буквы и цифры, а также символ подчёркивания " _ " ,
- максимальная длина имени макроса – 80 символов.

ПРИМЕЧАНИЕ 2: если нужно назначить имя, состоящее из нескольких частей (слов), лучше всего разделять эти части знаком подчёркивания " _ " или использовать внутри имени заглавные буквы, например, **ИмяНашегоМакроса.**

8.2. Процесс создания макроса

Рассмотрим практическое создание не очень сложного макроса.

Процесс создания макросов, их действия и содержание будем рассматривать в среде Excel как наиболее популярного приложения MS Office.

Пример. Предположим, что в рабочей книге (РК) Excel надо автоматизировать подготовку ввода данных по коммерческим и государственным банкам города Казани. Под этим будем понимать выполнение таких действий:

- в ячейки **B3:H3** какого-либо рабочего листа надо ввести (в соответствии с вышеупомянутыми требованиями) названия банков города Казани – ИнтехБанк, УрсаБанк, АкБарсБанк, СберБанк, ЭнергоБанк и СпуртБанк;

- отформатировать эти названия (задать шрифт, его размер и стиль "полужирный курсив", оформить толстыми рамками);

- на следующей строке проставить порядковые номера намеченных граф, отцентрировать их и оформить тонкими рамками;

- после этого подготовить РК Excel ко вводу данных по этим банкам, установив курсор в ячейку **B5**.

Предлагается такой порядок действий в Excel:

1) установить курсор в начальную ячейку **B3**;

2) выполнить одну из цепочек :

- **Сервис – Макросы – Начать запись** (в Excel ***2003***);

- вкладка **Вид** - группа **Макросы - Запись макроса** (в Excel ***2007***)

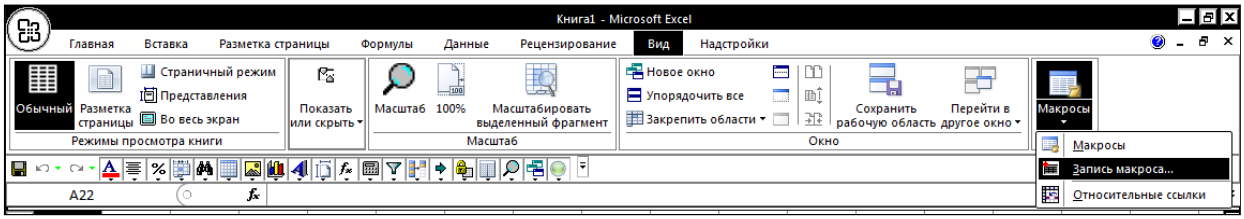


Рис. 8.1. Лента в Excel 2007 с выбранной командой **Запись макроса**

3) на экране – диалоговое окно **Запись макроса** (рис. 8.2):

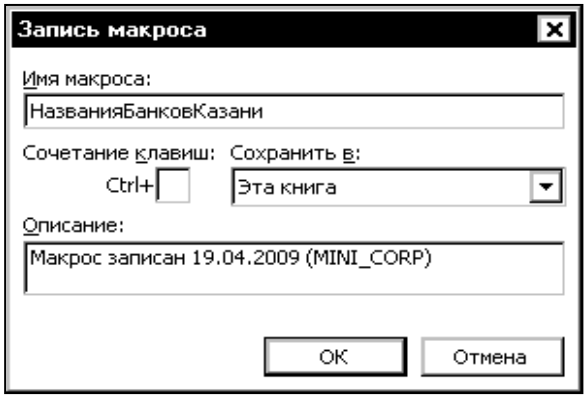


Рис. 8.2. Диалоговое окно для записи параметров (имени и местонахождения) нового макроса

4) в панели "Имя макроса" нужно ввести его будущее уникальное имя **НазванияБанковКазани**;

5) в панели "Сохранить в" оставить значение "**Эта книга**", заданное по умолчанию; это обеспечит доступ к макросу *только из д а н н о й* РК;

6) в этом же окне пользователю предлагается использовать поле "Сочетание клавиш", в котором можно указать желаемую комбинацию клавиш для быстрого вызова данного макроса (сочетание клавиш <Ctrl+*клавиша*>). Это

пригодится в том случае, если будет нужно часто использовать записываемый макрос.

В Н И М А Н И Е ! Допускается использование сочетаний CTRL+ *буква* (для *строчных* букв) или CTRL+SHIFT+ *буква* (для *прописных* букв), где *буква* — любая буквенная клавиша на клавиатуре. ***Н е л ь з я*** использовать сочетания клавиш с ***цифрами*** и ***специальными знаками***, такими как **@** или **#.**

ПРИМЕЧАНИЕ 3: в окне "Запись макроса" есть ещё необязательное поле "Описание". В него можно записать произвольный текст, т.е. комментарий о том, для чего предназначен данный макрос. По умолчанию VBA заполняет это окно информацией о том, где и кем был создан данный макрос (дата записи макроса и имя пользователя).

7) Когда все действия в окне «Запись макроса» будут закончены, можно нажать ОК.

8) Excel переходит в режим записи макроса. С этого момента надо быть ***особенно аккуратными*** - ведь запись макроса уже включена и будут фиксироваться все наши действия, в том числе и неверные**!** Последовательно производим все заданные нами действия. Когда все они будут выполнены и курсор будет установлен в ячейку **B5**, остановим запись, выполнив цепочку действий:

вкладка **Вид** - *группа* **Макросы - Остановить запись (**в Excel ***2007***).

ПРИМЕЧАНИЕ 4: помимо команд «Начать запись» и «Остановить запись» в выпадающем меню группы **Макросы** есть ещё параметр «Относительные ссылки».

Если его использовать, то все действия в макросе будут записываться *относительно* той ячейки, в которой в момент исполнения макроса находится курсор. Например, перед записью макроса, перемещающего курсор на 8 шагов из ячейки A1 в ячейку A8, была активна ячейка A1. Если данный параметр был **включён** *перед* записью макроса, то при его запуске из ячейки F1 он переместит курсор на те же 8 шагов, но уже в ячейку F8! Если же параметр был **отключен**, то откуда бы не запускался макрос, курсор всегда перейдёт в A1.

8.3. Запуск макроса на исполнение

Когда макрос записан, его можно запустить на исполнение.

1) Для этого сначала нужно открыть ту РК, в которой он сохранён.

2) Если это уже сделано, выполняем цепочку:

вкладка **Вид** - *группа* **Макросы - Макросы (**в Excel 2007**).**

3) на экран выводится диалоговое окно **Макрос** (рис. 8.3):

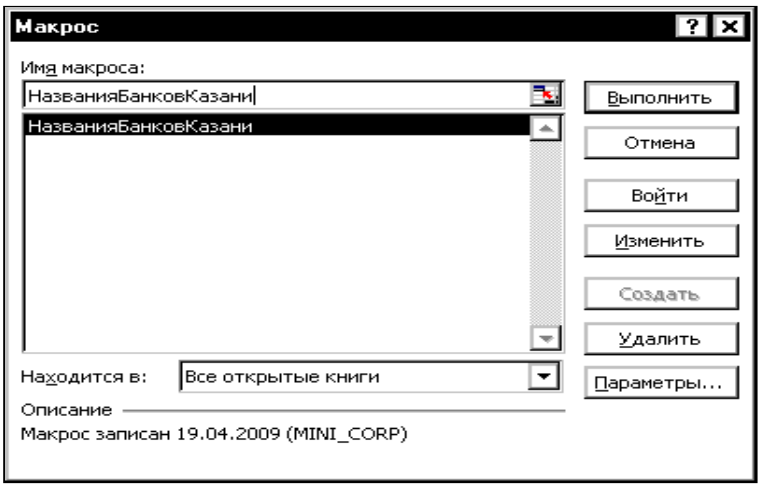


Рис. 8.3. Окно выбора макросов для выполнения

3) выбираем в панели "имя макроса" название нужного нам макроса **НазванияБанков Казани** и нажимаем кнопку **Выполнить.**

Если всё было сделано правильно, на рабочем листе РК появится нужная информация (рис. 8.4):

	A	B	C	D	E	F	G
1							
2							
3		ИнтехБанк	УрсаБанк	АкБарсБанк	СберБанк	ЭнергоБанк	СпутБанк
4		1	2	3	4	5	6
5							

Рис. 8.4. Результат выполнения макроса **НазванияБанковКазани**

Как и было задано, все названия размещены в указанном диапазоне ячеек, им придан стиль"полужирный курсив", они оформлены рамками, есть и номера граф. Работа макроса завершилась тем, что курсор остановился в ячейке B5.

8.4. Код (текст) программы макроса и пояснения к нему

А теперь посмотрим, как выглядит исходный текст (код) макроса **НазванияБанковКазани:**

```
Sub НазванияБанковКазани()  'начало программы-процедуры
'  апостроф – это начало комментария, т.е. пояснения к тексту
' НазванияБанковКазани Макрос
' Макрос записан 19.04.2009 (MINI_CORP)
'
    Range("B3").Select     ' сначала выбрана и активизирована ячейка B3
    ActiveCell.FormulaR1C1 = "ИнтехБанк"        ' текст "ИнтехБанк"
                                                размещён в активной ячейке B3
    Range("C3").Select
    ActiveCell.FormulaR1C1 = "УрсаБанк"
    Range("D3").Select
    ActiveCell.FormulaR1C1 = "АкбарсБанк"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "СберБанк"
    Range("F3").Select
    ActiveCell.FormulaR1C1 = "ЭнергоБанк"
    Range("G3").Select
    ActiveCell.FormulaR1C1 = "СпуртБанк"
    Columns("B:G").Select                   ' выбраны столбцы от B  до G
    Columns("B:G").EntireColumn.AutoFit ' автободбор ширины столбцов
    Range("B3,C3,D3,E3,F3,G3").Select       ' активизированы ячейки
                                                от B3 до G3
    Range("G3").Activate            ' активна ячейка G3 и в ней остановился
курсор
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
```

Запрограммированы аналогичные действия для ячеек C3 – G3

```
'   в последних двух операторах объектам и свойствам обрамления
"диагональ сверху вниз" и "диагональ снизу вверх" задано значение "нет"
   With Selection.Borders(xlEdgeLeft)
      .LineStyle = xlContinuous
      .Weight = xlMedium
      .ColorIndex = xlAutomatic
   End With
   With Selection.Borders(xlEdgeTop)
      .LineStyle = xlContinuous
      .Weight = xlMedium
      .ColorIndex = xlAutomatic
   End With
   With Selection.Borders(xlEdgeBottom)
      .LineStyle = xlContinuous
      .Weight = xlMedium
      .ColorIndex = xlAutomatic
   End With
   With Selection.Borders(xlEdgeRight)
      .LineStyle = xlContinuous
      .Weight = xlMedium
      .ColorIndex = xlAutomatic
   End With
   Selection.Font.Bold = True          ' выбран шрифт Bold - полужирный
   Selection.Font.Italic = True        ' выбран шрифт Italic - курсив
   Columns("B:G").Select               ' выделены столбцы от B до G
   Columns("B:G").EntireColumn.AutoFit      ' включена команда
                                              автоподбора ширины
                                              активных столбцов
```

Запрограммированы действия с рамками

```
Range("B4").Select
   ActiveCell.FormulaR1C1 = "1"
   Range("C4").Select
   ActiveCell.FormulaR1C1 = "2"
   Range("D4").Select
   ActiveCell.FormulaR1C1 = "3"
   Range("E4").Select
   ActiveCell.FormulaR1C1 = "4"
   Range("F4").Select
   ActiveCell.FormulaR1C1 = "5"
   Range("G4").Select
   ActiveCell.FormulaR1C1 = "6"
   Range("B4,C4,D4,E4,F4,G4").Select
   Range("G4").Activate
   Selection.Borders(xlDiagonalDown).LineStyle = xlNone
   Selection.Borders(xlDiagonalUp).LineStyle = xlNone
   With Selection.Borders(xlEdgeLeft)
      .LineStyle = xlContinuous
      .Weight = xlThin
      .ColorIndex = xlAutomatic
   End With
   With Selection.Borders(xlEdgeTop)
      .LineStyle = xlContinuous
      .Weight = xlThin
      .ColorIndex = xlAutomatic
   End With
   With Selection.Borders(xlEdgeBottom)
      .LineStyle = xlContinuous
```

Запрограммировано последовательное размещение номеров граф от 1 до 6

Снова запрограммированы действия с рамками

```
    .Weight = xlThin
    .ColorIndex = xlAutomatic
  End With
  With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
  End With
  With Selection      ' для выбранного -
    .HorizontalAlignment = xlCenter   'горизонтальное выравнивание
                                      содержимого по центру
    .VerticalAlignment = xlBottom ' вертикальное выравнивание - вниз
    .WrapText = False          ' запрещён переход на следующую строку
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
  End With
  Range("B5").Select        ' В заключение активизирована ячейка B5
End Sub
```

Продолжение действий с рамками

Рис. 8.5. Исходный текст (код) макроса.

Начало и конец кода макроса специально выделены автором более крупным шрифтом.

Кроме того, в текст программы специально внесены некоторые *поясне-ния* в виде комментариев к отдельным операторам, оформленные *курсивом*.

Готовая программа даже такого несложного макроса, составленная в автоматическом режиме, достаточно велика. Вполне очевидно, что разобраться в ней, не зная назначения и правил записи отдельных конструкций языка, невозможно. А уж вносить туда дополнения или поправки - и подавно! Однако, всё впереди, и дорогу осилит идущий!

8.5. Корректировка макросов

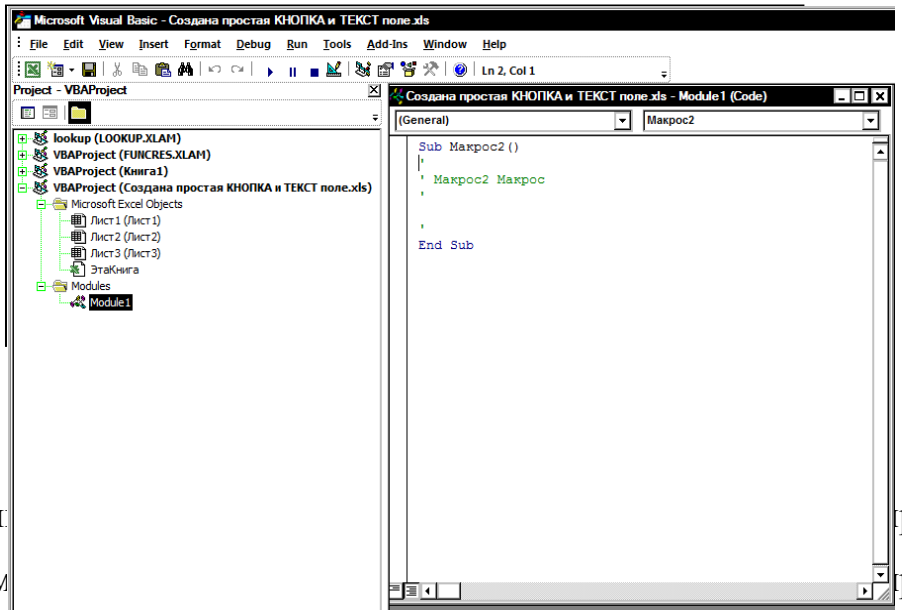
При записи макроса в любом из выбранных приложений (в Microsoft Word, Excel или PowerPoint) действия, выполненные во время записи, будут зафиксированы в виде последовательности соответствующих операторов языка VBA. Вся совокупность этих операторов будет называться исходным кодом, или кодом макроса. Полученный таким образом код в дальнейшем можно будет использовать при создании собственной программы. Причем в этой программе можно будет как корректировать уже имеющиеся, так и, при необходимости, добавлять новые операторы. Получить доступ к исходному тексту макроса, просмотреть или отредактировать его можно двумя способами.

Способ 1. Вызвать окно «Макросы» (см. рис. 8.3), выполнив цепочку действий: *вкладка* **Вид** - *группа* **Макросы - Макросы (**в Excel 2007**),** затем выбрать из списка нужный макрос и нажать в окне кнопку **Войти** или **Изменить.**

Способ 2. На вкладке **Разработчик**[2] в группе **Код** кнопкой **Visual Basic** включить редактор **Visual Basic.**

В обоих случаях на экране появится окно этого редактора (рис. 8.6):

[2] Если такой вкладки нет, то создать её можно, включив **Параметры Excel**, затем в категории **Основные** включив флажок **Показывать вкладку «Разработчик» на ленте** и нажав **ОК**.



с программн рования. В этом окне м резать или вставлять любые операторы на языке VBA, пользуясь командами меню Edit. Для сохранения программного кода отредактированного макроса выберите команду меню **File** => **Save** *Имя_Документа* (**Файл** =>Сохранить в...) редактора VBA или щелкните на кнопке **Save** *Имя_Документа* (Сохранить в...) панели инструментов **Edit**.

Если же был создан объект «Кнопка», то самый лёгкий и естественный способ – раскрыть исходный текст прямо с рабочего листа РК Excel, на котором и совершалась запись макроса. Для этого на вкладке **Разработчик** надо сначала включить режим **Конструктор** в группе **Элементы управления**. Затем, выделив кнопку, вызвать контекстное меню и выбрать там пункт «Исходный текст». На экране – окно редактора **Visual Basic** с исходным текстом макроса, приписанного к данной кнопке.

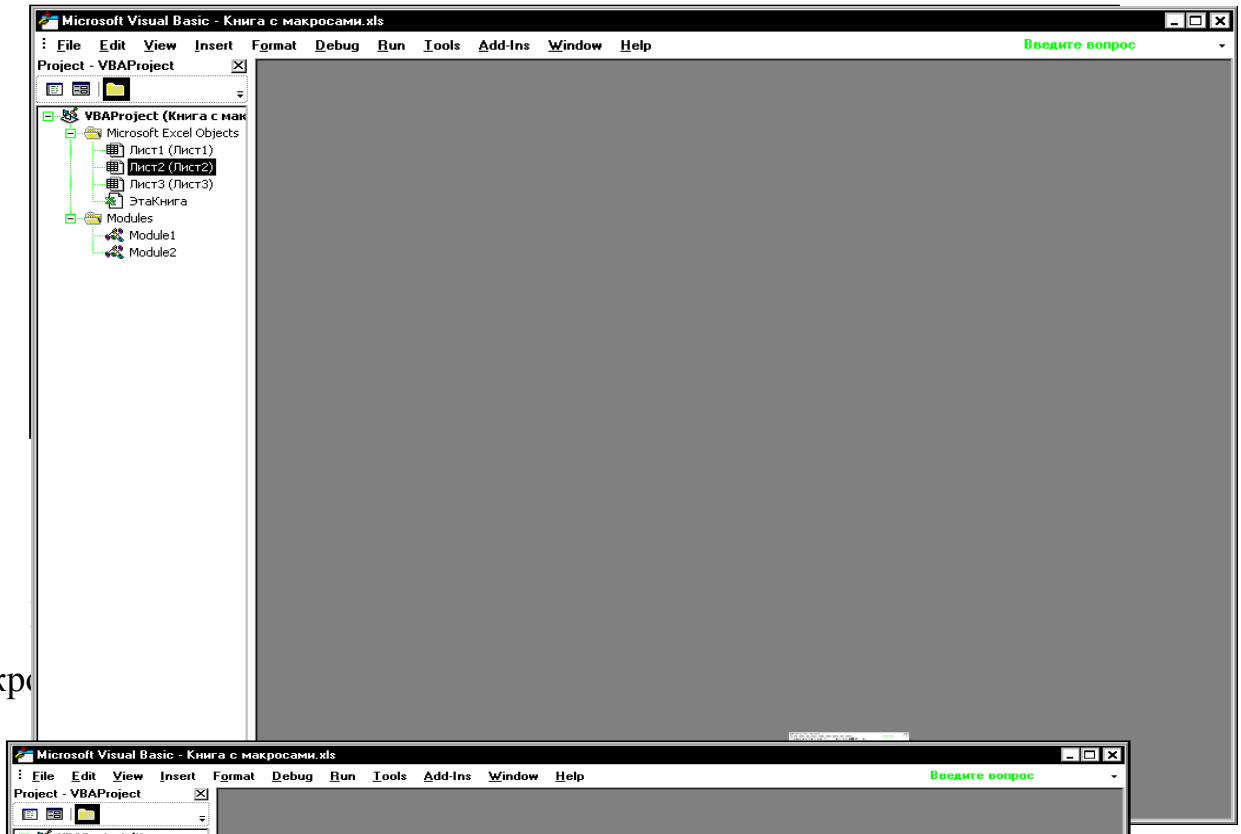
Простой пример составления и записи программы.

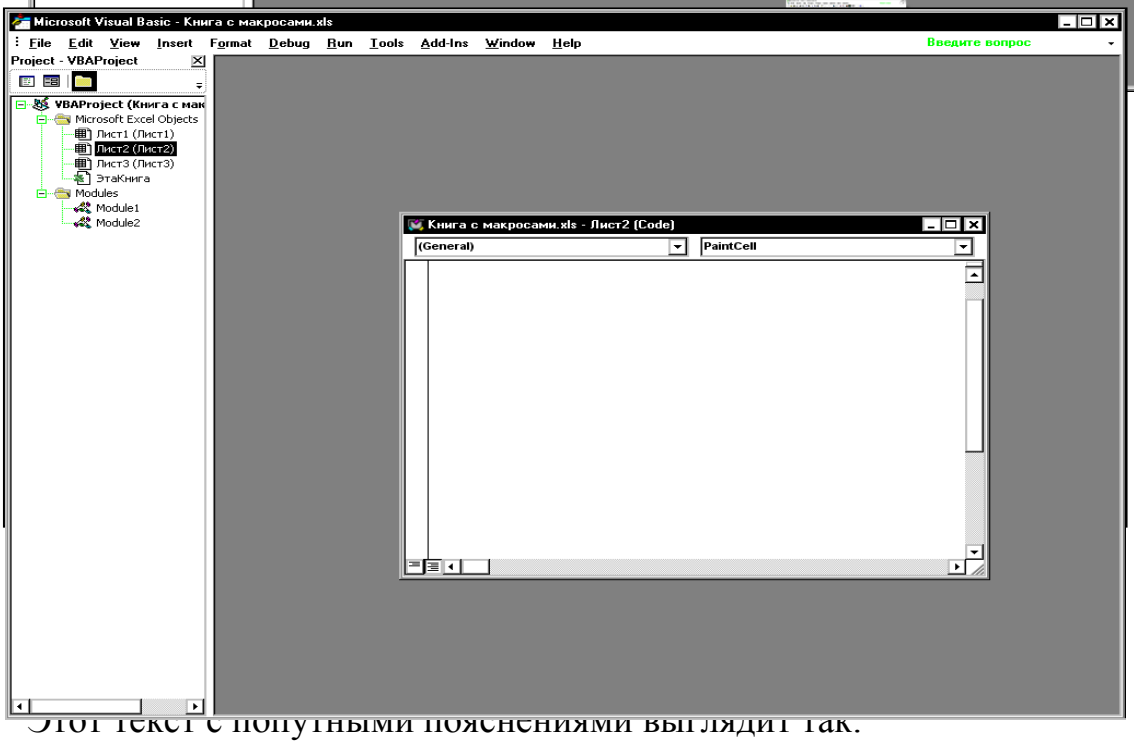
Начнём с очень простого примера: нужно задать цвет фона для любой активной ячейки на листе ***Лист2*** рабочей книги Excel.

Попробуем сделать это, воспользовавшись средой редактора VBA. Снова включим редактор **Visual Basic** на вкладке **Разработчик.**

Открывается окно редактора:

го откр





Этот текст с попутными пояснениями выглядит так:

Sub PaintCell() ' *Заголовок процедуры "Цвет ячейки"*

Dim *n* As Integer ' *переменная* ***n*** *(код цвета) представлена как целочисленная*

***n* = (InputBox("Введите код цвета:"))** ' *команда запроса ввода кода цвета*

ActiveCell.Interior.ColorIndex = *n* ' *введённый код* ***n*** *задаёт новое значение свойству цвета* ***активной*** *ячейки.*

' *Здесь* ***ActiveCell*** *– объект, активная ячейка,* ***Interior.ColorIndex*** *– "двухступенчатое" свойство, задающее цвет объекта* ***ActiveCell***

End Sub () ' *Конец процедуры "Цвет ячейки"*

Рис. 8.9. Текст (код) макроса для изменения цвета активной ячейки

ПРИМЕЧАНИЕ 5: здесь **полужирный** шрифт – операторы и свойства, *курсив* - пояснения к ним. Добавим: **Paint** – цвет, **Cell** - ячейка, **Dim** (**Dimension**) – придать (размеры), **Active** – активизация объекта, **ColorIndex** – числовое обозначение (шифр, код) цвета в среде VBA.

Разместим текст этой крошечной программы в окне Code, а затем нажмём клавишу запуска программы на исполнение **F5** (или выполним **Run - Run Sub/UserForm**).

Если до ввода кода программы был активен ***Лист2*** и любая ячейка на нём, то в площади листа появится окно запроса (рис. 8.9):

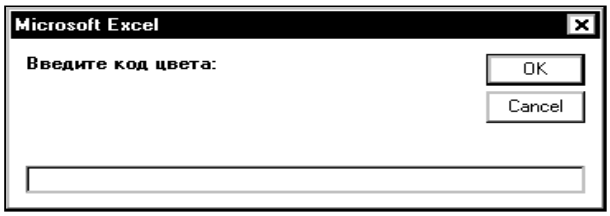


Рис. 8.10. Окно запроса на ввод кода цвета (значения кода - от 0 до 56)

Если мы теперь введём здесь число 6, то активная ячейка получит ЖЁЛТЫЙ цвет фона, если 8, то - ИЗУМРУДНЫЙ, если 0, то - БЕЛЫЙ, если 1, то - ЧЁРНЫЙ и т. д.

Чтобы убедиться в этом, введите вышеуказанный исходный текст в окно Code редактора **Visual Basic** и затем выполните эту программку.

Хотя эта программа и проста, но она уже имеет существенный недостаток: в ней описывается последовательный алгоритм и поэтому он выполняется всего 1 раз. Чтобы перепробовать несколько вариантов окрашивания ячейки, её каждый раз надо запускать заново. Для автоматического повторения выбора этих вариантов нужно превратить её в циклическую программу.

Изменим её, применив один из вариантов цикла. Наиболее подходит для этого оператор с предусловием: While <условие>… Wend. Он не ограничивает число повторов цикла и, вместе с тем, позволяет прервать их в любой момент, если правильно записать <условие>.

Поскольку известно, что коды цветов занимают закрытый интервал [1-56], то введем в рассмотрение переменную КодЦвета, будем придавать ей различные значения и проверять их по условию: КодЦвета >=0 и одновременно КодЦвета <=56.

Изменённая программа примет вид:

```
Sub PaintCell()
Dim n As Integer
n = 0
   While КодЦвета >= 0 and КодЦвета <=56
        n = (InputBox("Введите код цвета:"))
       ActiveCell.Interior.ColorIndex = n
  Wend
MsgBox ("До свидания. Работа закончена!")
End Sub            ' Конец  процедуры "Цвет ячейки"
```

Рис. 8.11. Текст (код) макроса цикла изменения цвета активной ячейки

Теперь программа будет работать и менять цвета в активной ячейке, пока в качестве значения **n** не будет введено отрицательное число.

Тогда вместо запроса на ввод кода цвета она выдаст в окне сообщения пользователю **MsgBox** фразу "**До свидания. Работа закончена!"**

8.6. Сохранение макросов в виде модулей

Текстовое представление операторов на языке VBA, содержащееся в исходном коде макроса, сохраняется в специальной части файла данных приложения, называемой ***модулем***. Файл документа соответствующего типа любого из VBA - приложений может содержать один или несколько модулей или не содержать их вообще. Модулям, сохраняемым в документах Excel, по умолчанию присваивается общее название – VBAProject (VBA - проект).

При записи макроса в Excel в окне ”Запись макроса” требуется указать рабочую книгу (формата . xlsm или . xlsb), в которой будет сохранен записанный макрос (рис. 8.2). Для этого в списке **Сохранить** в следует выбрать одно из доступных значений, определяющее тот документ, в котором будет сохранен записанный макрос: “*Личная книга макросов”*, “*Новая книга”* и “*Эта книга”*. Макросы, имеющие непосредственное отношение к текущей рабочей книге, рекомендуется сохранять, выбрав в списке значение “*Эта книга”*. После того как место хранения макроса будет выбрано, щелкните на кнопке ОК.

Если в документе, выбранном для хранения записываемого макроса, еще нет модуля, он будет создан с именем Module1.

Для того чтобы найти модуль с интересующим вас макросом в любом из приложений VBA, необходимо выполнить следующие действия.

1. Открыть редактор VBA, выбрав на ленте приложения вкладку Разработчик и щелкнув в группе Код на кнопке Visual Basic (крайняя слева).

2. Открыть окно проектов, выбрав в окне редактора VBA команду меню View → Project Explorer (Вид =>Окно проекта).

3. В окне Project Exp1орег найти в иерархии компонентов имя необходимого модуля и дважды щелкнуть на нем.

После выполнения указанных действий редактор VBA откроет окно кода с текстом выбранного модуля. В списке Declaration (Объявления) этого окна (справа вверху) выберите имя требуемого макроса в открытом модуле, после чего в данном окне можно будет просмотреть или отредактировать исходный код этого макроса.

ПРИМЕЧАНИЕ 6: при записи макросов в приложениях Microsoft Office 2007 их текст записывается в виде последовательности операторов языка VBA. Таким образом, можно использовать эти средства для создания *заготовок* элементов программ, которые предстоит написать на языке VBA. Вместо того, чтобы вручную программно описывать некоторую последовательность действий в том или ином приложении Office, можно просто записать соответствующий макрос, а затем скопировать его код в текст создаваемой программы.

Тестовые задания

Ниже приведено несколько тестовых заданий, предназначенных для закрепления изложенного в этой главе материала. В каждом задании предлагается вопрос и несколько вариантов ответа на него, один (или несколько) из которых является правильным, а остальные — нет. Укажите правильный ответ.

№ вопроса	Вопросы	Предлагаемые ответы
1	Язык VBA представляет собой…	а) самостоятельное приложение работы с данными, б) объектно-ориентированный язык программирования, в) процедурный язык программирования, г) инструмент обработки данных в приложениях Microsoft Office.
2	Редактор Visual Basic представляет собой…	а) интегрированную визуальную среду разработки, б) самостоятельное приложение Microsoft Office, в) самостоятельное приложение Microsoft Windows, г) транслятор с языка Visual Basic.

3	Макрос в языке VBA — это…	а) самостоятельная программа Microsoft Office, б) средство управления работой приложений Microsoft Office, в) поименованная запись последовательности действий пользователя для упрощения ее многократного повторения, г) фрагмент документа приложения Microsoft Office.
4	Код макроса на языке VBA сохраняется:	а) в теле документа приложения Microsoft Office, б) в шаблоне документа Microsoft Office, в) в отдельном файле с расширением .vba, г) в системных библиотеках.
5	Основными понятиями объектно-ориентированного программирования являются…	а) процедуры и функции, б) объекты и события, в) константы и переменные, г) свойства и методы, д) классы.
6	Какие программные элементы не относятся к VBA?	а) объекты, б) методы, в) процедуры, г) свойства.
7	Готовый программный элемент интерфейса «человек-компьютер» имеющий совокупность свойств и методов, это…	а) событие, б) объект, в) класс объектов, г) инкапсуляция.
8	Свойства характеризуют…	а) текущее состояние объекта, б) возможное состояние объекта, в) сходство с другим объектом, г) уникальность объекта.
9	Объектом VBA не может быть …	а) окно, б) вводимая информация, в) кнопка, г) флажок.
10	Метод – это…	а) ответ VBA на действия пользователя, б) подпрограмма для изменения свойств объекта, в) средство для изменения события, г) описание свойств объекта.
11	Классы объектов…	а) упорядочивают объекты по их важности, б) объединяют разные по назначению объекты, в) описывают сходные свойства различных по назначению объектов, г) описывают свойства сходных по назначению объектов интерфейса.
12	Макрос – это…	а) окно, б) экранная форма, в) набор автоматически зафиксированных

		и одинаково выполняемых операций, г) набор свойств объекта.
13	Имя макроса не должно содержать …	а) буквы, б) буквы и цифры, в) знак подчеркивания, г) пробелы.
14	При создании макроса не надо вводить…	а) длину его имени, б) имя макроса, в) указания на место его сохранения, г) описание макроса.
15	Укажите порядок действий при создании макроса:	а) ввести имя макроса, б) включать команды начала его записи, в) указать место его сохранения, г) включить команду конца его записи.
16	Включение параметра «Относительные ссылки» применяется	а) для автоматического перехода в ячейку, с которой должен выполняться макрос, б) для выполнения макроса с текущей ячейки, в) для выполнения макроса с произвольной ячейки, г) для полного описание макроса.
17	Активизация ячейки для работы с ней производится по команде	а) **Range**(“имя ячейки”).**Select**, б) **ActiveCell**, в) **Selection**.**Borders**(), г) **Selection**.**Font**.**Bold**.
18	Выбор нужного шрифта для вводимых данных происходит по команде	а) **Range**(“имя ячейки”).**Select**, б) **ActiveCell**, в) **Selection**.**Borders**(), г) **Selection**.**Font**.**Bold**.
19	Выделение нужного столбца(ов) для работы в нем (них) происходит по команде	а) **Range**(“имя ячейки”).**Select**, б) **Column** ().**Select**, в) **Selection**.**Borders**(), г) **Selection**.**Font**.**Bold**.
20	Назначение нужного стиля рамки для активной ячейки происходит по команде	а) **Range**(“имя ячейки”).**Select**, б) **ActiveCell**, в) **With Selection**.**Borders**(), г) **Selection**.**Font**.**Bold**.
21	Работая в Excel нельзя сохранить макрос в виде модуля	а) в личной книге макросов, б) в этой (текущей) книге, в) в новой книге, г) в оперативной памяти ПК.

Глава 9. Создание и выполнение VBA – программ

9.1. Понятие об общем цикле создания VBA – программы

Процесс создания программы обычно разбивается на ряд этапов:

1. Тщательный анализ задачи, решение которой предстоит запрограммировать.

2. Проектирование программы, т.е., подразделение её на отдельные блоки, каждый из которых будет предназначен для решения какой-нибудь локализованной подзадачи. Вместе с тем продумать, как эти блоки будут связаны друг с другом общим процессом решения всей задачи.

3. Поскольку мы имеем дело с ***визуальным*** программированием, то не обойтись без всевозможных окон (форм), в которых надо предусмотреть управляющие элементы и зоны для ввода или выбора данных.

4. Для каждого из элементов таких форм надо продумать, какие действия должны происходить в программе при воздействии на эти элементы. Вот тут уже придётся писать соответствующие подпрограммы (процедуры), которые эти действия будут производить в общей программе.

5. Когда этапы 1 - 4 пройдены, т.е., детальный проект составлен, можно переходить к его реализации:

- проектирование внешнего вида нужных форм и размещение на них элементов управления;
- описание свойств этих элементов в среде редактора VBA;
- написание программного кода (текста программ на языке VBA) необходимых процедур и т. д.

6. Следующий шаг – запуск программы на исполнение и выявление ошибок в ней.

7. При обнаружении ошибок в синтаксисе текста программы или ошибок в самом алгоритме (это выявляется посредством выполнения контрольного

примера с заранее известными результатами) наступает период *отладки* программы.

8. Отладка сводится к выявлению характера ошибки, её устранению и повторному исполнению программы. Эти действия повторяются до тех пор пока все ошибки не будут устранены. У программистов бытует такое шуточное высказывание об этом периоде работы над программой: "Ещё одна ПОСЛЕДНЯЯ ошибка!"

9. Последний этап – "прогон" проекта на реальных данных и сдача его в эксплуатацию.

Уточним некоторые подробности.

Создавать новую форму, размещать в ней надписи, поля ввода, командные кнопки и другие элементы управления, а также писать программный код необходимых процедур, удобнее всего с помощью визуальной среды разработки, предоставляемой редактором VBA.

Например, для создании новой формы потребуется лишь выбрать в меню редактора VBA команду Insert =>UserForm, и на экран будет выведено окно новой, пока еще пустой формы. Затем, копируя в форму заготовку элемента управления *надпись,* можно будет добавить в нее необходимые надписи, а с помощью копирования заготовки элемента управления *командная кнопка,* можно будет поместить в форму любые требуемые кнопки, — например, ту, после щелчка на которой выполнение вашей программы будет завершаться. Наконец, при выборе команды меню View => Code в редакторе VBA будет открыто окно программного кода, предназначенное для записи текста необходимых процедур обработки событий формы и ее компонентов, а также разнообразных вспомогательных процедур.

После того как программа, наконец, будет готова, следует перейти к третьему этапу работы — тестированию созданной вами программы. Этот процесс предполагает тщательный анализ функционирования только что созданной программы в самых различных режимах, назначение которого — убедиться,

что она работает в полном соответствии с вашими планами. Для этого удобнее всего воспользоваться командой меню редактора VBA Run ... (Выполнить ...). При выборе этой команды вновь созданная программа запускается на выполнение и появляется возможность проверить, как она реагирует на те или иные события, правильно ли обрабатываются введенные данные (как корректные, так и заведомо ошибочные), позволяет ли она решить ту задачу, для которой, собственно, и была создана.

При обнаружении любых ошибок или просто нежелательных отклонений от нормальной работы требуется перейти к четвертому этапу разработки программы — провести ее отладку. Суть этой процедуры состоит в том, что с помощью тех или иных средств выясняется причина появления ошибки в работе программы и принимается решение о способах ее устранения. Теперь потребуется вернуться ко второму этапу и внести в программу необходимые изменения, а затем вновь перейти к третьему этапу и повторить ее тестирование.

Этот цикл повторяется до тех пор, пока тестирование не покажет отсутствие каких-либо отклонений от ожидаемого поведения программы. Надо заметить, что редактор VBA предлагает несколько мощных и эффективных средств автоматизации процесса отладки.

9.2. Общие принципы построения VBA-программы

Программа *не является самостоятельным* структурным элементом в иерархии объектов языка VBA, и поэтому редактор VBA распознает по именам не программы, а *процедуры, модули* и *проекты.* Любая VBA-программа обязательно содержит хотя бы одну процедуру — по той простой причине, что компилятор языка VBA может выполнять только операторы, помещенные в процедуру. Однако выполняемая программа-процедура может, в свою очередь, обращаться к одной или нескольким другим процедурам, помещенным в один или несколько модулей, входящих в состав одного или нескольких проектов.

Другими словами, правильнее будет сказать, что в VBA строки программного кода организованы в процедуры, которые размещаются в модулях, а модули размещаются в проектах. Отсюда можно сделать вывод, что программный код VBA состоит из следующих "строительных блоков":

Оператор — это наименьшая единица VBA-кода. Он предназначен для определения переменной, установки параметров или выполнения какого-либо действия в программе.

Процедура — это отдельная единица программного кода VBA, *которую* можно вызывать по имени для выполнения; она может выполняться самостоятельно. Любая процедура содержит один или несколько операторов.

Модуль — это именованная единица, состоящая из одной или нескольких процедур и раздела объявлений, в котором объявляются переменные, константы и пользовательские типы данных, а также устанавливаются параметры компилятора (о них мы будем говорить ниже в этой главе).

Проект — включает в себя все модули, формы и связанные с приложением объекты, относящиеся к конкретному документу, причем проект сохраняется вместе с самим этим документом.

9.3. Написание новых макросов и процедур

Нам уже известно, что макросы представляют собой средство, с помощью которого можно описать произвольную последовательность действий пользователя для ее последующего многократного выполнения.

Обычно термин "макрос" применяется для инструкций, которые автоматически записываются с помощью специальных инструментов в приложениях Microsoft Office, а термин "процедура" применяется к коду VBA, который пишется пользователем вручную. Иначе говоря, ***процедура*** — это компьютерная программа, которая выполняет некоторые действия с объектами и сохраняется

в модуле VBA. С такой, "программной" точки зрения макрос — это тоже процедура типа Sub, не имеющая входных параметров.

Макросы образуют единственный класс процедур типа Sub, которые можно запускать на выполнение непосредственно вызовом их по имени либо из 66редактора VBA, либо из самого VBA-приложения. Написать новый макрос без использования специальных автоматических средств Microsoft Office, т.е. самостоятельно написать соответствующую процедуру типа Sub, можно либо создав в редакторе VBA новый модуль, который будет содержать эту процедуру, либо поместив эту процедуру в уже существующий модуль. Для создания нового модуля необходимо выполнить следующие действия:

1. Убедиться в том, что вы работаете с нужным проектом.

2. Активизировать редактор VBA, нажав клавиши <**Alt+Fl 1**>.

3. В окне Project Explorer выделить либо сам проект, либо один из его компонентов.

4. Выбрать команду Редактора VBA **Insert** =>**Module** (Добавить=> Модуль) или щелкнуть на кнопке Insert панели инструментов Standard и выберите в раскрывшемся меню команду Module.

В результате редактор VBA откроет на экране пустое окно программного кода, предназначенное для создания нового модуля (см. рис.8.8). Следует напомнить, что новому модулю при открытии окна программного кода редактор VBA по умолчанию присваивает типовое имя Module *n*, где *n* — порядковый номер модуля, созданного в данном сеансе работы с редактором VBA.

Для того чтобы модуль получил некоторое собственное имя, следует переименовать его. Для этого:

1. Выбрать команду меню **View** => **Properties Window** (Вид=>Окно свойств) или щелкнуть на кнопке Properties Window в панели инструментов Standard. На экране раскроется пустое окно свойств вновь создаваемого модуля (см. рис. 9.1).

2. Здесь текстовом поле Name на вкладке **Alphabetic** окна **Properties** введите новое имя модуля. Обратите внимание, что модули имеют только одно свойство — Name (Имя).

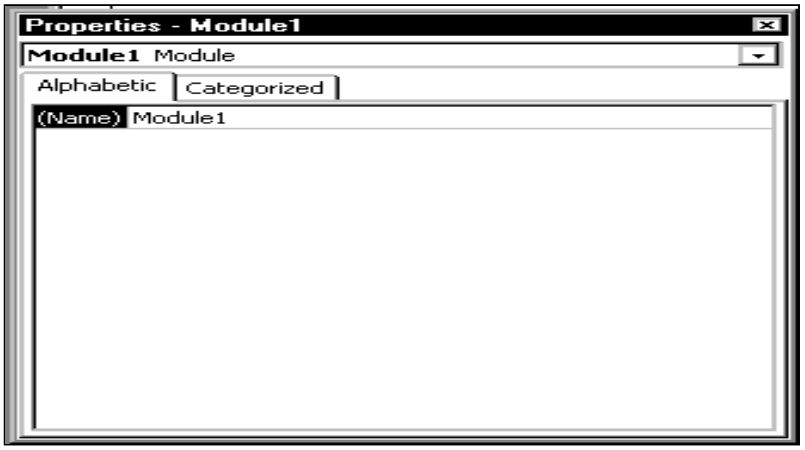


Рис. 9.1. Окно свойств модуля для ввода его имени

Прежде чем создать новую процедуру в *уже существующем* модуле, необходимо сначала открыть окно Code для того модуля, в котором предполагается поместить эту процедуру. Для этого щелкните дважды на требуемом модуле в окне Project Explorer либо выделите этот модуль в указанном окне и выберите команду меню **View** => **Code** (см. рис. 9.2):

Рис. 9.2. Окно для ввода кода (текста) модуля

Следует ещё раз напомнить, что модули не хранятся в отдельных файлах, а являются частями некоторого проекта, сохраняемого в соответствующем

документе VBA-приложения. Чтобы использовать один и тот же модуль в разных проектах (читай, в VBA-документах) его следует скопировать в них.

Для этой цели требуемый модуль можно предварительно экспортировать, создав отдельный файл с его текстом. Потом надо выделить его в Project Explorer, после чего выбрать команду меню **File** => **Export File** (Файл => Экспорт файла). В раскрывшемся на экране диалоговом окне нужно указать место сохранения экспортируемого файла и, при необходимости, изменить его имя (рис. 9.3):

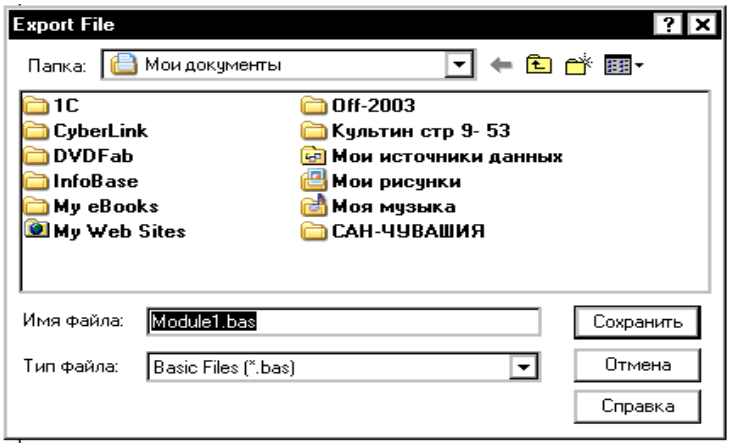


Рис. 9.3. Диалоговое окно Export File

Убедитесь, что в списке Тип файла выбрано значение Basic Files (*.bas), указывающее, что расширение имени файла определяет копируемый файл как файл исходного кода VBA (*.bas). Для завершения операции экспорта файла щелкните по кнопке "Сохранить".

Для того чтобы импортировать модуль, т.е. добавить экспортированный ранее файл типа *.bas* в любой из VBA-проектов, выберите в Project Explorer необходимый проект, а затем выберите команду меню **File** => **Import File** (Файл=>Импорт файла). В раскрывшемся: диалоговом окне Import File надо убедиться, что в списке Тип файла выбрано значение Files of Type (**.frm, *.bas,*

**.cls*) и дважды щелкнуть на имени того файла, который буде импортироваться (диалоговое окно "Import File" – такое же, как на рис. 9.3)

Если по какой либо причине потребуется удалить некоторый модуль из VBA-приложения, выполните следующие действия:

1. Щелчком мыши выделите в окне Project Explorer тот модуль, который требуется удалить.

2. Выберите команду меню **File** => **Remove** *object_name* (Файл =>Удалить *имя_объекта).* Другой способ — щелкните в окне Project Explorer правой кнопкой на удаляемом модуле и выберите в раскрывшемся контекстном меню Remove *object_name.*

В любом случае редактор VBA выведет на экран диалоговое окно с предложением предварительно экспортировать удаляемый модуль.

Поместить исходный код процедуры можно, в принципе, в любом месте модуля, независимо от того, добавляете ли вы процедуру в новый или уже существующий модуль.

Для этого поместите курсор в то место в модуле, где необходимо ввести текст новой процедуры. Ключевыми словами для процедур являются операторы **Sub** и **End Sub**, т.е. новая процедура должна начинаться оператором Sub, с указанием имени процедуры, и заканчиваться оператором End Sub, который редактор VBA *автоматически* добавляет в строку, сразу же после того, как был введен оператор Sub.

Если модуль содержит несколько процедур, то новая процедура должна начинаться после оператора End Sub предыдущей процедуры или перед оператором Sub, которым начинается следующая процедура в модуле. Надо обязательно отметить, что основными VBA-процедурами являются процедуры типа Sub (подпрограммы) и процедуры типа Function (функции).

Попробуем на практике создать простейшую процедуру в Word 2007 (или Excel). Предположим, требуется написать простейшую процедуру, которая будет выводить на экран приветствие. Текст ее будет выглядеть так:

Sub Приветствие ()

MsgBox "Привет! Я – ваша первая процедура! "

End Sub

Чтобы всё опробовать, создайте новый модуль с именем **Приветствие** и введите в окно его программного кода этот текст так, как это показано на рис. 9.4.

Для этого выполните следующие действия:

1. Откройте любой документ Word (или рабочую книгу Excel).

2. Активизируйте редактор VBA, нажав клавиши <Alt+Fl 1 >.

3. В окне Project Explorer выделите проект того документа или рабочей книги, в которой будет создана эта процедура.

4. Для добавления нового модуля к проекту выберите команду **Insert** => **Module**. На экране раскроется окно программного кода для вновь созданного модуля.

5. Присвойте новому модулю имя Приветствие и, убедившись, что курсор вставки текста находится в начале первой пустой строки, введите приведенный выше программный текст.

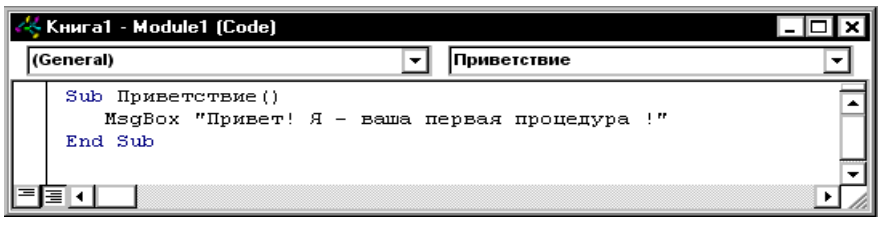


Рис 9.4. Создание процедуры "Приветствие"

ПРИМЕЧАНИЕ 7: чтобы компилятор VBA мог помогать обнаруживать синтаксические ошибки, советуем набирать ключевые слова (sub - процедура, msgbox – блок сообщения) б е з использования з а г л а в н ы х букв. Тогда,

если у вас всё верно, он сам в нужных местах перейдет к заглавным буквам. Если же этого не произойдёт, значит, допущена ошибка при записи!

Завершив ввод текста процедуры, надо попробовать ее выполнить. Для этого выберите команду меню редактора VBA **Tools** => **Macros** и в диалоговом окне Macros в списке Macro Name выберите процедуру "Приветствие", после чего щелкните на кнопке **Run** (Выполнить). Другой вариант: если вы уже в редакторе, выберите команду меню **Run** => **Run Sub / UserForm** или просто нажмите <**F5**>. В любом случае в окне Word (Excel) будет выведено окно сообщения с приветствием, записанным в тексте процедуры (рис. 9.5). Для закрытия этого окна щелкните на кнопке ОК.

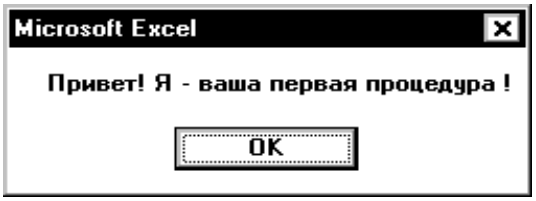


Рис. 9.5. Результат работы новой процедуры "Приветствие"

9.4. Выполнение VBA-программы

Итак, в самом общем случае VBA-программа представляет собой определенную последовательность команд, которые выполняются по порядку, одна за другой, при каждом запуске программы. Сама по себе любая VBA-программа является совершенно бесполезной, если не уметь её запускать на выполнение.

Один способ мы уже знаем: VBA-программу можно выполнить в окне редактора VBA, выбрав команду меню Run=>Run ..., щелкнув на кнопке Run ... панели инструментов Standard или нажав <F5>.

Несомненно, это очень удобно при тестировании программы, но совершенно не подходит для повседневной работы с ней. Запускать программу, готовую к выполнению, нужно (и гораздо удобнее) из того приложения, в кото-

ром вы работаете. При этом организовать ее запуск в приложении можно будет с помощью кнопки на панели инструментов, нажатия определенной комбинации клавиш, выбора команды меню или даже, предусмотрев ее автоматический запуск при открытии документа.

Повторим: макрос — это та же VBA-программа, а макрос, написанный самостоятельно (без использования специальных функций Microsoft Office), — это процедура. Поэтому выполнить VBA-программу фактически означает выполнить макрос или процедуру.

Программа может включать как одну, так и несколько процедур, но только одна из этих процедур будет выполняться *первой* при вызове программы на выполнение — имя этой первой выполняемой процедуры можно считать именем всей программы. Эта первая процедура может вызывать другие процедуры, но для выполнения *всей* программы вполне необходимо и достаточно запускать на выполнение только первую процедуру, а все остальные будут выполняться автоматически, по мере необходимости, непосредственно по ходу решения задачи.

При вызове программы на выполнение из диалогового окна Макрос следует найти в этом окне ее имя:

1) откройте диалоговое окно Макрос в том VBA-приложении, в котором будет выполняться ваша программа, т.е. перейдите на ленте приложения (для Office 2007) на вкладку Вид, раскройте в группе Макросы меню и выберите в нем команду Макросы. Вид диалогового окна Макрос приложения был показан ранее, однако некоторые детали в нем могут несколько отличаться, в зависимости от используемого VBA-приложения.

2) в диалоговом окне **Макрос** выбираем из списка доступных VBA-программ (или макросов) имя требуемой программы и щелкаем по кнопке Выполнить.

3) если в приведенном в окне списке макросов интересующая вас программа (или макрос) отсутствует, откройте раскрывающийся список *Макросы*

из (для Word) и выберите в нем вместо принимаемого по умолчанию значения *Активных шаблонов* то, которое описывает документ, в котором хранится интересующая вас программа. Для приложения Excel подобный список называется *Находится в* — по умолчанию в нем выбрано значение *Все открытые книги*.

9.5. Обработка ошибок

При написании кода VBA-программы безусловно могут быть допущены ошибки или случайные описки. Многие из подобных случайных ошибок редактор VBA автоматически отслеживает непосредственно во время написания или редактирования программного кода. Более сложные логические ошибки можно обнаружить компилятором при выполнении процедуры — он проверяет код процедуры на наличие ошибок непосредственно перед преобразованием программы в форму, понятную для компьютера. Если операция компиляции не будет успешной, появится окно с уведомлением об ошибке.

Данный уровень проверки намного строже, чем обычная проверка синтаксиса в каждой строке кода, так как на этом этапе проверяется правильность ссылок и тип переменных, а каждый оператор проверяется на корректность всех его параметров.

Однако ту часть ошибок, которые все еще остались необнаруженными компилятором придется отыскивать самостоятельно. Самыми простыми ошибками, с которыми вы непременно столкнетесь при написании процедур VBA, являются *ошибки синтаксиса* (syntax error). Если при написании программного кода была допущена синтаксическая ошибка, редактор VBA сообщит об этом сразу, не дожидаясь момента выполнения программы.

Всякий раз, когда редактор VBA не сможет понять и корректно интерпретировать введенную в строку последовательность символов в качестве того или иного оператора, этот текст будет выделен красным цветом. При попытке перейти с такой строки на другую, редактор VBA выдаст сообщение о наличии

ошибки с теми или иными разъяснениями. Для того чтобы описанный выше режим был активизирован, следует выбрать команду меню **Tools => Options** редактора VBA и в раскрывшемся диалоговом окне Option (Параметры) на вкладке **Editor** (Редактор) установить флажок опции Auto Syntax Check (автоматическая проверка синтаксиса).

Обнаруживая многие ошибки синтаксиса, редактор VBA информирует о пропущенных запятых, кавычках и т.д. Однако некоторые синтаксические ошибки редактор может обнаружить только тогда, когда программа начнет выполняться — например, если в операторе была указана ссылка на несуществующую в программе метку. Сообщение о такой ошибке будет выдано, когда начнется выполнение программы.

Подобные ошибки следует исправлять сразу же, как только редактор VBA их обнаруживает. В некоторых боле сложных случаях редактор VBA не может определить, что именно неверно в синтаксис данного оператора, и только сообщает о наличии неопознанной им ошибки.

Кроме ошибок компиляции, т.е. синтаксических ошибок, существуют еще и *логические ошибки* — неточности и несоответствия, допущенные при разработке самого алгоритма, воплощённого в программе. При наличии подобных ошибок программа делает не то, что было запланировано, или делает что-то не так, как это ожидалось. Причиной таких несоответствий чаще всего являются ошибки, заложенные в неверной логике алгоритма программы. Тогда не следует искать ошибки в записи операторов; необходимо еще раз тщательно продумать саму структуру программы и уточнить алгоритм ее поведения. Когда правильное решение будет найдено, вы обязательно получите требуемый результат.

Существует еще третий вид ошибок — это ошибки выполнения. Такие ошибки приводят к останову выполнения программы вследствие непредсказуемых ошибок в обрабатываемых данных или в действиях пользователя. Они вызывают возникновение непредусмотренных в программе ситуаций. Такие виды

ошибок далеко не всегда поддаются легкому обнаружению и исправлению. Здесь на помощь к разработчику приходят различные методы тестирования программ с последующей отладкой.

Вывод: чем тщательнее будет проведено тестирование, тем устойчивее и надежнее будет работать написанная вами программа.

Тестовые задания

Ниже приведено несколько тестовых заданий, предназначенных для закрепления изложенного в этой главе материала. В каждом задании вам предлагается вопрос и несколько вариантов ответа на него, один (или несколько) из которых является правильным, остальные — нет. Укажите правильный ответ.

№ п/п	Вопросы	Предлагаемые ответы
1	Общий цикл создания VBA-программ предусматривает такую последовательность действий:	а) анализ задачи, написание программного кода, тестирование и отладка, б) проектирование программы, написание программного кода, тестирование, отладка, в) анализ задачи, проектирование программы, реализация проекта, тестирование и отладка, г) анализ задачи, проектирование программы, написание программного кода, передача приложения в эксплуатацию.
2	Программными единицами, с которыми может работать редактор VBA, являются:	а) проекты, модули, процедуры, операторы, б) проекты, программы, макросы, в) программы, модули, процедуры, операторы, г) проекты, программы, модули, процедуры.
3	Обмен программными модулями между проектами осуществляется с помощью следующих операций:	а) копирования и вставки, б) экспорта и импорта файлов, в) совместного редактирования и слияния, г) пересылки и считывания.
4	Запуск VBA-программ можно организовать с помощью…	а) ярлыка на рабочем столе, б) сочетания клавиш, набираемых в окне VBA-приложения, в) кнопки на панели инструментов, г) команды в меню Пуск, д) команды в меню VBA-приложения.
5	Редактор VBA позволяет автома-	а) орфографические,

	тически обнаруживать следующие виды ошибок в программах:	б) синтаксические, в) логические, г) структурные.
6	Комментарии в программу помещаются для следующих целей:	а) для повышения "читабельности" кода, б) для пояснения смысла выполняемых действий, в) для защиты авторских прав, г) для описания назначения переменных и функций.

Глава 10. ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ VBA

Microsoft Visual Basic for Application - самая последняя версия популярного языка программирования, работающая под управлением операционных систем из семейства Windows. В VBA реализован визуальный стиль программирования.

Графический интерфейс формируется путём перетаскивания стандартных элементов управления в окно формы. Появилась возможность осуществлять непосредственное наблюдение за построением проекта, то есть проектировать приложение. Отпала необходимость в написании кода, нужного для обеспечения связи элемента управления с источниками данных - достаточно задать параметры в окне свойств элемента.

Среда Visual Basic прекрасно подходит для разработки приложений практически любого типа.

10.1 Типы данных в VBA.

В Visual Basic существуют следующие типы данных:

Табл. 10.1

Тип	***Размер***	***Диапазон значений***
Byte	1 байт	Целое от 0 до 255
Boolean (Булевый)	2 байта	True (Истина) или False (Ложь)
Integer (Целый)	2 байта	<= | 32 767 |
Long (Длинное целое)	4 байта	<= | 2 147 483 647 |
Single (С плавающей точкой одинарной точности)	4 байта	|-1,4013E-45 |<= Single <= |-3,4028E+38 | (отриц.числа) |1,4013E-45 | <= Single <= |3,4028E+38 | (полож.числа)
Double(С плавающей точкой двойной точности)	8 байт	От -1,8E+308 до -4,94E-324 - для чисел < 0, От 4,94E-324 до 1,8E+308 - для чисел > 0
Currency *(Денежный, масштабированное целое число)*	8 байт	Currency <= |922 377 203 685 477.5807|
Decimal (десятичное)	14 байт	28 цифр в дроб.части (для Ч≥0) или ±0,1E-15 (для Ч≤0)
Date (Дата)	8 байт	От 01.01.100 до 31.12.9999
Object (Объектный)	4Б (байта)	Любая ссылка на объект
String (Строка перемен. длины)	10 Б+длина строки	От 0 до 2 000 000 000 символов
String (Строка фиксир. длины)	Длина строки	От 0 до 65 400 символов
Variant (Вариантный числовой)	16 байт	Любое числ. знач-е вплоть до границ *Double*
Variant (Вариантный строковый)	22 Б+длина строки	От 0 до 2 000 000 000 символов
User-defined (Определяемый пользователем)	Любой	Определяется заданным типом данных

Коротко - о каждом из типов данных.

Первые 3 типа (Byte, Boolean, Integer) относятся к целым типам данных – это числа только с целой частью. Следующие 3 (*Single, Double, Currency)* – это числа с плавающей точкой, имеющие целую и дробную части. Тип *Currency* отличается ещё и тем, что после точки в дробной части всегда отображаются 4 разряда.

Назначение остальных типов данных следует из их названий. Отдельно надо сказать только о типе **Variant.**

Это - очень своеобразный, "коварный" тип. Он характерен тем, ***сам*** устанавливает ***тип*** данных ***в зависимости от содержимого*** (значения) переменной, которой он присвоен. Если значение – **число 5**, то переменная типа **Variant** принимает **тип Integer;** если **1,25,** то - **Single (**вещественный**);** если **текст**, то - **String (**строка**).** Кроме того, он может заключать в себе ещё и даты и объекты. Запомните ещё одну особенность**:** переменная типа **Variant** сама ***изменяет*** свой тип ***во время выполнения программы!*** Тип данных Decimal не самостоятелен, а всего лишь является подтипом **Variant.**

10.2. Переменные VBA.

Переменная - это поименованное место в оперативной памяти компьютера.

По ходу выполнения программы, значение, хранящееся в ней, может сохраняться или изменяться его.

Прежде, чем начать работать с переменной, ей надо присвоить имя, чтобы иметь возможность обращаться к ней. Затем задают и её тип.

Имя переменной можно выбирать произвольно, но при этом следует соблюдать следующие правила:

- имя обязательно должно быть *уникальным*;
- *максимальная* длина имени - 255 символов;
- имя должно *начинаться* с *буквы*;

- в качестве прочих символов имени д о п у с т и м ы *буквы, цифры и символ подчеркивания* (_); другие символы не допускаются;
- в качестве имени н е л ь з я использовать к л ю ч е в ы е (*зарезервированные*) слова языка Visual Basic (например, Print).

ПРИМЕЧАНИЕ 8: в имени можно использовать и строчные и заглавные буквы, однако, в отличие от Word или Excel, для Visual Basic это б е з р а з л и ч н о. Для него SumZarpl и sumzarpl – это одно и тоже.

В отличие от имени, переменная может быть приписана только к одному из ограниченного количества типов данных. Тип переменной зависит от того, данные какого типа она должна хранить во время исполнения программы.

10.3. Объявление переменных

Перед тем как впервые использовать переменную, поместив ее имя в оператор VBA, можно предварительно *объявить* ее, т.е. указать компилятору VBA, что требуется создать переменную определенного типа с данным именем.

Если переменная создается (т.е. используется в операторе) *без предварительного объявления* (в этом случае говорят, что она объявляется неявно), то такой переменной будет присвоен тип ***Variant***, принимаемый в языке VBA по умолчанию.

Для явного объявления переменной используется оператор **Dim**, с помощью которого в языке VBA резервируется область памяти, предназначенная для хранения данных указанного типа. На эту область и будет ссылаться переменная. Синтаксис оператора Dim таков:

Dim Имя_переменной [**As** Тип_данных]

Необязательная часть объявления переменной заключена в скобки [ ] и может быть опущена. Например, объявления переменных Name и Name1:

Dim Name **As String**

Dim Namel

В первом операторе Name — имя переменной, а String (*строчный)*— присвоенный ей тип данных.

Во втором операторе конструкция As String (*как строчный*) отсутствует, поэтому переменной Namel будет присвоен тип Variant.

П О М Н И Т Е : переменные типа *Variant* требуют для своего хранения гораздо б о л ь ш е памяти и работа с ними з а м е д л я е т выполнение программы. Хотя такие переменные не ограничены хранением какого-то одного типа данных, на практике ситуация, когда в одной переменной требуется хранить данные разных типов, встречается довольно редко.

В одной строке можно объявить и несколько переменных. (через запятую), при этом ключевое слово Dim задается только один раз, переменные перечисляются через запятую – каждая со своим, указанным для неё типом данных. Кроме того, в одном операторе Dim допускается смешивать объявления переменных различных типов:

Dim Name **As String,** TabNom **As Integer**

Как правило, используемые переменные описываются в начале процедур. Это необязательно, но группировка объявлений переменных в начале процедуры позволяет, если нужно, быстро отыскать их в операторах программы.

10.4. Область действия переменной

После объявления переменную можно неоднократно использовать в той части программы, где она будет доступной. Такая часть программы, в рамках которой некоторая переменная является доступной, называется *областью действия* данной переменной.

Любая переменная может использоваться только в области своего действия, поскольку вне этой области данная переменная считается просто несуществующей и программе о ней ничего неизвестно.

Область действия каждой переменной зависит от двух взаимосвязанных факторов:

1) места объявления этой переменной и

2) указанных при ее объявлении ключевых слов.

Для областей действия переменных можно выделить ***три*** уровня.

1. *Переменная, объявленная* ***в процедуре****, является доступной* ***только в этой процедуре.***

Если переменная используется без объявления или объявлена внутри процедуры с помощью ключевого слова Dim, то такую переменную можно использовать только внутри этой процедуры.

2. *Переменная, объявленная* ***в разделе объявлений модуля*** *с помощью ключевых слов* **Private** *или* **Dim***, является доступной* ***только в пределах этого модуля.***

Ключевое слово Private в этом случае работает точно так же, как и Dim. Для того чтобы переменная, объявленная с использованием этих ключевых слов, была доступна для всех процедур в данном модуле, ее объявление необходимо поместить в начало всего модуля, перед кодом всех его процедур.

3. *Переменная, объявленная в разделе объявления модуля* ***с помощью ключевого слова Public****, является доступной для* ***всех*** *модулей* ***во всех*** *проектах.*

При наличии в разделе объявлений модуля оператора Option Private Module переменные, объявленные с использованием ключевого слова Public, будут доступны всем модулям только данного проекта.

Всё это трудно понять до тех пор, пока не будет рассмотрен какой-нибудь конкретный пример, в котором бы встречались все 3 варианта области действия переменных.

Рассмотрим следующий абстрактный (не "привязанный" к конкретной задаче) пример (см. рис.10.1).

Здесь переменные ***a*** и ***b*** доступны в пределах всего модуля, в состав которого входит приведенный выше фрагмент. Переменная ***c*** будет доступна для

всех модулей, но только **данного** проекта (поскольку указан оператор Option Private Module).

Переменная ***x*** будет доступна **только внутри** самой **процедуры** NewSub, поскольку она объявлена непосредственно в ней.

```
Option Private Module
        Dim a As Integer
        Private b As Long
        Public c As Single
        . . . . . .
        Public Sub
NewSub()
        Dim x As Single
        . . . . . .
End Sub
```

Рис. 10.1. Способы объявления переменных в VBA-программе

10.5. Присвоение значения переменной

Когда переменная объявлена, можно поместить в эту переменную какую-то информацию, т.е., *присвоить* ей значение. Это присвоенное значение будет храниться в переменной до тех пор, пока ей не будет присвоено другое значение.

Значение переменной присваивается специальным оператором, называемым *оператором присваивания,* в котором имя переменной (слева) соединяется с присваиваемым ей значением (справа) с помощью знака равенства. Например, в операторе присваивания **x = 5** переменной x присваивается значение 5.

Такие явно указанные значения называются *литеральными* значениями, или просто *литералами,* однако точно так же можно присваивать значения, вычисляемые на основе значений других переменных. Рассмотрим следующий оператор:

```
Summa = CenaZaOdinKg * Wes
```

Здесь переменной *Summa* присваивается значение, равное произведению значений переменных *CenaZaOdinKg* и *Wes*. Отсюда следует, что в языке VBA оператор присваивания представляет собой конструкцию, в которой знак равенства (=) связывает переменную, имя которой указано слева от него, с выражением, определяющим новое значение этой переменной и указанным справа от знака равенства.

При выполнении оператора присваивания языка VBA сначала вычисляется выражение, стоящее справа от знака равенства, а затем полученный результат сохраняется в переменной, имя которой указано слева от знака равенства.

10.6. Константы

При написании программы иногда удобнее и правильнее пользоваться значениями, которые не будут меняться на протяжении всего хода выполнения программы - *константами*.

Использование констант делает программы более наглядными и упрощает внесение в них исправлений. В языке VBA используются константы двух типов — л и т е р а л ь н ы е и с и м в о л и ч е с к и е.

Литеральная *константа* — это такая константа, действительное значение которой (строка символов или число) записывается прямо в тексте программы. Такие константы можно изменять только при редактировании программного кода. В языке VBA строчные литеральные константы записываются в двойных кавычках (" "), а числовые — без специального форматирования.

```
Dim Имя As String
Dim Число As Single
Имя = "Саша"
Число = 12.5
```

Символическая *константа*, как и переменная, имеет своё имя, но, зато в отличие от переменной, значение такой константы никогда не меняется на всем протяжении выполнения программы.

Обычно символические константы применяются для таких значений, которые будут использоваться в программах многократно. Перед тем как использовать символическую константу в программе, ее следует обязательно объявить.

После объявления символическая константа может использоваться в любом месте программы наравне с переменными и с учетом обычных правил для области действия символических имен. Иначе говоря, если константа объявлена в процедуре, то областью ее действия будет только данная процедура. Если вы хотите, чтобы константа была доступна всем процедурам модуля, ее следует определить в области определений данного модуля. Для объявления констант используется ключевое слово Const:

Const СкидкаНаТовар = 12%

Const ВидУслуги = "Кредит"

Тип данных при определении констант указывать необязательно. Явно задавать тип данных объявляемой константы нужно только в том случае, когда ее необходимо сохранить с определенным типом данных.

Если при объявлении константы тип данных не указан, то VBA использует тот тип, который будет соответствовать заданному значению. Например, константа, содержащая строку символов, по умолчанию будет сохранена с типом String.

Следует также отметить, что язык VBA предоставляет в распоряжение пользователя набор *внутренних* констант, которые были определены разработчиками VBA специально. Для того чтобы увидеть полный список внутренних констант, доступных в языке VBA, нужно открыть любой документ Word (или рабочую книгу Excel) и войти в редактор VBA. Это можно сделать, нажав сочетание клавиш <Alt+Fll>.

Теперь в редакторе через меню View надо открыть диалоговое окно Object Browser - обозреватель объектов (рис. 10.2):

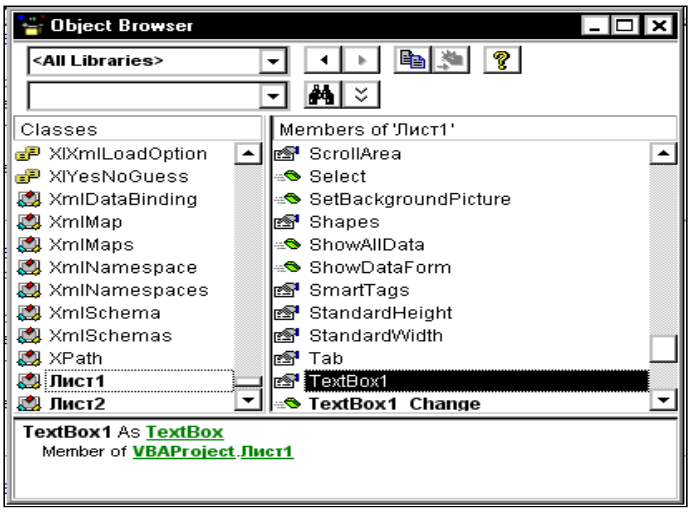


Рис. 10.2. Окно обозревателя (Browser) объектов

и в нём выполнить следующие действия:

- в левом верхнем раскрывающемся списке окна Object Browser выберите значение **VBA**.
- ниже (в поле списка **Classes)** выберите значение **Constants**. Справа, в поле списка Members of 'Constants' будет показан полный список внутренних констант языка VBA (рис. 10.3):

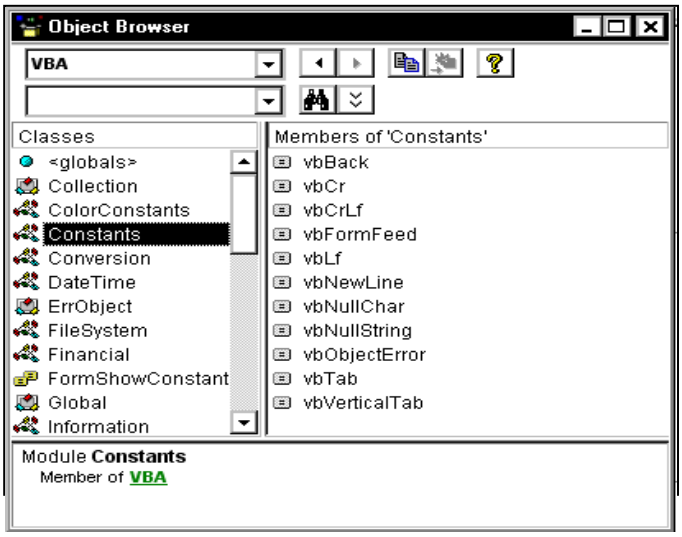


Рис. 10.3. Выделен тип и список объектов вида Constant .

➢ для того, чтобы узнать значение конкретной внутренней константы надо теперь выбрать ее имя в поле списка Members of 'Constants' - в нижнем поле диалогового окна Object Browser будет выведено имя этой константы и ее значение. Более подробные сведения об этой константе (только на английском языке!) можно получить, щелкнув по кнопке Help в окне Object Browser: (рис. 10.4):

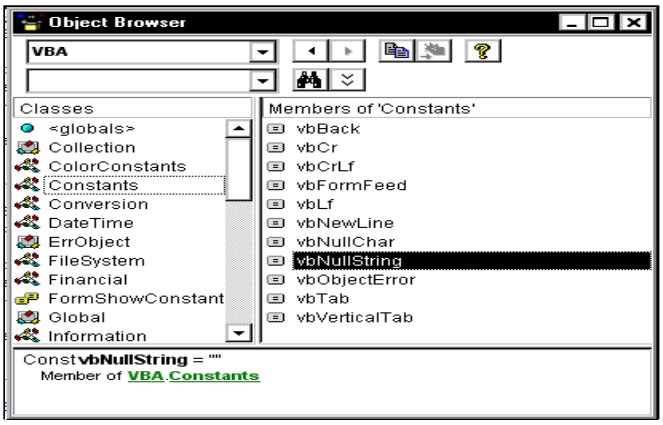


Рис. 10.4. Внизу – информация о значении константы: "" - пустая строка

Напоследок заметим, что работать с символической константой очень удобно, так как для изменения её значения нужно отредактировать лишь её объявление. Это можно сделать либо в процедуре, которая ее использует, либо в области объявления модуля (если эта константа используется в нескольких процедурах модуля).

10.7. Массивы

Массив — это набор элементов одинакового типа, имеющих общее имя.

Массивы позволяют работать с целым набором однотипных данных как с единым целым. В алгоязыке VBA каждый конкретный массив должен иметь своё собственное имя и сдержать один или более элементов.

Каждый из таких элементов занимает в массиве определенное, пронумерованное место. Поэтому, чтобы найти в массиве нужный элемент, доста-

точно указать имя этого массива и номер (индекс) того элемента, с которым требуется в данный момент работать.

10.7.1. Одномерные массивы

Пусть у нас есть набор чисел, представляющий собой список телефонных номеров:

5533467 2382219 5541235 2312278 2641710 2401199

Массив, представляющий собой *простой список* данных, называется *одномерным* массивом, а число в скобках, стоящее рядом с именем массива, называется *индексом элемента* данного массива.

Если присвоить этому массиву имя TelefNomer, то обращаться к его пятому элементу (2401199) надо так: TelefNomer (5).

Как же так? Ведь этот телефонный номер явно шестой по порядку!

Дело в том, что в языке VBA по умолчанию нумерация элементов в массиве начинается *с нуля,* однако с помощью оператора Option Base можно изменить начальный индекс элементов массива. Поэтому в нашем примере при стандартной нумерации элементов, записывая TelefNomer (5), мы обращаемся именно к пятому (считая от нулевого!) элементу от начала массива, т.е., к телефонному номеру 2401199. Если же перед именем массива поставить ещё и ключевое слово **Int**, то оно будет указывать уже на то, что в этом массиве хранятся только целые числа. Поэтому любое, хранящееся в массиве intTelefNomer значение также будет представлять собой целое число.

10.7.2. Многомерные массивы

Кроме одномерных массивов, которые подходят только для представления в программе простых списков с единственным индексом, в языке VBA можно создавать и *многомерные* массивы. Они могут иметь два и более индексов. В случае двух индексов сохраняемые в массиве данные можно представить в виде таблицы, состоящей из строк и столбцов.

Например, **Dim** Matrica (1 to 4, 1 to 6) — это матрица размером 4x6 элементов, состоящая из четырёх строк и шести столбцов. Количество размерностей массива в языке VBA не ограничено, т.е., разрешается создавать массивы как одно- или двухмерные, так и шестидесятимерные и т.д. Но чаще всего используются одномерные, двух- и трёхмерные массивы.

Иногда значение, хранимое в элементе массива, может случайно совпасть (или иметь сходство) с его индексом. Чтобы не запутаться, надо чётко понимать, что индекс в VBA – всего лишь средство обнаружения места, где находится соответствующее значение. Причём, это место определяется от начального адреса массива в оперативной памяти, где все его элементы размещаются последовательно.

В оперативной памяти каждому элементу отводится столько байтов, сколько определенно его типом. Все переменные могут ссылаться только на один адрес в памяти, в том числе и переменные массивов. Они ссылаются на адрес первого элемента.

10.8. Статические и динамические массивы

Язык VBA поддерживает два типа массивов — *статические* и *динамические*.

Статическими называют такие массивы, *размерность* которых была указана *непосредственно* при их *объявлении*.

В этом случае размер массива остается ф и к с и р о в а н н ы м на протяжении в с е г о в ы п о л н е н и я программы. Для объявления массивов используются те же самые операторы Dim и Public, что и при объявлении обычной переменной, причем с теми же правилами определения его области действия. Объявить массив фиксированного размера можно, указав в скобках после его имени конкретные значения для каждого его измерения. Например,

```
'Одномерный массив из 51 элемента
Dim Ml(50) As Integer
```

'Двухмерный массив из 11 строк по 21 элементу

Dim M2(10,20) **As Integer**

В этом примере первый оператор объявляет одномерный массив Ml из 51 целого числа. Как уже говорилось, в языке VBA по умолчанию принято нумерацию индексов начинать с нуля. Поэтому элементы массива M1будут пронумерованы от 0 до 50. Используя этот же принцип, аналогично, второй оператор объявляет двухмерный массив M2, который будет содержать 11 строк по 21 целому числу.

Итак, доступ к элементам объявленного массива осуществляется с помощью указания его индекса(ов) - целых чисел. Индекс может быть литералом, значением целочисленной переменной или же значением указанного элемента массива целых чисел. Самое важное в том, чтобы указанная величина в данный момент была целым числом - индексом нужного нам элемента массива.

Например, чтобы присвоить значение 215 **тридцать девятому** элементу массива Ml можно воспользоваться любым из трех указанных ниже способов:

```
Ml (38)  =  215
```

или

```
x =  38
M1(x)  =  215
```

или

```
x =  38
M1(0) = x
M1(M1(0))  =  215
```

Для того чтобы сослаться на элемент двухмерного массива, необходимо указать два индекса: M2 (9, 12). В этом примере мы обращаемся к 13 элементу 10-й строки.

Если вариант нумерации элементов в массиве, начиная от нуля, почему-либо не подходит, можно поместить в начало модуля оператор **Option Base 1**, указывающий, что нумерация индексов массивов должна начинаться с 1. Но

надо учитывать, что действие этого оператора распространяется только на те массивы, которые находятся в н у т р и данного модуля.

Такого же эффекта в нумерации индексов можно достичь с помощью ключевого слова **To** (см. пример в пункте "Многомерные массивы"). Он более универсален и позволяет задавать любое начальное значение индексам массива:

Dim Stroka(10 **To** 25, 30 **To** 40) **As String**

Здесь происходит объявление двухмерного массива строковых переменных с именем Stroka, в котором по первому измерению имеется 16 позиций (от 10-й по 25-ю), а по второму измерению — 11 позиций (от 30-й по 40-ю), т.е. всего 176 строковых переменных.

В отличие от статических массивов, ***динамические*** *массивы* имеют п е р е м е н н о е количество элементов, т.е., динамические массивы могут увеличиваться или сокращаться в зависимости от того, какое число элементов нужно в заданный момент исполнения участка программы. Объявлять динамический массив целесообразно в следующих случаях:

1) когда требуемый размер массива неизвестен заранее, до выполнения программы;

2) если наоборот - заранее известно, что по ходу исполнения программы размер массива должен меняться;

3) если после завершения использования массива в программе нужно освободить занимаемую им память.

При объявлении ***динамического*** массива его размерность (индексы) в скобках после имени массива ***не указываются***.

Dim D1( ) **As String**

Прежде чем использовать динамический массив в какой-то процедуре, обязательно нужно поместить в нее же оператор **ReDim** (*переопределение* массива), который бы задавал действительную размерность этого массива.

Оператор **ReDim** может быть указан для одного и того же массива сколько угодно раз, всякий раз задавая для него новую размерность и число элементов.

```
Dim Ml ()
Dim M2 ()
Dim M3()
. . . . . .
ReDim Ml (5)   '1 измерение,  6  элементов
ReDim M2 (5  To  11)'1 измерение,  7  элементов
ReDim M3 (1  To  6,  1  To  10)   '2  измерения,  60  элементов
```

При *о б ы ч н о м переопределении* массива его содержимое *полностью у н и ч т о ж а е т с я.*

Если же при этом необходимо *сохранить* уже существующие в нем данные, надо ввести необязательное ключевое слово ***Preserve.***

```
Dim M()  As  Single
ReDim M (l  To  10,  1  To  20)  j
ReDim Preserve M (l  To  10,  1  To  50)
```

Здесь сначала объявляется динамический массив M, затем он переопределяется как двухмерный массив заданного размера. Затем его размер увеличивается по второму измерению с сохранением уже имевшихся к этому моменту в нём данных.

Надо учитывать, однако, что использование ключевого слова Preserve не всегда "спасает положение":

- при сокращении размеров массива д а н н ы е, оказавшиеся з а п р е д е л а м и его новых размеров, будут п о т е р я н ы.
- р а з м е р н о с т ь массива (количество измерений) н е л ь з я и з м е н и т ь.
- в м н о г о м е р н ы х массивах можно менять размер только п о с л е д н е г о измерения.

В VBA можно также передавать значения элементов одного массива другому. Для этого надо применить простой оператор присваивания, указав в нем в качестве операндов имена этих массивов:

МассивДанных = МассивИсходныхДанных

Но для того, чтобы не было "неожиданных эффектов", нужно

- либо *совпадение типов и количества элементов* в обоих массивах,
- либо массив, которому присваиваются значения (слева от знака равенства), *должен быть динамическим*, и сохраняемые в нем данные должны иметь тип данных исходного массива.

В противном случае компилятор VBA автоматически заменит тип данных, размер и размерность результирующего массива на соответствующие характеристики исходного массива.

Для очистки и удаления массивов в языке VBA применяется оператор **Erase**. После заполнения элементов массива данные в них хранятся до тех пор, пока программа не присвоит им новые значения или пока VBA не удалит этот массив из памяти.

Оператор **Erase** позволяет очистить все элементы массива при применении его к статическому массиву. В случае динамического массива оператор Erase удаляет из памяти сам массив, освобождая при этом ту область памяти, которая ранее использовалась этим массивом.

Чтобы вновь использовать удаленный динамический массив, его следует переопределить с помощью оператора **ReDim**.

- Что касается статических массивов, то действие оператора Erase зависит от конкретного типа элементов данного массива:
  - для любого числового типа оператор **Erase** записывает во все элементы массива значение **нуль**;
  - для строкового типа с *переменной* длиной строки оператор **Erase** помещает во все элементы массива строку **нулевой** длины, а строки *фиксированной* длины **полностью заполняются символами пробелов**;

- для типа **Variant** оператор **Erase** устанавливает все элементы массива на **Empty** (пусто);
- для типа **Object** оператор **Erase** устанавливает элементы массива на **Nothing** (ничто, ничего);
- для пользовательского типа оператор **Erase** устанавливает числовые типы на **0**, строковые — на **строки нулевой длины**, Variant — на **Empty**, а Object — на **Nothing**.

10.9. Структура текста программы и комментарии

По ходу выполнения проекта неоднократно придется возвращаться к ранее написанным программам, чтобы проверять и изменять их.

Поэтому каждая программа должна быть написана таким образом, чтобы в ней легко можно было находить (причем не только ее создателю, но и другим программистам) места, в которые требуется внести те или иные изменения. Иначе говоря, текст программы должен быть написан максимально понятно для любого человека, которому впоследствии потребуется в ней разобраться, включая и самого ее автора. Эту мысль можно выразить более кратко — ваша программа должна быть "читабельна".

При написании программы надо взять за правило: для взаимосвязанных по смыслу операторов использовать одинаковые отступы.

Приведём пример такого написания и использования комментариев:

```
Do                          ' Начало общего цикла программы
    Switch = False
    For i = 1 to m          ' Цикл по заданному количеству циклов
       If  b(i) > b(i+1) then  ' оператор сравнения i-го и  i+1 члена
          bam = b(i)
          b(i) = b(i + 1)
          b(i + 1) = bam
          Switch = true
        End if              ' конец оператора сравнения
    Next i                  ' переход к новому циклу с увеличением i
Loop                        ' конец общего цикла программы
```

Рис. 10.5. Пример правильной записи фрагмента программы с выделением операторов, принадлежащих одной группе

Здесь ярко видно, что операторы внутри конструкции If ... Then ... End If — это операторы одной группы: для них установлен один и тот же отступ. Поскольку в данной программе оператор switch = False (switch – переключатель) и структура For ... Next выполняется при каждом проходе структуры Do ... Loop, то у них другой отступ.

Это позволяет с первого взгляда четко определять саму структуру и содержащиеся в ней подчиненные операторы. Для добавления отступа в строке можно использовать клавишу пробела или клавишу <TaЬ>.

При работе с программой также очень удобно использовать *комментарии* — текст, предназначенный для человека и не являющийся программным кодом и потому игнорируемый компилятором. Внесение в текст программы комментариев, словесно описывающих выполняемые в ней действия, является хорошей практикой.

В языке VBA комментарии записываются после символа апострофа «'», который можно поместить в любом месте строки (см. вышеприведённый при-

мер программы). При этом **все** символы после апострофа (и до конца строки) будут восприниматься как **комментарии!**

Комментарии можно размещать на отдельных строках или ставить их после операторов программы. Очень важно применять комментарии при написании сложных программ, смысл которых со временем может стать непонятным. В каждой строке желательно кратко пояснять, что и как делает данная программа. Это поможет вам при отладке программы, а также при добавлении в нее новых фрагментов, когда требуется четко знать, в какой процедуре и в какое ее место следует поместить новый фрагмент.

Поскольку каждая написанная процедура может использоваться многократно, для удобства ее дальнейшего применения целесообразно в начало каждой процедуры помещать комментарии, описывающие назначение и способ использования этой процедуры. Эти же рекомендации справедливы и в отношении комментариев, помещаемых в начало модулей. Комментарии также можно использовать для описания назначения переменных.

Вот рекомендации по использованию комментариев, которые могут помочь любому:

- будьте по возможности кратки;
- грамотно составляйте предложения;
- старайтесь писать ясно и понятно;
- умеренно используйте знаки пунктуации.
- обязательно поясняйте аргументы, передаваемые между процедурами;
- отмечайте версии фрагментов программы — когда, как и зачем выполнялись изменения;
- пишите комментарии параллельно вводу текста программы, пока суть задачи еще свежа в памяти.

Тестовые задания

Ниже размещено несколько тестовых заданий. В каждом задании предлагается вопрос и несколько вариантов ответа на него, один (или несколько) из которых является правильным, а остальные — нет. Укажите правильный(е) ответ(ы).

№ п/п	Вопросы	Предлагаемые ответы
1	В языке VBA тип данных определяет следующие характеристики некоторого элемента информации:	а) назначение элемента и смысл сохраняемой в нем информации б) способ представления в памяти в) скорость выполнения операций г) набор допустимых операций д) категорию информации и ее владельца
2	Язык VBA поддерживает следующие типы числовых данных:	а) размеры в метрических и дюймовых единицах измерения; б) целочисленные; в) дата и время; г) количественные; д) с плавающей точкой е) комплексные
3	Язык VBA поддерживает следующие типы нечисловых данных:	а) строки символов фиксированной и произвольной длины б) текстовые в) графические г) логические д) валюты
4	Массивы в языке VBA могут быть следующих типов:	а) линейные и плоские б) одномерные в) статические г) переменной длины д) циклические
5	Комментарии в программу помещаются для следующих целей:	а) для повышения читабельности кода; б) для пояснения смысла выполняемых действий; в) для защиты авторских прав; г) для описания назначения переменных и функций.

Глава 11. Примеры реализации различных макросов и фрагментов программ

11.1. Варианты реализации макросов

11.1.1. Порядок создания макросов в Excel

1. Прежде всего, надо *обдумать и записать*, *какие* действия должен выполнять создаваемый макрос.

2. Записать – *по п у н к т а м !* – порядок выполнения всех действий, которые будут зафиксированы в макросе.

3. Заранее определиться:
   - с *именем* РК, в которой он будет сохранён,
   - с его *собственным названием*,
   - с *сочетанием клавиш* "быстрого вызова" макроса (если он должен вызываться часто),
   - выбрать, *г д е* он будет сохраняться:
     - в "Личной книге макросов",
     - в "Новой книге",
     - в"Этой книге" (имеется в виду та РК, в которой вы будете создавать макрос).

4. Теперь можно начинать работу по созданию и записи макроса:
   - Открыть пустую РК, *активизировать* РЛ (например, Лист1),
   - *Переименовать* РЛ и сохранить РК под назначенным ранее именем,
   - Установить курсор в *произвольную* ячейку РЛ,
   - Выполнить: **Сервис – Макрос – Начать запись,**
   - Точно по пунктам выполнить все запланированные действия,
   - В заключение, когда всё сделано, выполнить

     Сервис – Макрос – Остановить запись,

5. Для проверки работы макроса:

Сервис – Макрос – Макросы – выбрать название нужного макроса **– Выполнить.**

6. Если нужно просмотреть, изменить или дополнить код (текст) программы макроса, то

Сервис – Макрос – Макросы – выбрать название нужного макроса **– Войти / Изменить.**

Произойдёт переход в редактор VBA, где код программы будет выведен в окно **Code.**

ПРИМЕЧАНИЕ 9: если в данной РК будут храниться *несколько* макросов, то в ячейке A1 весьма целесообразно разместить *примечание*, в котором для каждого макроса ввести название и "горячие" клавиши для его быстрого вызова. Тогда по прошествии времени не придётся гадать, есть ли макросы в этой книге и какие именно.

11.1.2. Задания на создание макросов в Excel

Задание 1: для начала попробуйте создать простейший макрос, который в РК "***Работа с ячейками***" только один раз перемещал бы курсор из произвольной ячейки РЛ Лист1 в заданную ячейку ***F7*** и окрашивал бы её в синий цвет (внутренний код синего цвета – 23).

Задание 2: находясь в произвольной ячейке РЛ Лист1 последовательно задать на нём заливку:

- ячейки E5 - в жёлтый цвет,
- ячейки F5 - в зелёный цвет,
- ячейки G5 - в красный цвет,
- ячейку H5 – оставить без заливки и прекратить работу макроса.

Выбрать для него название "ЦветаЯчеекE5_F5_G5отмена_H5" и задать "горячие" клавиши Ctrl+W.

Решение. После выполнения всех этих действий в процессе записи мак-

роса "ЦветаЯчеекE5_F5_G5отмена_H5" вы должны получить макрос, код которого приведён ниже:

```
Sub ЦветаЯчеекE5_F5_G5отмена_H5()
'
' ЦветаЯчеекE5_F5_G5отмена_H5 Макрос
' Макрос записан 02.05.2009 (MINI_CORP)
'
' Сочетание клавиш: Ctrl+w
'
   Range("E5").Select           ' выбрать ячейку E5
   With Selection.Interior      ' для выбранного назначить
                                двойное свойство
   .ColorIndex = 6          ' Interior.ColorIndex = 6 (жёлтый цвет)
      .Pattern = xlSolid    ' образец заливки = сплошной
   End With                 ' конец для выбранного
   Range("F5").Select
   With Selection.Interior
      .ColorIndex = 4       4 – зелёный цвет
      .Pattern = xlSolid
   End With                                      } пояснения аналогичны
   Range("G5").Select
   With Selection.Interior
      .ColorIndex = 3       3 – красный цвет
      .Pattern = xlSolid
   End With
   Range("H5").Select    '
   Selection.Interior.ColorIndex = xlNone   ' xlNone - без цвета
End Sub
```

Рис. 11.1 Текст (код) макроса "ЦветаЯчеек…"

Теперь этот макрос можно запустить тремя способами:

1) **Сервис** – **Макрос** – **Макросы** – выбрать имя макроса из списка – **Выполнить,**
2) из среды редактора Visual Basic**,** нажав клавишу **F5,**
3) из Excel (если редактор VBA закрыт) нажатием **Ctrl+W.**

Запустите макрос на исполнение и проанализируйте результат.

Задание 3. Усложним работу, добавив в задание 2 требование не только окрасить ячейки в нужные цвета, но и оформив их тонкими рамками.

Если вы правильно выполните заданное, то в новом макросе "Цвета_и_рамки_ячеек" вы увидите новые операторы и свойства:

```
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
```

Рис. 11.2. Фрагмент кода макроса "Цвета_и_рамки_ячеек"

Все они связаны с *обрамлением* ячеек *рамками*. К прежним пояснениям теперь добавим:

Selection.Borders – *выбор обрамления*,

xlDiagonalDown – *диагональ сверху вниз*, (**xlDiagonalUp** – *диагональ снизу вверх*),

LineStyle – *стиль* (тип) линии, **xlNone** – *нет*

Edge – *кромка*, **Left**, **Right**, **Top**, **Bottom** – *слева*, *справа*, *сверху*, *снизу*,

Weight- *толщина* (линии), **Thin** - *тонкий*

Остаётся заметить, что подобные конструкции сопровождают выбор каждой ячейки. Так что длина кода существенно увеличилась.

Задание 4. Попробуйте теперь выполнить пример из п. 8.2 главы 8 (стр. 158). Старайтесь не отступать от предлагаемых там рекомендаций.

11.2. Варианты реализации разветвляющихся алгоритмов

С этого момента надо работать только в среде редактора VBA. Текст программ вводится в окно **Code**, после чего запускается на выполнение, если по ходу ввода программы компилятор VBA не выдавал сообщений о синтаксических ошибках. Впрочем, ошибки могут быть обнаружены и позже - на стадии выполнения.

Задание 5. Составить программу, определяющую принадлежность человека к определённой социальной группе по его возрасту.

Пусть **Возраст –** целочисленная переменная. Программа должна предлагать ввести возраст и по нему выдавать сообщения "Дошкольник", "Школьник", "Студент", "Специалист" или "Пенсионер".

Решение. Предлагается такой код программы:

```
Sub СоциальнаяГруппа()
Dim Возраст As Integer
```

```
Возраст = InputBox("Укажите, пожалуйста, ваш возраст:")
If Возраст < 7 Then
   MsgBox "Ты ещё дошкольник."
ElseIf Возраст < 17 Then
   MsgBox "Ты уже школьник!"
ElseIf Возраст < 23 Then
   MsgBox "Вы - студент."
ElseIf Возраст < 55 Then
   MsgBox "О, специалист со стажем!"
Else
   MsgBox "Пенсионер, Вы заслужили отдых!"
End If
End Sub
```

Рис. 11.3. Фрагмент кода макроса "СоциальнаяГруппа"

Задание 6. Измените программу так, чтобы вводился не возраст, а год рождения. Возраст же должен каждый раз вычисляться от текущего года.

Задание 7. В VBA есть очень удобный оператор **Select Case (***выбор***).** Его общий вид (формат) таков:

Select Case <*значение*>
    Case <*условие1*>
        <*группа операторов1*>
    Case < *условие2*>
        < *группа операторов2*>
. . . . . .
    Case < *условиеN*>
        < *группа операторовN*>
End Select

Принцип его работы прост. Сначала вычисляется *значение*, затем оно последовательно сравнивается с *условиями*. Как только *значение* удовлетворит очередное *условие*, тут же выполняется соответствующая *группа операторов* и

программ выходит из оператора **Select Case** и выполняет уже те операторы, что идут вслед за **End Select**.

Применим этот оператор к решению задачи о выполнении плана предприятием.

Пусть плановая величина прибыли равна 100 миллионов. Как будет реагировать программа на разные значения прибыли?

Предлагается такой текст программы:

```
Sub РеакцияНаПрибыль()
   Dim Прибыль As Single
   Прибыль = InputBox ("Какова общая прибыль предприятия за месяц (млн.руб.)?")
   Select Case Прибыль
      Case Is = 100
      MsgBox "Молодцы, но нельзя ли побольше?"
      Case Is < 100
      MsgBox "Негоже! В чём дело?"
      Case Is > 100
      MsgBox "Молодцы, здорово! Будем премировать!"
   End Select
End Sub
```

Рис. 11.4. Фрагмент кода макроса " РеакцияНаПрибыль "

11.3. Варианты реализации циклических алгоритмов

Задание 8. Написать программу расчета и вывода на экран в виде сообщения последовательности нечётных чисел от 1 до 27.

Решение. Для решения этой задачи более всего подходит цикл с заданным числом повторов (см. рис. 6.8):

For Сч= <начальное значение> ***to*** <конечное значение> ***step*** <шаг>

Next Сч

Применим его:

```
Sub НечетныеЧисла()
   Dim НечЧисло As String   'переменная НечЧисло – строка символов
   For k = 1 To 27 Step 2
      НечЧисло = НечЧисло & k & " "
   Next k
   MsgBox "Вот эти нечётные числа:" & _
   Chr(10) & НечЧисло
End Sub
```

В итоге на экран будет выведено сообщение вида:

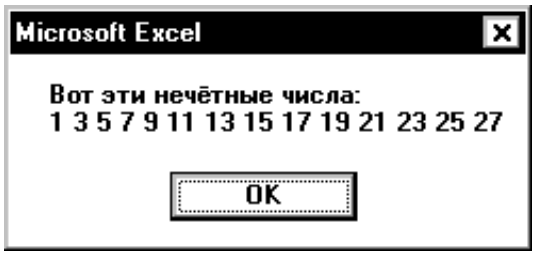


Рис. 11.5. Результат работы программы " НечетныеЧисла "

Задание 9. Составить программу перевода сантиметров в дюймы (1 дюйм = 2, 51 см) как циклическую с дробным шагом для счетчика.

Решение. Для этой задачи тоже подходит цикл For, но только уже с дробным шагом 2,51 для каждого нового значения сантиметра.

```
Sub Дюймы_в_см()
   Dim ЗначДюйм As Single
   Dim Дюйм_см As String
   Dim дюйм As Integer
   дюйм = 0
   For ЗначДюйм = 2.51 To 14 Step 2.51
      дюйм = дюйм + 1
      Дюйм_см = Дюйм_см & дюйм & " = " & ЗначДюйм & " см" _
      & Chr(13)
   Next ЗначДюйм
   MsgBox "Таблица перевода дюймов в сантиметры:" & _
        Chr(13) & ЗначДюйм
End Sub
```

Рис. 11.6. Текст (код) макроса пересчета сантиметров в дюймы

В итоге на экране – сообщение с таблицей перевода:

Рис. 11.7. Результат выполнения макроса (программы) "Дюймы"

11.4. Вариант реализации смешанного алгоритма

Задание 10. Ранее описанные алгоритмы (задания 5 - 7) страдают одним общим недостатком: они выполняются всего один раз. Чтобы снова с ними поработать, надо снова и снова запускать программы на выполнение.

Учитывая некоторый опыт, который вы приобрели, решая задачи с разветвляющимися и циклическими алгоритмами, попробуйте из программ заданий 5 – 7 составить программы смешанного типа – циклические с разветвлениями. В качестве примера возьмите фрагмент программы в конце п. 8.5 о закраске ячейки разными цветами, сравнив её с предыдущим вариантом той же программы (стр. 168-170).

Желаем успехов!

РАЗДЕЛ 3. Основы информационной безопасности

Глава 12. Введение в информационную безопасность

12.1. Понятие информационной безопасности

Развитие и широкое применение электронной вычислительной техники в промышленности, управлении, связи, научных исследованиях, образовании, сфере услуг, коммерческой, финансовой и других сферах человеческой деятельности являются в настоящее время приоритетным направлением научно-технического прогресса. Эффект, достигаемый за счет применения вычислительной техники, возрастает при увеличении масштабов обработки информации. Масштабы и сферы применения вычислительной техники стали таковы, что наряду с проблемами надежности и устойчивости ее функционирования возникает проблема обеспечения безопасности циркулирующей в ней информации.

Проблема защиты информации актуальна для любой организации и частного лица, владеющих, использующих или предающих какую-либо информацию. Однако особую актуальность проблемы защиты информации приобретают в системах электронной обработки данных. ***Системы электронной обработки данных*** – это системы любой структуры и функционального назначения, в которых информация обрабатывается с помощью средств электронной вычислительной техники.

В общем смысле ***информационная безопасность*** – это защищённость информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры.

Информационная безопасность не сводится исключительно к защите от несанкционированного доступа к информации, это принципиально более широкое понятие. Субъект информационных отношений может пострадать

(понести убытки и/или получить моральный ущерб) не только от несанкционированного доступа, но и, например, от поломки системы, вызвавшей перерыв в работе. Согласно определению, информационная безопасность зависит не только от способа хранения и обработки информации, но и от всей инфраструктуры, обеспечивающей информационные процессы. К инфраструктуре можно отнести системы водо-, тепло-, электроснабжения, кондиционирования, коммуникации, а также персонал, обслуживающий все эти системы, либо непосредственно занимающийся сбором, обработкой и использованием информации.

В проблеме информационной безопасности можно выделить следующие аспекты:

- Обеспечение целостности информации. Под целостностью информации понимается её физическая сохранность, защищённость от разрушения и искажения, а также её актуальность и непротиворечивость.

- Обеспечение доступности информации. Доступность информации подразумевает, что субъект информационных отношений (пользователь) имеет возможность за приемлемое время получить требуемую информационную услугу. Следует отметить, что системы электронной обработки данных создаются именно для предоставления определённых информационных услуг. Если предоставление таких услуг становится невозможным, то это наносит ущерб всем субъектам информационных отношений. Поэтому, не противопоставляя доступность остальным аспектам, её выделяют как важнейший элемент информационной безопасности.

- Обеспечение конфиденциальности информации. Конфиденциальная информация – это информация, на доступ к которой имеет право ограниченный круг лиц. Если же доступ к конфиденциальной информации получает лицо, не имеющее такого права, то такой доступ называется несанкционированным и рассматривается как нарушение защиты конфиденциальной информации. Лицо, получившее или пытающееся получить несанкционирован-

ный доступ к конфиденциальной информации, называется злоумышленником.

- Соблюдение авторских и иных прав. Этот аспект информационной безопасности стал исключительно актуальным в последнее время в связи с принятием ряда международных правовых актов по защите интеллектуальной собственности. Данный аспект касается в основном предотвращения нелегального использования программ. Кроме того, данный аспект касается использования информации, полученной из электронных источников. Эта проблема стала наиболее актуальной в связи с развитием сети Интернет. Сложилась ситуация, когда пользователь Интернет рассматривает всю размещенную там информацию как свою личную собственность, и пользуется ей без каких-либо ограничений, зачастую выдавая за собственный интеллектуальный продукт.

Защита информации – это комплекс мероприятий, направленных на обеспечение информационной безопасности.

12.2. Угрозы безопасности информации

Угроза безопасности – это действие или событие, которое может привести к разрушению, искажению или несанкционированному использованию информационных ресурсов, включая хранимую и обрабатываемую информацию, а также программные и аппаратные средства.

Угрозу отождествляют обычно либо с характером (видом, способом) дестабилизирующего воздействия на информацию, либо с последствиями (результатами) такого воздействия. Однако такого рода термины могут иметь много трактовок. Возможен и иной подход к определению угрозы безопасности информации, базирующийся на понятии «угроза». Угроза - это намерение нанести физический, материальный или иной вред общественным или личным интересам, возможная опасность.

Классификация возможностей реализации угроз (атак), представляет собой совокупность возможных вариантов действий источника угроз опреде-

ленными методами реализации с использованием уязвимостей, которые приводят к реализации целей атаки. Цель атаки может не совпадать с целью реализации угроз и может быть направлена на получение промежуточного результата, необходимого для достижения в дальнейшем реализации угрозы. В случае такого несовпадения атака рассматривается как этап подготовки к совершению действий, направленных на реализацию угрозы, т.е. как «подготовка к совершению» противоправного действия. Результатом атаки являются последствия, которые являются реализацией угрозы и/или способствуют такой реализации.

Сам подход к анализу и оценке состояния безопасности информации основывается на вычислении весовых коэффициентов опасности для источников угроз и уязвимостей, сравнения этих коэффициентов с заранее заданным критерием и последовательном сокращении (исключении) полного перечня возможных источников угроз и уязвимостей до минимально актуального для конкретного объекта.

Исходными данными для проведения оценки и анализа служат результаты анкетирования субъектов отношений, направленные на уяснение направленности их деятельности, предполагаемых приоритетов целей безопасности, задач, решаемых автоматизированной системой и условий расположения и эксплуатации объекта. Благодаря такому подходу возможно:

- установить приоритеты целей безопасности для субъекта отношений;

определить перечень актуальных источников угроз;

- определить перечень актуальных уязвимостей;
- оценить взаимосвязь угроз, источников угроз и уязвимостей;
- определить перечень возможных атак на объект;
- описать возможные последствия реализации угроз.

Угрозы безопасности информации делятся на *естественные* и *искусственные*.

Естественные угрозы (угрозы, вызванные воздействием на информационную систему объективных физических процессов, стихийных природных явлений, не зависящих от человека) делятся на:

- природные (стихийные бедствия, магнитные бури, радиоактивное излучение, осадки);
- технические. Связаны с надежностью технических средств обработки информации и систем обеспечения.

Искусственные делят на:

- случайные (непреднамеренные) - совершенные по незнанию и без злого умысла, из любопытности или халатности;
- умышленные (преднамеренные) - результат активного воздействия человека на объекты и процессы с целью умышленной дезорганизации функционирования информационной технологии, вывода ее из строя, проникновения в систему и несанкционированного доступа к информации.

Источником случайных угроз могут быть:

- отказы и сбои аппаратных средств в случае их некачественного исполнения и физического старения;
- помехи в каналах и на линиях связи от воздействия внешней среды;
- форсмажорные ситуации (пожар, выход из строя электропитания и т.д.);
- схемные системотехнические ошибки и просчеты разработчиков и производителей технических средств;
- алгоритмические и программные ошибки;
- неумышленные действия пользователей, приводящие к частичному или полному отказу технологии или разрушению аппаратных, программных, информационных ресурсов (неумышленная порча оборудования , удаление, искажение файлов с важной информацией или программ , в том числе системных и т. д.);
- неправомерное включение оборудования или изменение режимов работы устройств и программ;

- неумышленная порча носителей информации;
- запуск технологических программ, способных при некомпетентном использовании вызывать потерю работоспособности системы (зависания или зацикливания) или необратимые изменения в информационной технологии (форматирование или реструктуризация носителей информации, удаление данных и т. д.);
- нелегальное внедрение и использование неучтенных программ (игровых, обучающих, технологических и др., не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей) с последующим необоснованным расходованием ресурсов (загрузка процессора, захват оперативной памяти и памяти на внешних носителях информации и т. д.);
- заражение компьютерными вирусами;
- неосторожные действия, приводящие к разглашению конфиденциальной информации или делающие ее общедоступной;
- разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т. д.);
- проектирование архитектуры технологии, разработка прикладных программ с возможностями, представляющими угрозу для работоспособности информационной технологии и безопасности информации;
- вход в систему в обход средств защиты (загрузка посторонней операционной системы со сменных носителей информации и т. д.); некомпетентное использование, настройка или неправомерное отключение средств защиты персоналом службы безопасности экономического объекта;
- пересылка данных по ошибочному адресу абонента или устройства;
- ввод ошибочных данных;
- неумышленное повреждение каналов связи и т. д.

Умышленные угрозы, в свою очередь, делятся на пассивные и активные.

Пассивные угрозы – это угрозы, направленные на несанкционированное использование информационных ресурсов, не оказывая при этом влияния на функционирование информационной системы. К ним относится, например, попытка получения информации, циркулирующей в каналах связи, посредством их прослушивания.

Активные угрозы – это угрозы, имеющие целью нарушение нормального функционирования информационной системы посредством целенаправленного воздействия на аппаратные, программные и информационные ресурсы. К ним относятся, например, разрушение или радио-электронное подавление каналов связи, вывод из строя рабочих станций сети, искажение сведений в базах данных либо в системной информации в информационных технологиях и т. д.

Среди умышленных угроз выделяют также следующие виды:

1. Внутренние. Возникают внутри управляемой организации. Они чаще всего сопровождаются социальной напряженностью и тяжелым моральным климатом на экономическом объекте, который провоцирует специалистов выполнять какие-либо правонарушения по отношению к информационным ресурсам

Внутренними источниками являются:

- противозаконная деятельность политических, экономических и криминальных структур и отдельных лиц в области формирования, распространения и использования информации, направленная в т.ч. и на нанесение экономического ущерба государству;
- неправомерные действия различных структур и ведомств, приводящие к нарушению законных прав работников в информационной сфере;
- нарушения установленных регламентов сбора, обработки и передачи информации;
- преднамеренные действия и непреднамеренные ошибки персонала автоматизированных систем, приводящие к утечке, уничтожению, искаже-

нию, подделке, блокированию, задержке, несанкционированному копированию информации;

- отказы технических средств и сбои программного обеспечения в информационных и телекоммуникационных системах;
- каналы побочных электромагнитных излучений и наводок технических средств обработки информации.

2. Внешние. Направлены на информационную технологию извне. Такие угрозы могут возникать из-за злонамеренных действий конкурентов, экономических условий и других причин (например, стихийных бедствий). К внешним источникам относятся:

- недружественная политика иностранных государств в области информационного мониторинга, распространения информации и новых информационных технологий;
- деятельность иностранных разведывательных и специальных, направленная против интересов Российской Федерации;
- деятельность иностранных экономических структур, направленная против интересов Российской Федерации;
- преступные действия международных групп, формирований и отдельных лиц; стихийные бедствия и катастрофы.

Практика функционирования информационных технологий показывает, что в настоящее время существует большое количество угроз безопасности информации. К основным угрозам безопасности информации и нормального функционирования информационной технологии относятся большое количество различных угроз, которые могут иметь локальный характер или интегрированный, т. е. совмещаться, комбинироваться или совпадать по своим действиям с другими видами угроз безопасности .

В целом можно выделить следующие умышленные угрозы безопасности данных в информационных технологиях (включая активные, пассивные, внутренние и внешние):

- взлом системы;

- компрометация информации;
- нарушение информационного обслуживания;
- незаконное использование привилегий;
- несанкционированное использование информационных ресурсов;
- несанкционированный доступ;
- отказ от информации;
- утечка конфиденциальной информации.

Взлом системы – это умышленное проникновение в информационную технологию, когда взломщик не имеет санкционированных параметров для входа. Способы взлома могут быть различными, и при некоторых из них происходит совпадение с ранее описанными угрозами. Например, использование пароля пользователя информационной технологии, который может быть вскрыт, например, путем перебора возможных паролей.

Основную нагрузку защиты системы от взлома несет программа входа. Алгоритм ввода имени и пароля, их шифрование, правила хранения и смены паролей не должны содержать ошибок. Противостоять взлому системы поможет, например, ограничение попыток неправильного ввода пароля (т. е. исключить достаточно большой перебор) с последующей блокировкой персонального компьютера (рабочей станции) и уведомлением администратора в случае нарушения. Кроме того, администратор безопасности должен постоянно контролировать активных пользователей системы: их имена, характер работы, время входа и выхода и т. д. Такие действия помогут своевременно установить факт взлома и предпринять необходимые действия.

Необходимо отметить, что особую опасность в настоящее время представляет проблема компьютерных вирусов и вредоносных программ, т. к. эффективной защиты против них разработать не удалось.

Этот вид угроз может быть непосредственно связан с понятием атака, который в настоящее время широко используется нарушителями против информационных технологий различных экономических объектов.

Атаки **–** это злонамеренные действия взломщика, попытки реализации им любого вида угрозы. Например, атакой является применение любой из вредоносных программ.

Среди атак на информационные системы и технологии часто выделяют «маскарад» и «взлом системы», которые могут быть результатом реализации разнообразных угроз (или комплекса угроз).

Маскарад - это выполнение каких-либо действий одного пользователя от имени другого. При этом такие действия другому пользователю могут быть разрешены. Нарушение заключается в присвоении прав и привилегий. Цель маскарада скрыть какие-либо действия за именем другого пользователя или присвоение прав и привилегий другого пользователя для доступа к его данным или для использования его привилегий. Примерами маскарада могут служить вход в систему под именем и паролем другого пользователя, создание и использование программ, которые в определенном месте могут изменить данные пользователя, передача данных в сети от имени другого пользователя и др.

Для предотвращения маскарада нужно использовать надежные методы идентификации контроль входа в систему, блокировку попыток взлома системы.

Условием, способствующим реализации многих видов угроз информации является наличие *люков*.

Люк **–** это скрытая, недокументированная точка входа в программный модуль, входящий в состав программного обеспечения информационной системы и информационной технологии. Люк вставляется в программу обычно на этапе отладки для облегчения работы, Этот модуль можно вызывать из разных частей программы, что позволяет обрабатывать их независимо. Забытый в программе люк позволяет вызвать программу нестандартным образом, что существенно влияет на состояние системы защиты. Люк может остаться в программе ненамеренно (забыли убрать) или намеренно (с целью тайного доступа к программе после ее установки). Люки представляют собой боль-

шую угрозу безопасности информации, но их сложно обнаружить, если не знать о них заранее. Защита от люков заключается в анализе текстов исходных программ при их приемке на наличие люков.

Компрометация информации – это один из видов информационных инфекций. Реализуется, как правило, посредством несанкционированных изменений в базе данных, в результате чего ее потребитель вынужден либо отказаться от нее, либо предпринимать дополнительные усилия для выявления изменений и восстановления истинных сведений. При использовании скомпрометированной информации потребитель подвергается опасности принятия неверных решений.

Нарушение информационного обслуживания – это весьма существенная и распространенная угроза, источником которой является сама автоматизированная информационная технология. Задержка с предоставлением информационных ресурсов абоненту может привести к тяжелым для него последствиям. Отсутствие у пользователя своевременных данных, необходимых для принятия решения, может вызвать его нерациональные действия.

Незаконное использование привилегий – еще один вид умышленных угроз безопасности информации. Любая защищенная технология содержит средства, используемые в чрезвычайных ситуациях, или средства, которые способны функционировать с нарушением существующей политики безопасности. Например, на случай внезапной проверки пользователь должен иметь возможность доступа ко всем наборам системы. Обычно эти средства используются администраторами, операторами, системными программистами и другими пользователями, выполняющими специальные функции.

Большинство систем защиты в таких случаях используют наборы привилегий, т. е. для выполнения определенной функции требуется определенная привилегия. Обычно пользователи имеют минимальный набор привилегий, администраторы – максимальный.

Наборы привилегий охраняются системой защиты. Несанкционированный (незаконный) захват привилегий возможен при наличии ошибок в си-

стеме защиты, но чаще всего происходит в процессе управления системой защиты, в частности, при небрежном пользовании привилегиями .

Строгое соблюдение правил управления системой защиты, а также принципа минимума привилегий позволяет избежать таких нарушений.

Большинство из перечисленных технических путей утечки информации поддаются надежной блокировке при правильно разработанной и реализуемой на практике системе обеспечения безопасности.

Несанкционированное использование информационных ресурсов – это, с одной стороны, последствие утечки информации и средство ее компрометации. С другой стороны, оно имеет самостоятельное значение, так как может нанести большой ущерб управляемой системе (вплоть до полного выхода информационной технологии из строя) или ее абонентам.

Несанкционированный доступ – это нарушение установленных правил разграничения доступа, последовавшее в результате случайных или преднамеренных действий пользователей или других субъектов системы разграничений.

Несанкционированный доступ к информации выражается в противоправном преднамеренном овладении конфиденциальной информацией лицом, не имеющим права доступа к охраняемым сведениям.

Наиболее распространенными путями несанкционированного доступа к информации являются:

- перехват электронных излучений;
- принудительное электромагнитное облучение (подсветка) линий связи с целью получения паразитной модуляции несущей;
- применение подслушивающих устройств (закладок);
- дистанционное фотографирование;
- перехват акустических излучений и восстановление текста принтера;
- чтение остаточной информации в памяти системы после выполнения санкционированных запросов;
- копирование носителей информации с преодолением мер защиты;

• маскировка под зарегистрированного пользователя («маскарад»);

• использование недостатков языков программирования и операционных систем;

• маскировка под запросы системы;

• использование программных ловушек;

• незаконное подключение к аппаратуре и линиям связи специально разработанных аппаратных средств, обеспечивающих доступ к информации ;

• злоумышленный вывод из строя механизмов защиты;

• расшифровка специальными программами зашифрованной информации; информационные инфекции.

Перечисленные пути несанкционированного доступа требуют достаточно больших технических знаний и соответствующих аппаратных или программных разработок со стороны взломщика. Например, используются технические каналы утечки – это физические пути от источника конфиденциальной информации к злоумышленнику, посредством которых возможно получение охраняемых сведений. Причиной возникновения каналов утечки являются конструктивные и технологические несовершенства схемных решений либо эксплуатационный износ элементов. Все это позволяет взломщикам создавать действующие на определенных физических принципах преобразователи, образующие присущий этим принципам канал передачи информации – канал несанкционированного доступа.

Отказ от информации – это непризнание получателем или отправителем этой информации фактов ее получения или отправки. Это позволяет одной из сторон расторгать заключенные финансовые соглашения «техническим» путем, формально не отказываясь от них, нанося тем самым второй стороне значительный ущерб.

Утечка конфиденциальной информации – это бесконтрольный выход конфиденциальной информации за пределы информационной технологии или круга лиц, которым она была доверена по службе или стала известна в процессе работы.

Конфиденциальная информация – это информация, исключительное право на пользование которой принадлежит определенным лицам или группе лиц.

Раскрытие конфиденциальной информации может быть следствием разглашения конфиденциальной информации; утечки информации по различным, главным образом техническим, каналам (по визуально-оптическим, акустическим, электромагнитным и др.); несанкционированного доступа к конфиденциальной информации различными способами.

Можно выделить следующие пути утечки информации при обработке и передаче данных в автоматизированных информационных системах и технологиях:

- хищение носителей информации, незаконное считывание и копирование информации;
- использование программных ловушек;
- внедрение компьютерных вирусов;
- неправильная идентификация, отсутствие контроля ошибок, ошибки в программах;
- маскировка под пользователя, подбор пароля;
- ошибочная коммутация;
- неисправности аппаратуры;
- перехват информации в технических каналах ее утечки, внедрение электронных устройств перехвата информации в технические средства передачи информации и помещения;
- перехват, дешифрование и внедрение ложной информации в сетях передачи данных и линиях связи;
- радиоэлектронное подавление линий связи и систем управления;
- персонал: ошибки в работе оператора, искажение программной защиты, организация люков, ошибочная коммутация, бесконтрольное считывание, использование недостаточной защиты;

• примитивные пути несанкционированного доступа: хищение документальных отходов; инициативное сотрудничество; склонение к сотрудничеству со стороны взломщика; выпытывание; подслушивание; наблюдение и другие пути.

Любые способы утечки конфиденциальной информации могут привести к значительному материальному и моральному ущербу как для организации, где функционирует информационная технология, так и для ее пользователей.

Кто является нарушителем безопасности информации?

Нарушитель – это субъект, совершивший противоправные действия по отношению к информации. Нарушителями в информационных системах и технологиях экономического объекта являются, прежде всего, пользователи и работники, имеющие доступ к информации.

Для определения потенциального нарушителя следует определить предполагаемую категорию лиц, к которым может принадлежать нарушитель, мотивы действий нарушителей, квалификацию нарушителей и их техническую оснащенность.

Нарушители безопасности информации могут быть внутренними или внешними.

Внутренними нарушителями могут быть лица из следующих категорий персонала:

• специалисты (пользователи) информационной технологии;

• сотрудники-программисты, сопровождающие системное, общее и прикладное программное обеспечение;

• персонал, обслуживающий технические средства (инженерные работники информационной технологии);

• другие сотрудники, имеющие санкционированный доступ к ресурсам информационной технологии (в том числе подсобные рабочие, уборщицы, электрики, сантехники и т. д.);

• сотрудники службы безопасности информационной технологии; руководители различного уровня управления.

Доступ к ресурсам информационной технологии других посторонних лиц, не принадлежащих к указанным категориям, может быть ограничен организационно-режимными мерами. Однако следует также учитывать следующие категории посторонних лиц.

К внешним нарушителями относятся:

• посетители (лица, приглашенные по какому-либо поводу);

• клиенты (представители сторонних организаций или граждане, с которыми работают специалисты организации);

• представители организаций, взаимодействующих по вопросам обеспечения жизнедеятельности экономического объекта (энерго-, водо-, теплоснабжения и т. д.), представители конкурирующих организаций, иностранных спец- служб, лиц, действующих по их заданию и т. д.;

• лица, случайно или умышленно нарушившие пропускной режим (даже без цели нарушения безопасности);

• любые лица за пределами контролируемой территории.

Можно выделить следующие мотивы действий нарушителей – корыстный интерес, безответственность, самоутверждение.

Всех нарушителей можно классифицировать по четырем классификационным признакам.

1. По уровню знаний об информационной технологии различают нарушителей: знающих функциональные особенности информационной технологии, умеющих пользоваться штатными средствами; обладающих высоким уровнем знаний и опытом работы с техническими средствами информационной технологии и их обслуживания; обладающих высоким уровнем знаний в области программирования и вычислительной техники, проектирования и эксплуатации информационных технологий; знающих структуру, функции и механизм действия средств защиты , их сильные и слабые стороны.

2. По уровню возможностей различают нарушителей: применяющих агентурные методы получения сведений; применяющих пассивные средства (технические средства перехвата без модификации компонентов информационной технологии); использующих только штатные средства и недостатки систем защиты для ее преодоления (несанкционированные действия с использованием разрешенных средств), а также компактные машинные носители информации, которые могут быть скрытно пронесены через посты охраны; применяющих методы и средства активного воздействия (модификация и подключение дополнительных устройств и пр.).

3. По времени действия различают нарушителей: действующих в процессе функционирования информационной технологии (во время работы компонентов системы); действующих в нерабочее время, во время плановых перерывов в работе информационной технологии, перерывов для обслуживания и ремонта и т. д., как в процессе функционирования информационной технологии, так и в нерабочее время.

4. По месту действия различают нарушителей: имеющих доступ в зону управления средствами обеспечения безопасности информационной системы; имеющих доступ в зону данных; действующих с автоматизированных рабочих мест (рабочих станций); действующих внутри помещений, но не имеющие доступа к техническим средствам; действующих с контролируемой территории без доступа в здания и сооружения; не имеющих доступа на контролируемую территорию организации.

Для построения надежной системы защиты информации требуются значительные материальные и финансовые затраты. Поэтому необходимо не просто разрабатывать частные механизмы защиты информации, а использовать целый комплекс мер, т.е. использовать специальные средства, методы и мероприятия с целью предотвращения потери данных. Для того, чтобы принятые меры оказались эффективными, необходимо определить:

- что такое угроза безопасности информации;

• выявить каналы утечки данных и пути несанкционированного доступа

• к защищаемой информации;

• определить потенциального нарушителя;

• построить эффективную систему защиты данных в информационных системах и технологиях.

12.3. Объекты и элементы защиты информации в компьютерных системах обработки данных

При создании средств защиты информации важно определить природу угроз, форму и пути их возможного проявления и осуществления, перечень объектов и элементов, которые с одной стороны, могут быть подвергнуты угрозам с целью нарушения защищенности информации, а с другой – достаточно четко локализованы для организации эффективной защиты информации.

Согласно Государственному стандарту РФ ГОСТ Р 52069.0-2003 "Защита информации. Система стандартов. Основные положения", объект защиты - это информация или носитель информации или информационный процесс, в отношении которых необходимо обеспечивать защиту в соответствии с поставленной целью защиты информации.

В соответствии с Государственным стандартом объекты защиты информации включают в себя следующие подгруппы:

а) продукцию;

б) технологии;

в) процессы (работы);

г) объекты капитального строительства;

д) услуги;

е) документы.

В специальной литературе выделяют объекты и элементы защиты информации.

Под объектом защиты понимается такой структурный компонент системы, в котором находится или может находиться информация, подлежащая защите. Под элементом защиты понимается совокупность данных, которая может содержать сведения, подлежащие защите.

В качестве объектов защиты информации можно выделить следующие:

- терминалы пользователей (персональные компьютеры, рабочие станции сети);
- терминал администратора сети;
- узел связи;
- средства отображения информации;
- средства документирования информации;
- машинный зал (компьютерный или дисплейный) и хранилище носителей информации;
- внешние каналы связи и сетевое оборудование;
- накопители и носители информации.
- В соответствии с приведенным определением в качестве элементов защиты выступают блоки (порции, массивы, потоки и др.) информации в объектах зашиты, в частности:
- данные и программы в основной памяти компьютера;
- данные и программы на внешнем машинном носителе;
- данные, отображаемые на экране монитора;
- данные, выводимые на принтер при автономном и сетевом использовании ПК;
- пакеты данных, передаваемые по каналам связи;
- данные, размножаемые (тиражируемые) с помощью копировально-множительного оборудования;
- отходы обработки информации в виде бумажных и магнитных носителей;

- журналы назначения паролей и приоритетов зарегистрированным пользователям;
- служебные инструкции по работе с комплексами задач;
- архивы данных и программное обеспечение и др.

Вопросы для самоконтроля

1. Что понимается под угрозой безопасности информации?
2. Как классифицируются угрозы безопасности информации?
3. Что представляют собой естественные и искусственные угрозы безопасности информации?
4. Назовите источники случайных угроз.
5. Что понимается под пассивными и активными угрозами безопасности информации?
6. Назовите основные виды умышленных угроз.
7. Перечислите наиболее распространенные пути несанкционированного доступа к информации.
8. Что представляет собой нарушитель безопасности информации? Охарактеризуйте основные категории нарушителей.
9. Что понимается под объектом защиты информации? Что понимается под элементом защиты информации?
10. Перечислите элементы защиты информации.
11. Что относится к объектам защиты информации?
12. Что понимается под информационной безопасностью?
13. Что понимается под защитой информации?

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	Бесконтрольный выход конфиденциальной информации за пределы информационной технологии или круга лиц, которым она была доверена по службе или стала известна в процессе работы, – это:	• раскрытие конфиденциальной информации • несанкционированный доступ • компрометация информации
2.	Действие или событие, которое может при-	• вредоносная программа

	вести к разрушению, искажению или несанкционированному использованию информационных ресурсов, включая хранимую и обрабатываемую информацию, а также программные и аппаратные средства, – это:	• угроза безопасности информации • троянский конь
3.	Информация, преимущественное право на использование которой принадлежит одному лицу или группе лиц, – это:	• секретная информация • конфиденциальная информация • информация для служебного доступа
4.	Непризнание получателем или отправителем информации фактов ее получения или отправки – это	• отказ от информации • нарушение информационного обслуживания • незаконное использование привилегий
5.	Результатом реализации угроз информационной безопасности может быть…	• внедрение дезинформации в периферийные устройства • уничтожение устройств ввода-вывода информации • несанкционированный доступ к информации • изменение конфигурации периферийных устройств
6.	Выполнение каких-либо действий одного пользователя от имени другого называется …	• маскарад • люк • компрометация информации • нарушение информационного обслуживания
7.	Скрытая, недокументированная точка входа в программный модуль, входящий в состав программного обеспечения информационной системы называется …	• пользовательский вход • люк • код доступа • контрольный вход
8.	Что относится к умышленным угрозам безопасности информации?	• помехи в каналах и на линиях связи от воздействия внешней среды • алгоритмические и программные ошибки • изменение режимов работы устройств и программ • компрометация информации
9.	К объектам защиты информации относятся…	• накопители и носители информации • данные, отображаемые на экране монитора • данные и программы в основной памяти компьютера • пакеты данных, передаваемые по каналам связи
10.	К элементам защиты информации относятся…	• данные и программы в основной памяти компьютера

		• терминал администратора сети; • узел связи; • средства отображения информации
11.	Что не относится к целостности информации?	• физическая сохранность • доступность • защищенность от разрушения • актуальность
12.	Какие угрозы относятся к естественным?	• случайные • умышленные • технические • программные ошибки
13.	К пассивным угрозам безопасности информации относятся…	• прослушивание каналов связи • искажение сведений в базах данных • разрушение каналов связи • воздействие на программные ресурсы
14.	Физическая сохранность, защищенность от разрушения и искажения, актуальность и непротиворечивость информации – это	• целостность информации • конфиденциальность информации • доступность информации • аутентичность информации
15.	К естественным угрозам безопасности информации не относятся:	• технические угрозы • стихийные бедствия • магнитные бури • алгоритмические и программные ошибки
16.	Алгоритмические и программные ошибки относится к:	• активным угрозам • пассивным угрозам • случайным угрозам • естественным угрозам
17.	К умышленным угрозам относится:	• нарушение информационного обслуживания • ошибки разработчиков технических средств • ввод ошибочных данных • утрата атрибутов разграничения доступа
18.	Умышленное проникновение в информационную технологию без санкционированных параметров для входа называется:	• нарушением информационного обслуживания • взломом системы • технической угрозой безопасности информации • нарушением безопасности информации
19.	Помехи в каналах связи от воздействия	• активным угрозам

	внешней среды относятся к:	• пассивным угрозам • случайным угрозам • естественным угрозам
20.	Классификационным признаком классификации нарушителей безопасности информации не является:	• уровень знаний информационной технологии • время действия • уровень возможностей • степень воздействия на информационную технологию

Глава 13. Методы и средства защиты информации

13.1. Механизмы, методы и средства защиты информации

Выделяют следующие механизмы защиты информации:

- формирование и опознание подписи;
- контроль и разграничение доступа;
- система регистрации и учета информации;
- обеспечение целостности данных;
- обеспечение аутентификации;
- подстановка трафика;
- управление маршрутизацией;
- арбитраж или освидетельствование.

1. Формирование и опознание подписи. Ее механизм основывается на алгоритмах асимметричного шифрования и включает две процедуры: формирование подписи отправителем и ее опознание (верификацию) получателем. Первая процедура обеспечивает шифрование блока данных или его дополнение криптографической контрольной суммой, причем в обоих случаях используется секретный ключ отправителя. Вторая процедура основывается на использовании общедоступного ключа, знание которого достаточно для опознавания отправителя.

2. Контроль и разграничение доступа. Осуществляет проверку полномочий объектов (программ и пользователей) на доступ к ресурсам сети. В основе контроля доступа к данным лежит система разграничения доступа специалистов информационной технологии к защищаемой информации.

3. Система регистрации и учета информации. Отвечает за ведение регистрационного журнала, позволяет проследить за тем, что происходило в прошлом, и соответственно перекрыть каналы утечки информации. В регистрационном журнале фиксируются все осуществленные или неосуществленные попытки доступа к данным или программам. Содержание регистрационного журнала может анализироваться как периодически, так и непре-

рывно. В регистрационном журнале ведется список всех контролируемых запросов, осуществляемых специалистами, а также учет всех защищаемых носителей информации с помощью их маркировки, с регистрацией их выдачи и приема.

Система регистрации и учета является одним из эффективных методов увеличения безопасности в информационных системах и технологиях.

4. Обеспечение целостности данных. Применяется как к отдельному блоку, так и к потоку данных. Целостность блока является необходимым, но не достаточным условием целостности потока. Целостность блока обеспечивается выполнением взаимосвязанных процедур шифрования и дешифрования отправителем и получателем. Отправитель дополняет передаваемый блок криптографической суммой, а получатель сравнивает ее с криптографическим значением, соответствующим принятому блоку. Несовпадение свидетельствует об искажении информации в блоке. Однако описанный механизм не позволяет вскрыть подмену блока в целом. Поэтому необходим контроль целостности потока данных, который реализуется посредством шифрования с использованием ключей, изменяемых в зависимости от предшествующих блоков.

5. Обеспечение аутентификации. Это механизм установления подлинности, т.е. проверка, является ли объект (субъект) действительно тем, за кого себя выдает. Механизмы аутентификации подразделяются на одностороннюю и взаимную аутентификацию. При использовании односторонней аутентификации один из взаимодействующих объектов проверяет подлинность другого. Во втором случае – проверка является взаимной.

6. Подстановка трафика (подстановка текста). Используются для реализации службы засекречивания потока данных. Они основываются на генерации объектами информационной системы фиктивных блоков, их шифровании и организации передачи по каналам связи. Тем самым нейтрализуется возможность получения информации об информационной технологии и об-

служиваемых ее пользователей посредством наблюдения за внешними характеристиками потоков информации, циркулирующих по каналам связи.

7. Управление маршрутизацией. Обеспечивают выбор маршрутов движения информации по коммуникационной сети таким образом, чтобы исключить передачу секретных сведений по скомпрометированным (небезопасным), физически ненадежным каналам.

8. Арбитраж. Обеспечивает подтверждение характеристик данных, передаваемых между объектами информационной системы, третьей стороной (арбитром). Для этого вся информация, отправляемая или получаемая объектами, проходит и через арбитра, что позволяет ему впоследствии подтверждать упомянутые характеристики.

В основе механизмов защиты лежат методы защиты информации. К основным методам защиты относятся:

- маскировка;
- побуждение;
- препятствие;
- принуждение;
- регламентация;
- управление доступом.

Маскировка – это метод защиты информации путем ее криптографического закрытия. Этот метод сейчас широко применяется как при обработке, так и при хранении информации, в том числе и на переносных носителях. При передаче информации по каналам связи большой протяженности данный метод является единственно надежным.

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

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

Принуждение – это метод защиты, когда специалисты и персонал информационной технологии вынуждены соблюдать правила обработки, передачи и использования защищаемой информации под угрозой материальной, административной или уголовной ответственности.

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

Управление доступом – это метод защиты информации с помощью использования всех ресурсов информационной технологии. Управление доступом включает следующие функции защиты:

- идентификация специалистов, персонала и ресурсов информационной технологии (присвоение каждому объекту персонального идентификатора);
- опознание (установление подлинности) объекта или субъекта по предъявленному им идентификатору;
- проверка полномочий (соответствие дня недели, времени суток, запрашиваемых ресурсов и процедур установленному регламенту);
- разрешение и создание условий работы в пределах установленного регламента;
- регистрация (протоколирование) обращений к защищаемым ресурсам;
- реагирование (сигнализация, отключение, задержка работ, отказ в запросе) при попытке несанкционированных действий.

Методы обеспечения безопасности реализуются на практике за счет применения средств защиты, которые делятся на формальные и неформальные.

Неформальные средства защиты – это средства защиты, которые определяются целенаправленной деятельностью человека, либо регламентируют эту деятельность

К основным неформальным средствам защиты относятся организационные, законодательные, морально-этические средства.

1. Организационные средства. Представляют собой организационно-технические и организационно-правовые мероприятия, осуществляемые в процессе создания и эксплуатации вычислительной техники, аппаратуры телекоммуникаций для обеспечения защиты информации в информационных системах. Организационные мероприятия охватывают все структурные элементы аппаратуры на всех этапах их жизненного цикла (строительство и оборудование помещений экономического объекта, проектирование информационной системы, монтаж и наладка оборудования, испытания, эксплуатация и т. д.). К ним можно отнести, например, охрану серверов, тщательный подбор персонала, исключение случаев ведения особо важных работ только одним человеком, наличие плана восстановления работоспособности сервера после выхода его из строя, универсальность средств защиты от всех пользователей (включая высшее руководство).

2. Законодательные средства. Определяются законодательными актами страны, в которых регламентируются правила пользования, обработки и передачи информации ограниченного доступа и устанавливаются меры ответственности за нарушения этих правил.

3. Морально-этические средства. Реализуются в виде всевозможных норм, которые сложились традиционно или складываются по мере распространения вычислительной техники и средств связи. Эти нормы большей частью не являются обязательными как законодательные меры, однако несоблюдение их ведет к утечке информации и нарушению секретности.

Формальные средства защиты – это средства, выполняющие защитные функции строго по заранее предусмотренной процедуре без непосредственного участия человека.

К основным формальным средствам защиты, которые используются для защиты информации в информационных системах, относятся программные и технические средства.

1. Программные средства. Представляют собой программное обеспечение, специально предназначенное для выполнения функций защиты информации.

2. Технические средства. Реализуются в виде электрических, электромеханических и электронных устройств. Все технические средства делятся аппаратные и физические. Аппаратные средства представляют собой устройства , встраиваемые непосредственно в вычислительную технику, или устройства, которые сопрягаются с подобной аппаратурой по стандартному интерфейсу. Физические средства представляют собой автономные устройства и системы, создающие физические препятствия для злоумышленников (замки, решетки, охранная сигнализация и т.д.). К ним можно отнести, например, резервирование особо важных компьютерных подсистем, организацию вычислительных сетей с возможностью перераспределения ресурсов в случае нарушения работоспособности отдельных звеньев, установку оборудования обнаружения и тушения пожара, оборудования обнаружения воды, принятие конструкционных мер защиты от хищений, саботажа, диверсий, взрывов, установку резервных систем электропитания, оснащение помещений замками, установку сигнализации и многое другое.

13.2. Средства опознания и разграничения доступа к информации

Для того чтобы обеспечить безопасность информационных ресурсов, устранить возможность несанкционированного доступа, усилить контроль за санкционированным доступом к конфиденциальной либо к подлежащей засекречиванию информации, внедряются различные системы опознавания, установления подлинности субъекта (объекта) и разграничения доступа. В

основу построения подобных систем закладывается принцип допуска и выполнения только таких обращений к информации, в которых присутствуют соответствующие признаки разрешенных полномочий.

Одним из механизмов обеспечения безопасности информации в информационных системах является механизм контроля доступа, осуществляющий проверку полномочий объектов информационной системы (программ и пользователей) на доступ к ресурсам сети. В основе контроля доступа к данным лежит система разграничения доступа специалистов к защищаемой информации.

Реализация систем разграничения доступа представляет собой программу, которая должна закрыть все входы в операционную систему, как стандартные, так и всевозможные нестандартные. Запуск системы разграничения доступа осуществляется на стадии загрузки операционной системы, после чего вход в систему и доступ к ресурсам возможен только через систему разграничения доступа. Кроме этого, система разграничения доступа содержит ряд автономных утилит, которые позволяют настраивать систему и управлять процессом разграничения доступа.

Система разграничения доступа контролирует действия субъектов доступа по отношению к объектам доступа и, на основании правил разграничения доступа, может разрешать и запрещать требуемые действия.

Для успешного функционирования системы разграничения доступа в информационных технологиях решаются следующие задачи:

- невозможность обхода системы разграничения доступа действиями, находящимися в рамках выбранной модели;
- гарантированная идентификация специалиста информационной технологии, осуществляющего доступ к данным (аутентификация пользователя).

Основными понятиями в этой системе являются идентификация и аутентификация.

Идентификация – это присвоение какому-либо объекту или субъекту уникального имени или образа.

Аутентификация – это установление подлинности, т.е. проверка, является ли объект (субъект) действительно тем, за кого себя выдает. Механизмы аутентификации подразделяются на одностороннюю и взаимную аутентификацию. При использовании односторонней аутентификации один из взаимодействующих объектов проверяет подлинность другого. Во втором случае – проверка является взаимной.

К объектам идентификации и аутентификации относятся:

- люди (пользователи, операторы и др.);
- технические средства (мониторы, рабочие станции, абонентские пункты);
- документы (распечатки, рукописные и др.);
- внешние магнитные носители информации;
- внутрикомпьютерные информационные ресурсы и др.

Конечная цель процедур идентификации и аутентификации объекта (субъекта) – допуск его к информации в случае положительного результата проверки либо отказ в допуске в случае отрицательного исхода проверки.

Можно выделить следующие методы аутентификации:

- метод вопрос-ответ. Пользователь при входе отвечает на m ориентированных и n стандартных вопросов. Стандартные вопросы не касаются пользователя и вводятся в систему заранее;
- метод секретного алгоритма. Система выдает случайное число. Пользователь, зная секретный алгоритм, сообщает системе результаты вычислений по алгоритму;
- метод пароля и его модификация. Это один из наиболее распространенных методов аутентификации – присвоение лицу или другому имени пароля и хранение его значения в вычислительной системе.

Пароль – это совокупность символов, определяющих объект (субъект).

Пароль вводится пользователем в начале работы с компьютерной системой, а иногда в конце сеанса (в особо ответственных случаях пароль нормального выхода может отличаться от входного). Для усиления подтвержде-

ния правомочности пользователя можно предусмотреть ввод пароля через определенные промежутки времени.

При выборе пароля возникают вопросы о его размере, стойкости к несанкционированному подбору, способам применения. Чем больше длина пароля, тем сложнее его угадать. Выбор длины пароля во многом зависит от возможностей технических средств (их элементной базы и быстродействия). Например: для распознавания четырехзначного десятичного числа, компьютеру нужно перебрать числа от 0000 до 9999, т.е. 9999 комбинаций. Четырехзначный пароль, в котором применяются цифровые символы и 26 букв латинского алфавита (т. е. всего 36 возможных знаков), требует более трудоемкого процесса распознавания, потому что он допускает 364 уникальных комбинаций. Увеличивая длину пароля и число используемых символов, можно увеличить число возможных комбинаций, повысив время, которое потребуется на взлом пароля. Для усиления подтверждения правомочности пользователя можно предусмотреть ввод пароля через определенные промежутки времени. Надежность пароля значительно повышается, если разделить пароль на две части: одна состоит из комбинации чисел, легко запоминаемой пользователем, другая содержит количество знаков, определяемое требованиями к защите и возможностями технической реализации системы. Вторая часть хранится на специальном физическом носителе, устанавливаемом пользователем в считывающее устройство.

Учитывая важность пароля, рекомендуется соблюдать следующие меры предосторожности:

- периодически менять пароль;
- применять сочетание символов верхнего и нижнего регистров клавиатуры;
- использовать комбинации из двух простых слов, соединенных специальными символами, например, "+";
- не хранить пароли в вычислительной системе в незашифрованном виде;

- не печатать и не отображать пароли в явном виде на мониторе;
- не использовать в качестве пароля личную информацию (свое имя или имена родственников, дату рождения, номер телефона, номер машины и др.).

Для идентификации пользователей применяются сложные системы, обеспечивающие установление подлинности пользователя на основе анализа его индивидуальных параметров: отпечатков пальцев, геометрии руки, радужной оболочки глаз, особенностям речи, тембра голоса, ритму работы на клавиатуре и др.

Широкое распространение получили физические методы идентификации с использованием носителей кодов паролей: пропусков; пластиковых карточек с кодом владельца и подписью; пластиковых карточек с магнитной полосой, содержащей информацию, которая считывается специальным считывающим устройством (карточки для банкоматов и пр.); пластиковых карточек со встроенной микросхемой и др.

Одно из современных направлений обеспечения безопасности информации – идентификация и установление подлинности документов на основе электронной цифровой подписи.

13.3. Криптографические методы защиты информации

13.3.1. Основные понятия криптографии

История криптографии насчитывает несколько тысяч лет. Потребность скрывать написанное появилась у человека почти сразу, как только он научился писать. Предполагается, что криптография была известна в древнем Египте и Вавилоне. До нашего времени дошли указания на то, что искусство секретного письма использовалось в древней Греции. Широко известным историческим примером криптосистемы является так называемый шифр Цезаря, который представляет из себя простую замену каждой буквы открытого текста третьей следующей за ней буквой алфавита (с циклическим перено-

сом, когда это необходимо). Например, "A" заменялась на "D", "B" на "E", "Z" на "C".

Несмотря на значительные успехи математики за века, прошедшие со времён Цезаря, тайнопись вплоть до середины 20 века не сделала существенных шагов вперёд. В ней бытовал ненаучный подход. В 20 веке широко применялись "книжные" шифры, в которых в качестве ключа использовалось какое-либо массовое печатное издание.

Появление первых электронно-вычислительных машин кардинально изменило ситуацию:

- объем циркулирующей в обществе информации стал возрастать по экспоненциальному закону - он примерно удваивается каждые пять лет;
- доступ к определенным данным позволяет контролировать значительные материальные и финансовые ценности; информация приобрела стоимость, которую во многих случаях даже можно подсчитать;
- характер обрабатываемых данных стал многообразным и не сводится к исключительно текстовым данным;
- характер информационных взаимодействий усложнился: наряду с классической задачей защиты передаваемых текстовых сообщений от несанкционированного прочтения и искажения возникли новые задачи сферы защиты информации, например, подпись под электронным документом;
- субъектами информационных процессов теперь являются не только люди, но и созданные ими автоматические системы, действующие по заложенной в них программе;
- вычислительные способности современных компьютеров подняли на совершенно новый уровень как возможности по реализации шифров, ранее немыслимых из-за своей высокой сложности, так и возможности аналитиков по их взлому.

Перечисленные изменения привели к тому, что криптография сделала в своем развитии огромный скачок. Учёные вплотную занялись проблемами криптографии и криптоанализа.

Криптография (иногда употребляют термин криптология) – это область знаний, изучающая тайнопись (криптография) и методы ее раскрытия – криптоанализ - Криптография считается разделом математики.

В настоящее время термин "криптография" далеко ушел от своего первоначального значения - "тайнопись", "тайное письмо". Сегодня эта дисциплина объединяет методы защиты информационных взаимодействий совершенно различного характера, опирающиеся на преобразование данных по секретным алгоритмам, включая алгоритмы, использующие секретные параметры.

Защита информации методами криптографии заключается в приведении ее к неявному виду путем преобразования составных частей (букв, цифр, слогов, слов) с помощью специальных алгоритмов либо аппаратных средств и кодов ключей.

Ключ — это секретная информация, используемая криптографическим алгоритмом при шифровании/расшифровке сообщений, постановке и проверке цифровой подписи, вычислении кодов аутентичности (MAC). При использовании одного и того же алгоритма результат шифрования зависит от ключа. Для современных алгоритмов сильной криптографии утрата ключа приводит к практической невозможности расшифровать информацию.

Цель криптографической системы заключается в том, чтобы зашифровать осмысленный исходный текст (также называемый открытым текстом) и получить в результате совершенно бессмысленный на взгляд шифрованный текст (шифротекст, криптограмма). Получатель, которому он предназначен, должен быть способен расшифровать (говорят также "дешифровать") этот шифротекст, восстановив, таким образом, соответствующий ему открытый текст. При этом противник (называемый также криптоаналитиком) должен быть неспособен раскрыть исходный текст.

Существует важное отличие между расшифрованием (дешифрованием) и раскрытием шифротекста.

Раскрытием криптосистемы называется результат работы криптоаналитика, приводящий к возможности эффективного раскрытия любого, зашифрованного с помощью данной криптосистемы, открытого текста. Степень неспособности криптосистемы к раскрытию называется ее стойкостью.

Криптография обладает той особенностью, что на "вскрытие" шифра зачастую нужно затратить на несколько порядков больше средств, чем на его создание. Однако при этом не исключён случай, когда профессионалы долго, но безуспешно бились над шифром, а некий новичок применил нестандартный подход - и шифр дался ему легко.

Дополнительным обеспечением надёжности шифра служит секретность алгоритма. Но на самом деле, если алгоритм известен разработчикам, он уже не может считаться секретным, если только пользователь и разработчик - не одно лицо. К тому же, если вследствие некомпетентности или ошибок разработчика алгоритм оказался нестойким, его секретность не позволит проверить его независимым экспертам. Нестойкость алгоритма обнаружится только, когда он будет уже взломан, а то и вообще не обнаружится.

Поэтому криптограф должен руководствоваться правилом, впервые сформулированным голландцем Керкхоффом: стойкость шифра должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь механизм шифрования, кроме значения секретного ключа априори считается известным противнику.

Криптография часто используется и для других целей:

- проверка подлинности. Получатель сообщения может проверить его источник. Злоумышленник не сможет замаскироваться под кого-либо;
- целостность. Получатель сообщения может проверить, не было ли сообщение изменено в процессе доставки;
- неотрицание авторства. Отправитель не сможет ложно отрицать отправку сообщения.

Существует, также, метод защиты информации (строго говоря, не относящийся к криптографии), когда скрывается не алгоритм шифровки, а сам

факт того, что сообщение содержит зашифрованную (скрытую в нём) информацию. Такой приём называют маскировкой информации.

Криптоанализ - это наука получения открытого текста не имея ключа. Успешно проведенный криптоанализ может раскрыть открытый текст или ключ. Раскрытие ключа не криптологическим способом называют компрометацией. Попытка криптоанализа называется вскрытием. Обычно различают следующие виды криптоанализа:

1. Вскрытие с использованием только шифротекста. У криптоаналитика есть шифротексты нескольких сообщений, зашифрованных одним и тем же алгоритмом шифрования. Задача криптоаналитика состоит в раскрытии открытого текста как можно большего числа сообщений или получения ключа, использованного для шифрования других сообщений, зашифрованных тем же ключом.

2. Вскрытие с использованием открытого текста. У криптоаналитика есть доступ не только к шифротекстам нескольких сообщений, но и к открытому тексту этих сообщений. Его задача состоит в получении ключа, использованного для шифрования сообщения, для дешифрования других сообщений, зашифрованных тем же ключом.

3. Вскрытие с использованием выбранного открытого текста. У криптоаналитика не только есть доступ к шифротекстам и открытым текстам нескольких сообщений, но и возможность выбирать открытый текст для шифрования.

4. Адаптивное вскрытие с использованием открытого текста. Это частный случай вскрытия с использованием выбранного открытого текста. Криптоаналитик не только может выбирать шифруемый текст, но также может строить свой последующий выбор на базе полученных результатов.

5. Вскрытие с использованием выбранного шифротекста. Криптоаналитик может выбрать различные шифротексты для шифрования и имеет доступ к дешифрованным открытым текстам.

13.3.2. Криптографические ключи и методы защитных преобразований

Криптографические ключи различаются согласно алгоритмам, в которых они используются.

Секретные (Симметричные) ключи — это ключи, используемые в симметричных алгоритмах (шифрование, выработка кодов аутентичности). Главное свойство симметричных ключей: для выполнения как прямого, так и обратного криптографического преобразования (шифрование/расшифровывание, вычисление MAC/проверка MAC) необходимо использовать один и тот же ключ (либо же ключ для обратного преобразования легко вычисляется из ключа для прямого преобразования, и наоборот). С одной стороны, это обеспечивает более высокую конфиденциальность сообщений, с другой стороны, создаёт проблемы распространения ключей в системах с большим количеством пользователей.

Асимметричные ключи — это ключи, используемые в асимметричных алгоритмах (шифрование, ЭЦП). Они разделены на два ключа:

- закрытый ключ — ключ, известный только своему владельцу. Только сохранение пользователем в тайне своего закрытого ключа гарантирует невозможность подделки злоумышленником документа и цифровой подписи от имени заверяющего;

- открытый ключ — ключ, который может быть опубликован и используется для проверки подлинности подписанного документа, а также для предупреждения мошенничества со стороны заверяющего лица в виде отказа его от подписи документа. Открытый ключ подписи вычисляется, как значение некоторой функции от закрытого ключа, но знание открытого ключа не дает возможности определить закрытый ключ.

Криптографический алгоритм, также называемый шифром, представляет собой математическую функцию, используемую для шифрования и дешифрования. Обычно это две связанные функции: одна для шифрования, другая - для дешифрования.

Основные требования, предъявляемые к методам защитного преобразования:

- применяемый метод должен быть достаточно устойчив к попыткам раскрыть исходный текст имея только зашифрованный текст;
- объем ключа не должен затруднять его запоминание и пересылку;
- алгоритм преобразования информации и ключ, используемый для шифрования и дешифрования не должны быть очень сложными. Затраты на защитные преобразования должны быть приемлемы при заданном уровне сохранности информации;
- ошибки в шифровании не должны вызывать потерю информации. Из-за появления ошибок передачи шифрованного сообщения по каналам связи не должна исключаться возможность надежной расшифровки текста на приемном конце;
- длина зашифрованного текста не должна превышать длину исходного текста;
- необходимые временные и стоимостные ресурсы на шифрование и дешифрование информации определяются требуемой степенью защиты информации.

Множество современных методов защитных преобразований можно классифицировать на 4 большие группы:

1. Перестановки.
2. Замены.
3. Аддитивные.
4. Комбинированные.

Методы перестановки и подстановки обычно характеризуются короткой длиной ключа, а надежность их защиты определяется сложностью алгоритмов преобразования.

Для аддитивных методов характерны простые алгоритмы преобразования, а их надежность основана на увеличении длины ключа.

Все перечисленные методы относятся к симметричному шифрованию - для шифрования и дешифрования используется один и тот же ключ.

При асимметричном шифровании для шифрования используется один ключ - открытый, а для дешифрования другой - закрытый.

1. Метод перестановки. Заключается в том, что входной поток исходного текста делится на блоки, в каждом из которых выполняется перестановка символов. Простейшим примером перестановки является запись исходного текста по строкам некоторой матрицы и чтение его по столбцам этой матрицы. Последовательность заполнения строк и чтение столбцов может быть любой и задается ключом. Для методов перестановки характерны простота алгоритма, возможность программной реализации и низкий уровень защиты. Недостаток этого метода - легкое раскрытие, если удается направить в систему для шифрования несколько специально подобранных сообщений.

2. Метод замены (подстановки). Заключается в том, что символы исходного текста, записанные в одном алфавите, заменяются символами другого алфавита в соответствии с принятым ключом преобразования. Одним из простейших методов является прямая замена исходных символов их эквивалентом из вектора замен. Для очередного символа исходного текста отыскивается его местоположение в исходном алфавите. Эквивалент из вектора замены выбирается как отстоящий на полученное смещение от начала алфавита. При дешифровании поиск производится в векторе замен, а эквивалент выбирается из алфавита. Полученный таким образом текст имеет низкий уровень защиты.

Более стойкой в отношении раскрытия является схема шифрования, основанная на использовании таблицы Вижинера. Таблица представляет собой квадратную матрицу с числом элементов k, где k - количество символов в алфавите.

В первой строке матрицы записываются буквы в порядке очередности их в алфавите, во второй - та же последовательность букв, но со сдвигом влево на одну позицию, в третьей - со сдвигом на 2 позиции и т.д. Освободив-

шиеся места справа заполняются вытесненными влево буквами, записанными в естественной последовательности.

Для шифрования текста устанавливается ключ, представляющий собой некоторое слово или набор букв. Далее, из полной матрицы выбирается подматрица шифрования, включающая, например, первую строку и строку матрицы, начальные буквы которой являются последовательной буквой ключа.

Пример:

Ключ - МОРЕ

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

……

МНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЁЖЗИЙКЛ

ОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЁЖЗИЙКЛМН

РСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЁЖЗИЙКЛМНОП

ЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГД

Исходный текст:

ЗАЩИТА ИНФОРМАЦИИ

МОРЕМО РЕМОРЕМОРЕ

Зашифрованный текст:

УОИОЭО ШТЯЫЯСМГШО

Процесс шифрования включает следующую последовательность действий:

1. Под каждой буквой шифруемого текста записываются буквы ключа, повторяющие ключ требуемой число раз.

2. Шифруемый текст по подматрице заменяется буквами, расположенными на пересечении линий, соединяющих буквы текста первой строки подматрицы и буквы ключа, находящиеся под ней.

Расшифровка текста выполняется в следующей последовательности:

1. Над буквами шифрованного текста последовательно записываются буквы ключа.

2. В строке подматрицы таблицы Вижинера для каждой буквы ключа отыскивается буква, соответствующая знаку шифрованного текста. Находящаяся над ней буква первой строки и будет знаком расшифрованного текста.

3. Полученный текст группируется в слова по смыслу.

Один из недостатков шифрования по таблице Вижинера - ненадежность шифрования при небольшой длине ключа и сложность формирования длинных ключей. С целью повышения надежности шифрования текста применяется усовершенствованный вариант таблицы Вижинера, который заключается в следующем:

1. Во всех строках, кроме первой, буквы алфавита располагаются в произвольном порядке.

2. Выбирается 10, не считая первой, строк, пронумерованных натуральными числами от 0 до 9.

3. В качестве ключа используются величины, выраженные бесконечным рядом чисел (например, число Пи).

Шифрование и расшифрование осуществляется в той же последовательности, что и в случае простой таблицы Вижинера.

3. Аддитивные методы. В качестве ключа в этих методах используется некоторая последовательность букв того же алфавита и такой же длины, что и в исходном тексте.

Шифрование выполняется путем сложения символов исходного текста и ключа по модулю, равному числу букв в алфавите.

Примером такого же метода является гаммирование, т.е. наложение на исходный текст некоторой последовательности кодов, называемой гаммой. Процесс наложения осуществляется следующим образом:

1. Символы исходного текста и гамма представляются в двоичном коде и располагаются один под другим.

2. Каждая пара двоичных знаков заменяется одним двоичным знаком шифрованного текста в соответствии с принятым алгоритмом.

3. Полученная последовательность двоичных знаков шифрованного текста заменяется символами алфавита в соответствии с выбранным кодом.

Если ключ шифрования выбирается случайным образом, например, формируется с помощью датчика псевдослучайных чисел, то раскрыть информацию, не зная ключа практически невозможно.

13.3.3. Криптографические системы

Различают криптографические системы с открытым ключом и симметричные криптосистемы.

Криптографическая система с открытым ключом (или Асимметричное шифрование, Асимметричный шифр) — это система шифрования информации, при которой ключ, которым зашифровывается сообщение и само зашифрованное сообщение передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу. Для генерации открытого ключа и для прочтения зашифрованного сообщения получатель использует секретный ключ. Криптографические системы с открытым ключом в настоящее время широко применяются в различных сетевых протоколах. Последовательность действий при использовании данной системы следующая:

1. Получатель генерирует ключ. Ключ разбивается на открытую и закрытую часть. При этом открытый ключ не должен передаваться по открытому каналу. Либо его подлинность должна быть гарантирована некоторым сертифицирующим органом
2. Отправитель с помощью открытого ключа шифрует сообщение.
3. Получатель с помощью закрытого ключа дешифрует сообщение.

Преимущество асимметричных шифров состоит в отсутствии необходимости передачи секретного ключа. Сторона, желающая принимать зашифрованные тексты, в соответствии с используемым алгоритмом вырабатывает пару «открытый ключ — закрытый ключ». Значения ключей связаны между собой, однако вычисление одного значения из другого должно быть невозможным с практической точки зрения. Открытый ключ публикуется в открытых справочниках и используется для шифрования информации контраген-

том. Закрытый ключ держится в секрете и используется для расшифровывания сообщения, переданного владельцу пары ключей.

Асимметричные криптосистемы не лишены недостатков. Они требуют существенно больших вычислительных ресурсов.

Симметричные криптосистемы (также симметричное шифрование, симметричные шифры) – это способ шифрования, в котором для (за)шифрование и расшифрование применяется один и тот же криптографический ключ. До изобретения схемы асимметричного шифрования единственным существовавшим способом являлось симметричное шифрование. Ключ алгоритма должен сохраняться в секрете обеими сторонами. Ключ алгоритма выбирается сторонами до начала обмена сообщениями.

Важным свойством симметричных шифров является невозможность их использования для подтверждения авторства, так как ключ известен каждой стороне.

К достоинствам этой системы можно отнести скорость (по сравнению с асимметричной криптосистемой примерно на 3 порядка выше), простота реализации (за счёт более простых операций), изученность.

К недостаткам симметричной криптосистемы относятся сложность управления ключами в большой сети, сложность обмена ключами. Для применения этой системы необходимо решить проблему надёжной передачи ключей каждому абоненту, так как нужен секретный канал для передачи каждого ключа обеим сторонам.

Для компенсации недостатков симметричного шифрования в настоящее время широко применяется комбинированная (гибридная) криптографическая схема, где с помощью асимметричного шифрования передаётся сеансовый ключ, используемый сторонами для обмена данными с помощью симметричного шифрования.

Криптографическое преобразование – один из наиболее эффективных методов, резко повышающий безопасность передачи данных в компьютерных сетях, данных, хранящихся в удаленных устройствах памяти, информа-

ции при обмене между удаленными объектами. Криптография, в отличие от мер физической защиты, обладает тем уникальным свойством, что при правильном выборе метода затраты на обеспечение защиты информации много меньше затрат на преодоление этой защиты.

13.4. Электронная цифровая подпись

Электронная цифровая подпись (ЭЦП) – это аналог собственноручной подписи физического лица, представленный как последовательность символов, полученная в результате криптографического преобразования электронных данных с использованием закрытого ключа ЭЦП, которая позволяет пользователю открытого ключа установить целостность и неизменность этой информации, а также установить владельца закрытого ключа ЭЦП.

Электронная цифровая подпись является реквизитом электронного документа и предназначена для удостоверения источника данных и защиты данного электронного документа от подделки.

Защита электронного документа в системе ЭЦП обеспечивает:

- подтверждение того, что документ исходит от конкретного пользователя системы (подтверждение авторства документа);
- проверку подлинности и целостности документа;
- предотвращение несанкционированного доступа к документу в процессе информационного обмена;
- идентификацию пользователя системы, подписавшего электронный документ.

Возможны следующие угрозы цифровой подписи:

- злоумышленник может попытаться подделать подпись для выбранного им документа;
- злоумышленник может попытаться подобрать документ к данной подписи, чтобы подпись к нему подходила;

- злоумышленник может попытаться подделать подпись для какого-нибудь документа;
- злоумышленник, укравший закрытый ключ, может подписать любой документ от имени владельца ключа;
- злоумышленник может обманом заставить владельца подписать какой-либо документ, например, используя протокол слепой подписи;
- злоумышленник может подменить открытый ключ владельца (см. управление ключами) на свой собственный, выдавая себя за него.

К основным характеристикам ЭЦП относятся следующие.

1. Для формирования ЭЦП используется ассиметричная ключевая система. Это значит, что каждый абонент системы имеет открытый ключ и закрытый ключ. Закрытый ключ сохраняется абонентом в тайне - именно это не позволяет злоумышленнику выдать свой документ за документ абонента.

2. Длина ключа составляет 1024 бит. Сертификат ФСБ гарантирует невозможность подделки электронной цифровой подписи.

3. Специальные алгоритмы позволяют определить изменялся ли документ (например, файл с декларацией) после его подписания ЭЦП или не изменялся.

4. Для проверки ЭЦП на электронном документе используется открытый ключ абонента, поэтому этот открытый ключ есть, например, в налоговой инспекции абонента, а также у специализированного оператора связи.

5. Практически все действия по подписанию документов ЭЦП выполняются автоматически. Абоненту достаточно вставить съемный носитель информации с личным закрытым ключом непосредственно перед отправкой сообщений.

Алгоритмы ЭЦП делятся на два больших класса:

- обычные цифровые подписи;
- цифровые подписи с восстановлением документа.

Обычные цифровые подписи пристыковывают к подписываемому документу. К этому классу относятся, например, алгоритмы, основанные на эллиптических кривых.

Цифровые подписи с восстановлением документа содержат в себе подписываемый документ: в процессе проверки подписи автоматически вычисляется и тело документа. К этому классу относится один из самых популярных алгоритмов - RSA.

Следует различать электронную цифровую подпись и код аутентичности сообщения, несмотря на схожесть решаемых задач (обеспечение целостности документа и неотказуемости авторства). Коды аутентичности вычисляются по симметричным схемам.

Алгоритмы ЭЦП относятся к классу асимметричных алгоритмов (асимметричное шифрование, асимметричный шифр). В данной криптографической системе с открытым ключом открытый ключ передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу, и используется для проверки ЭЦП и для шифрования сообщения. Для генерации ЭЦП и для расшифрования сообщения используется секретный ключ.

Схема электронной подписи обычно включает в себя:

- алгоритм генерации ключевых пар пользователя;
- функцию вычисления подписи;
- функцию проверки подписи.

Функция вычисления подписи на основе документа и секретного ключа пользователя вычисляет собственно подпись.

Функция проверки подписи проверяет, соответствует ли данная подпись данному документу и открытому ключу пользователя. Открытый ключ пользователя доступен всем, так что любой может проверить подпись под данным документом.

Поскольку подписываемые документы — переменной (и достаточно большой) длины, в схемах ЭЦП зачастую подпись ставится не на сам документ, а на его хэш. Для вычисления хэша используются криптографические

хэш-функции, что гарантирует выявление изменений документа при проверке подписи. Хэш-функции не являются частью алгоритма ЭЦП, поэтому в схеме может быть использована любая надёжная хэш-функция.

Управление ключами заключается в необходимости обеспечить доступ любого пользователя к подлинному открытому ключу любого другого пользователя, защитить эти ключи от подмены злоумышленником, а также организовать отзыв ключа в случае его компрометации.

Задача защиты ключей от подмены решается с помощью сертификатов. Сертификат позволяет удостоверить заключённые в нём данные о владельце и его открытый ключ подписью какого-либо доверенного лица. В централизованных системах сертификатов используются центры сертификации, поддерживаемые доверенными организациями.

Управлением ключами занимаются центры распространения сертификатов. Обратившись к такому центру пользователь может получить сертификат какого-либо пользователя, а также проверить, не отозван ли ещё тот или иной открытый ключ.

В России юридически значимый сертификат электронной подписи выдаёт удостоверяющий центр на основании государственной лицензии. Правовые условия использования электронной цифровой подписи в электронных документах регламентирует Федеральный Закон от 10.01.2002 № 1-ФЗ «Об электронной цифровой подписи» (ред. от 08.112007).

В Росси с 2005-го года активно стала развиваться инфраструктура электронного документооборота между налоговыми органами и налогоплательщиками. При помощи электронной подписи можно отправить налоговую декларацию. Общие принципы организации информационного обмена при представлении налогоплательщиками налоговой декларации в электронном виде по телекоммуникационным каналам связи определяет Приказ Министерства по налогам и сборам Российской Федерации от 2 апреля 2002 г. N БГ-3-32/169 «Порядок представления налоговой декларации в электронном виде по телекоммуникационным каналам связи».

ЭЦП используют многие силовые структуры, Федеральная налоговая служба и другие государственные учреждения для придания электронным документам юридической силы.

В настоящее время в Республике Татарстан в рамках проекта Электронное правительство Республики Татарстан внедряется система электронного документооборота в органах исполнительной власти РТ с использованием электронной цифровой подписи. Ключи для ЭЦП выданы руководителям и служащим всех республиканских министерств.

В Республике Татарстан применяется «усовершенствованная подпись КРИПТО-ПРО». В ней преодолен ряд проблем, присущих для "классической" ЭЦП, а именно:- отсутствие доказательства момента подписи, трудность доказывания статуса сертификата открытого ключа подписи на момент подписи (или действителен, или аннулирован, или приостановлен). Новый формат подписи обеспечивает:

- доказательство момента подписи документа и действительности сертификата ключа подписи на этот момент;
- отсутствие необходимости сетевых обращений при проверке подписи;
- архивное хранение электронных документов;
- простоту встраивания и отсутствие необходимости контроля встраивания.

В формате усовершенствованной подписи вся необходимая информация для проверки подлинности ЭЦП находится в реквизитах документа. Для сохранения юридической значимости электронных документов при архивном хранении остаётся только обеспечить их целостность организационно-техническими мерами. В этом случае подлинность ЭЦП может быть подтверждена через сколь угодно долгое время, в том числе и после истечения срока действия сертификата ключа подписи.

Вопросы для самоконтроля

1. Охарактеризуйте основные механизмы защиты информации.

2. Какие методы защиты информации вы знаете?

3. Дайте характеристику основных неформальных средств защиты информации.

4. Дайте характеристику основных формальных средств защиты информации.

5. Охарактеризуйте управление доступом как способ защиты информации. Каковы его роль и значение?

6. Что понимается под идентификацией и аутентификацией в системах обеспечения безопасности информации?

7. Что относится к объектам идентификации и аутентификации ?

8. Что такое пароль? Какие правила следует соблюдать при выборе пароля?

9. Назовите основные методы аутентификации.

10. Понятие криптографии и криптоанализа.

11. Назначение криптографических методов защиты информации.

12. Какие требования предъявляются к методам защитного преобразования?

13. Понятие ключа и виды ключей в криптографии.

14. Назовите и охарактеризуйте методы защитных преобразований.

15. Как выполняется шифрование текста с использованием схемы шифрования, основанной на таблице Вижинера?

16. Назовите основные виды криптоанализа.

17. Охарактеризуйте криптографическую систему с открытым ключом.

18. Каковы особенности симметричных криптосистем?

19. Понятие и назначение электронной цифровой подписи (ЭЦП).

20. Перечислите основные характеристики ЭЦП.

21. Охарактеризуйте особенности защиты электронного документа с помощью ЭЦП.

22. Назовите возможные угрозы для цифровой подписи.

23. Что включает в себя схема ЭЦП?

24. К какому классу алгоритмов относятся алгоритмы ЭЦП?

25. Чем вызвана необходимость управления ключами в технологии ЭЦП?

26. Охарактеризуйте особенности и возможности нового формата усовершенствованной ЭЦП.

27. Назовите законодательные акты, регулирующие применение ЭЦП.

28. Приведите примеры использования ЭЦП на федеральном уровне и в Республике Татарстан.

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	Какие средства защиты информации реализуются в виде всевозможных норм, которые сложились традиционно или складываются по мере распространения вычислительной техники и средств связи?	• организационные • морально-этические • законодательные
2.	Какой механизм безопасности информации обеспечивает подтверждение характеристик данных, передаваемых между объектами информационной технологии, третьей стороной?	• арбитража • аутентификации • управления маршрутизацией
3.	Метод защиты информации путем ее криптографического закрытия – это	• препятствие • регламентация • маскировка
4.	Методы шифрования с открытым ключом для шифрования документов используют …	• два разных закрытых ключа • два разных ключа: открытый и закрытый • один и тот же закрытый ключ • два разных открытых ключа
5.	Одинаковые ключи для шифрования и дешифрования имеет _____ криптология.	• асимметричная • двоичная • симметричная • хеширующая
6.	Основное отличие симметричной криптографии от асимметричной заключается в том, что…	• симметричные криптоалгоритмы используют один и тот же ключ при шифровании и расшифровании, а асимметричные – разные ключи • симметричные криптоалгоритмы работают быстрее

		• симметричные криптоалгоритмы более стойкие • симметричные криптоалгоритмы используют ключи меньшей длины
7.	Открытые ключи используются…	• в симметричных методах с несколькими циклами шифрования • в асимметричной криптологии • в методах блочных перестановок • при симметричном шифровании
8.	Электронно-цифровая подпись (ЭЦП) документа позволяет получателю …	• только удостовериться в том, что документ не изменен во время передачи • либо удостовериться в корректности отправителя документа, либо удостовериться в том, что документ не изменен во время передачи • только удостовериться в истинности отправителя документа, но не проверить подлинность документа • удостовериться в корректности отправителя документа и удостовериться в том, что документ не изменен во время передачи
9.	Электронно-цифровая подпись (ЭЦП), как правило, реализуется на базе…	• какого-либо асимметричного криптоалгоритма • какого-либо симметричного криптоалгоритма • только на базе алгоритмов подстановки • только на базе алгоритма DES и ему подобных
10.	Идентификация – это…	• установление подлинности объекта (субъекта) • присвоение какому-либо объекту или субъекту уникального имени или образа • шифрование
11.	Секретная информация, используемая криптографическим алгоритмом при шифровании/расшифровке сообщений, постановке и проверке цифровой подписи – это…	• люк • точка входа • пароль • ключ
12.	К механизмам защиты информации не относится:	• электронная цифровая подпись

		• переустановка программного обеспечения • управление маршрутизацией • контроль и разграничение доступа
13.	Метод физического преграждения пути злоумышленнику к защищаемой информации это:	• маскировка • управление доступом • препятствие • арбитраж
14.	К формальным средствам защиты информации относятся:	• программные средства • законодательные средства • технические средства • организационные средства
15.	К методам аутентификации не относится:	• метод пароля • метод секретного алгоритма • метод вопрос-ответ • программный метод
16.	Какие ключи используются в асимметричных алгоритмах?	• закрытый и открытый • закрытый и секретный • открытый и секретный • симметричный
17.	К неформальным средствам защиты информации относятся:	• программные средства • законодательные средства • морально-этические средства • организационные средства
18.	К методам защиты информации не относятся:	• маскировка • препятствие • управление маршрутизацией • подстановка трафика
19.	К механизмам защиты информации относится:	• электронная цифровая подпись • переустановка программного обеспечения • управление маршрутизацией • подстановка трафика
20.	Какой механизм защиты информации основывается на генерации объектами информационной системы фиктивных блоков, их шифровании и организации передачи по каналам связи?	• электронная цифровая подпись • целостность данных • подстановка трафика • маршрутизация

Глава 14. Компьютерные вирусы и спам

14.1. Понятие вредоносных программ

Первые сообщение о программах, которые при наступлении определенных условий начинают производить вредные действия, появились в середине XX века, когда американские ученые Дж. фон Нейман и Норберт Винер, занимаясь проблемами алгоритмического обеспечения и программного управления ЭВМ, открыли возможность саморазмножения искусственных алгоритмических конструкций, т.е. программного кода.

В настоящее время выделяют целый ряд разновидностей вредоносных программ, которые являются существенной угрозой безопасности информации в информационных системах. К ним относятся:

- бактерии;
- захватчики паролей;
- компьютерные вирусы;
- логические бомбы;
- троянские кони;
- черви.

Бактерии – это программы, которые делают копии самих себя и становятся паразитами, перегружая память и микропроцессор персонального компьютера или рабочей станции сети.

Захватчики паролей – это программы, специально предназначенные для воровства паролей. При попытке обращения пользователя к рабочей станции на экран выводится информация, необходимая для окончания сеанса работы. Пытаясь организовать вход, пользователь вводит имя и пароль, которые пересылаются владельцу программы-захватчика, после чего выводится сообщение об ошибке, а ввод и управление возвращаются к операционной системе. Пользователь, думающий, что допустил ошибку при наборе пароля, повторяет вход и получает доступ к системе. Однако его имя и пароль уже известны владельцу программы-захватчика. Перехват пароля возможен и

другими способами. Для предотвращения этой угрозы перед входом в систему необходимо убедиться, что вы вводите имя и пароль именно системной программе ввода, а не какой-нибудь другой. Кроме того, необходимо неукоснительно придерживаться правил использования паролей и работы с системой. Большинство нарушений происходит из-за элементарной небрежности. Соблюдение специально разработанных правил использования паролей – необходимое условие надежной защиты.

Компьютерные вирусы – это специально написанные, обычно небольшие по размерам программы, способные самопроизвольно присоединяться к другим программам (т. е. заражать их), создавать свои копии (не обязательно полностью совпадающие с оригиналом) и внедрять их в файлы, системные области персонального компьютера и в другие объединенные с ним компьютеры с целью нарушения нормальной работы программ, порчи файлов и каталогов, создания различных помех при работе на компьютере.

Логические бомбы – это программы, используемые для искажения или уничтожения информации. Реже с их помощью совершаются кража или мошенничество. Логическую бомбу иногда вставляют во время разработки программы, а срабатывает она при выполнении некоторого условия (время, дата, кодовое слово).

Манипуляциями с логическими бомбами могут заниматься чем-то недовольные сотрудники организации; это могут быть консультанты, служащие с определенными политическими целями и т. п.

Троянские кони – это программы, выполняющие в дополнение к основным, т. е. запроектированным и документированным действиям, действия дополнительные, не описанные в документации. Троянский конь представляет собой дополнительный блок команд, тем или иным образом вставленный в исходную безвредную программу, которая затем передается, продается, подменяется) пользователям информационной системы. Этот блок команд может срабатывать при наступлении некоторого условия (даты, времени, по команде извне и т. д.). Запустивший такую программу подвергает опасности как

свои файлы, так и всю информационную систему в целом. Троянский конь действует обычно в рамках полномочий одного пользователя, но в интересах другого пользователя или вообще постороннего человека, личность которого установить порой невозможно.

Наиболее опасные действия троянский конь может выполнять, если запустивший его пользователь обладает расширенным набором привилегий . В таком случае злоумышленник, составивший и внедривший троянского коня и сам этими привилегиями не обладающий, может выполнять несанкционированные привилегированные функции чужими руками .

- Для защиты от этой угрозы желательно, чтобы привилегированные и непривилегированные пользователи работали с различными экземплярами прикладных программ, которые должны храниться и защищаться индивидуально. Оптимальным способом защиты от этой угрозы является создание замкнутой среды использования программ.

Черви – это программы, распространяющие свои копии по локальным и/или глобальным сетям с целью:

- проникновения на удаленные компьютеры;
- запуска своей копии на удаленном компьютере;
- дальнейшего распространения в другие компьютеры в сети.

Для своего распространения сетевые черви используют разнообразные компьютерные и мобильные сети: электронную почту, системы обмена мгновенными сообщениями, файлообменные (P2P) и IRC-сети, LAN, сети обмена данными между мобильными устройствами (телефонами, карманными компьютерами) и т. д.

Большинство известных червей распространяется в виде файлов: вложение в электронное письмо, ссылка на зараженный файл на каком-либо веб- или FTP-ресурсе в ICQ- и IRC-сообщениях, файл в каталоге обмена P2P и т. д.

Некоторые черви (так называемые «бесфайловые» или «пакетные» черви) распространяются в виде сетевых пакетов, проникают непосредственно в память компьютера и активизируют свой код.

Для проникновения на удаленные компьютеры и запуска своей копии черви используют различные методы: социальный инжиниринг (например, текст электронного письма, призывающий открыть вложенный файл), недочеты в конфигурации сети (например, копирование на диск, открытый на полный доступ), ошибки в службах безопасности операционных систем и приложений.

Некоторые черви обладают также свойствами других разновидностей вредоносного программного обеспечения. Например, некоторые черви содержат троянские функции или способны заражать выполняемые файлы на локальном диске, т. е. имеют свойство троянской программы и/или компьютерного вируса.

Email-Worm — это почтовые черви.

К данной категории червей относятся те из них, которые для своего распространения используют электронную почту. При этом червь отсылает либо свою копию в виде вложения в электронное письмо, либо ссылку на свой файл, расположенный на каком-либо сетевом ресурсе (например, URL на зараженный файл, расположенный на взломанном или хакерском веб-сайте).

В первом случае код червя активизируется при открытии (запуске) зараженного вложения, во втором — при открытии ссылки на зараженный файл. В обоих случаях эффект одинаков — активизируется код червя.

Для отправки зараженных сообщений почтовые черви используют различные способы. Наиболее распространены:

- прямое подключение к SMTP-серверу, используя встроенную в код червя почтовую библиотеку;
- использование сервисов MS Outlook;

Различные методы используются почтовыми червями для поиска почтовых адресов, на которые будут рассылаться зараженные письма. Почтовые черви:

- рассылают себя по всем адресам, обнаруженным в адресной книге MS Outlook;
- считывает адреса из адресной базы;
- сканируют «подходящие» файлы на диске и выделяет в них строки, являющиеся адресами электронной почты;
- отсылают себя по всем адресам, обнаруженным в письмах в почтовом ящике (при этом некоторые почтовые черви «отвечают» на обнаруженные в ящике письма).

Многие черви используют сразу несколько из перечисленных методов. Встречаются также и другие способы поиска адресов электронной почты.

IM-Worm — это черви, использующие интернет-пейджеры.

Известные компьютерные черви данного типа используют единственный способ распространения — рассылку на обнаруженные контакты (из контакт-листа) сообщений, содержащих URL на файл, расположенный на каком-либо веб-сервере. Данный прием практически полностью повторяет аналогичный способ рассылки, использующийся почтовыми червями.

IRC-Worm — это черви в IRC-каналах.

У данного типа червей, как и у почтовых червей, существуют два способа распространения червя по IRC-каналам, повторяющие способы, описанные выше. Первый заключается в отсылке URL-ссылки на копию червя. Второй способ — отсылка зараженного файла какому-либо пользователю сети. При этом атакуемый пользователь должен подтвердить прием файла, затем сохранить его на диск и открыть (запустить на выполнение).

Net-Worm — это прочие сетевые черви.

Существуют прочие способы заражения удаленных компьютеров, например:

- копирование червя на сетевые ресурсы;

- проникновение червя на компьютер через уязвимости в операционных системах и приложениях;
- проникновение в сетевые ресурсы публичного использования;
- паразитирование на других вредоносных программах.

Первый способ заключается в том, что червь ищет удаленные компьютеры и копирует себя в каталоги, открытые на чтение и запись (если такие обнаружены). При этом черви данного типа или перебирают доступные сетевые каталоги, используя функции операционной системы, и/или случайным образом ищут компьютеры в глобальной сети, подключаются к ним и пытаются открыть их диски на полный доступ.

Для проникновения вторым способом черви ищут в сети компьютеры, на которых используется программное обеспечение, содержащее критические уязвимости. Для заражения уязвимых компьютеров червь посылает специально оформленный сетевой пакет или запрос (эксплойт уязвимости), в результате чего код (или часть кода) червя проникает на компьютер-жертву. Если сетевой пакет содержит только часть кода червя, он затем скачивает основной файл и запускает его на исполнение.

Отдельную категорию составляют черви, использующие для своего распространения веб- и FTP-сервера. Заражение происходит в два этапа. Сначала червь проникает в компьютер-сервер и необходимым образом модифицирует служебные файлы сервера (например, статические веб-страницы). Затем червь «ждет» посетителей, которые запрашивают информацию с зараженного сервера (например, открывают зараженную веб-страницу), и таким образом проникает на другие компьютеры в сети.

Существуют сетевые черви, паразитирующие на других червях и/или троянских программах удаленного администрирования (бэкдорах). Данные черви используют тот факт, что многие бэкдоры позволяют по определенной команде скачивать указанный файл и запускать его на локальном диске. То же возможно с некоторыми червями, содержащими бэкдор-процедуры. Для заражения удаленных компьютеров данные черви ищут другие компьютеры

в сети и посылают на них команду скачивания и запуска своей копии. Если атакуемый компьютер оказывается уже зараженным «подходящей» троянской программой, червь проникает в него и активизирует свою копию.

Следует отметить, что многие компьютерные черви используют более одного способа распространения своих копий по сетям, использующие два и более метода атаки удаленных компьютеров.

P2P-Worm — это черви для файлообменных сетей.

Механизм работы большинства подобных червей достаточно прост — для внедрения в P2P-сеть червю достаточно скопировать себя в каталог обмена файлами, который обычно расположен на локальной машине. Всю остальную работу по распространению вируса P2P-сеть берет на себя — при поиске файлов в сети она сообщит удаленным пользователям о данном файле и предоставит весь необходимый сервис для скачивания файла с зараженного компьютера.

Существуют более сложные P2P-черви, которые имитируют сетевой протокол конкретной файлообменной системы и на поисковые запросы отвечают положительно — при этом червь предлагает для скачивания свою копию.

Подходящей средой распространения червя является сеть, все пользователи которой считаются дружественными и доверяют друг другу, а защитные механизмы отсутствуют. Наилучший способ защиты от червя – принятие мер предосторожности против несанкционированного доступа к сети.

14.2. Понятие компьютерного вируса

Компьютерный вирус – это специальная программа, предназначенная для выполнения разрушительных действий в вычислительной системе или сети.

Термин «компьютерный вирус» первым употребил сотрудник Лехайского университета Фред Коэн в своем сообщении на национальной конференции США по компьютерной безопасности 1984 года. Ф Коэн является автором первой серьезной работы, посвященной математическим исследовани-

ям жизненного цикла и механизмов размножения компьютерных вирусов. В то время специалисты не придали сообщению большого значения. Но уже с 1985 г. стали появляться сообщения о реальных фактах проявления компьютерных вирусов. С 1987 г. были зафиксированы факты появления компьютерных вирусов и в нашей стране.

Масштабы реальных проявлений «вирусных эпидемий» в настоящее время оцениваются сотнями тысяч случаев заражения персональных компьютеров. Интенсивность вирусных инцидентов постоянно растет. Увеличивается и ущерб, наносимый мировому сообществу этими программами. Особенно опасны вирусы для компьютеров, входящих в состав локальных вычислительных сетей.

При заражении компьютерным вирусом важно его обнаружить. Для этого следует знать об основных признаках проявления вирусов. К ним можно отнести:

- замедление или прекращение работы компьютера;
- частые зависания и сбои в работе компьютера;
- искажение содержимого файлов и папок;
- невозможность загрузки операционной системы;
- прекращение работы или неправильная работа ранее успешно функционирующей программы пользователя;
- увеличение количества файлов на диске;
- изменение размеров файлов;
- нарушение работоспособности операционной системы, что требует ее периодической перезагрузки;
- появление на экране монитора непредусмотренных сообщений;
- подача непредусмотренных звуковых сигналов;
- заметное возрастание времени доступа к винчестеру;
- изменение даты и времени создания файлов;

- разрушение файловой структуры (исчезновение файлов, искажение каталогов);
- существенное уменьшение объема оперативной памяти;
- загорание сигнальной лампочки дисковода, когда к нему нет обращения пользователя;
- форматирование диска без команды пользователя.

Основными путями заражения компьютерными вирусами являются съемные носители и компьютерные сети. Одна из основных причин заражения компьютерными вирусами – отсутствие в операционных системах эффективных средств защиты информации от несанкционированного доступа.

Следует отметить, что вышеперечисленные проблемы необязательно вызываются компьютерными вирусами. Они могут быть следствием некоторых других причин, поэтому вычислительные средства следует периодически комплексно диагностировать.

Прежде всего, вирус – это программа, обладающая способностью к самовоспроизведению. Такая способность является единственным средством, присущим всем типам вирусов. Но не только вирусы способны к самовоспроизведению. Операционные системы и ряд других программ способны создавать свои копии. Но копии вируса не обязаны полностью совпадать с оригиналом, они могут даже полностью отличаться от него.

Способ функционирования большинства вирусов – это такое изменение системных файлов, чтобы вирус начинал свою деятельность при каждой загрузке персонального компьютера. Некоторые вирусы инфицируют файлы загрузки системы, другие специализируются на различных программных файлах. Наиболее часто вирусами заражаются загрузочный сектор диска и файлы, имеющие расширения .exe, .com, .sys, .bat, а также текстовые файлы.

Всякий раз, когда пользователь копирует файлы на машинный носитель информации или посылает инфицированные файлы по сети, переданная копия вируса пытается установить себя на новый диск. Некоторые вирусы разрабатываются так, чтобы они появлялись, когда происходит определенное

событие вызова: например, конкретная дата, определенное число перезагрузок какого-либо конкретного приложения, процент заполнения винчестера и т.д.

После того, как вирус выполнит нужные ему действия, он передает управление той программе, в которой он находится, и ее работа некоторое время не отличается от работы незараженной программы.

Существует понятие жизненного цикла вируса, позволяющее провести анализ действия компьютерных вирусов. Жизненный цикл компьютерного вируса можно представить следующей цепочкой:

Внедрение → Инкубационный период → Репродуцирование (самовоспроизведение) → Деструкция (искажение и/или уничтожение информации)

Для реализации каждого из этапов жизненного цикла компьютерного вируса в его структуру включают несколько взаимосвязанных элементов:

- часть вируса, ответственная за внедрение и инкубационный период;
- часть вируса, осуществляющая его копирования и добавление к другим файлам (программам);
- часть вируса, в которой реализуется проверка условия активизации его деятельности;
- часть вируса, содержащая алгоритм деструктивных действий;
- часть вируса, реализующая алгоритм саморазрушения.

Часто названные части вируса хранятся отдельно друг от друга, что затрудняет борьбу с ними.

14.3. Классификация компьютерных вирусов

Поскольку разнообразие компьютерных вирусов слишком велико, то они, как и их биологические прообразы, нуждаются в классификации. В настоящее время не существует единой системы классификации и именования вирусов (хотя попытка создать стандарт была предпринята еще в 1991 году). Принято разделять вирусы по поражаемым объектам (файловые вирусы, загрузочные вирусы, скриптовые вирусы, макро-вирусы, сетевые черви),

по поражаемым операционным системам и платформам (Microsoft Windows, Unix, Linux), по технологиям, используемым вирусом (полиморфные вирусы, стелс-вирусы), по языку, на котором написан вирус (ассемблер, высокоуровневый язык программирования, скриптовый язык и др.).

По среде обитания вирусы можно разделить на:

- файловые вирусы - внедряются в исполняемые файлы (*.com, *.exe, *.sys, *.bat);
- загрузочные вирусы - внедряются в загрузочный сектор диска (Boot-сектор) или в сектор, содержащий системный загрузчик винчестера (Master Boot Record);
- сетевые вирусы - обитают в компьютерных сетях;
- системные вирусы - проникают в системные модули и драйверы периферийных устройств, таблицы размещения файлов и таблицы разделов;

Существуют и сочетания - например, файлово-загрузочные вирусы, заражающие как файлы, так и загрузочные сектора. Такие вирусы, как правило, имеют довольно сложный алгоритм работы, часто применяют оригинальные методы проникновения в систему и их труднее обнаружить.

По способу заражения файловые вирусы (вирусы, внедряющие свой код в исполняемые файлы: командные файлы, программы, драйверы, исходный код программ и др.) разделяют на перезаписывающие, паразитические, вирусы-звенья, вирусы-черви, компаньон-вирусы, а так же вирусы, поражающие исходные тексты программ и компоненты программного обеспечения:

- перезаписывающие вирусы - записывают своё тело вместо кода программы, не изменяя названия исполняемого файла, вследствие чего исходная программа перестаёт запускаться. При запуске программы выполняется код вируса, а не сама программа;
- вирусы-компаньоны - как и перезаписывающие вирусы, создают свою копию на месте заражаемой программы, но в отличие от перезаписываемых не уничтожают оригинальный файл, а переименовывают или переме-

щают его. При запуске программы вначале выполняется код вируса, а затем управление передаётся оригинальной программе.

Возможно существование и других типов вирусов-компаньонов, использующих иные оригинальные идеи или особенности других операционных систем. Например, PATH-компаньоны, которые размещают свои копии в основном каталоге Windows, используя тот факт, что этот каталог является первым в списке PATH, и файлы для запуска Windows, в первую очередь, будет искать именно в нём. Данными способом самозапуска пользуются также многие компьютерные черви и троянские программы.

- *вирусы-звенья* - как и компаньон-вирусы, не изменяют код программы, а заставляют операционную систему выполнить собственный код, изменяя адрес местоположения на диске заражённой программы, на собственный адрес. После выполнения кода вируса управление обычно передаётся вызываемой пользователем программ;
- «полиморфик»-вирусы (самошифрующиеся или вирусы-призраки, polymorphic) - достаточно труднообнаруживаемые вирусы, не содержащие ни одного постоянного участка кода. В большинстве случаев два образца одного и того же полиморфик-вируса не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика;
- паразитические вирусы - изменяют содержимое файла, добавляя в него свой код. При этом заражённая программа сохраняет полную или частичную работоспособность. Код может внедряться в начало, середину или конец программы. Код вируса выполняется перед, после или вместе с программой, в зависимости от места внедрения вируса в программу;
- вирусы, поражающие исходный код программ - поражают исходный код программы или её компоненты (.OBJ, .LIB, .DCU), а также VCL и ActiveX-компоненты. После компиляции программы оказываются встроенными в неё. В настоящее время широкого распространения не получили.

По степени воздействия компьютерных вирусов на ресурсы компьютерных систем и сетей выделяют:

- безвредные вирусы - не оказывают разрушительного влияния на работу персонального компьютера, но могут переполнять оперативную память в результате своего размножения;
- неопасные вирусы - не разрушают файлы, но уменьшают свободную дисковую память, выводят на экран графические эффекты, создают звуковые эффекты и т. д.;
- опасные вирусы - нередко приводят к различным серьезным нарушениям в работе персонального компьютера и всей информационной системы;
- разрушительные - приводят к стиранию информации, полному или частичному нарушению работы прикладных программ и пр.

По способам заражения вирусы бывают резидентные и нерезидентные:

- резидентный вирус - при инфицировании компьютера оставляет в оперативной памяти свою резидентную часть, которая затем перехватывает обращение операционной системы к объектам заражения и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера;
- нерезидентные вирусы - не заражают память компьютера и являются активными лишь ограниченное время.

По алгоритмической особенности построения *в*ыделяют следующие виды вирусов:

- репликаторы - благодаря своему быстрому воспроизводству приводят к переполнению основной памяти, при этом уничтожение программ-репликаторов усложняется, если воспроизводимые программы не являются точными копиями оригинала;
- мутирующие - со временем видоизменяются и самовоспроизводятся. При этом они воссоздают копии, которые явно отличаются от оригинала;

- стэлс-вирусы (невидимки) - перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо себя незараженные объекты. Использование СТЕЛС-алгоритмов позволяет вирусам полностью или частично скрыть себя в системе. Наиболее распространенным стелс-алгоритмом является перехват запросов операционной системы на чтение/запись зараженных объектов. Стелс-вирусы при этом либо временно лечат их, либо "подставляют" вместо себя незараженные участки информации. Один из первых файловых стелс-вирусов - вирус "Frodo", первый загрузочный стелс-вирус - "Brain".Такие вирусы при обращении к файлам используют достаточно оригинальные алгоритмы, позволяющие «обманывать» резидентные антивирусные программы;
- макровирусы - используют возможности макроязыков, встроенных в офисные программы обработки данных (текстовые редакторы, электронные таблицы и т. д.). Наиболее популярный способ действия этих вирусов - запрет вызовов меню просмотра макросов;

14.4. Программы борьбы с компьютерными вирусами

Наиболее эффективны в борьбе с компьютерными вирусами антивирусные программы. Несмотря на все разнообразие современных антивирусных программных продуктов принципы их работы одинаковы. К основным функциям современных антивирусов относятся:

- сканирование памяти и содержимого дисков по расписанию;
- сканирование памяти компьютера, а также записываемых и читаемых файлов в реальном режиме времени с помощью резидентного модуля;
- выборочное сканирование файлов с измененными атрибутами - размером, датой модификации, контрольной суммой и прочими;
- сканирование архивных файлов;
- распознавание поведения, характерного для компьютерных вирусов;
- удаленная установка, настройка и администрирование антивирусных программ с консоли системного администратора; оповещение системного

администратора о событиях, связанных с вирусными атаками, по электронной почте, по сотовой связи и так далее;

- принудительная проверка подключенных к корпоративной сети компьютеров, инициируемая системным администратором;
- удаленное обновление антивирусного программного обеспечения и баз данных с информацией о вирусах, в том числе автоматическое обновление баз данных по вирусам посредством Интернет;
- фильтрация трафика Интернет на предмет выявления вирусов в программах и документах, передаваемых посредством протоколов SMTP, FTP, HTTP;
- выявление потенциально опасных Java-апплетов и модулей ActiveX;
- функционирование на различных серверных и клиентских платформах, а также в гетерогенных корпоративных сетях;

В настоящее время на рынке программных продуктов имеется довольно большое число специальных антивирусных программ. В основе работы большинства их лежит принцип поиска сигнатуры вирусов.

Вирусная сигнатура – это некоторая уникальная характеристика вирусной программы, которая выдает присутствие вируса в вычислительной системе.

Обычно в антивирусные программы входит периодически обновляемая база данных сигнатур вирусов. Антивирусная программа просматривает компьютерную систему, проводя сравнение и отыскивая соответствие с сигнатурами в базе данных. Когда программа находит соответствие, она пытается убрать обнаруженный вирус.

Существуют следующие основные виды антивирусных программ.

Детекторы (сканеры) – это специальные программы, предназначенные для просмотра всех возможных мест нахождения вирусов (файлы, операционная система, внутренняя память и т. д.) и сигнализирующие об их наличии. Устаревший вариант названия таких программ — «полифаги».

Принцип работы антивирусных сканеров основан на проверке файлов, секторов и системной памяти и поиске в них известных и новых (неизвестных сканеру) вирусов. Для поиска известных вирусов используются так называемые «маски». Маской вируса является некоторая постоянная последовательность кода, специфичная для этого конкретного вируса. Если вирус не содержит постоянной маски, или длина этой маски недостаточно велика, то используются другие методы. Примером такого метода является алгоритмический язык, описывающий все возможные варианты кода, которые могут встретиться при заражении подобного типа вирусом. Такой подход используется некоторыми антивирусами для детектирования полиморфик - вирусов. Сканеры также можно разделить на две категории — «универсальные» и «специализированные». Универсальные сканеры рассчитаны на поиск и обезвреживание всех типов вирусов вне зависимости от операционной системы, на работу в которой рассчитан сканер. Специализированные сканеры предназначены для обезвреживания ограниченного числа вирусов или только одного их класса, например макро-вирусов. Специализированные сканеры, рассчитанные только на макро-вирусы, часто оказываются наиболее удобным и надежным решением для защиты систем документооборота в средах MS Word и MS Excel.

Сканеры также делятся на «резидентные» (мониторы, сторожа), производящие сканирование «на-лету», и «нерезидентные», обеспечивающие проверку системы только по запросу. Как правило, «резидентные» сканеры обеспечивают более надежную защиту системы, поскольку они немедленно реагируют на появление вируса, в то время как «нерезидентный» сканер способен опознать вирус только во время своего очередного запуска. С другой стороны резидентный сканер может несколько замедлить работу компьютера в том числе и из-за возможных ложных срабатываний. К достоинствам сканеров всех типов относится их универсальность, к недостаткам — относительно небольшую скорость поиска вирусов. Наиболее распростране-

ны в России следующие программы: AVP - Касперского, Dr.Weber – Данилова, Norton Antivirus фирмы Semantic.

Принцип работы CRC-сканеров основан на подсчете CRC-сумм (контрольных сумм) для присутствующих на диске файлов/системных секторов. Эти CRC-суммы затем сохраняются в базе данных антивируса, как, впрочем, и некоторая другая информация: длины файлов, даты их последней модификации и т.д. При последующем запуске CRC-сканеры сверяют данные, содержащиеся в базе данных, с реально подсчитанными значениями. Если информация о файле, записанная в базе данных, не совпадает с реальными значениями, то CRC-сканеры сигнализируют о том, что файл был изменен или заражен вирусом.

CRC-сканеры не способны поймать вирус в момент его появления в системе, а делают это лишь через некоторое время, уже после того, как вирус разошелся по компьютеру. CRC-сканеры не могут определить вирус в новых файлах (в электронной почте, на дискетах, в файлах, восстанавливаемых из backup или при распаковке файлов из архива), поскольку в их базах данных отсутствует информация об этих файлах. Более того, периодически появляются вирусы, которые используют эту "слабость" CRC-сканеров, заражают только вновь создаваемые файлы и остаются, таким образом, невидимыми для них.

Фильтры (сторожа) – это резидентные программы, обнаруживающие свойственные для вирусов действия и требующие от пользователя подтверждения на их выполнение. В качестве проверяемых действий выступают:

- обновление программных файлов и системной области диска;
- форматирование диска;
- резидентное размещение программы в оперативной памяти и т. д.

Пользователь в ответ на это должен либо разрешить выполнение действия, либо запретить его. Главными недостатками этих программ является то, что подобные часто повторяющиеся запросы могут мешать пользователю, а объем оперативной памяти уменьшается из-за необходимости постоянного

нахождения в ней вирус-фильтра. К тому же эти программы не лечат файлы или диски, для этого необходимо использовать другие антивирусные программы.

Доктора (дезинфекторы) – это программы, осуществляющие удаление вируса из программного файла или памяти ПК. Если это возможно, то дезинфектор восстанавливает нормальное функционирование ПК. Однако ряд вирусов искажает систему так, что ее исходное состояние дезинфектор восстановить не может.

Иммунизаторы (программы-викцины) – это резидентные программы, изменяющие прививаемый файл таким образом, чтобы вирус, против которого делается прививка, уже считал файл заражённым. В современных условиях, когда количество возможных вирусов измеряется десятками тысяч, этот подход малоэффективен.

Иммунизаторы делятся на два типа: иммунизаторы, сообщающие о заражении, и иммунизаторы, блокирующие заражение. Первые обычно записываются в конец файлов (по принципу файлового вируса) и при запуске файла каждый раз проверяют его на изменение. Недостаток у таких иммунизаторов всего один, но он летален: абсолютная неспособность сообщить о заражении стелс-вирусом. Поэтому такие иммунизаторы, как и блокировщики, практически не используются в настоящее время.

Второй тип иммунизации защищает систему от поражения вирусом какого-то определенного вида. Файлы на дисках модифицируются таким образом, что вирус принимает их за уже зараженные. Для защиты от резидентного вируса в память компьютера заносится программа, имитирующая копию вируса. При запуске вирус натыкается на нее и считает, что система уже заражена.

Такой тип иммунизации не может быть универсальным, поскольку нельзя иммунизировать файлы от всех известных вирусов.

Существуют, также, интегрированные программы, такие как полидетекторы-дезинфекторы, позволяющие выявить вирусы в персональном ком-

пьютере, обезвредить их и по возможности восстановить пораженные файлы и программы. В некоторых случаях программы этого семейства позволяют блокировать зараженный файл от открытия и перезаписи.

Следует отметить, что не существует антивирусных программ, гарантирующих стопроцентную защиту от вирусов, и заявления о существовании таких систем можно расценить как либо недобросовестную рекламу, либо непрофессионализм. Таких систем не существует, поскольку на любой алгоритм антивируса всегда можно предложить контр-алгоритм вируса, невидимого для этого антивируса (обратное, к счастью, тоже верно: на любой алгоритм вируса всегда можно создать антивирус). Самыми популярными и эффективными антивирусными программами являются антивирусные сканеры (другие названия: фаг, полифаг, программа-доктор). Следом за ними по эффективности и популярности следуют CRC-сканеры (также: ревизор, checksumer, integrity checker). Часто оба приведенных метода объединяются в одну универсальную антивирусную программу, что значительно повышает ее мощность.

Одной из основных характеристик современных вирусных атак является их высокая скорость распространения и высокая частота появления новых атак. Следовательно, современное антивирусное программное обеспечение должно обновляться как можно чаще, повышая качество защиты, то есть учитывая все актуальные на текущий момент времени вирусные угрозы. Наличие антивирусного программного обеспечения - необходимое, но недостаточное условие для отражения вирусной атаки. Мало иметь в своем распоряжении средство, требуется продумать методы его использования.

14.5. Меры и средства защиты от компьютерных вирусов

Среди комплекса мер и средств защиты, применяемых для противодействия компьютерным вирусам и другим типам вредоносных программ, можно выделить следующие виды:

1. Юридические меры защиты от компьютерных вирусов.

Для успешной борьбы с распространением вирусов и других типов вредоносных программ в нашей стране необходимо совершенствовать отечественное законодательство в этой области.

2. Административные и организационные меры защиты от компьютерных вирусов. На современном этапе являются более действенными. Они заключаются в составлении четких планов профилактических мероприятий и планов действия на случай возникновения заражений. В подразделениях предприятий и организаций, связанных с эксплуатацией программного обеспечения, должны применяться жесткие административные и организационные меры для предупреждения заражения вирусами.

Возможны следующие способы проникновения вируса или другого типа вредоносной программы в эксплуатируемую систему:

- вирус или другая вредоносная программа поступает вместе с программным обеспечением, предназначенным для последующего использования в работе;
- вирус или другой тип вредоносной программы поступает в систему при приеме сообщений по сети;
- вирус или другая вредоносная программа приносятся персоналом
- с программами, не относящимися к эксплуатируемой системе;
- вирус или другой тип вредоносной программы преднамеренно создаются обслуживающим персоналом.
- Источник вируса легко выявляется, если в эксплуатируемой информационной системе производится разграничение доступа пользователей к привилегированным функциям и оборудованию, присутствуют надежные средства регистрации процесса всего технологического цикла, включая регистрацию внутримашинных процессов. Особенно важными являются меры разграничения доступа в вычислительных сетях.

3. Программно-аппаратные меры защиты основаны на использовании программных антивирусных средств и специальных аппаратных средств (специальных плат), с помощью которых производится контроль зараженно-

сти вычислительной системы, контроль доступа, шифрование данных и регистрация попыток обращения к данным.

Наиболее часто в информационных системах используются два метода защиты от вирусов с помощью использования специального программного обеспечения:

1. Применение «иммуностойких» программных средств, защищенных от возможности несанкционированной модификации (разграничение доступа, методы самоконтроля и самовосстановления).

2. Применение специальных антивирусных программных средств, осуществляющих:

- постоянный контроль возникновения отклонений в деятельности прикладных программ;
- периодическую проверку наличия других возможных следов вирусной активности (например, обнаружение нарушений целостности программного обеспечения);
- входной контроль новых программ и файлов перед их использованием (по характерным признакам наличия в их теле вирусных образований).

Защита от вирусов должна быть элементом политики безопасности организации, которую понимают и соблюдают все пользователи системы.

Для того чтобы сформулировать главные принципы антивирусной политики безопасности, необходимо вспомнить основные моменты, относящиеся к вирусной атаке.

1. Вирусная атака состоит из двух фаз - фаза заражения и фаза распространения (и, возможно, выполнения деструктивных действий).

2. Современные вирусы часто распространяются не только с помощью исполняемых файлов, но и с помощью файлов-документов популярных программ.

3. Современные вирусы при атаке часто используют возможности Интернет.

Очевидно, что лучший способ борьбы с атакой - предотвращение. Решение этой задачи предусматривает ряд действий.

1. Необходимо соответствующим образом сконфигурировать антивирусное программное обеспечение. Для этого произвести следующие установки антивируса:

- сканирование в режиме реального времени, в фоновом или аналогичном режиме, должно быть разрешено;
- при старте системы должны сканироваться память, загрузочный сектор и системные файлы;
- своевременно обновлять вирусные базы данных;
- желательно сканировать все типы файлов или как минимум *.com, *.exe файлы, а также файлы типа *.vbs, *.shs, *.ocx;
- установить аудит всех действий антивирусных программ.

2. Использовать только лицензионное программное обеспечение. Программное обеспечение, полученное из неизвестного источника, может быть зараженно.

3. Ограничить набор программ, которые пользователь способен установить в системе (посторонние программы могут быть заражены вирусами или служить причиной успеха других атак). Особо следует обратить внимание на различные сервисы Интернет и, в первую очередь, на программы передачи сообщений, такие как IRC, ICQ, Microsoft Chat (они могут передавать файлы и служить источником заражения системы).

4. Желательно устранить известные уязвимости в используемом программном обеспечении (они обычно публикуются в списках рассылки Internet, а также на специальных сайтах). В качестве источника информации об уязвимостях можно порекомендовать базу данных на сайте www.securityfocus.com.

5. Контролировать использование накопителей. В идеале вся информация, содержащаяся на носителях, должна быть проверена на наличие вирусов

до того, как к ней будет осуществлен доступ со стороны пользователей вычислительной системы.

6. Разработать политику обработки электронной почты (как составной элемент политики безопасности). Сообщения электронной почты - один из самых популярных и быстрых способов распространения вирусов. Для защиты от проникновения вирусов через сообщения электронной почты каждый пользователь системы должен:

- никогда не открывать сразу почтовое вложение в пришедшем ему сообщении, а сохранять его в определенном "карантинном" каталоге;
- если отправитель неизвестен, сообщение с вложением может быть удалено, но даже когда отправитель известен, сообщение может содержать вирус, поэтому общее правило формулируется следующим образом: никогда не открывать почтовых вложений, которые не были запрошены или о которых не было уведомления от отправителя;
- перед открытием вложения обязательно проверить его с помощью антивирусного программного обеспечения;
- если после выполнения всех этих процедур остались сомнения, стоит связаться с отправителем и выяснить у него информацию о посланном вложении;
- устранить возможные уязвимости в клиентском почтовом программном обеспечении.

7. Разработать политику безопасности приложений (особенно если в организации используется семейство продуктов Microsoft Office), обрабатывающих документы с интерпретируемыми языками (как составной элемент политики безопасности).

Что делать, если заражение уже произошло?

Первый шаг при обнаружении атаки на систему - идентификация. Для успешной идентификации атаки необходимо наличие загрузочного диска, создаваемого при установке системы, и осуществление загрузки системы с его помощью.

Если атака идентифицируется антивирусом, проблема решается фактически моментально. Но, если вы имеете дело с неизвестным вирусом, во многих случаях критичным является время, за которое была идентифицирована атака. Поэтому большое значение имеет способность пользователя быстро обнаружить вирусную атаку (признаками могут служить массовая рассылка почты, уничтожение файлов и т.д.). Сложность идентификации часто зависит от сложности самой атаки. На данном этапе желательно установить как минимум следующие признаки: сам факт атаки, ее тип (сетевая или локальная) и источник происхождения.

Вне зависимости от типа операционной системы необходимо обращать внимание на следующую активность в системе:

- целостность программного обеспечения, используемого для обнаружения нарушителя;
- целостность критичных для безопасности системы программ и данных;
- операции в системе и сетевой трафик.

Если вы смогли определить факт вирусного заражения неизвестным вирусом (или у вас есть такие небезосновательные подозрения), то желательно обратиться к производителю используемого антивирусного программного обеспечения.

И, наконец, необходимо провести анализ последствий вирусной атаки. Если в вашей системе обрабатывались какие-то ценные данные, то настоятельно рекомендуется иметь их резервную копию. Для этого в организации должны быть разработаны правила резервного копирования. К сожалению, если резервная копия отсутствует, данные могут быть утеряны (это уже зависит не от вас, а от злоумышленника, написавшего вирус).

После оценки ущерба нужно устранить причины, приведшие к проникновению вируса в систему с использованием способов, описанных выше.

В любом случае необходимо помнить: наличие адекватных средств защиты и дисциплины их применения позволяет если не избежать вирусной атаки, то, по крайней мере, минимизировать ее последствия.

В целях защиты информационной технологии от компьютерных вирусов необходимо соблюдать следующие правила:

Правило первое. Следует осторожно относиться к программам и документам, полученным из глобальных сетей. Перед тем, как запустить файл на выполнение или открыть документ, базу данных и прочее, необходимо в обязательном порядке проверить их на наличие вирусов.

Правило второе. Для уменьшения риска заразить файл на сервере локальной вычислительной сети следует активно использовать стандартные возможности защиты сетей:

- ограничение прав пользователей;
- установку атрибутов «только для чтения» или «только на запуск» для выполняемых файлов;
- скрытие (закрытие) важных разделов диска и директорий.

В локальных вычислительных сетях следует использовать специализированные антивирусные средства, проверяющие все файлы, к которым идет обращение. Если это по каким-либо причинам невозможно, необходимо регулярно проверять сервер обычными антивирусными средствами. Необходимо также перед тем, как запустить новое программное обеспечение, проверить его на тестовом персональном компьютере , не подключенном к общей сети.

Правило третье. Следует приобретать дистрибутивные копии программных продуктов у официальных поставщиков. При этом значительно снижается вероятность заражения.

Правило четвертое. Следует хранить дистрибутивные копии программного обеспечения (в том числе копии операционной системы), причем копии желательно хранить на защищенных от записи машинных носителях.

Следует пользоваться только хорошо зарекомендовавшими себя источниками программного обеспечения.

Правило пятое. Не следует запускать непроверенные файлы, в том числе полученные из компьютерной сети. Желательно использовать только программы, полученные из надежных источников. Перед запуском новых программ обязательно следует проверять их одним или несколькими антивирусными средствами.

Правило шестое. Необходимо пользоваться утилитами проверки целостности информации. Такие утилиты сохраняют в специальных базах данных информацию о системных областях дисков (или целиком системные области) и информацию о файлах. Следует периодически сравнивать информацию, хранящуюся в подобной базе данных, с информацией , записанной на винчестере. Любое несоответствие может служить сигналом о появлении вируса.

Правило седьмое. Следует периодически сохранять на внешнем носителе файлы, с которыми ведется работа.

Для эффективности защиты информации от компьютерных вирусов необходимо использовать комплекс всех известных способов и средств, выполняя мероприятия непрерывно.

14.6. Защита от спама

Слово «спам» сейчас знакомо практически каждому компьютерному пользователю. В первую очередь термин «спам» относится к электронным письмам. За последние годы рост объема незапрошенных массовых коммерческих рассылок превзошел самые смелые прогнозы и на текущий момент спам представляет собой крупномасштабную угрозу нормальному функционированию электронной почты. В настоящее время доля вирусов и спама в общем трафике электронной почты составляет по разным оценкам от 70 до 95 процентов.

В Рунете встречаются спамерские сообщения практически на любом языке. Более половины спамерских сообщений написаны на иностранных

языках. Лидирует английский язык. Преобладание русского или иностранных языков в спаме, приходящем на конкретный почтовый адрес, зависит от того, в какую спамерскую базу (или базы) попал этот адрес - в базу рассылок по Рунету или в «международную».

***Спам* (*spam*)** — это массовая неперсонифицированная рассылка коммерческой, политической и иной рекламы или иного вида сообщений лицам, не выражавшим желания её получать.

В законодательстве большинства стран закреплена легальность массовой рассылки некоторых видов сообщений без согласия получателей, например, для сообщений о надвигающихся стихийных бедствиях, о массовой мобилизации граждан и т. п.

Есть и другой подход к определению термина спам. Спам — это анонимная массовая незапрошенная рассылка. Анонимность рассылки подтверждается тем, что в настоящее время не существует спамеров, которые не скрывали бы своего адреса и места рассылки, т. е спам это, как правило, автоматическая рассылка со скрытым или фальсифицированным обратным адресом. Массовость рассылки**:** именно массовые рассылки являются настоящим бизнесом для спамеров и настоящей проблемой для пользователей. Небольшая рассылка, сделанная по ошибке человеком, не являющимся профессиональным спамером, может быть нежелательной почтой, но не спамом. Непрошеная рассылка**:** очевидно, подписные рассылки и конференции не должны попадать в категорию «спама» (хотя условие анонимности и так в значительной мере это гарантирует).

В это определение спама не включены словосочетания «рекламная рассылка» или «коммерческое предложение», так как значительная часть спама не преследует рекламных или коммерческих целей.

Спамеры собирают e-mail адреса с помощью специального робота или вручную (редко), используя веб-страницы, конференции Usenet, списки рассылки, электронные доски объявлений, гостевые книги, чаты… Такая программа-робот способна собрать за час тысячи адресов и создать из них базу

данных для дальнейшей рассылки по ним спама. Некоторые компании занимаются только сбором адресов, а базы потом продают. Некоторые компании продают спамерам e-mail адреса своих клиентов, заказавших у них товары или услуги по электронной почте. Есть ещё один способ получить большой список работающих e-mail адресов: адреса сначала генерируются случайным образом по заданным шаблонам (от тысячи до миллиона), а потом просто проверяются специальной программой-валидатором на их валидность (существование).

Для рассылки спама используются подключённые к Интернет плохо защищённые или неправильно настроенные компьютеры.

Спам может распространяться не только через Интернет. Это могут быть рекламные сообщения, присылаемые на мобильные телефоны с помощью SMS-сообщений, при этом часто пользователь должен платить за каждое сообщение.

Существуют рассылки политического и агитационного спама, есть также «благотворительные» спамерские письма (призывающие помочь каким-нибудь несчастным). Целью такой рассылки является сбор e-mail адресов.

Отдельную категорию составляют мошеннические письма («нигерийские письма», потому что большое количество таких писем приходило из Нигерии) с предложениями обналичить большую сумму денег или вовлекающие в финансовые пирамиды.

«Фишинг» (phishing от fishing - рыбалка) - ещё один способ мошенничества с помощью спама. Это попытка спамеров выманить у получателя письма номера его кредитных карточек или пароли доступа к системам онлайновых платежей. Такое письмо обычно маскируется под официальное сообщение от администрации банка. В нём говорится, что получатель должен подтвердить сведения о себе, иначе его счёт будет заблокирован, и приводится адрес сайта (принадлежащего спамерам) с формой, которую надо заполнить. Среди данных, которые требуется сообщить, присутствуют и те, кото-

рые нужны мошенникам. Для того, чтобы жертва не догадалась об обмане, оформление этого сайта также имитирует оформление официального сайта банка.

Еще бывают так называемые «цепочечные письма», например, письма с просьбой переслать их знакомым.

Существуют вирусные письма, содержащие завлекательный текст, и вирусы под видом игрушек, картинок, программ.

Все эти письма, как правило, нельзя отнести к рекламе, хотя они являются очевидным спамом.

Коммерческое предложение, явно направленное на адрес получателя и с реальным обратным адресом, - это не спам. Например, не считается спамом непрошеное рекламное письмо, например, приглашение на семинар, посланное лично директору фирмы, а также предложение круиза с настоящим обратным адресом турфирмы. Такие письма во многих случаях тоже можно распознать и отфильтровать технически, наряду со спамом. Например, Kaspersky Anti-Spam имеет рубрики «Семинары/Конференции», «Туризм» и тому подобные. Рекомендуется следующее: прежде чем удалять письма данных категорий, системному администратору стоит согласовать политику обработки спама с отделом маркетинга. Вполне возможно, что им нужны подобные письма. Например, коммерческие сотрудники туристических фирм часто с интересом читают туристические предложения и даже спам, а организаторы семинаров и сотрудники кадровых отделов хотели бы получать все приглашения на семинары.

Кроме спама и целевых коммерческих предложений существует еще один вид почтовых сообщений, который часто путают со спамом. Это нежелательная почта. В некоторых случаях незапрошенное и ненужное сообщение спамом не является.

Примеры нежелательной почты, которую получатель не заказывал и/или не желает получать:

- разного рода ошибки*:* ошибки автоматических рассыльщиков; запросы на подтверждение подписки на рассылку; ошибки людей (человек ищет однокурсника, а получатель имеет ту же фамилию и похожий адрес);
- разнообразная техническая корреспонденция: сообщения о недоставке письма; автоматические сообщения от антивирусных программ о вирусах в отправленном с вашего адреса письме; сообщения от администраторов сервисов (например, о том, что почтовый сервис будет недоступен, или о появлении вируса);
- новые возможности общения и бизнеса: деловое письмо от частного лица (фирмы) частному лицу (фирме). Такое письмо часто может служить началом нового контракта, дела, бизнеса;
- личные письма от тех, с кем получатель никогда ранее не переписывался: письма от старых знакомых, друзей и т.п.

Любое из этих писем является незапрошенным, так как принимающая сторона его явно не запрашивала. С другой стороны, выбрасывать подобную почту без прочтения нельзя.

Все незапрошенные сообщения, попавшие в почтовый ящик, можно разделить на следующие категории.

1. Спам, имеющий все признаки анонимной массовой рассылки.
2. Целевые коммерческие предложения.
3. Нежелательная почта.

Спам нужно фильтровать, а затем сохранять в особых папках или помещать в карантин, а иногда сразу удалять - в соответствии с политикой компании. Вторую и третью категорию писем также можно распознавать и фильтровать, но с ними нужно обращаться более осторожно. В компании могут быть разные отделы, которые хотели бы получать различные категории непрошеной почты (администраторам нужны сообщения от сервисов и антивирусов, кадровикам — приглашения на семинары).

Массовая рассылка спама имеет низкую себестоимость для отправителя. Однако огромное количество бесполезных сообщений наносит очевидный вред получателям.

В первую очередь это время, потраченном впустую на отсеивание ненужной почты и выискивание среди неё отдельных нужных писем. Очень часто интернет-трафик стоит дорого, и пользователю приходится платить за очевидно ненужные письма. Считается, что спам может быть выгоден провайдерам, так как приводит к повышенному трафику. На самом деле, провайдеры также несут дополнительные затраты из-за повышения бесполезной нагрузки на каналы и оборудование. Именно провайдерам приходится тратить ресурсы на избыточное оборудование и системы защиты от спама (избыточное оборудование, избыточная ёмкость каналов, специальное программное обеспечение для распознавания спама). Спам может использоваться в недобросовестной конкуренции и «чёрном» пиаре.

Очевидно, что спам приносит экономическую выгоду его заказчикам. Это означает, что пользователи, несмотря на неприязнь к спаму, всё-таки пользуются рекламируемыми посредством спама услугами. До тех пор, пока отдача от спама превышает затраты на преодоление защиты, спам не исчезнет. Таким образом, самым надёжным способом борьбы является отказ от услуг, рекламируемых посредством спама. Встречаются предложения о применении общественного осуждения, вплоть до прекращения общения, против лиц, покупающих рекламируемые спамом товары и услуги.

Другие способы направлены на затруднение спамерам доступа к пользователям.

Самый надёжный способ борьбы со спамом — не позволить спамерам узнать электронный адрес.

Проблемы со спамом у частного пользователя начинаются в тот момент, когда его email-адрес попадает в базу данных к спамерам. Email-адреса пользователей спамеры находят сканируя веб-сайты; сканируя доски объяв-

лений, форумы, чаты и так далее; подбирая «легкие» адреса по словарю имен и частых слов; подбирая «короткие» адреса простым перебором.

Исходя из этого, можно порекомендовать следующие меры:

1. Завести два адреса — частный, для переписки (приватный и малоизвестный, который вы никогда не публикуете в общедоступных источниках), и публичный — для публичной деятельности (форумов, чатов и так далее).

2. Адрес для переписки никогда не публиковать в открытом доступе.

3. Адрес для переписки не должен быть легким в запоминании или «красивым». Чем длиннее адрес и чем менее он удобочитаем - тем лучше.

4. Если нужно сообщить свой приватный адрес (в конференции, на сайте), то сделать это способом, непригодным для автоматического прочтения сборщиком адресов. Если речь идет о публикации на сайте, можно опубликовать адрес в виде картинки.

5. Адрес для публикации нужно заранее считать временным. Как правило, спам начинает приходить на него через несколько дней после публикации. Поскольку этот адрес могут использовать не только спамеры (туда будет приходить и нормальная почта), стоит его периодически просматривать (раз в неделю или раз в месяц).

6. При регистрациях на сайтах рекомендуется всегда указывать публичный адрес. Он все равно может считаться потерянным. Можно на каждую регистрацию заводить новый адрес на бесплатных почтах, тогда можно узнать, кто из магазинов и форумов «продал» ваш адрес спамерам.

7. Никогда не отвечать спамеру, так как ответ возможно прочитает «робот» и пометит ваш адрес как «живой» — в результате спама будет приходить еще больше.

8. Рекомендуется использовать антиспам-фильтр: на сервере, выбрав провайдера с услугой фильтрации спама, или у себя на компьютере, выбрав средство, подходящее для вашего почтового клиента. Современные фильтры обладают достаточно высоким качеством (процент фильтруемого спама у хо-

роших и хорошо настроенных фильтров достигает 95-99%), и их использование резко снизит остроту проблемы.

У всех методик сокрытия адреса есть принципиальный недостаток: они создают неудобства реальным адресатам.

Распространённым методом борьбы со спамом стало отсеивание из входящего потока почты. На настоящее время этот метод - основной и наиболее широко используемый.

Существует программное обеспечение (ПО) для автоматического определения спама – спам-фильтры. Оно может быть предназначено для конечных пользователей или для использования на серверах. Это ПО использует два основных подхода.

Первый заключается в том, что анализируется содержание письма и делается вывод, спам это или нет. Письмо, классифицированое как спам, отделяется от прочей корреспонденции: оно может быть помечено, перемещено в другую папку, удалено. Такое ПО может работать как на сервере, так и на компьютере клиента. В последнем случае пользователь не видит отфильтрованного спама, но продолжает нести издержки, связанные с его приемом, так как фильтрующее ПО получает каждое письмо и только потом решает, показывать его или нет. С другой стороны, если ПО работает на сервере, пользователь не несёт издержек по передаче его на свой компьютер.

Второй подход заключается в том, чтобы, применяя различные методы, опознать отправителя как спамера, не заглядывая в текст письма. Это ПО может работать только на сервере, который непосредственно принимает письма. При таком подходе дополнительный трафик тратится только сервером на общение со спамерскими почтовыми программами (т.е. на отказы принимать письма) и обращения к другим серверам (если таковые нужны) при проверке.

Существуют также специализированные online-сервисы, например, «Лаборатория Касперского» (сервис Kaspersky Hosted Security), Outcom («СПАМОРЕЗ»), ИНКАП («Антиспам-Пост»), предоставляющие платную

защиту от спама. Указанные сервисы позволяют перенаправить почту для защищаемого домена на специализированный почтовый сервер, где она очищается от спама и вирусов, а затем направляется на корпоративный почтовый сервер. Метод подходит для корпоративных пользователей и не годится для обладателей почтовых ящиков в публичных почтовых системах.

Проблема автоматической фильтрации в том, что она может по ошибке отмечать как спам полезные сообщения. Поэтому многие почтовые сервисы и программы по желанию пользователя могут не стирать те сообщения, которые фильтр счёл спамом, а помещать их в отдельную папку.

Программы автоматической фильтрации используют статистический анализ содержания письма для принятия решения, является ли оно спамом. Наибольшего успеха удалось достичь с помощью алгоритмов, основанных на теореме Байеса. Для работы этих методов требуется предварительное «обучение» фильтров путем передачи ему рассортированных вручную писем для выявления статистических особенностей нормальных писем и спама.

Метод очень хорошо работает при сортировке текстовых сообщений (в т.ч. HTML). После обучения на достаточно большой выборке удаётся отсечь до 95 -97 % спама. Для обхода таких фильтров спамеры иногда помещают содержательную часть в картинку, вложенную в письмо, текст же либо отсутствует, либо случаен, что не позволяет фильтру составить статистику для распознавания таких писем. В этом случае необходимо пользоваться программами распознавания текста (большинство современных почтовых программ этого не поддерживают), либо использовать другие методы.

Залог надежной работы байесовского метода — постоянное дообучение фильтра и указание ему на совершаемые ошибки. В почтовых программах для этого вводится возможность ручной пометки сообщения «спам/не-спам», а в почтовых сервисах в интернете — кнопка «пожаловаться на спам».

Многие программы и почтовые сервисы в интернете позволяют пользователю задавать собственные фильтры. Такие фильтры могут состоять из слов или, реже, регулярных выражений, в зависимости от наличия или отсут-

ствия которых сообщение попадает или не попадает в мусорный ящик. Однако такая фильтрация трудоёмкая и негибкая, кроме того, требует от пользователя известной степени знакомства с компьютерами. С другой стороны, она позволяет эффективно отсеять часть спама, и пользователь точно знает, какие сообщения будут отсеяны и почему.

Были предложены различные способы для подтверждения того, что компьютер, отправляющий письмо, действительно имеет на это право (Sender ID, SPF, Caller ID, Yahoo DomainKeys), но они пока не получили широкого распространения. Кроме того, эти технологии ограничивают некоторые распространённые виды функциональности почтовых серверов: становится невозможно автоматически перенаправлять корреспонденцию с одного почтового сервера на другой.

Среди провайдеров распространена политика, согласно которой клиентам разрешается устанавливать SMTP-соединения (англ. Simple Mail Transfer Protocol - простой протокол передачи почты) только с серверами провайдера. В этом случае становится невозможно использовать некоторые из механизмов авторизации.

Общие ужесточения требований к письмам и отправителям, например, отказ в приеме писем с неправильным обратным адресом (письма из несуществующих доменов), проверка доменного имени по IP-адресу компьютера, с которого идет письмо, и т. п. С помощью данных мер отсеивается только самый примитивный спам - небольшое число сообщений. Однако не нулевое, поэтому смысл в их применении остается.

Сортировка писем по содержанию полей заголовка письма даёт возможность избавиться от некоторого количества спама. Некоторые клиентские программы (например, Mozilla Thunderbird или The Bat!) дают возможность проанализировать заголовки, не скачивая с сервера всё письмо целиком, и таким образом сэкономить трафик.

Системы типа «вызов-ответ» позволяют убедиться, что отправитель - человек, а не программа-робот. Использование этого метода требует от от-

правителя выполнения определённых дополнительных действий, часто это может быть нежелательно. Многие реализации таких систем создают дополнительную нагрузку на почтовые системы, во многих случаях они присылают запросы на поддельные адреса, поэтому в профессиональных кругах такие решения не пользуются уважением. Кроме того, такая система не может отличить робота, рассылающего спам, от любых других, например тех, которые рассылают новости.

Системы определения признаков массовости сообщения, такие как Razor и Distributed Checksum Clearinghouse (DCC). Встраиваемые в программное обеспечение почтового сервера модули подсчитывают контрольные суммы каждого проходящего через них письма и проверяют их на серверах служб Razor или DCC, которые сообщают количество появлений письма в сети Интернет. Если письмо появилось, например, несколько десятков тысяч раз - вероятно, это спам. С другой стороны, массовое сообщение может быть и легитимной почтовой рассылкой. Кроме того, спамеры могут варьировать текст сообщения, например, добавляя в конец случайный набор символов.

В ряде стран принимаются законодательные меры против спамеров. Попытки законодательного запрещения или ограничения деятельности спамеров наталкиваются на целый ряд сложностей. Непросто определить в законе, какая рассылка является законной, а какая нет. В России спам запрещён федеральным законом «О рекламе» от 13.03.2006 № 38-ФЗ (редакция от 27.10.2008), статья 18, пункт1.

Распространение рекламы по сетям электросвязи, в том числе посредством использования телефонной, факсимильной, подвижной радиотелефонной связи, допускается только при условии предварительного согласия абонента или адресата на получение рекламы и регулируется постановлением Правительства РФ «Об утверждении Правил оказания телематических услуг связи» от 10.09.2007 № 575 (редакция от 16.02.2008).

При этом реклама признается распространенной без предварительного согласия абонента или адресата, если рекламораспространитель не докажет, что такое согласие было получено.

В официальных комментариях Федеральной антимонопольной службы, уполномоченной осуществлять функции контроля за соблюдением этого закона, указывалось на применимость данной нормы к интернет-рассылкам. За нарушение статьи 18 закона «О рекламе» рекламораспространитель несёт ответственность в соответствии с законодательством об административных правонарушениях. Однако ФАС не имеет полномочий для проведения оперативно-розыскных мероприятий по установлению лица, ответственного за спам, а уполномоченные на это органы не могут их проводить в связи с отсутствием в российском административном и уголовном законодательстве ответственности за рассылку спама. Поэтому, несмотря на периодические публикации материалов о привлечении нарушителей к ответственности, в настоящее время данная законодательная норма малоэффективна.

Вопросы для самоконтроля

1. Назовите основные виды вредоносных программ.
2. Что такое компьютерный вирус?
3. Укажите основные признаки заражения компьютера вирусом.
4. Какие существуют способы проникновения вируса или других вредоносных программ в информационную систему?
5. Что представляет собой жизненный цикл компьютерного вируса?
6. Какие элементы образуют структуру компьютерного вируса?
7. Как классифицируются вирусы по способу заражения?
8. Как классифицируются вирусы по среде обитания?
9. Как классифицируются вирусы по степени воздействия?
10. Как классифицируются вирусы по способу заражения среды обитания?

11. Как классифицируются вирусы по алгоритмической особенности построения?

12. Перечислите основным функциям современных антивирусных программ.

13. Как классифицируются антивирусные программы?

14. Назовите основные меры и средства защиты от компьютерных вирусов.

15. Какие правила следует соблюдать в целях защиты информационной технологии от компьютерных вирусов?

16. Дайте определение понятию спам. Раскройте содержание различных подходов к этому термину.

17. Назовите и дайте характеристику наиболее распространенным видам спама.

18. Назовите основные категории незапрошенных сообщений, получаемых по электронной почте.

19. Целевые коммерческие предложения. Какие особенности существуют при работе с подобными рассылками?

20. Какой вид почтовых сообщений называют нежелательной почтой? Приведите примеры нежелательной почты.

21. В чем заключается политика обращения со спамом и нежелательной почтой?

22. В чем заключаются негативные последствия рассылки спама?

23. Перечислите способы борьбы со спамом.

24. Назовите основные мероприятия, позволяющие избежать попадания email-адреса в базу данных к спамерам.

25. В чем заключается суть методики автоматической фильтрации спам?

26. Что такое спам-фильтр и где он применяется?

27. Назовите основные подходы, используемые при автоматической фильтрации спама.

28. В чем заключается суть неавтоматической фильтрации спама? Укажите достоинства и недостатки этого метода.

29. Что понимается под авторизациий почтовых серверов? Каковы преимущества и недостатки этой технологии в борьбе со спамом?

30. Охарактеризуйте юридические аспекты проблемы борьбы со спамом.

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	Какие вирусы поражают загрузочные секторы дисков и файлы прикладных программ?	• мутирующие • файлово-загрузочные • стэлс-вирусы
2.	Какие вирусы со временем видоизменяются?	• мутирующие • репликаторные • макровирусы
3.	Какие меры зашиты от компьютерных вирусов заключаются в составлении четких планов профилактических мероприятий и планов действия на случай возникновения заражений?	• программно-аппаратные • юридические • административные и организационные
4.	Какие программы относятся к антивирусным?	• AVP, DrWeb, Norton AntiVirus • Windows, Skype, AVP • Skype, IRC, Norton Commander
5.	На чем основано действие антивирусной программы?	• на ожидании начала вирусной атаки. • на сравнение программных кодов с известными вирусами • на удалении зараженных файлов
6.	Некоторая уникальная характеристика вирусной программы, которая выдает присутствие вируса в вычислительной системе – это:	• деструкция • вирусная сигнатура • репродуцирование
7.	Основными типами компьютерных вирусов является –	• аппаратные, программные, загрузочные • файловые, загрузочные, макровирус, сетевые • файловые, программные, макровирусы
8.	Программа, выполняющая в дополнение к основным, т. е. запроектированным и документированным действиям, действия дополнительные, не описанные в докумен-	• троянский конь • логическая бомба • захватчик паролей

	тации, – это:	
9.	Что такое компьютерный вирус?	• прикладная программа • системная программа • программы, которые могут «размножаться» и скрытно внедрять свои копии в файлы, загрузочные секторы дисков и документы • утилита
10.	Резидентные программы, обнаруживающие свойственные для вирусов действия и требующие от пользователя подтверждения на их выполнение, называются …	• фильтры (сторожа) • черви • детекторы • иммунизаторы
11.	Массовая неперсонифицированная анонимная рассылка сообщений лицам, не выражавшим желания её получать – это…	• вирус • спам • электронное письмо • прикрепленный файл
12.	Фишинг – это…	• способ заражения компьютерным вирусом • способ мошенничества с помощью спама • нарушение конфиденциальности информации
13.	К компьютерным вирусам не относятся:	• логические бомбы • захватчики паролей • репликаторы • спам
14.	IM-Worm, Email-Worm, IRC-Worm - это:	• логические бомбы • черви • троянские кони • спам
15.	Черви - это:	• программы, распространяющие свои копии по сетям • блоки команд, вставляемых в исходную безвредную программу • фрагмент программы, срабатывающий при выполнении некоторого условия
16.	Кто первым употребил термин «компьютерный вирус»?	• Дж. Фон Нейман • Билл Гейтс • Фред Коэн • Клод Шеннон
17.	Какие периоды включает жизненный цикл компьютерного вируса?	• инкубационный период • импликация • деструкция • репродуцирование
18.	Вирусы классифицируются по следующим признакам:	• по среде обитания • по способу заражения • по степени воздействия

		• по времени срабатывания
19.	К основным видам антивирусных программ не относятся:	• детекторы • импликаторы • фильтры • иммунизаторы
20.	Методы автоматической и неавтоматической фильтрации относятся:	• к методам борьбы с компьютерными вирусами • к методам борьбы со спамом • к методам гибернации • к методам идентификации

Глава 15. Защита информации в корпоративных системах

15.1. Цели и задачи корпоративной системы информационной безопасности

Главной целью любой системы обеспечения информационной безопасности является обеспечение устойчивого функционирования предприятия, предотвращение угроз его безопасности, защита законных интересов предприятия от противоправных посягательств, недопущение хищения финансовых средств, разглашения, утраты, утечки, искажения и уничтожения служебной информации, обеспечение нормальной торговой и производственной деятельности всех подразделений предприятия.

Еще одной целью системы информационной безопасности является повышение качества предоставляемых услуг и гарантий безопасности имущественных прав и интересов клиентов.

Основными задачами любой системы информационной безопасности предприятия являются:

- отнесение информации к категории ограниченного доступа (служебной или коммерческой тайне);
- прогнозирование и своевременное выявление угроз безопасности информационным ресурсам, причин и условий, способствующих нанесению финансового, материального и морального ущерба, нарушению его нормального функционирования и развития;
- создание условий функционирования с наименьшей вероятностью реализации угроз безопасности информационным ресурсам и нанесения различных видов ущерба;
- создание механизма и условий оперативного реагирования на угрозы информационной безопасности и проявления негативных тенденций в функционировании, эффективное пресечение посягательств на ресурсы на основе правовых, организационных и технических мер и средств обеспечения безопасности;

- создание условий для максимально возможного возмещения и локализаций ущерба, наносимого неправомерными действиями физических и юридических лиц, ослабление негативного влияния последствий нарушения информационной безопасности на достижение стратегических целей.

15.2. Политики информационной безопасности

15.2.1. Основные понятия политик безопасности

Политика информационной безопасности компании – это формальное изложение правил поведения лиц, получающих доступ к конфиденциальным данным в корпоративной информационной системе. При этом различают:

- общую стратегическую политику безопасности компании, взаимоувязанную со стратегией развития бизнеса и информационно-технологической стратегией компании;
- частные тактические политики безопасности, детально описывающие правила безопасности при работе с соответствующими информационно-технологическими системами и службами компании.

В соответствии с рекомендациями ведущих международных стандартов в области планирования информационной безопасности и управления ею политики безопасности должны содержать следующее:

- предмет, основные цели и задачи политики безопасности;
- условия применения политики безопасности и возможные ограничения;
- описание позиции руководства компании в отношении выполнения политики безопасности и организации режима информационной безопасности компании в целом;
- права и обязанности, а также степень ответственности сотрудников за выполнение политики безопасности компании;

- порядок действия в чрезвычайных ситуациях в случае нарушения политики безопасности.

Актуальность разработки политик безопасности для отечественных компаний и организаций объясняется необходимостью формирования основ планирования информационной безопасности и управления ею на современном этапе. В настоящее время большинством российских компаний определены следующие приоритетные задачи развития и совершенствования своей деятельности:

- минимизация рисков бизнеса путем защиты своих интересов в информационной сфере;
- обеспечение безопасного, доверенного и адекватного управления предприятием;
- планирование и поддержка непрерывности бизнеса;
- повышение качества деятельности по обеспечению информационной безопасности;
- снижение издержек и повышение эффективности инвестиций в информационную безопасность;
- повышение уровня доверия к компании со стороны акционеров, партнеров, уполномоченных государственных органов и др.

Успешное выполнение перечисленных задач проблематично. Это связано с. возрастающей необходимостью повышения уровня информационной безопасности и недостаточной проработанностью политик информационной безопасности в отечественных компаниях.

При разработке политик безопасности важно иметь в виду:

- для достижения разумного баланса между стоимостью и эффективностью разрабатываемых правил политик безопасности необходимо учитывать в равной мере нормативные, экономические, технологические, технические и организационно-управленческие аспекты планирования информационной безопасности и управления ею;

- политики безопасности российских компаний не должны противоречить отечественной нормативной базе в области защиты информации в автоматизированных системах на территории РФ;
- при создании политик безопасности желательно учесть текущие реформы действующей Государственной системы стандартизации;
- при отражении в политиках безопасности нормативного аспекта рекомендуется следовать требованиям новой российской национальной системы стандартизации. Следование этим требованиям позволит устранить существующие технические барьеры для отечественных компаний в торговле и обеспечения конкурентоспособности продукции;
- использование в политиках безопасности современных подходов и принципов обеспечения информационной безопасности, основанных на лучшем мировом и отечественном опыте. Это позволит выработать концептуальную схему обеспечения информационной безопасности, а также требуемые модели постановки проблем в области управления информационной безопасностью и предложить разумно достаточные решения этих проблем;
- при отражении в разрабатываемых политиках безопасности отечественных компаний экономического подхода к планированию информационной безопасности и управлению ею на основе концепции управления рисками рекомендуется обратить внимание на методы прикладного информационного анализа; расчета потребительского индекса; расчета добавленной экономической стоимости; определения исходной экономической стоимости; управления портфелем активов; оценки действительных возможностей; поддержки жизненного цикла искусственных систем; расчета системы сбалансированных показателей; расчета совокупной стоимости владения; функционально-стоимостного анализа;
- при разработке детальных технических политик безопасности следует определить техническую архитектуру корпоративных систем защиты конфиденциальной информации компаний, в частности: определить цели создания технической архитектуры корпоративной системы защиты информа-

ции; разработать эффективную систему обеспечения информационной безопасности на основе управления информационными рисками; рассчитать совокупности детализированных показателей для оценки соответствия информационной безопасности заявленным целям; выбрать требуемый инструментарий обеспечения информационной безопасности и оценки ее текущего состояния; реализовать требуемые методики мониторинга и управления информационной безопасностью;

- политики безопасности должны представлять собой законченные нормативные документы, содержащие единые нормы и требования по обеспечению информационной безопасности, обязательные для утверждения и применения соответствующими органами управления, руководством служб безопасности, руководством служб информационно-технологического обеспечения отечественных компаний.

Современный рынок средств защиты информации можно условно разделить на две группы:

- средства защиты для государственных структур, позволяющие выполнить требования нормативно-правовых документов (федеральных законов, указов Президента РФ, постановлений Правительства РФ), а также требования нормативно-технических документов(государственных стандартов, руководящих документов Гостехкомиссии (ФСТЭК) России, силовых ведомств РФ;
- средства защиты для коммерческих компаний и структур, позволяющие выполнить требования и рекомендации федеральных законов, указов Президента РФ, постановлений Правительства РФ и некоторых международных стандартов.

Например, к защите конфиденциальной информации в органах исполнительной власти могут предъявляться следующие требования.

1. Выбор конкретного способа подключения к сети Интернет, в совокупности обеспечивающего межсетевое экранирование с целью управления доступом, фильтрации сетевых пакетов и трансляции сетевых адресов для

сокрытия структуры внутренней сети, а также проведение анализа защищенности интернет-узла, использование средств антивирусной защиты и централизованное управление, должны производиться на основании действующих рекомендаций Гостехкомиссии РФ.

2. Автоматизированные системы организации должны обеспечивать защиту информации от несанкционированного доступа в соответствии с действующим руководящим документом Гостехкомиссии РФ.

3. Средства вычислительной техники, программные средства автоматизированных систем должны удовлетворять требованиям действующего руководящего документа Гостехкомиссии РФ.

4. Программно-аппаратные средства межсетевого экранирования, применяемые для изоляции корпоративной сети от сетей общего пользования, должны удовлетворять требованиям действующего руководящего документа Гостехкомиссии РФ.

5. Информационные системы должны удовлетворять требованиям действующего ГОСТ по защищенности информационных систем в рамках заданных профилей защиты.

6. Программно-аппаратные средства криптографической защиты конфиденциальной информации, в том числе используемые для создания виртуальных защищенных сетей - Virtual Private Network (VPN), должны быть легитимны (т.е. соответствовать действующему законодательству).

7. Обязательным является использование средств электронно-цифровой подписи (ЭЦП) для подтверждения подлинности документов.

8. Для использования персональных цифровых сертификатов и поддержки инфраструктуры открытых ключей, средств ЭЦП и шифрования необходимо создать легитимный удостоверяющий центр (систему удостоверяющих центров).

9. Политика информационной безопасности должна предусматривать обязательное включение в технические задания на создание коммуникационных и информационных систем требований информационной безопасности.

10. Должен быть регламентирован порядок ввода в эксплуатацию новых информационных систем, их аттестации по требованиям информационной безопасности.

Для выполнения перечисленных требований и надлежащей защиты конфиденциальной информации в госструктурах принято использовать сертифицированные средства, например, средства защиты от несанкционированного доступа, межсетевые экраны и пр.

Средства защиты информации для коммерческих структур более многообразны и включают в себя средства:

- управления обновлениями программных компонент;
- межсетевого экранирования;
- построения VPN;
- контроля доступа;
- обнаружения вторжений и аномалий;
- резервного копирования и архивирования;
- централизованного управления безопасностью;
- контроля деятельности сотрудников в сети Интернет;
- анализа содержимого почтовых сообщений;
- анализа защищенности информационных систем;
- защиты от спама;
- защиты от атак класса «отказ в обслуживании»;
- контроля целостности;
- инфраструктуры открытых ключей и пр.

Можно привести следующую краткую характеристику основных средств защиты информации.

1. Средства управления обновлениями.

Внедрение средств управления обновлениями программных компонент автоматизированных систем, например Microsoft Software Update Services, позволяет уменьшить объем интернет-трафика предприятия в целом, так как

становится возможным организовать и контролировать необходимые обновления программных компонент автоматизированных систем предприятия с одной точки – выделенного внутреннего сервера. При этом предприятие получает следующие преимущества:

- уменьшаются расходы по оплате трафика;
- увеличивается надежность функционирования программных компонент;
- уменьшается время на техническую поддержку и сопровождение программных компонент;
- повышается защищенность автоматизированной системы в целом, в частности уменьшается количество инцидентов, связанных с вирусами.

2. Средства межсетевого экранирования.

Межсетевые экраны используются как средства защиты от несанкционированного доступа периметра сети и основных критичных компонент автоматизированных систем. Межсетевые экраны позволяют организовать защиту на уровне доступа к компонентам и сети в целом, на сетевом уровне (контроль IP-адресов), на транспортном уровне и на прикладном уровне.

3. Средства построения VPN.

Средства построения виртуальных частных сетей (VPN) используются для организации защиты трафика данных, передаваемых по открытым каналам связи. При этом защита организуется на физическом уровне (защита кабелей, экранизация наводок), на сетевом уровне (например, шифрование трафика от компьютера до компьютера на основе протокола IPsec), на транспортном уровне (например, шифрование данных, передаваемых одним приложением другому приложению на другом компьютере, на основе протокола SSL) на прикладном уровне (например, шифрование данных самостоятельно приложением).

4. Средства контроля доступа.

Данные средства осуществляют аутентификацию (точное опознание) подключающихся к автоматизированным системам (АС) пользователей и

процессов, авторизацию (наделение определенными полномочиями) пользователей и процессов, регламентируя доступ множества пользователей к приложениям и информационным ресурсам предприятия.

5. Средства обнаружения вторжений и аномалий.

Средства обнаружения вторжений (Intrusion Detection Systems, IDS) позволяют с помощью некоторого регламента проверок контролировать состояние безопасности корпоративной сети в реальном масштабе времени. Это программные или аппаратные средства, предназначенные для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет. Системы обнаружения вторжений используются для обнаружения некоторых типов вредоносной активности, которая может нарушить безопасность компьютерной системы. К такой активности относятся сетевые атаки против уязвимых сервисов, атаки, направленные на повышение привилегий, неавторизованный доступ к важным файлам, а также действия вредоносного программного обеспечения.

6. Средства резервного копирования и архивирования.

Средства резервного копирования и архивирования применяются для обеспечения целостности хранилищ в случаях аппаратных и программных сбоев, ошибочных действий администраторов и пользователей, а также отказов средств вычислительной техники.

7. Средства централизованного управления безопасностью.

Средства централизованного управления информационной безопасностью позволяют эффективно создавать, проверять и поддерживать технические политики безопасности программных компонент автоматизированной системы.

8. Средства предотвращения вторжений на уровне серверов.

Так как серверы компании обычно являются основной целью атак злоумышленников (на них обрабатывается основная часть конфиденциальной

информации компании), то необходимо использовать средства предотвращения вторжений на уровне серверов.

9. Средства мониторинга безопасности.

Большое количество средств обеспечения информационной безопасности (межсетевые экраны, системы обнаружения вторжений, маршрутизаторы, средства создания виртуальных частных сетей, журналы безопасности серверов, системы аутентификации, средства антивирусной защиты и т.д.) генерирует огромное количество сообщений. Для успешного мониторинга и управления этими средствами рекомендуется использовать соответствующие средства аудита безопасности.

10. Средства контроля деятельности сотрудников в Интернете.

В настоящее время одной из серьезных проблем в работе отечественных служб безопасности является предотвращение попыток использования интернет-ресурсов компании в личных целях (загрузка видео, аудио, картинок, нелицензированного программного обеспечения). Нецелевое использование Интернета приводит к потере продуктивности сотрудников компании приблизительно на 30-40%.

Для предупрежден6ия подобных действий рекомендуется применять соответствующие средства, например Websense, которые позволяют анализировать и формировать отчеты по использованию сотрудниками компании ресурсов Интернета и программного обеспечения на рабочих местах, а также проводить анализ сетевой активности и пропускной способности сети компании в целом.

11. Средства анализа содержимого почтовых сообщений.

Средства анализа содержимого почтовых сообщений предназначены для обнаружения и предотвращения передачи конфиденциальной информации с помощью корпоративной электронной почты.

12. Средства анализа защищенности.

Анализ защищенности АС является одним из ключевых аспектов построения надежной системы обеспечения информационной безопасности предприятия и основан на применении сканеров безопасности.

Основной особенностью наиболее продаваемых и используемых коммерческих сканеров является возможность как минимум еженедельно обновлять базы данных уязвимостей путем взаимодействия с крупнейшими центрами по сбору новых уязвимостей и с ведущими производителями сетевого оборудования и программного обеспечения.

13. Средства защиты от спама.

Спам наносит предприятию значительный ущерб. Приходится тратить время на просмотр и удаление таких сообщений. Спам также содержит вирусы, программы-шпионы и пр. Для предотвращения получения сотрудниками сообщений, содержащих спам, можно воспользоваться одним из продуктов следующих фирм: Symantec (использует технологию фирмы Brightmail), Trend Micro (использует технологию фирмы Postini) или «Лаборатории Касперского».

14. Средства защиты от атак класса «отказ в обслуживании».

В связи с тем, что атаки класса «отказ в обслуживании» приносят значительные убытки отечественным и западным компаниям, можно воспользоваться специальными средствами защиты, например продуктами компании Cisco Systems.

15. Средства контроля целостности.

Внесение некорректного изменения в конфигурацию сервера или маршрутизатора может привести к выходу из строя необходимого сервиса или целой сети. Очень важно предупредить и отследить несанкционированные изменения. Для быстрого реагирования на такую ситуацию нужно иметь средство отслеживания всех производимых изменений. Данную возможность предоставляет, например, серия продуктов компании Tripwire.

16. Средства инфраструктуры открытых ключей.

Внедрение инфраструктуры открытых ключей очень трудоемкая задача, требующая тщательной проработки и анализа. При решении этой задачи можно воспользоваться продуктами компании RSA Security (Keon) и отечественной компании «КриптоПро» («Криптопровайдер»).

15.2.2. Основные причины создания политик безопасности

Политики информационной безопасности определяют стратегию и тактику построения корпоративной системы защиты информации. В российской терминологии документ, определяющий стратегию, часто называют концепцией, а документ определяющий тактику, - политикой. На западе принято создавать единый документ, включающий в себя стратегию и тактику защиты. Политики безопасности компании являются основой для разработки целого ряда документов по обеспечению безопасности: стандартов, руководств, процедур, практик, регламентов, должностных инструкций и пр.

К мотивам разработки политики информационной безопасности отечественными предприятиями и компаниями относятся:

- выполнение требований компании;

Как правило, руководство компании проявляет внимание к проблемам информационной безопасности после нескольких серьезных инцидентов, повлекших за собой остановку или замедление работы компании. Например, в результате вирусной атаки или атаки «отказ в обслуживании», разглашения конфиденциальной информации или кражи компьютеров с ценной информацией.

- выполнение требований российской нормативной базы в области защиты информации;

Каждая компания обладает информацией, представляющей некоторую ценность и, по понятным причинам, она не желала бы разглашения. Политики информационной безопасности позволяют определить правила, в соответствии с которыми информация будет отнесена к категории коммерческой или служебной тайны. Это позволит компании юридически защитить информа-

цию (ст.139 Гражданского кодекса и закон «О коммерческой тайне»). В зависимости от сферы компании она должна выполнять требования существующего законодательства, применимого к ее отрасли.

В целом автоматизированные системы отечественных компаний должны удовлетворить требованиям российской нормативной базы в области защиты информации, нормативно-правовым документам (федеральным законам, указам Президента, постановлениям Правительства) и нормативно-техническим документам (государственным стандартам, руководящим документам Гостехкомиссии (ФСТЭК) России, отраслевым и ведомственным стандартам).

- выполнение требований клиентов и партнеров;

Клиенты и партнеры компании часто желают получить гарантии того, что их конфиденциальная информация защищена надлежащим образом и могут потребовать юридического подтверждения этого в контрактах. В этом случае политики информационной безопасности компании и являются доказательством предоставления подобных гарантий, так как в политиках безопасности декларируется намерения компании относительно качества обеспечения информационной безопасности.

- подготовка к сертификации по ISO 9001, ISO 15408 и ISO 17799;

Сертификация по одному из вышеперечисленных стандартов подтверждает необходимый уровень обеспечения информационной безопасности компании. В настоящее время фокус создания продуктов и услуг смещается в страны дешевой рабочей силой, и одним из доказательств того, что компании этих стран смогут адекватно защитить передаваемую информацию производителей, является сертификация на соответствие требованиям стандартов по информационной безопасности, например ISO 17799 (BS 7799-2).

- устранение замечаний аудиторов;

Любая внешняя аудиторская проверка обращает внимание на необходимость защищенности бизнес-процессов компании, в том числе особое внимание уделяется наличию политик информационной безопасности.

- получение конкурентного преимущества на рынке;

Правильно разработанные и реализованные политики безопасности позволяют увеличить время доступности и коэффициент готовности сервисов компании. Таким образом увеличивается общая жизнеспособность компании и обеспечивается непрерывность бизнеса.

- демонстрация заинтересованности руководства компании;

Вовлечение руководства в организацию режима информационной безопасности компании значительно увеличивает приоритет безопасности, что положительно сказывается на общем уровне безопасности компании. Без демонстрации заинтересованности руководства компании сотрудники не станут воспринимать политики информационной безопасности всерьез. Цель любой политики безопасности – разъяснение и доведение позиции руководства в соответствии с принципами безопасности и бизнес–целями компании.

- создание корпоративной культуры безопасности;

Сотрудники компании должны понимать, что обеспечение информационной безопасности – обязанность всех сотрудников. Это достигается путем введения процедуры ознакомления с требованиями политик безопасности и подписания соответствующего документа о том, что сотрудник ознакомлен, ему понятны все требования политик, и он обязуется их выполнить. Политики безопасности позволяют ввести требования по поддержанию необходимого уровня безопасности в перечень обязанностей каждого сотрудника. В процессе выполнения трудовых обязанностей для сотрудников необходимо периодически проводить ознакомление с вопросами обеспечения информационной безопасности и обучение. Чем крупнее компания, тем более важной становится информационная поддержка сотрудников по вопросам безопасности.

- уменьшение стоимости страхования;

Страхование – важная составляющая управления информационными рисками. Наличие политик информационной безопасности является необходимым и обязательным условием страхования. В России уже появились фир-

мы, предлагающие страховать информационные риски, например «Ингосстрах» и «РОСНО». Стоимость страхования страховая компания определяет путем проведения аудита в этой области.

Таким образом, политики обеспечения информационной безопасности необходимы для успешной организации режима информационной безопасности любой отечественной компании. Политики безопасности минимизируют влияние «человеческого фактора» и недостатки существующих технологий защиты информации. Кроме того, политики безопасности дисциплинируют сотрудников компании и позволяют создать корпоративную культуру безопасности.

15.2.3. Разработка политик безопасности

Разработка политик безопасности предполагает решение ряда вопросов.

1. Выбор уровня доверия.

От правильного выбора уровня доверия к сотрудникам зависит успех или неудача реализации политики безопасности компании. При разработке политики безопасности обычно используют следующие модели доверия:

- доверять всем и всегда – самая простая модель доверия, но, к сожалению, непрактичная;
- не доверять никому и никогда – самая ограниченная модель доверия и также непрактичная;
- доверять избранным на время – модель доверия подразумевает определение разного уровня доверия на определенное время. При этом доступ к информационным ресурсам компании предоставляется по необходимости для выполнения служебных обязанностей, а средства контроля доступа используются для проверки уровня доверия к сотрудникам компании.

Самая реалистичная модель доверия – «доверять некоторым из сотрудников компании на время».

2. Проведение работы с персоналом.

Внедрение политики безопасности часто приводит к возникновению напряженности во взаимоотношениях между сотрудниками компании. Это в основном связано с тем, сотрудники часто стараются не следовать каким-либо правилам безопасности, так как не хотят себя ограничивать в своих действиях. Другая причина в том, что каждый сотрудник имеет свое представление о необходимости и способах организации режима информационной безопасности в компании. Получить одобрение всех положений политики безопасности у всех групп сотрудников компании – трудная и практически неосуществимая задача. Поэтому лучше всего попробовать достигнуть некоторого компромисса.

3. Определение круга заинтересованных лиц.

Политики безопасности затрагивают практически каждого сотрудника компании. Сотрудники службы поддержки будут осуществлять и поддерживать правила безопасности компании. Менеджеры заинтересованы в обеспечении безопасности информации для достижения своих целей. Юристы компании и аудиторы заинтересованы в поддержании репутации компании и предоставлении определенных гарантий безопасности клиентам и партнерам компании. Рядовых сотрудников компании политики безопасности затрагивают больше всего, поскольку правила безопасности накладывают ряд ограничений на поведение сотрудников и затрудняют выполнение работы.

4. Определение состава группы но разработке политик безопасности.

Рекомендуемый состав рабочей группы по разработке политик безопасности:

- член совета директоров;
- представитель руководства компании (финансовый директор, директор по развитию);
- директор службы автоматизации;
- директор по информационной безопасности;
- аналитик службы безопасности;
- представитель юридического отдела;

- представитель от пользователей.

5. Ознакомление сотрудников с политикой безопасности.

До начала внедрения новой политики безопасности желательно предоставить сотрудникам текст политики на одну-две недели для ознакомления и внесения поправок и комментариев. Сотрудники, на которых распространяются правила безопасности, должны обладать всеми необходимыми полномочиями для того, чтобы выполнять эти правила.

6. Определение основных требований к политике безопасности.

Политика безопасности должна быть реалистичной и выполнимой, краткой и понятной, а также не приводить к существенному снижению общей производительности подразделений компании. Политика безопасности должна содержать основные цели и задачи организации режима информационной безопасности, четкое описание области действия, а также указывать на ответственных и их обязанности. Следует учитывать, как политика безопасности будет влиять на уже существующие информационные системы компании. Как только политика утверждена, она должна быть представлена сотрудникам компании для ознакомления. Политику безопасности необходимо пересматривать ежегодно, чтобы отражать текущие изменения в развитии бизнеса компании.

7. Выбор политик безопасности.

Хорошо написанные политики безопасности компании должны позволять балансировать между достигаемым уровнем безопасности и получаемым уровнем производительности корпоративных информационных систем компании. Здесь решающую роль играют необходимость организации режима информационной безопасности, а также бизнес-культура компании. При этом, если правила политики безопасности слишком жесткие, то они будут либо игнорироваться, либо сотрудники компании найдут способ обойти средства безопасности.

В настоящее время ряд ведущих компаний в области безопасности выделяют следующие политики:

- допустимого шифрования,
- допустимого использования,
- аудита безопасности,
- оценки рисков,
- классификации данных,
- управления паролями,
- использования ноутбуков,
- безопасности рабочих станций и серверов,
- антивирусной защиты,
- безопасности маршрутизаторов и коммутаторов,
- безопасности беспроводного доступа,
- организации удаленного доступа,
- построения виртуальных частных сетей (VPN) и пр.,
- безопасности периметра.

Например:

- политика допустимого использования информационных ресурсов компании определяет права и ответственность сотрудников компании за надлежащую защиту конфиденциальной информации компании. В частности, политика допустимого использования определяет, могут ли сотрудники компании читать и копировать файлы, владельцами которых они не являются, но к которым имеют доступ. Также эта политика устанавливает правила допустимого использования корпоративной электронной почты, служб новостей и процедур доступа к сети компании;

- политика организации удаленного доступа определяет допустимые способы удаленного соединения с корпоративной информационной системой. Представляет собой основной документ безопасности в крупных транснациональных компаниях с географически разветвленной сетью. Должна описывать все доступные способы удаленного доступа к внутренним инфор-

мационным ресурсам компании: доступ по коммутируемым сетям, доступ через Интернет, выделенную линию и пр.

Политика удаленного доступа определяет, кто из сотрудников может иметь высокоскоростной удаленный доступ. При этом определяются ограничения по организации удаленного доступа;

- политика безопасности периметра описывает порядок и правила получения привилегированного доступа к системам безопасности периметра корпоративной сети компании. Кроме того, описывает процедуру инициации и обработки запросов на изменение конфигурации систем безопасности периметра сети, а также порядок и периодичность проверки этих конфигураций;

- политика управления паролями определяет правила и порядок создания и изменения паролей сотрудников компании. Например, все пароли системного уровня, пароли администраторов приложений и пр. должны периодически изменяться. Все пароли системного уровня должны быть частью глобальной базы данных управления паролями отдела защиты информации.

Также как и политики безопасности важны и процедуры безопасности. Если политики безопасности определяют что должно быть защищено, то процедуры безопасности определяют как защитить информационные ресурсы компании.

Наиболее важные процедуры безопасности:

- процедура управления конфигурацией обычно определяется на уровне отдела или на уровне компании. Процедура управления изменениями должна определять процесс документирования и запроса на изменения конфигурации всех масштабов. Служба защиты информации должна проводить анализ изменений и контролировать запросы на изменения. Процесс управления изменениями важен по нескольким ключевым причинам: документированные изменения обеспечивают возможность проведения аудита безопасности; в случае возможного простоя из-за изменения проблема будет быстро

определена; обеспечивается способ координирования изменений таким образом, чтобы одно изменение не влияло на другое изменение;

- процедуры резервного копирования информации и хранения резервных копий вне офиса могут потребоваться из-за требований клиентов и партнеров по бизнесу. Число сотрудников компании, имеющих доступ к резервным данным за пределами компании, должно быть сведено к минимуму. Необходимо регулярно проверять возможность восстановления информации из резервных носителей для проверки целостности резервных копий;
- процедура обработки инцидентов определяет порядок обработки и расследования инцидентов. Необходимо иметь описание порядка реагирования на основные типы инцидентов: сканирование портов, атаки типа «отказ в обслуживании», взлом компьютеров, взлом пароля учетной записи и несоответствующее использование информационных систем.

15.2.4. Пример постановки задачи разработки политики информационной безопасности предприятия

Задача: разработка политики информационной безопасности предприятия.

Основание: целевая программа предприятия «Разработка и реализация мероприятий по обеспечению информационной безопасности объектов информатизации предприятия».

Сроки выполнения: начало работ - по договору. Окончание работ - по договору.

Основные требования к составу работ. Разработка политики информационной безопасности объектов информатизации предприятия.

При разработке политики информационной безопасности объектов информатизации необходимо обеспечить:

- универсальность решений и полноту требований по информационной безопасности действующих информационных и коммуникационных си-

стем объектов информатизации предприятия в соответствии с законодательными и нормативными актами Российской Федерации;

- формирование системы взглядов, требований и решений, обеспечивающих единство, интегрированность и совместимость реализации мероприятий по созданию системы обеспечения информационной безопасности предприятия (создание и развитие удостоверяющего центра, выдающего сертификаты ключей ЭЦП, разработка концепции программно-аппаратного комплекса мониторинга и защиты информационных ресурсов объектов информатизации предприятия от внешних и внутренних угроз информационной безопасности, создание и развитие защищенного центра резервного хранения данных информационных ресурсов предприятия).

При разработке политики информационной безопасности необходимо:

- разработать требования по обеспечению безопасности информационных ресурсов и систем предприятия;
- обследовать защищаемые объекты информатизации;
- разработать проект Технического задания на создание единой системы информационной безопасности предприятия;
- определить перечень первоочередных работ по защите программными и аппаратными способами информационных ресурсов и систем предприятия от несанкционированного доступа, искажения или потери;
- подготовить испытательный стенд для отработки типовых решений в области защиты информации; разработать технико-экономическое обоснование реализации базовых технологий по обеспечению информационной безопасности в рамках проектов планируемого года;
- подготовить и выпустить комплект необходимой документации, представить ее на экспертизу.

Решения, полученные в результате разработки политики информационной безопасности объектов информатизации предприятия, должны обеспечивать:

- возможность создания единой системы информационной безопасности предприятия;
- единство принципов и интеграцию технологических решений по защите информации предприятия при реализации проектов в области защиты информации.

Разработка политики информационной безопасности предприятия должна быть осуществлена с учетом следующих требований:

- существующих общих проблем и тенденций обеспечения информационной безопасности информационно-коммуникационных технологий на основе мирового и отечественного опыта;
- законодательной и нормативной основы обеспечения информационной безопасности информационно-коммуникационных технологий;
- особенностей обеспечения информационной безопасности объектов информатизации предприятия.

В результате разработки политики информационной безопасности предприятия должны быть рассмотрены:

- потенциальные угрозы информационным ресурсам и системам, возможные последствия их реализации;
- требования и методы противодействия угрозам информационной безопасности;
- технологии обеспечения информационной безопасности защищаемых ресурсов и систем;
- функциональные подсистемы и организационные процедуры обеспечения информационной безопасности ресурсов и систем объектов информатизации предприятия;
- органы и процедуры управления деятельностью по обеспечению информационной безопасности;
- вопросы мониторинга и анализа состояния дел в сфере информационной безопасности.

В результате обследования должны быть оценены решения и разработаны типовые требования по обеспечению информационной безопасности предприятия.

Документирование проекта: Отчетная документация оформляется в соответствии с общими требованиями к текстовым документам по действующему ГОСТ.

В процессе выполнения работ Исполнителем разрабатывается следующая документация:

- политика информационной безопасности предприятия;
- итоговый научно-технический отчет;
- предложения по реализации Концепции информационной безопасности предприятия на заданный период (по результатам обследования);
- проект Технического задания на создание комплексной системы информационной безопасности объектов информатизации предприятия;
- требования по обеспечению информационной безопасности объектов информатизации предприятия;
- технико-экономическое обоснование реализации базовых технологий по обеспечению информационной безопасности предприятия.

15.2.5. Особенности разработки политик безопасности в России

Темпы развития современных информационных технологий значительно опережают темпы разработки рекомендательной и нормативно-правовой базы руководящих документов, действующих на территории России. Поэтому решение вопроса о разработке политики информационной безопасности на современном предприятии связано с проблемой выбора критериев и показателей защищенности, а также эффективности корпоративной системы защиты информации. Вследствие этого в дополнение к требованиям и рекомендациям стандартов, Конституции и федеральным законам, руководящим документам Гостехкомиссии (ФСТЭК) России приходится использовать ряд международных рекомендаций. В том числе адаптировать к отече-

ственным условиям и применять на практике методики международных стандартов, а также использовать методики управления информационными рисками в совокупности с оценками экономической эффективности инвестиций в обеспечение защиты информации предприятия.

Современные методики управления рисками позволяют в рамках политик безопасности отечественных компаний поставить и решить ряд задач перспективного стратегического развития:

- количественно оценить текущий уровень информационной безопасности предприятии;
- разработать политику безопасности и планы совершенствования корпоративной системы защиты информации с целью достижения приемлемого уровня защищенности информационных активов компании.

Для этого необходимо:

- обосновать и произвести расчет финансовых вложений в обеспечение безопасности на основе технологий анализа рисков, соотнести расходы на обеспечение безопасности с потенциальным ущербом и вероятностью его возникновения;
- выявить и провести первоочередное блокирование наиболее опасных уязвимостей до осуществления атак на уязвимые ресурсы;
- определить функциональные отношения и зоны ответственности при взаимодействии подразделений и должностных лиц по обеспечению информационной безопасности компании, создать необходимый пакет организационно-распорядительной документации;
- разработать и согласовать со службами организации, надзорными органами проект внедрения необходимых комплексов защиты, учитывающий современный уровень и тенденции развития информационных технологий;
- обеспечить поддержание внедренного комплекса защиты в соответствии с изменяющимися условиями работы организации, регулярными доработками организационно-распорядителъной документации, модификацией технологических процессов и модернизацией технических средств защиты.

Решение названных задач политик безопасности открывает новые широкие возможности перед должностными лицами разного уровня.

Руководителям верхнего звена это поможет объективно и независимо оценить текущей уровень информационной безопасности компании, обеспечить формирование единой стратегии безопасности, рассчитать, согласовать и обосновать затраты на защиту компании. На основе полученной оценки начальники отделов и служб смогут выработать и обосновать необходимые организационные меры (состав и структуру службы информационной безопасности, положение о коммерческой тайне, пакет должностных инструкций и инструкции по действиям в нештатных ситуациях).

Менеджеры среднего звена смогут обоснованно выбрать средства защиты информации, а также адаптировать и использовать в своей работе количественные показатели оценки информационной безопасности, методики оценки и управления безопасностью с привязкой к экономической эффективности компании.

Практические рекомендации по нейтрализации и локализации выявленных уязвимостей системы, полученные в результате аналитических исследований, помогут в работе над проблемами информационной безопасности на разных уровнях и, что особенно важно, помогут определить основные зоны ответственности , в том числе материальной, за ненадлежащее использование информационных активов компании. При определении масштабов материальной ответственности за ущерб, причиненный работодателю, в том числе связанный с разглашением коммерческой тайны, следует руководствоваться положениями Трудового кодекса РФ.

В соответствии с Федеральным законом «Об информации, информатизации и защите информации» целями защиты информации являются в том числе: предотвращение утечки, хищения, утраты, искажения, подделки информации ; предотвращение несанкционированных действий по уничтожению, модификации, искажению, копированию, блокированию информации; предотвращение других форм незаконного вмешательства в информацион-

ные ресурсы и информационные системы.

Поэтому главной целью политик безопасности отечественных компаний является обеспечение устойчивого функционирования предприятия: предотвращение угроз его безопасности, защита законных интересов владельца информации от противоправных посягательств, в том числе уголовно наказуемых деяний в рассматриваемой сфере отношений, предусмотренных Уголовным кодексом РФ, обеспечение нормальной производственной деятельности всех подразделений объекта. Другая задача политик безопасности сводится к повышению качества предоставляемых услуг и гарантий безопасности имущественных прав и интересов клиентов.

Для этого необходимо:

- отнести информацию к категории ограниченного доступа (коммерческой тайне);
- прогнозировать и своевременно выявлять угрозы безопасности информационным ресурсам, причины и условия, способствующие нанесению финансового, материального и морального ущерба, нарушению нормального функционирования и развития ресурсов;
- создать условия функционирования с наименьшей вероятностью реализации угроз безопасности информационным ресурсам и нанесения различных видов ущерба;
- создать механизм и условия оперативного реагирования на угрозы информационной безопасности и проявления негативных тенденций в функционировании автоматизированной системы, а также пресечения посягательств на ресурсы на основе правовых, организационных и технических мер и средств обеспечения безопасности;
- создать условия для максимально возможного возмещения и локализации ущерба, наносимого неправомерными действиями физических и юридических лиц и тем самым ослабить негативное влияние последствий нарушения информационной безопасности.

Для создания эффективных политик безопасности отечественных ком-

паний предлагается первоначально провести анализ рисков в области информационной безопасности. Затем определить оптимальный уровень риска для предприятия на основе заданного критерия. Политику безопасности и соответствующую корпоративную систему защиты информации предстоит построить таким образом, чтобы достичь заданного уровня риска.

Важно помнить, что прежде чем внедрять какие-либо решения по защите информации, необходимо разработать политики безопасности, адекватные целям и задачам современного предприятия. В частности, политики безопасности должны описывать порядок предоставления и использования прав доступа пользователей, а также требования отчетности пользователей за свои действия в вопросах безопасности. Система информационной безопасности окажется эффективной, если она будет надежно поддерживать выполнение правил политик безопасности, и наоборот. Этапы построения требуемых политик безопасности – это внесение в описание объекта автоматизации структуры ценностей, проведение анализа риска, определение правил для любого процесса пользования данным видом доступа к ресурсам объекта автоматизации. При этом политики безопасности желательно оформить в виде отдельных документов и утвердить у руководства компании.

15.3. Аудит безопасности корпоративных систем Интенет/Интранет

15.3.1. Понятие аудита безопасности

Понятие аудит информационной безопасности компании появилось сравнительно недавно. Примерно с 1995 года в ряде высокотехнологичных стран мира, главным образом в США, Великобритании, Германии и Канаде, проводятся ежегодные слушания и совещания специально созданных комитетов и комиссий по вопросам аудита информационной безопасности корпоративных систем. Подготовлено более десятка различных стандартов и спецификаций, посвященных аудиту информационной безопасности, среди которых наибольшую известность приобрели международные стандарты ISO

17799 (BS 7799), BSI и COBIT.

В настоящее время аудит информационной безопасности корпоративных систем Интернет/Интранет представляет собой одно из наиболее актуальных и динамично развивающихся направлений стратегического и оперативного менеджмента в области безопасности корпоративных систем Интернет/Интранет.

Основная задача аудита безопасности – объективно оценить текущее состояние информационной безопасности компании, а также ее адекватность поставленным целям и задачам бизнеса с целью увеличения эффективности и рентабельности экономической деятельности компании.

Аудит информационной безопасности корпоративной системы Интернет/Интранет – это системный процесс получения объективных качественных и количественных оценок о текущем состоянии информационной безопасности компании в соответствии с определенными критериями и показателями безопасности.

В настоящее время возрос объем конфиденциальных данных, обрабатываемых в корпоративной информационной системе Интернет/Интранет. В связи с этим актуальность аудита информационной безопасности резко также возрастает. Можно выделить две основные причины роста уязвимости корпоративных систем Интернет/Интранет.

Во-первых, повысилась уязвимость собственно корпоративных информационных систем за счет обоснованного усложнения аппаратно-программных элементов этих систем, увеличения структурной и функциональной сложности системного и прикладного программного обеспечения, применения новых технологий обработки, передачи и хранения данных.

Во-вторых, расширился спектр угроз корпоративным информационным системам из-за передачи информации по открытым каналам сетей общего назначения, «информационных войн и электронных диверсий» конкурирующих организаций, активного промышленного шпионажа с привлечением профессионалов в области IT-security и пр.

Есть два варианта построения системы информационной безопасности:

- полная замена системы корпоративной защиты информации, требующая больших капиталовложений;
- модернизация существующих систем безопасности. Этот вариант является наименее затратным, но несет проблемы совместимости старых, оставляемых из имеющихся аппаратно-программных средств безопасности, и новых элементов системы защиты информации; обеспечения централизованного управления разнородными средствами обеспечения безопасности.

Существенной причиной необходимости проведения аудита безопасности является то, что при модернизации и внедрении новых технологий защиты информации их потенциал полностью не реализуется. Аудит позволяет:

- оценить текущую безопасность функционирования корпоративной информационной системы;
- оценить и прогнозировать риски;
- управлять влиянием рисков на бизнес-процессы компании;
- обоснованно подойти к вопросу обеспечения безопасности ее информационных активов (стратегических планов развития, маркетинговых программ, финансовых и бухгалтерских ведомостей, содержимого корпоративных баз данных).

Важно то, что аудит информационной безопасности ориентирован как на специалистов в области безопасности корпоративных систем Интернет/Интранет, так и на специалистов в области менеджмента. Такой подход устраняет существующее недопонимание специалистов в области информационной безопасности ТОП - менеджерами компании. В данном случае они объединяются в единую команду, ориентированную на повышение экономической эффективности и рентабельности бизнес-деятельности компании.

В настоящее время многие поставщики средств защиты информации декларируют поставку полного, законченного решения в области безопасности корпоративных систем Интернет/Интранет. Однако, в лучшем случае все сводится к проектированию и поставке соответствующего оборудования и

программного обеспечения. При этом фактически не создается корпоративная система безопасности. Для построения такой системы необходимо ответить на следующие вопросы:

- соответствует ли корпоративная система информационной безопасности целям и задачам бизнеса компаний;
- как контролировать реализацию и выполнение политики безопасности в компании;
- когда необходимо провести модернизацию системы безопасности; как обосновать необходимость модернизации и затрат;
- как быстро окупятся инвестиции в корпоративную систему безопасности;
- насколько правильно и корректно сконфигурированы и настроены штатные средства обеспечения информационной безопасности компании;
- как убедиться в том, что существующие в компании средства защиты – межсетевые экраны (firewalls), системы обнаружения вторжений (IDS), антивирусные шлюзы, VPN-шлюзы – эффективно справляются со своими задачами;
- как решаются вопросы обеспечения конфиденциальности, доступности и целостности;
- есть ли необходимость постоянно обучать сотрудников службы информационной безопасности компании, какие бюджетные средства для этого нужны;
- как управлять информационными рисками компании, какие инструментальные средства для этого необходимо задействовать;
- удовлетворяет ли организация информационной безопасности компании требованиям международных стандартов оценки и управления безопасностью, например ISO 15408, ISO 17799 (BS 7799).

Только объективный и независимый аудит безопасности корпоративной системы Интернет/Интранет позволит получить ответы на поставленные вопросы. Такой аудит, который позволит комплексно проверить все основ-

ные уровни обеспечения информационной безопасности компании: нормативно-правовой, организационный, технологический и аппаратно-программный.

15.3.2. Аудит безопасности для корпоративных пользователей

Несмотря на постоянное развитие технологий безопасности Интернет/Интранет, положение дел в практике обеспечения информационной безопасности в российских компаниях вызывает беспокойство. Это вызвано следующими причинами:

- возрастает роль информационных технологий в поддержке бизнес–процессов отечественных компаний, повышаются требования к качеству и безопасности процессов обработки, хранения и передачи данных, возрастает структурная и функциональная сложность корпоративных информационных систем, а следовательно, и возрастает цена ошибок и сбоев информационных систем. Эта причина нейтрализуется способностью компании обеспечить возрастающие требования в области информационной безопасности;
- эволюционное развитие Интернет/Интранет–технологий приводит к появлению все большего числа уязвимостей операционной среды, многочисленных служб и сервисов, а также протоколов TCP/IP, которые на практике ранее были не известны и не изучены, что в свою очередь приводит к росту уязвимости и незащищенности корпоративных информационных систем. Данная проблема решается постоянным отслеживанием и анализом выявленных уязвимостей с целью дальнейшего их оперативного устранения;
- постоянное усложнение компьютерных информационных систем повышает квалификационные требования к обслуживающему персоналу, приводит к усложнению процедур выбора решений и выполнения политики безопасности компании, обеспечивающих приемлемый уровень информационной безопасности при допустимом уровне затрат. Эта проблема решается в рамках кадровой политики и определяется возможностью получения объективной информации о состоянии системы.

Рассмотрим актуальность аудита безопасности для корпоративных пользователей на примере наиболее используемых сервисов, реализованных в компьютерных сетях.

1. Безопасность электронной почты.

Безопасность электронной почты должна обеспечиваться как на уровне администрации сети, так на уровне конечного пользователя. В общем случае пользователь не является специалистом по технологиям Интернет и обучен работать с определенной почтовой программой без понимания деталей того, что происходит во время приема, отправки и чтения сообщений, и не может идентифицировать и анализировать нештатную ситуацию. Служба электронной почты предприятия должна быть организована так, чтобы администратор мог перехватить как можно большее число потенциальных инцидентов еще до начала работы пользователя.

На уровне конечного пользователя опасность для компьютера могут представлять только запущенные на этом компьютере программы, пересылка текстовых сообщений совершенно безвредна, но любая программа, содержащаяся во вложении к письму и неосторожно (либо автоматически) запущенная при его прочтении, может причинить компьютеру вред. Избежать этого можно следуя нескольким простым правилам:

- никогда не конфигурировать свою почтовую программу на автоматическое открытие приложений. При получении письма с вложением его следует извлекать в отдельный файл на диске и проверять антивирусной программой. При получении письма с неизвестным вам типом файла или с неожиданным для данного отправителя приложением, следует до открытия файла попросить у отправителя разъяснений. по поводу этого приложения.

Нужно иметь в виду, что не только .exe - файлы, но и файлы Visual Basic Script (vbs), Microsoft Office (Word, Excel), файлы HTML, Postscript (.ps), Program Information File (.pif) в общем случае являются программами и могут содержать вредоносный код.;

- убедиться, что почтовая программа не опускает расширение и не сокращает слишком длинное имя. Система (по крайне мере, MS Windows) будет интерпретировать файл по его последнему расширению; ни имя файла, ни расширения, предшествующие последнему, значения не имеют, то есть файл Документ.txt.exe будет интерпретирован как исполнимый .exe-файл;
- своевременно обновлять базу данных антивирусной программы и проводить периодическую проверку всех файлов системы. Некоторые современные почтовые серверы производят автоматическую проверку вложений в проходящих через них письмах на наличие вредоносных программ. Следует проконсультироваться у сетевого администратора или провайдера, производится ли такая проверка и какие именно типы приложений проверяются.

Проблема безопасности электронной почты состоит также в возможности фальсификации адреса отправителя в протоколе SMTP, с помощью которого письма пересылаются через Интернет. Частично эту проблему можно решить правильной конфигурацией почтовых серверов, но полностью ликвидировать эту угрозу нельзя.

Фальсификация адреса отправителя относится к типу атак, называемых social engineering. Для борьбы с подобными угрозами всем пользователям сети должна быть четко разъяснена политика безопасности на предприятии и, в частности, то, что администратор *никогда* не попросит пользователя сообщить свой пароль.

Поскольку почтовые сообщения передаются через Интернет в открытом виде, пользователю нужно иметь в виду, что любое отправленное им письмо может быть прочитано злоумышленником, и любое полученное им письмо может оказаться фальсификацией. Эти проблемы могут быть решены только с помощью шифрования сообщений и цифровой подписи.

Еще одна проблема безопасности – открытая передача имени пользователя и пароля при доступе пользователя к серверу электронной почты. Для доступа к серверу почтовая программа пользователя использует действующие версии протокола POP. Для получения почты с сервера пользователь

должен предоставить пароль, который передается через сеть на сервер. Иногда пароль требуется и при отправке сообщения (в этом случае используется протокол SMTP). Перехват пароля позволит злоумышленнику не только свободно читать адресованные пользователю письма или удалять их с сервера до того, как к ним получит доступ адресат.

Часто «почтовый» пароль пользователя совпадает с «системным». Многие пользователи из соображений удобства вообще используют один и тот же пароль для получения почты и входа в различные системы предприятия (WWW-серверы, сеть Windows). Перехватив такой пароль, злоумышленник получит доступ к другим, возможно, лучше защищенным и более ответственным, системам.

Защититься от перехвата пароля можно применяя методы аутентификации, не требующие передачи пароля в открытом виде (APOP, SASL), или шифруя все передаваемые между клиентом и сервером данные (SSL). И POP-клиент, и сервер должны поддерживать используемый протокол APOP или SSL.

На уровне администратора проблема безопасности электронной почты представляется следующим образом.

Система электронной почты предприятия должна быть организована так, чтобы весь почтовый трафик (по крайней мере, весь почтовый трафик между корпоративной сетью и Интернет) проходил через один почтовый сервер.

В этом случае администратор имеет возможность контролировать проходящие сообщения на предмет опасных вложений и блокировать спам.

Известные производители антивирусных программ (например, Лаборатория Касперского) предлагают специальные модули для почтовых серверов, которые производят проверку корреспонденции на наличие вирусов.

Администратор не может предотвратить фальсификацию почтовых сообщений, т.к. нет никаких прямых средств борьбы с этой проблемой. В общем случае только использование цифровой подписи может гарантировать

подлинность сообщения.

Со стороны администратора требует особого внимания обеспечение надежной аутентификации, серьезно затрудняющей возможность перехвата пароля пользователя.

Администратор должен также уделять серьезное внимание предотвращению атак, направленных против почтового сервера. Атаки на почтовый сервер реализуются через почтовое программное обеспечение. Поэтому от администратора требуется своевременное обновление программного обеспечения.

2. Безопасность WWW.

С точки зрения пользователя WWW – это огромная библиотека текстографических документов, распределенных по множеству серверов и связанных друг с другом перекрестными ссылками. При этом не все информационные ресурсы WWW могут быть открыты для всеобщего просмотра.

Для того чтобы ограничить доступ к какому-либо ресурсу, используется аутентификация клиента, то есть «клиент должен предоставить имя пользователя и пароль, прежде чем его запрос будет обслужен HTTP-сервером. Но эта мера не обеспечивает защиту передаваемых данных от перехвата.

Для пользователя WWW опасна также, как и электронная почта, а именно:

- прослушивание передаваемых данных и паролей;
- загрузка и исполнение на компьютере пользователя вредоносных программ: при этом может либо произойти автоматическая загрузка программного кода браузером без ведома пользователя при просмотре последним определенной Web-странницы, либо пользователь находит на каком-либо сайте ссылку на исполнимую программу и загружает ее. В последнем случае нужно понимать, что загруженная программа может содержать любой вредоносный код. Загрузка известной программы с известного сайта, не дает полной гарантии безопасности. Наличие цифровой подписи говорит только о

том, что программа написана определенным автором и не была изменена. Подпись не гарантирует того, что после запуска программа не начнет стирать файлы с жесткого диска. Конечно, программа, подписанная широко известной компанией, вряд ли содержит умышленно введенный вредоносный код, но может содержать ошибки, которые потом могут быть использованы злоумышленником. Что касается программ, содержащих подпись с ничего не говорящей вам фамилией, то от них можно ожидать любых действий. К тому же, если браузер доверяет одной подписанной программе, то он автоматически доверяет всем программам, подписанным тем же автором;

- подлог документов (ресурсов). Технология обмана пользователя, известная также под названием mirror world, состоит в том, что злоумышленник создает на подконтрольном ему сервере копию другого сайта (или его части). После этого злоумышленник обманом заставляет пользователя обратиться к своему серверу. В результате пользователь, полагая, что работает на сайте, скажем, банка, вводит в HTML-форму номер кредитной карты, который немедленно попадает к злоумышленнику. Аутентификация в этом случае не поможет, поскольку она предназначена для защиты ресурсов сервера, а не пользователя. Более того, использование аутентификации для доступа на сфальсифицированный сервер приведет к сдаче пароля злоумышленнику.

15.3.3. Возможности аудита безопасности

Своевременно проведенный аудит безопасности корпоративной системы Интернет/Интранет должен помочь свести воедино существующие меры безопасности, которые в свою очередь могут помочь в борьбе с атаками разного рода злоумышленников.

Системный администратор, исходя из политики сетевой безопасности в своей организации и имея четкое представление о возможных инцидентах и их последствиях, определит, какие меры являются необходимыми и приемлемыми для его сети. Например, такими мерами на уровне TCP/IP могут быть:

- фильтрация на маршрутизаторе;
- анализ сетевого трафика;
- защита маршрутизатора;
- защита хоста;
- превентивное сканирование.

1. Фильтры на маршрутизаторе, соединяющем сеть предприятия с Интернет, применяются для запрета пропуска датаграмм, которые могут быть использованы для атак как на сеть организации из Интернет, так и на внешние сети злоумышленником, находящимся внутри организации.

Для этого необходимо:

- запретить пропуск датаграмм с широковещательным адресом назначения между сетью организации и Интернет;
- запретить пропуск датаграмм, направленных из внутренней сети (сети организации) в Интернет, но имеющих внешний адрес отправителя;
- запретить пропуск датаграмм, прибывающих из Интернет, но имеющих внутренний адрес отправителя;
- на сервере доступа клиентов по коммутируемой линии – разрешить пропуск датаграмм, направленных только с/на IP-адрес, назначенный клиенту.

2. Анализ сетевого трафика проводится для обнаружения атак, предпринятых злоумышленниками, находящимися как в сети организации, так и в Интернет.

Для этого следует:

- сохранять и анализировать статистику работы маршрутизаторов, особенно – частоту срабатывания фильтров;
- применять специализированное программное обеспечение для анализа трафика с целью выявления выполняемых атак. Выявлять узлы, занимающие ненормально большую долю полосы пропускания, и другие аномалии в поведении сети;

- применять специальные программы выявления узлов, использующих нелегальные IP- или MAC-адреса;
- применять специальные программы выявления узлов, находящихся в режиме прослушивания сети.

3. Мероприятия по защите маршрутизатора проводятся с целью предотвращения атак, направленных на нарушение схемы маршрутизации датаграмм или на захват маршрутизатора злоумышленником.

Для осуществления защиты маршрутизатора необходимо:

- использовать аутентификацию сообщений протоколов маршрутизации с помощью специальных алгоритмов;
- осуществлять фильтрацию маршрутов, объявляемых сетями-клиентами, провайдером или другими автономными системами. Фильтрация выполняется в соответствии с маршрутной политикой организации; маршруты, не соответствующие политике, игнорируются;
- отключить на маршрутизаторе все ненужные сервисы;
- ограничить доступ к маршрутизатору консолью или выделенной рабочей станцией администратора, использовать парольную защиту; не использовать Telnet для доступа к маршрутизатору в сети, которая может быть прослушана;
- применять последние версии и обновления программного обеспечения, следить за бюллетенями по безопасности, выпускаемыми производителем.
- и др.

4. Защита хоста проводится с целью предотвращения атак, имеющих целью перехват данных, отказ в обслуживании или проникновение злоумышленника в операционную систему.

Для защиты хоста следует:

- запретить обработку запросов, направленных на широковещательный адрес;
- отключить все ненужные сервисы;

- использовать программы, позволяющие отследить попытки скрытного сканирования;
- применять средства безопасности используемых на хосте прикладных сервисов;
- использовать последние версии и обновления программного обеспечения, следить за бюллетенями по безопасности, выпускаемыми производителем и др.

Администратор сети должен знать и использовать методы и инструменты злоумышленника и проводить превентивное сканирование сети организации для обнаружения слабых мест в безопасности до того, как это сделает злоумышленник. Для этой цели имеется также специальное программное обеспечение – сканеры безопасности.

15.3.4. Практические шаги аудита безопасности

Реализация задач обеспечения безопасности корпоративной системы Интернет/Интранет становится возможной в ходе следующих практических шагов аудита безопасности.

1. Комплексный анализ информационной системы (ИС) предприятия и подсистемы информационной безопасности на методологическом, организационно-управленческом, технологическом и техническом уровнях. Анализ рисков.

1.1. Исследование и оценка состояния информационной безопасности корпоративной информационной системы (КИС) и подсистемы информационной безопасности предприятия.

1.1.1. Комплексная оценка соответствия типовых требований Руководящих документов (РД) Гостехкомиссии РФ системе информационной безопасности предприятия.

1.1.2. Комплексная оценка соответствия типовых требований международных стандартов ISO системе информационной безопасности предприятия.

1.1.3. Комплексная оценка соответствия специальных требований За-

казчика системе информационной безопасности предприятия.

1.2. Работы на основе анализа рисков.

1.2.1. Анализ рисков. Уровень управления рисками на основе качественных оценок рисков.

1.2.2. Анализ рисков. Уровень управления рисками на основе количественных оценок рисков.

1.3. Инструментальные исследования.

1.3.1. Инструментальное исследование элементов инфраструктуры компьютерной сети и корпоративной информационной системы на наличие уязвимостей.

1.3.2. Инструментальное исследование защищенности точек доступа предприятия в Интернет.

1.4. Анализ документооборота предприятия.

2. Разработка комплексных рекомендаций по методологическому, организационно-управленческому, технологическому, общетехническому и программно-аппаратному обеспечению режима информационной безопасности предприятия.

2.1. Разработка концепции обеспечения информационной безопасности предприятия.

2.2. Разработка корпоративной политики обеспечения информационной безопасности предприятия на организационно-управленческом, правовом, технологическом и техническом уровнях.

2.3. Разработка плана защиты предприятия Заказчика.

2.4. Дополнительные работы по анализу и созданию методологического, организационно-управленческого, технологического, инфраструктурного и технического обеспечения режима информационной безопасности предприятия Заказчика.

3. Организационно-технологический анализ ИС предприятия.

3.1. Оценка организационно-управленческого уровня безопасности.

3.1.1. Оценка соответствия типовым требованиям руководящих доку-

ментов РФ к системе информационной безопасности предприятия в области организационно-технологических норм.

3.1.2. Анализ документооборота предприятия категории «конфиденциально» на соответствие требованиям концепции информационной безопасности, положению о коммерческой тайне, прочим внутренним требованиям предприятия по обеспечению конфиденциальности информации.

3.1.3. Дополнительные работы по исследованию и оценке информационной безопасности объекта.

3.2. Разработка рекомендаций по организационно-управленческому технологическому, общетехническому обеспечению режима информационной безопасности предприятия.

3.2.1. Разработка элементов концепции обеспечения информационной безопасности предприятия.

3.2.2. Разработка элементов корпоративной политики обеспечения информационной безопасности предприятия на организационно-управленческом, правовом и технологическом уровнях.

4. Экспертиза решений и проектов.

4.1. Экспертиза решений и проектов автоматизации на соответствие требованиям по обеспечению информационной безопасности экспертно-документальным методом.

4.2. Экспертиза проектов подсистем информационной безопасности на соответствие требованиям по безопасности экспертно-документальным методом.

5. Работы по анализу документооборота и поставке типовых комплектов организационно-распорядительной документации.

5.1. Анализ документооборота предприятия категории «конфиденциально» на соответствие требованиям концепции информационной безопасности, положению о коммерческой тайне, прочим внутренним требованиям предприятия по обеспечению конфиденциальности информации.

5.2. Поставка комплекта типовой организационно-распорядительной

документации в соответствии с рекомендациями корпоративной политики информационной безопасности предприятия на организационно-управленческом и правовом уровне.

6. Работы, поддерживающие практическую реализацию плана защиты.

6.1. Разработка технического проекта модернизации средств защиты КИС, установленных у Заказчика по результатам проведенного комплексного аналитического исследования корпоративной сети.

6.2. Разработка системы поддержки принятия решений на предприятии Заказчика по обеспечению информационной безопасности предприятия на основе CASE-систем и программных СППР.

6.3. Подготовка предприятия к аттестации.

6.3.1. Подготовка «под ключ» предприятия к аттестации объектов информатизации заказчика на соответствие требованиям РД РФ.

6.3.2. Подготовка предприятия к аттестации КИС на соответствие требованиям по безопасности международных стандартов ISO 15408, ISO 17799, стандарта ISO 9001 при обеспечении требований информационной безопасности предприятия.

6.4. Разработка организационно-распорядительной и технологической документации.

6.4.1. Разработка расширенного перечня сведений ограниченного распространения как части политики безопасности.

6.4.2. Разработка пакета организационно-распорядительной документации (ОРД) в соответствии с рекомендациями корпоративной политики информационной безопасности (ИБ) предприятия на организационно-управленческом и правовом уровне.

6.4.3. Поставка комплекта типовой организационно-распорядительной документации в соответствии с рекомендациями корпоративной политики ИБ предприятия на организационно-управленческом и правовом уровнях.

7. Повышение квалификации и переподготовка специалистов.

7.1. Тренинги в области организационно-правовой составляющей защиты информации.

7.2. Обучение основам экономической безопасности.

7.3. Тренинги в области технологии защиты информации.

7.4. Тренинги по применению продуктов (технических средств) защиты информации .

7.5. Обучение действиям при попытке взлома информационных систем.

7.6. Обучение и тренинги по восстановлению работоспособности системы после нарушения штатного режима ее функционирования, а также по восстановлению данных и программ из резервных копий.

8. Сопровождение системы информационной безопасности после проведенного комплексного анализа или анализа элементов системы ИБ предприятия.

9. Ежегодная переоценка состояния ИБ.

Здесь под термином аудит информационной безопасности корпоративной системы Интернет/Интранет понимается системный процесс получения объективных качественных и количественных оценок о текущем состоянии информационной безопасности компании в соответствии с определенными критериями и показателями безопасности на всех основных уровнях обеспечения безопасности: методологическом, организационно-управленческом, технологическом и техническом. Таких оценок, которые позволяют выработать практические рекомендации по управлению и обеспечению информационной безопасности компании, адекватные поставленным целям и задачам развития бизнеса.

В целом независимо от своей разновидности, состава и объема аудит безопасности корпоративной системы Интернет/Интранет должен позволить решить следующие актуальные задачи каждой проверяемой компании:

- обеспечить (при необходимости повысить) информационную безопасность предприятия;

- снизить потенциальные потери предприятия путем повышения устойчивости функционирования корпоративной сети;
- защитить конфиденциальную информацию, передаваемую по открытым каналам связи;
- защитить информацию от умышленного искажения (разрушения), несанкционированного копирования, доступа или использования;
- обеспечить контроль действий пользователей в корпоративной сети предприятия;
- своевременно оценить и переоценить информационные риски бизнес - деятельности компании;
- выработать оптимальные планы развития и управления предприятием.

15.4. Проектирование системы обеспечения информационной безопасности предприятия

Проектирование системы обеспечения информационной безопасности предприятия может состоять из следующих этапов.

1. Построение профиля защиты.

На этом этапе разрабатывается план проектирования системы защиты информационной среды предприятия. Производится оценка доступных средств, осуществляется анализ и планирование разработки и интеграции средств защиты. Необходимым элементом работы является утверждение допустимого риска объекта защиты.

Обеспечение повышенных требований к информационной безопасности предполагает соответствующие мероприятия на всех этапах жизненного цикла информационных технологий. Планирование этих мероприятий производится по завершении этапа анализа рисков и выбора контрмер. Обязательной составной частью этих планов является периодическая проверка соответствия существующего режима ИБ политике безопасности, сертификация информационной системы (технологии) на соответствие требованиям опреде-

ленного стандарта безопасности.

Работа по построению плана защиты объекта начинается с построения профиля защиты данного объекта. При этом часть этой работы уже была проделана при проведении анализа рисков.

2. Формирование организационной политики безопасности.

Организационная политика безопасности описывает порядок предоставления и использования прав доступа пользователей, а также требования отчетности пользователей за свои действия в вопросах безопасности.

Система информационной безопасности (СИБ) объекта окажется эффективной, если она будет надежно поддерживать выполнение правил политики безопасности, и наоборот. Шагами построения организационной политики безопасности являются:

- внесение в описание объекта автоматизации структуры ценности и проведение анализа риска;
- определение правил для любого процесса пользования данным видом доступа к ресурсам объекта автоматизации, имеющим данную степень ценности. Организационная политика безопасности оформляется в виде отдельного документа, который согласовывается и утверждается на предприятии.

3. Оформление условий безопасного использования информационных технологий (ИТ).

Система обеспечения безопасности предприятия, соответствующая выбранному профилю защиты, может обеспечивать требуемый уровень безопасности только в том случае, если она установлена, управляется и используется в соответствии с выработанными правилами. Операционная среда должна управляться согласно принятой для данного профиля защиты нормативной документации, а также инструкциям администраторов и пользователей.

Выделяются следующие виды условий безопасного использования ИТ:

- физические условия;
- условия для персонала;
- условия соединений.

Физические условия касаются размещения ресурсов объекта, а также защиты аппаратных средств и программного обеспечения, критичных к нарушению политики безопасности.

Условия для персонала содержат организационные вопросы управления безопасностью и отслеживания полномочий пользователей.

Условия соединений не содержат явных требований для сетей и распределенных систем, но, например, условие равенства положения означает наличие единой области управления всей сетью объекта.

Условия безопасного использования объекта автоматизации оформляются в виде отдельного документа, который согласовывается и утверждается на предприятии.

4. Формулирование целей безопасности объекта.

В этом разделе профиля защиты дается детализованное описание общей цели построения системы безопасности предприятия, выражаемое через совокупность факторов или критериев, уточняющих цель. Совокупность факторов служит базисом для определения требований к системе (выбор альтернатив).

Факторы безопасности, в свою очередь, могут распределяться на технологические, технические и организационные.

5. Определение функциональных требований безопасности.

Функциональные требования профиля защиты определяются на основе набора хорошо известных, отработанных и согласованных функциональных требований безопасности. Все требования к функциям безопасности можно разделить на два типа: управление доступом к информации и управление потоками информации.

На этом этапе предстоит правильно определить для объекта компонен-

ты функций безопасности. Компонент функции безопасности описывает определенный набор требований безопасности – наименьший выбираемый набор для включения в профиль защиты. Между компонентами могут существовать зависимости.

6. Требования гарантии достигаемой защищенности.

Структура требований гарантии аналогична структуре функциональных требований и включает классы, семейства, компоненты и элементы гарантии, а также уровни гарантии. Классы и семейства гарантии отражают такие вопросы, как разработка, управление конфигурацией, рабочая документация, поддержание этапов жизненного цикла, тестирование, оценка уязвимости и другие вопросы.

Требования гарантии достигаемой защиты выражаются через оценки функций безопасности СИБ объекта. Оценка силы функции безопасности выполняется на уровне отдельного механизма защиты, а ее результаты позволяют определить относительную способность соответствующей функции безопасности противостоять идентифицированным угрозам. Исходя из известного потенциала нападения, сила функции защиты определяется, например, категориями «базовая», «средняя», «высокая».

Потенциал нападения определяется путем экспертизы возможностей, ресурсов и мотивов побуждения нападающего.

Уровни гарантии. Предлагается использовать табличную сводку уровней гарантированности защиты. Уровни гарантии имеют иерархическую структуру, где каждый следующий уровень предоставляет большие гарантии и включает все требования предыдущего.

7. Формирование перечня требований.

Перечень требований к системе информационной безопасности (СИБ), Эскизный проект, План защиты (далее – техническая документация, ТД) содержат набор требований безопасности информационной среды предприятия, которые могут ссылаться на соответствующий профиль защиты, а также со-

держать требования, сформулированные в явном виде.

В общем виде разработка ТД включает:

- уточнение функций защиты;
- выбор архитектурных принципов построения СИБ;
- разработку логической структуры СИБ (четкое описание интерфейсов);
- уточнение требований функций обеспечения гарантоспособности СИБ;
- разработка методики и программы испытаний на соответствие сформулированным требованиям.

8. Оценка достигаемой защищенности.

На этом этане производится оценка меры гарантии безопасности информационной среды объекта автоматизации. Мера гарантии основывается на оценке, с которой после выполнения рекомендованных мероприятий можно доверять информационной среде объекта.

Базовые положения методики должны предполагать, что степень гарантии следует из эффективности усилий при проведении оценки безопасности. Увеличение усилий оценки предполагает:

- значительное число элементов информационной среды объекта, участвующих в процессе оценивания;
- расширение типов проектов и описаний деталей выполнения при проектировании системы обеспечения безопасности;
- строгость, заключающуюся в применении большего числа инструментов поиска и методов, направленных на обнаружение менее очевидных уязвимостей или на уменьшение вероятности их наличия.

Вопросы для самоконтроля

1. Что является целью системы обеспечения информационной безопасности?

2. Перечислите основные задачи системы информационной безопасности предприятия.

3. Дайте определение термину аудит информационной безопасности корпоративной системы Интернет/Интранет.

4. В чем заключается основная задача аудита безопасности?

5. Назовите основные причины роста уязвимости корпоративных систем Интернет/Интранет.

6. Назовите причины, подтверждающие актуальность аудита безопасности российских компаний.

7. Перечислите основные проблемы безопасности электронной почты для конечного пользователя.

8. Назовите рекомендательные меры, повышающие уровень безопасности электронной почты для конечного пользователя.

9. В чем состоит суть проблемы безопасности электронной почты на уровне администратора?

10. Какие опасности существуют для пользователя информационных ресурсов WWW?

11. Перечислите меры, необходимые для обеспечения безопасности сети.

12. Назовите меры, необходимые для обеспечения безопасности хоста.

13. Охарактеризуйте основные практические шаги аудита безопасности.

14. Из каких этапов состоит проектирование системы обеспечения информационной безопасности предприятия?

15. Что понимается под политикой информационной безопасности компании?

16. Назовите основные причины создания политик безопасности.

17. Какими принципами следует руководствоваться при разработке политик безопасности?

18. Какие требования предъявляются к защите конфиденциальной информации в органах исполнительной власти?

19. Перечислите средства защиты информации, используемые в коммерческих структурах.

20. Какие политики информационной безопасности выделяют в настоящее время?

Контрольные тесты

№ п/п	Вопрос	Возможные ответы
1.	Программными средствами для защиты информации в компьютерной сети являются: а) Firewall б) Brandmauer в) Sniffer г) Backup	• в, г • а, г • а, б • б, в
2.	Протокол SSL (Secure Socket Layer) является системой шифрования…	• с симметричными ключами • с открытыми ключами • на основе метода гаммирования • на основе методов случайного шифрования
3.	Формальное изложение правил поведения лиц, получающих доступ к конфиденциальным данным в корпоративной информационной системе – это…	• аудит безопасности • политика информационной безопасности • инструкция по безопасности
4.	Что такое VPN?	• виртуальная частная сеть • протокол передачи данных • метод шифрования
5.	Средством обеспечения информационной безопасности является …	• USB-порт • межсетевой экран • BSI
6.	Системным процессом получения объективных качественных и количественных оценок о текущем состоянии информационной безопасности компании в соответствии с определенными критериями и показателями безопасности называется	• политика информационной безопасности • аудит безопасности • проектирование информационной базы данных
7.	Межсетевые экраны позволяют организовать защиту на следующих уровнях:	• на прикладном • на сетевом • на гостевом • доступа к сети в целом
8.	При разработке политики безопасности используются следующие модели доверия:	• доверять всем и всегда • доверять избранным и всегда • не доверять никому и никогда

		• доверять избранным на время
9.	Управление паролями, антивирусная защита, организация удаленного доступа это:	• этапы проектирования информационной системы • политики информационной безопасности • этапы разработки сервисного программного обеспечения
10.	Определяет права и ответственность сотрудников компании за надлежащую защиту конфиденциальной информации следующая политика:	• политика управления паролями • политика допустимого использования • политика безопасности периметра
11.	В чем отличие политики безопасности от процедуры безопасности?	• политика безопасности определяет, что должно быть защищено, а процедура – как защитить информационные ресурсы • процедура безопасности определяет, что должно быть защищено, а политика – как защитить информационные ресурсы • между ними нет различий
12.	К процедурам безопасности относятся:	• процедура управления конфигурацией • процедура резервного копирования • процедура обработки инцидентов • процедура гибернации
13.	Построение профиля защиты, оформление условий безопасного использования информационных технологий, формулирование целей безопасности объекта относятся к этапам:	• разработки автоматизированной информационной технологии • проектирования системы обеспечения информационной безопасности предприятия • жизненного цикла автоматизированной информационной системы
14.	К средствам защиты от несанкционированного доступа периметра сети и основных компонент автоматизированных систем относятся:	• средства межсетевого экранирования • концентраторы • репитеры • маршрутизаторы
15.	Межсетевые экраны не позволяют…	• организовать защиту на уровне доступа к компонентам и сети в целом • выполнять контроль IP-адресов

		• организовать защиту трафика данных, передаваемых по открытым каналам связи
16.	IDS (Intrusion Detection Systems) – это…	• средства обнаружения вторжений, позволяющие контролировать состояние безопасности сети в реальном времени • средства аутентификации пользователей • средства построения виртуальных частных сетей
17.	Внутренняя сеть организации, построенная на использовании протокола IP для обмена и совместного использования некоторой части информации внутри этой организации – это…	• Internet • Intranet • Fidonet • Ethernet
18.	Что не относится к видам условий безопасного использования ИТ?	• физические условия • условия для персонала • условия соединений • эргономические условия
19.	К политикам информационной безопасности относятся…	• административно-ресурсная, общая тактическая • общая стратегическая, частная тактическая • системно-аналитическая, информационно-стратегическая
20.	Среди политик информационной безопасности компании различают:	• общая стратегическая • частная тактическая • общеадминистративная • инновационная

ГЛОССАРИЙ

Алгебра логики	Математический аппарат, с помощью которого записывают, вычисляют, упрощают и преобразовывают логические высказывания.
Алгоритм	Точное предписание, определяющее процесс, ведущий от варьируемых начальных данных к искомому результату.
Алгоритмические языки программирования	Включают группу языков, предназначенных для отражения структуры алгоритма и независящих от архитектуры компьютера (*Паскаль, Фортран, Бейсик* и др.).
Аналоговые вычислительные машины	Вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).
Антивирусные программы	Комплекс программ для предотвращения заражения компьютера вирусами и ликвидации последствий заражения.
Аппарат булевой алгебры	Формальная математическая система состоит из трех множеств: *элементов, операций* над ними и а*ксиом*. Наименьшим элементом алгебры логики является ***0***, наибольшим элементом - ***1***. Основные операции булевой алгебры: И (AND), ИЛИ (OR) и НЕ (NOT).
Аппаратная платформа ЭВМ	Совокупность технических средств, определяющих среду функционирования конкретных программ обработки данных. В основу аппаратной платформы были положены совокупность интерфейсной системы передачи данных и тип используемого процессора.
Аппаратное обеспечение вычислительной системы	Совокупность входящих в состав системы аппаратных средств, необходимых для ее функционирования. Аппаратное обеспечение составляет компьютер, внешние устройства, линии связи и т. д. Они технически обеспечивают эффективную работу системы, ее способность предоставлять пользователю определенные виды обслуживания.
Аппаратные средства вычислительной системы (ВС)	Электрические , электронные и механические схемы, блоки, приборы и устройства, составляющие материальную часть ВС Аппаратными средствами являются, например, компьютер и микросхемы, его составляющие. К аппаратным средствам относятся дисплей, дисковод, принтер и прочие.
Аппаратные средства ПК	Представляют собой совокупность электронных, электромеханических, электромагнитных и электронно-оптических устройств. Каждое устройство выполняет определенный набор функций, определяемых комбинацией входных управляющих электрических сигналов – команд.
Арифметико-логическое устройство	Часть процессора, предназначенная для выполнения арифметических и логических операций над данными

Архитектура компьютера	С точки зрения пользователя – совокупность основных характеристик компьютера, таких как система команд, организация памяти, система адресации, операции ввода/вывода, управления и т. п. Компьютеры , имеющую одинаковую архитектуру, с точки зрения программиста, являются совместимыми.
Базовая система ввода-вывода BIOS (Basic Input Output System)	Комплект программ, находящихся в ПЗУ. Программы **BIOS** проводят проверку основных систем компьютера после включения, обеспечивают взаимодействие с клавиатурой и монитором, выполняют проверку дисководов, позволяют выполнить некоторые настройки чипа материнской платы и процессора.
Базовое программное обеспечение (*Base Software*)	Минимальный набор программных средств, обеспечивающих работу компьютера.
Байт	Наименьшая единица памяти компьютера, равная 8 битам, или 8-значному двоичному числу: **1 байт = 8 бит.** Одним байтом можно закодировать 256 объектов.
Бит	Минимальная единица количества информации, равная одному двоичному разряду.
Буфер	Запоминающее устройство для временного хранения данных и согласования скоростей взаимодействия устройств с разными возможностями.
Векторный метод кодирования цветного изображения	Изображение представляется в виде совокупности линий и кривых. С помощью этой технологии описываются различные шрифты, поддерживаемые современными принтерами и мониторами.
Внешние устройства	Комплекс устройств, обеспечивающих эффективное взаимодействие компьютера с пользователями, объектами управления, другими машинами. В состав внешних устройств входят внешняя память и устройства ввода-вывода.
Внешние устройства компьютера	Устройства, обеспечивающие ввод и вывод данных из основных устройств компьютера (устройства ввода-вывода) и долговременное хранение информации, не обрабатываемой процессором в данный момент времени.
Внешняя память	Память, к содержимому которой можно обратиться только при помощи операций ввода/вывода. Внешняя память реализуется набором внешних запоминающих устройств вычислительной системы.
Внутренняя память	Память компьютера, непосредственно связанная с центральным процессором.
Высказывание	Некоторое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.
Вычислительная система	Система из разнообразного состава компьютеров, процессоров, программного обеспечения и периферийного оборудования, организована для совместного выполнения информационно-вычислительных процессов.
Гибридные вычислительные машины	Вычислительные машины комбинированного действия, работают с информацией, представленной и в цифровой и в аналоговой форме.
Данные	Факты, которые выступают в качестве средства представ-

	ления информации и обеспечивают возможность хранения, передачи и обработки информации.
Двоичная система счисления	Позиционная система счисления, состоящая из 2-х разных цифр: **0, 1,**
Десятичная система счисления	Позиционная система счисления, состоящая из 10 разных цифр: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9.**
Дополнительный код двоичного числа	В *n*-разрядном двоичном коде инвертировать все разряды (все его 0 заменить на 1, а 1 заменить на 0), затем в полученном числе прибавить 1 к младшему разряду. Используется для сложения отрицательных чисел в арифметико-логическом устройстве процессора.
Драйвер	Программа, расширяющая возможности операционной системы. Драйвер служит для управления работой периферийных устройств компьютера: дисководов, дисплеем, клавиатурой, принтером, манипулятором “мышь” и др.
Емкость памяти	Максимальное количество информации, которое может храниться в запоминающем устойстве.
Жесткий диск, Винчестер (HDD-Hard Disc Drive)	Основное устройство для долговременного хранения больших объемов информации: данных и программ. К основным параметрам жестких дисков относятся емкость и производительность.
Инструментальное программное обеспечение	Средство разработки и развития программного обеспечения.
Инструментарий технологии программирования	Программные продукты поддержки (обеспечения) технологии программирования, включают следующие группы программных продуктов: средства для создания приложений; средства для создания информационных систем(CASE-технологии); локальные средства и интегрированные среды разработчиков программ.
Интегрированные среды разработки программ	Комплексы программ, предназначенные для повышения производительности труда программистов путем автоматизации создания кодов программ, обеспечивающих интерфейс пользователя графического типа; автоматизации разработки приложений для архитектуры клиент-сервер; автоматизации создания запросов и отчетов.
Интерпретатор	Программа, которая последовательно преобразуетт каждый отдельный оператор входной программы в в машинный код и сразу его выполняет.
Интерфейс компьютера	Представляет собой совокупность стандартизованных аппаратных и программных средств, обеспечивающих обмен информацией (сигналами) между устройствами. Наличие стандартных интерфейсов позволяет унифицировать передачу информации в виде сигналов между устройствами независимо от их особенностей.
Информатизация общества	Организованный, социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования инфор-

	мационных ресурсов.
Информатика	Комплексная научно-техническая дисциплина, занимающаяся изучением структуры, общих свойств информации и информационных процессов, разработкой на этой основе информационной техники и технологии, а также решением научных и инженерных проблем создания, внедрения и эффективного использования компьютерной техники и технологии во всех сферах человеческой деятельности
Информационная емкость	Способность запоминающего устройства разместить определенное количество информации. Измеряется максимальным количеством единиц данных (битов, байтов и т. д.), которое может храниться в запоминающем устройстве.
Информационная культура	Умение целенаправленно работать с информацией и использовать ее для получения, обработки и передачи данных, применяя компьютерную информационную технологию, современные технические и программные средства.
Информационное общество	Общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формы.
Информационные объекты	Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств.
Информационные процессы	Процессы, связанные с определенными операциями над информационными объектами.
Информационные ресурсы	Отдельные документы и отдельные массивы документов, а так же документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных).
Информационные услуги	Услуги, связанные с получением и предоставлением в распоряжение пользователя информационных продуктов.
Информационный потенциал общества	Совокупность средств, методов и условий, позволяющих использовать информационные ресурсы.
Информационный продукт	Совокупность данных, сформированная производителем для распространения в вещественной или невещественной форме.
Информация	Сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают неполноту знаний о них, степень неопределенности.
Каналы связи (внутримашинный интерфейс)	Служат для сопряжения центральных узлов машины с е внешними устройствами.
Качество информации	Совокупность свойств экономической информации, характеризующих степень ее соответствия потребностям пользователей. Так при принятии управленческих решений учитываются следующие свойства: содержательность, репрезентативность, доступность, достаточность, актуальность, своевременность, точность, достоверность и др.
Кмоп-память, CMOS-память	Микросхема энергонезависимой памяти, устанавливается на материнской плате. Характеризуется высокой плотностью размещения элементов, высокой скоростью и низким потреблением энергии. Данные в память можно заносить и стирать самостоятельно. Содержимое памяти не стирается после выключения компьютера. Микросхема постоянно

	подпитывается от небольшой батарейки, расположенной на материнской плате. Часто используется в генераторе тактовой частоты.
Код	Правило отображения одного набора объектов или знаков в другой набор знаков без потери информации.
Код ASCII (American Standard Code for Information Interchange)	Семиразрядный код, обеспечивающий 128 различных битовых комбинаций, самая распространенная и универсальная компьютерная кодовая таблица. Расширенная таблица относится к символам с номерами от 128 до 255 и может отличаться на компьютерах разного типа.
Кодирование	Представление, моделирование одного набора знаков другим с помощью кода.
Кодирование целых чисел со знаком	Способ представление данных в компьютере. Целые числа со знаком обычно занимают в памяти компьютера один, два или четыре байта. Обычно используются три формы кодирования: *прямой код, дополнительный код, обратный код*.
Кодовая таблица	Соответствие между набором знаков и их кодами, обычно разными числами.
КОИ-8	Код обмена информацией восьмизначный. Используется в компьютерных сетях на территории России и в некоторых службах российского сектора Интернета, является стандартом в сообщениях электронной почты и телеконференций.
Команда	Это инструкция, предписывающая компьютеру выполнять ту или иную операцию (умножить два числа, записать данные на диск и т.д.).
Компилятор	Программа, формирующая полный текст программы в машинные коды. После окончания процесса компиляции программа готова к выполнению.
Компьютеризация общества	Направление развития общества при котором основное внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение результатов переработки и накопление информации.
Компьютеры высокой производительности	Одно- или многопроцессорные машины, предназначенные для индивидуального применения при решении задач повышенной сложности либо при обслуживании локальных или региональных компьютерных сетей.
Компьютеры ординарной производительности	Однопроцессорные машины, служат для решения несложных задач индивидуальных пользователей или работают в составе небольших компьютерных сетей. Они относятся к классу массовых персональных компьютеров.
Компьютеры сверхординарной (сверхвысокой) производительности	Многопроцессорные машины или многомашинные вычислительные комплексы, целью эксплуатации которых является решение задач большой сложности (метеорология, управление космическими объектами, моделирование микро- и макроэкономических процессов, обслуживание больших компьютерных сетей и др.).
Кэш память	«Сверхоперативная» буферная память, предназначена для промежуточного хранения наиболее часто используемых процессором данных. Кэш-память служит для частичной компенсации разницы в скорости процессора и основной памяти.
Лисп	Язык функционального программирования, разработанный

	для обработки символьной информации и исследований по проблематике искусственного интеллекта.
Логическая схема И	Схема реализует **конъюнкцию** двух или более логических значений. Если хотя бы на одном входе ***схемы* И** появится нуль, то на выходе будет тоже нуль (ложь). Если на всех входах *схемы И* появится единица, то на выходе будет тоже единица (истина).
Логическая схема ИЛИ	Схема реализует **дизъюнкцию** двух или более логических значений. Когда хотя бы на одном входе *схемы ИЛИ* будет единица, на ее выходе также будет единица (истина).
Логический подход к структуре экономической информации	Связан со смысловым содержанием информации, характеризуется наличием следующих единиц измерения: реквизит, показатель, экономический документ, информационный массив, информационный поток, информационная система.
Логический элемент ПК	Часть электронной логической схемы, которая реализует электронную логическую функцию.
Локальные средства разработки программ	Совокупность языков и систем программирования плюс инструментальная среда пользователя
Математический сопроцессор	Специализированная интегральная схема, работающая во взаимодействии с центральным процессором и предназначенная для выполнения математических операций с плавающей точкой
Материнская плата	Основная плата компьютера. На ней размещаются: процессор, микропроцессорный комплект (чипсет), шины, оперативная память, ПЗУ, слоты.
Машинное слово	Наибольшая последовательность бит, которую ЭВМ может обрабатывать как единое целое**.** Длина машинного слова зависит от разрядности процессора и может быть равной 16, 32 битам и т.д. Для кодирования символов достаточно одного байта. При этом можно представить 256 символов (с десятичными кодами от 0 до 255).
Машинно-ориенированные языки программирования	Языки, которые отражают структуру конкретного типа компьютера (*Ассемблер*).
Машинные языки программирования	Машинные коды, воспринимаемые аппаратной частью компьютера.
Микропроцессор	Процессор, выполненный в одном или нескольких взаимосвязанных полупроводниковых кристаллах интегральных схем
Многомашинная вычислительная система	Вычислительная система, построенная на основе нескольких компьютеров.
Непозиционная система счисления	Система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от местоположения (позиции) в изображении числа. Римская система счисления является непозиционной.
Обратный код двоичного числа	В *n*-разрядном двоичном коде инвертировать все разряды (все его 0 заменить на 1, а 1 заменить на 0). Используется для представления отрицательных чисел. Все операции с отрицательными числами выполняются в формате машинного слова (4 байта). Поэтому к двоичному числу слева дописываются ноли до нужного количества.
Оперативная память	Запоминающее устройство, используемое для оперативного

	хранения и обмена информацией с другими узлами компьютера
Операционные оболочки	Специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы.
Операция И	В алгебре логики означает **логическое умножение (конъюнкция),** обозначается знаком умножения { ×, ^}.
Операция ИЛИ	В алгебре логики означает **логическое сложение (дизъюнкция),** обозначается знаком сложения { +, v}.
Операция НЕ	В алгебре логики называется ***логическим отрицанием*** или ***инверсией*** (дополнением) и обозначается знаком { -, ¬}.
Основание системы счисления	Количество цифр позиционной системы счисления. Позиционные системы отличаются друг от друга своим количеством цифр, и поэтому именуется по своему основанию. Например, десятичная система счисления, двоичная система.
Пакет прикладных программ (ППП)	Комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области. **ППП** по сфере применения можно разделить на проблемно-ориентированные, общего назначения и интегрированные пакеты.
Память	Обобщающее название устройств в компьютере, предназначенное для хранения данных.
Позиционная система счисления	Система, в которой значение цифры определяется ее местоположением (позицией) в изображении числа. Наибольшее распространение получила десятичная система счисления. В вычислительной технике применяется двоичная, восьмеричная и шестнадцатеричная системы счисления. В компьютерах применяются две формы представления двоичных чисел: естественная форма, или форма с фиксированной запятой (точкой); нормальная форма, или форма с плавающей запятой (точкой).
Поле данных	Последовательность нескольких битов и байтов, используется для хранения числовой, текстовой и графической информации в памяти компьютера. Поля постоянной длины бывают следующих типов: полуслово (2 байта); слово (4 байта); двойное слово (8 байт). С
Постоянное запоминающее устройство (ПЗУ)	Микросхема памяти, предназначенная только для чтения. Программы, размещенные в ПЗУ, называются «защитными» - их записывают туда на этапе изготовления микросхемы.. Предназначена для хранения *базовой системы ввода-вывода (BIOS).*
Прагматический подход к оценке экономической информации	Связан с определением *ценности, полезности* использования информации при выработке потребителем решения для достижения своей цели.
Представление двоичного числа с плавающей запятой	Каждое число изображается в виде двух групп цифр. Первая группа цифр называется *мантиссой,* вторая – *порядком*, причем абсолютная величина мантиссы должна быть меньше 1, а порядок – целым числом. Такая форма представления чисел называется нормальной и является основной в современных ПК.
Представление двоичного числа с фиксированной запятой	Числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой (точкой), отделяющей целую часть от дробной.

Прикладное программное обеспечение (*Application Sortware)*	Совокупность программ для решения определенного класса задач конкретной предметной области.
Прикладное программное обеспечение (ППО)	Класс программных средств, предназначенных для решения определенных функциональных задач пользователя. Условно **ППО** делится на Пакеты прикладных программ и Прикладные программы пользователя.
Проблемно-ориентированные компьютеры	Предназначены для решения более узкого круга задач, связанных, как правило, с управлением технологическими процессами, обрабатывают относительно небольшие объемы данных по несложным алгоритмам.
Программа	Формализованное описание последовательности команд (действий) устройств компьютера по реализации той или иной задачи. Программа указывает, в каком порядке, над какими данными и какие операции должны быть выполнены компьютером, и в какой форме должен быть выдан результат.
Программно-аппаратный, логический контроль ПК	Основан на использовании избыточного кода исходных и промежуточных данных ПК, что позволяет находить ошибки при изменении значения отдельных бит данных.
Программное обеспечение (*Software*)	Совокупность программ, и необходимой документации, обеспечивающих обработку или передачу данных. По сфере использования программное обеспечение подразделяется на три класса: системное программное обеспечение; прикладное программное обеспечение; инструментальное программное обеспечение.
Программы архивирования данных	Позволяют сжимать информацию на дисках и создавать архивы данных. Архивирование данных упрощает их хранение.
Программы диагностики работоспособности компьютера	Совокупность программных средств ПК для обнаружения сбоев в работе компьютера.
Программы обеспечения компьютерной безопасности	Программные продукты, предназначенные для пассивной и активной защиты данных от повреждения а также от несанкционированного доступа, просмотра и изменения данных.
Программы обслуживания дисков	Группа программ, выполняющая функции сжатия и копирования информации на дисках, защиты и восстановления данных, антивирусные программы.
Программы обслуживания сети	Предназначены для создания и функционирования компьютерных сетей.
Пролог	Язык логического программирования, предназначенный для решения логических задач моделирования , связанных слогическими умозаключениями специалиста.
Процедурно-ориентированные языки программирования	Группа языков, ориентированных на решение задач определенного класса, например искусственного интеллекта (*Пролог, Лисп, Симула* и др.).
Процессор	Устройство компьютера, служащее для выполнения команд. В состав процессора входят арифметико-логическое устройство (АЛУ), устройство управления (УУ).
Процессор (*центральный процессор*)	Основной вычислительный блок компьютера, содержит важнейшие функциональные устройства: *устройство*

	управления с интерфейсом процессора; арифметико-логическое устройство; процессорную память.
Процессор MISP	Процессор, работающий с минимальным набором длинных команд.
Процессор RISP	Процессор, работающий с сокращенным набором длинных команд.
Прямой код двоичного числа	В *n*-разрядном двоичном коде отводится один, как правило, самый старший разряд для знака, а оставшиеся *n-1* разрядов – для значащих цифр. Используется для сложения положительных чисел в арифметико-логическом устройстве процессора
Растровый метод кодирования цветного изображения	Представление изображения как совокупность точек, называемых *пикселями* (элемент изображения). Линейные координаты и индивидуальные свойства каждой точки (яркость) кодируются двоичными целыми числами.
Режим работы компьютера	Это способ функционирования системы. Режим работы определяется тем, как пользователь может участвовать в процессе обработки данных на компьютере: иметь непосредственный доступ к системе либо принимать решение после завершения машинного задания
Семантика языка представления данных	Характеризует информацию, предназначенную для хранения, передачи и обработки данных.
Семантический подход к оценке экономической информации	Предполагает учет *смыслового содержания информации*. При семантическом подходе к измерению смыслового содержания информации используется *тезаурусная мера*. **Тезаурус** – это систематизированная совокупность сведений и знаний, с указанием смысловых связей между ними, которыми располагает пользователь или система.
Сервисное программное обеспечение	Программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя. Обычно выделяют группы программ: программы диагностики работоспособности компьютера; антивирусные программы; программы обслуживания дисков; программы архивирования данных; программы обслуживания сети; программы обеспечения компьютерной безопасности.
Сетевые операционные системы	Комплекс программ, обеспечивающий обработку, передачу и хранение данных в сети.
Сигнал	Это способ обмена информацией в технических устройствах и системах, отражающий физические характеристики объектов и процессов. (С.82). Существующие в технических устройствах сигналы делятся на ***непрерывные*** (или аналоговые) и ***дискретные.***
Синтаксис языка представления данных	Способ представления информации для хранения, передачи и обработки данных.
Синтаксис языка программирования	Совокупность правил, определяющих допустимыее конструкции языка.
Синтаксический подход к оценке экономической информации	Связан со способом представления, передачи и хранения информации. При синтаксическом подходе не рассматривается смысловое содержание информации. Для измерения количества информации в синтаксическом подходе используют **энтропийный подход**. Единица количества информа-

	ции – **бит.**
Система CMY *(*Cyan-Magenta-Yellow)	Цветовая система - голубой-пурпурный-желтый, использующая растровый метод представления изображения. ***CMY*** применяется для получения цветных изображений на белой поверхности. Эта система используется в большинстве устройств вывода, таких как лазерные и струйные принтеры. Новые цвета в системе *CMY* получают вычитанием цветовых составляющих из белого цвета.
Система HSV	Цветовая система, использующая растровый метод представления изображения. В системе **HSV** для представления новых цветов не смешивают основные цвета, а изменяют их свойства. Насыщенность определяется количеством белого цвета в оттенке.
Система RGB	8-ми разрядная система кодирования цветных графических изображений. В качестве основных составляющих произвольного цвета используются три цвета: красный (*Red, R*), зеленый (*Green, G*) и синий (*Blue, B*). Для кодирования яркости каждой из основных составляющих используется по 256 значений (восемь двоичных разрядов), а на кодирование цвета одной точки используется 24 разряда.
Система UNICODE	Универсальная система кодирования текстовых данных. В системе **UNICODE** символы кодируются 16-разрядными двоичными числами. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65536 различных символов – этого достаточно для размещения в одной таблице всех широко употребляемых языков.
Системное программное обеспечение (*System Sortware*)	Совокупность программ и программных комплексов для обеспечения работы компьютеров и вычислительной системы в целом.
Системы счисления	Совокупность символов и правил написания чисел. Все системы счисления можно разделить на позиционные и непозиционные.
Слоты	Разъемы для подключения дополнительных устройств к компьютеру. Слоты размещаются на материнской плате.
Специализированные компьютеры	Предназначены для решения сравнительно узкого класса задач или реализации строго регламентированной группы функций. Сфера применения машин: управление техническими устройствами; маршрутизация потоков данных и согласование работы узлов компьютерных сетей и т.д.
Средства диагностики ПК	Совокупность программно-аппаратных средств ПК для автоматического поиска ошибок и выявления неисправностей с определенной локализацией их в ПК и его отдельных модулях.
Средства для создания приложений	Совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.
Структура экономической информации	Характеристика экономической информации, определяющая два взаимосвязанных аспекта: состав элементов, образующих структуру информации; взаимосвязь элементов структуры.
Структура электрон-	Это модель, устанавливающая состав основных частей

ной вычислительной машины (ЭВМ)	ЭВМ и способы установления связей между ними. В классической структуре ЭВМ выделяют арифметико-логическое устройство, устройство управления, запоминающее устройство (оперативная и внешняя память), а также устройства ввода-вывода.
Структурный подход к оценке экономической информации	Связан с рассмотрением только количественных характеристик информационных единиц, при этом содержательности и ценности информации не принимается во внимание. Элементарными неделимыми единицами экономической информации являются реквизиты, показатели, массивы, информационная база. Важной стороной *оценки информации* является определение ее качества.
Таблица истинности	Табличное представление вычислительной (логической) схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.
Тестовый контроль ПК	Осуществляется с помощью специальных тестов для проверки правильности работы ПК или его отдельных устройств.
Транслятор	Специальная программа перевода исходной программы на машинный язык компьютера Трансляторы реализуются в виде компиляторов и интерпретаторов.
Триггер	Электронное логическое устройство с памятью, обладающее двумя состояниями равновесия, соответствующими логической “1” и логическому “0”.
Универсальные (общего назначения) компьютеры	Предназначены для решения разнообразных по реализуемым алгоритмам задач (экономических, информационно-поисковых, научно-технических и др.).
Устройство управления	Часть процессора, обеспечивает автоматическое выполнение программы путем принудительной координации работы всех остальных устройств ЭВМ.
Утилиты	Программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров. Утилиты расширяют и дополняют соответствующие возможности операционной системы, либо решают самостоятельные важные задачи.
Физический подход к структуре экономической информации	Обусловлен автоматизированной обработкой экономической информации, характеризуется наличием следующих единиц измерения: символ, запись, файл. Все структурные единицы информации обрабатываются с помощью технических средств. Обрабатываемая информация измеряется в технических единицах: байт и Кбайт.
Центральный процессор (ЦП)	Главный рабочий процессор компьютера или вычислительной системы, выполняющий основные функции по обработке данных и управлению работой других устройств. В персональном компьютере – это микросхема, управляющая работой компьютером и производящая основные вычисления. Функционально ЦП подразделяется на устройство управления и арифметико-логическое устройство, выполняющее операции в соответствии с программой.
Цифровой сигнал	Дискретный сигнал, значения которого выражены опреде-

	ленными конечными числами.
Цифровые вычислительные машины	Вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, то есть цифровой форме.
Чипсет	Набор микросхем, размещенных на материнской плате. Чипсет необходим для взаимодействия процессора с другими электронными схемами.
Шины материнской платы	Группа проводников, предназначенных для связи оперативной памяти и процессора с остальными устройствами компьютера. Шины подразделяются на шину данных, адресную шину и командную шину
Экономическая информация	Информация, отражающая и обслуживающая процессы производства, распределения, обмена и потребления материальных благ. Она включает в себя сведения о материальных, трудовых и стоимостных аспектах процессов, воспроизводимых в экономике, и устраняющих неопределенность в отношении исходов этих процессов.
Электронно-вычислительная машина (ЭВМ), компьютер	Комплекс технических средств, предназначенных для автоматизированной обработки информации в процессе решения вычислительных и информационных задач.
Язык представления данных	Формальная знаковая система, обеспечивающая возможность хранения, передачи и обработки данных.
Язык программирования	Формализованный язык для описания алгоритмаа решени задачи на компьютере.
CASE-технологии создания информационных систем	Специальный программный комплекс для проектирования, анализа программного обеспечения и сопровождения сложных программных систем.
Активные угрозы	Имеют целью нарушение нормального функционирования информационной технологии посредством целенаправленного воздействия на аппаратные, программные и информационные ресурсы.
Асимметричный ключ	Ключ, используемый в асимметричных алгоритмах (шифрование, электронная цифровая подпись).
Атака	Злонамеренные действия взломщика, попытки реализации им любого вида угрозы.
Аудит информационной безопасности	В рамках аудита проводится проверка имеющихся элементов защиты информации и разработка недостающих.
Аудит информационной безопасности корпоративной системы Internet/Intranet	Системный процесс получения объективных качественных и количественных оценок о текущем состоянии информационной безопасности компании в соответствии с определенными критериями и показателями безопасности на всех основных уровнях обеспечения безопасности: методологическом, организационно-управленческом, технологическом

	и техническом.
Аутентификация	Процедура проверки правильности введенной пользователем регистрационной информации для входа в систему.
Верификация	Установление подлинности пользователей по отпечаткам пальцев и картам.
Взлом системы	Умышленное проникновение в информационную технологию, когда взломщик не имеет санкционированных параметров для входа.
Вирусная сигнатура	Некоторая уникальная характеристика вирусной программы, которая выдает присутствие вируса в вычислительной системе.
Вирус-фильтр (сторож)	Резидентная программа, обнаруживающая свойственные для вирусов действия и требующая от пользователя подтверждения на их выполнение.
Датаграмма	(Datagram) Блок информации, посланный как пакет сетевого уровня через передающую среду без предварительного установления соединения и создания виртуального канала. Датаграмма представляет собой единицу информации в протоколе (protocol data unit, PDU) для обмена информацией на сетевом (в случае протокола IP, IP-датаграммы) и транспортном (в случае протокола UDP, UDP-датаграммы) уровнях эталонной модели OSI. Использование термина датаграмма подчёркивает такое свойство обоих протоколов, как негарантированная доставка данных.
Дезинфектор (доктор)	Программа, осуществляющая удаление вируса из программного файла или памяти ПК.
Детектор (сканер)	Специальные программы, предназначенные для просмотра всех возможных мест нахождения вирусов (файлов, операционная система, внутренняя память и т.д.) и сигнализирующие об их наличии.
Дешифрование	Процесс извлечения открытого текста без знания криптографического ключа на основе известного шифрованного. Термин дешифрование обычно применяют по отношению к процессу криптоанализа шифротекста (криптоанализ сам по себе, вообще говоря, может заключаться и в анализе шифросистемы, а не только зашифрованного ею открытого сообщения).
Домен	Определенная зона в системе доменных имён Интернет, выделенная владельцу домена (какой-либо стране, региону, юридическому или физическому лицу) для целей обеспечения доступа к предоставляемой в Интернете информации принадлежащей владельцу домена. Здесь доменом называется группа ресурсов информационной сети, которые работают или под одним компьютером, или под одной сетевой рабочей машиной или сетевым узлом. Примеры доменов: ru, com, org и т.д.
Захватчик паролей	Программы, специально предназначенные для воровства паролей.
Зашифрование	Процесс нормального применения криптографического преобразования открытого текста на основе алгоритма и ключа в результате которого возникает шифрованный текст.

Защита информации	Применение различных средств и методов, принятие мер и осуществление мероприятий с целью системного обеспечения надежности передаваемой, хранимой и обрабатываемой информации.
Защита информации в информационной технологии	Процесс создания и поддержания организованной совокупности средств, способов, методов и мероприятий, предназначенных для предупреждения, искажения, уничтожения и несанкционированного использования данных, хранимых и обрабатываемых в электронном виде.
Идентификация	Присвоение какому-либо объекту или субъекту уникального имени или образа.
Информационная безопасность	Состояние защищенности в информационной сфере, определяющееся совокупностью сбалансированных интересов личности, общества и государства.
Клиент-сервер	(Client-server) Сетевая архитектура, в которой устройства являются либо клиентами, либо серверами. Клиентом (front end) является запрашивающая машина (обычно ПК), сервером (back end) - машина, которая отвечает на запрос. Оба термина (клиент и сервер) могут применяться как к физическим устройствам, так и к программному обеспечению.
Ключ	Параметр шифра, определяющий выбор конкретного преобразования данного текста. В современных шифрах алгоритм шифрования известен и криптографическая стойкость шифра целиком определяется секретностью ключа (Принцип Керкгоффса).
Код аутентичности сообщения MAC	(Message Authentication Code) Специальный набор символов, который добавляется к сообщению и предназначен для обеспечения его целостности и аутентификации источника данных в протоколах аутентификации сообщений с доверяющими друг другу участниками.
Коллизия хеш-функции	В общем случае однозначного соответствия между исходными данными и хеш-кодом нет. Поэтому существует множество массивов данных, дающих одинаковые хеш-коды - так называемые коллизии. Вероятность возникновения коллизий играет немаловажную роль в оценке «качества» хеш-функций.
Компьютерный вирус	Специальная программа, предназначенная для выполнения разрушительных действий в вычислительной системе или в сети.
Конфиденциальная информация	Информация, исключительное право на пользование которой принадлежит определенным лицам или группе лиц.
Корпоративная информационная система	Автоматизированная система управления крупными, территориально рассредоточенными предприятиями, имеющими несколько уровней управления.
Криптоанализ	Наука, изучающая математические методы нарушения конфиденциальности и целостности информации.
Криптоаналитик	Человек, создающий и применяющий методы криптоанализа.
Криптографическая атака	Попытка криптоаналитика вызвать отклонения в атакуемой защищенной системе обмена информацией. Успешную криптографическую атаку называют взломом или вскрытием.

Криптографическая стойкость	Способность криптографического алгоритма противостоять криптоанализу.
Криптология	Наука, исследующая криптографические преобразования. В криптологии различают направления: криптографию и криптоанализ
Криптосистема	Семейство обратимых преобразований открытого текста в шифрованный.
Люк	Скрытая, недокументированная точка входа в программный модуль, входящий в состав программного обеспечения информационной технологии.
Маршрутизатор	Программные или программно-аппаратные средства определения маршрута передачи данных между узлами сети.
Маскировка	Метод защиты информации путем ее криптографического закрытия.
Несанкционированный доступ к информации	Нарушение установленных правил разграничения доступа, последовавшее в результате случайных или преднамеренных действий пользователей или других субъектов системы разграничений.
Неформальные средства защиты	Средства защиты, которые определяются целенаправленной деятельностью человека либо регламентируют эту деятельность.
Открытый (исходный) текст	Данные (не обязательно текстовые), передаваемые с использованием криптографии.
Пароль	Совокупность символов, определяющих объект (субъект).
Пассивные угрозы	Направлены на несанкционированное использование информационных ресурсов, не оказывая при этом влияния на функционирование информационной технологии.
Полидетектор-дезинфектор	Интегрированные программы, позволяющие выявить вирусы в персональном компьютере, обезвредить их, по возможности восстановить пораженные файлы и программы.
Принуждение	Метод защиты, когда специалисты и персонал информационной технологии вынуждены соблюдать правила обработки, передачи и использования защищаемой информации под угрозой материальной, административной или уголовной ответственности.
Протокол	Набор правил, определяющих взаимодействие устройств, программ, систем обработки данных, процессов или пользователей.
Протокол IP	Правила передачи сообщений в Интернет. Протокол IP (Internet Protocol) гарантирует, что коммуникационный узел определит наилучший маршрут доставки пакета.
Расшифрование	Процесс нормального применения криптографического преобразования шифрованного текста в открытый.
Регламентация	Метод защиты информации, создающий по регламенту в информационных технологиях такие условия автоматизированной обработки, хранения и передачи защищаемой информации, при которых возможности несанкционированного доступа к ней сводились бы к минимуму.
Регламентный режим	Режим обработки данных, при котором обработка информации производится в заранее определенные сроки (по регламенту).
Резидентная	Программа, постоянно находящаяся в оперативной памяти

программа	персонального компьютера.
Секретный (симметричный) ключ	Ключ, используемый в симметричных алгоритмах (шифрование, выработка кодов аутентичности)
Служба WWW	(World Wide Web) Гипертекстовая система поиска ресурсов в Internet и доступа к ним.
Спам	Массовая анонимная рассылка незатребованной корреспонденции.
Транспортный протокол TCP	(Transmission Control Protocol) Протокол передачи данных, разбивающий сообщение на пакеты. Собирает принимаемое сообщение из пакетов, следит за целостностью передаваемого пакета и контролирует доставку всех пакетов сообщения.
Трафик	Поток сообщений в сети передачи данных; рабочая нагрузка линии связи.
Троянский конь	Программа, выполняющая в дополнение к основным, т.е. запроектированным и документированным действиям, действия дополнительные, не описанные в документации.
Угроза безопасности информации	Действия или событие, которое может привести к разрушению, искажению или несанкционированному использованию информационных ресурсов, включая хранимую и обрабатываемую информацию, а также программные и аппаратные средства.
Управление доступом	Метод защиты информации с помощью использования всех ресурсов информационной технологии.
Утилита	Специальная программа, выполняющая определенные сервисные функции
Формальные средства защиты	Средства, выполняющие защитные функции строго по заранее предусмотренной процедуре без непосредственного участия человека.
Хеширование	(Hashing) - преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хеш-функциями или функциями свёртки, а их результаты называют хешем, хеш-кодом или дайджестом сообщения (англ. message digest).
Хост-компьютер	(Host computer) Компьютер, обслуживающий сеть, управляющий передачей сообщений и предоставляющий удаленный доступ к своим ресурсам.
Червь	Программа, распространяющаяся через сеть и не оставляющая своей копии на магнитном носителе.
Шифрованный (закрытый) текст	Данные, полученные после применения криптосистемы с указанным ключом.
Электронная почта	Форма передачи электронных сообщений на расстоянии.
BSI	(British Standards Institution) Британский институт стандартов. Группа BSI начала свою деятельность в 1901 г. как комитет инженеров, которые устанавливали стандарты на сталь. Эти стандарты использовались британскими промышленниками для производства более качественной и конкурентоспособной продукции. BSI — член Международной организации по стандартизации (ISO).
Cobit	(Control OBjectives for Information Technology) Контрольные объекты для информационных и смежных технологий.

	Является набором стандартов и рекомендаций для аудита и управления информационными технологиями. В большей степени ориентирован на аудит IT-инфраструктуры.
CRC	(Cyclic redundancy code, циклический избыточный код) Алгоритм вычисления контрольной суммы - способ цифровой идентификации некоторой последовательности данных, который заключается в вычислении контрольного значения её циклического избыточного кода. CRC является типом хеш-функции, используемой для вычисления контрольного кода - небольшого количества бит внутри большого блока данных, например сетевого пакета или блока компьютерного файла, применяемого для обнаружения ошибок при передаче или хранении информации. Результат вычисления добавляется в конец блока данных непосредственно перед началом передачи или сохранения данных на каком-либо носителе информации. Впоследствии он проверяется для подтверждения её целостности. Подобная проверка легко реализуема в двоичном коде оборудовании, легко анализируется и хорошо подходит для обнаружения общих ошибок, вызванных наличием шума в каналах передачи данных.
Firewalls	Межсетевой экран или сетевой экран (также брандмауэр или файервол). Это комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов на различных уровнях модели OSI в соответствии с заданными правилами. Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача - не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.
FTP-сервер	Компьютер, на котором содержатся файлы, предназначенные для открытого доступа.
HTTP	(HyperText Transfer Protocol - протокол передачи гипертекста») Протокол прикладного уровня передачи данных (изначально - в виде гипертекстовых документов). Основой HTTP является технология клиент-сервер, то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.
ICQ	Коммерческая услуга, обеспечивающая ведение личной переписки через сеть Интернет посредством собственных службы мгновенного обмена, компьютерного клиента для этой службы, а также Интернет-портала.
ICQ (клиент)	Компьютерная прикладная программа с графическим интерфейсом пользователя, официальный клиент службы мгновенного обмена сообщениями ICQ.
IDS	(Intruder Detection System) Системы обнаружения сетевых атак и вторжений. Предупреждает администратора сети о подозрительных и сомнительных действиях и явлениях в

	компьютерной сети.
IMAP	(Internet Message Access Protocol) Протокол прикладного уровня для доступа к электронной почте Интернета. Аналогичен POP3, т.е. служит для работы со входящими письмами, однако обеспечивает дополнительные функции, в частности, возможность провести поиск по ключевому слову, не сохраняя почту в локальной памяти.
Internet	Глобальная компьютерная сеть, объединяющая ПК отдельных пользователей и локальные вычислительные сети предприятий и организаций.
Intranet	В отличие от сети Интернет, это внутренняя частная сеть организации, построенная на использовании протокола IP для обмена и совместного использования некоторой части информации внутри этой организации.
IPS	(Intruder Prevention System) Системы предупреждения сетевых атак и вторжений. Уполномочивает администратора предпринять меры по сохранению безопасности системы
IPsec	(IP Security) – набор протоколов для обеспечения защиты данных, передаваемых по межсетевому протоколу IP, позволяет осуществлять подтверждение подлинности и/или шифрование IP-пакетов. IPsec также включает в себя протоколы для защищённого обмена ключами в сети Интернет.
IP-адрес	(Internet Protocol Address) - уникальный идентификатор устройства (обычно компьютера), подключённого к локальной сети и (или) Интернету.
IRC	(Internet Relay Chat - ретранслируемый Интернет-чат) Сервисная система, при помощи которой можно общаться через сеть Интернет с другими людьми в режиме реального времени.
ISO	(International Organization for Standardization) Международная организация по стандартизации – международная организация, занимающаяся выпуском стандартов.
IT	(Information Technology) Информационные технологии (ИТ). Широкий класс дисциплин и областей деятельности, относящихся к технологиям управления и обработки данных, в том числе, с применением вычислительной техники.
LAN	(Local Area Network - лока́льная вычисли́тельная сеть, ЛВС) Компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт). Также существуют локальные сети, узлы которых разнесены географически на расстояние более 12500 км (космические станции и орбитальные центры). Несмотря на такое расстояние, подобные сети относят к локальным.
MAC	(Message authentication code) Код аутентичности сообщения) в протоколах аутентификации сообщений с доверяющими друг другу участниками - специальный набор символов, который добавляется к сообщению и предназначен для обеспечения его целостности и аутентификации источника данных. Обычно применяется для обеспечения целостности передаваемой информации. Для проверки целостности сообщения на отправляющей стороне к сообщению добавля-

	ется значение хеш-функции от этого сообщения, на приемной стороне также вырабатывается хеш от полученного сообщения. Выработанный на приёмной стороне и полученный хеш сравниваются, если они равны то считается, что полученное сообщение дошло без изменений.
P2P	(Peer-to-peer) Однора́нговые, децентрализо́ванные или пи́ринговые сети. Это компьютерные сети, основанные на равноправии участников. В таких сетях отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и сервером. В отличие от архитектуры клиент-сервер, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов.
POP (POP3)	(Post Office Protocol Version 3 - протокол почтового отделения, версия 3) используется почтовым клиентом для получения сообщений электронной почты с сервера. Обычно используется в паре с протоколом SMPT. POP, POP2 являются устаревшими версиями протокола POP3.
RSA	(буквенная аббревиатура от фамилий Rivest, Shamir и Adleman) Криптографический алгоритм с открытым ключом. RSA стал первым алгоритмом такого типа, пригодным и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений.
SASL (Cyrus SASL)	(Simple Authentication and Security Layer) Метод для добавления поддержки аутентификации в протоколы соединения. Для использования SASL протокол включает команду для идентификации и аутентификации пользователя на сервере и для опциональной защиты переговоров последующей интерактивности протокола. Если это используется в переговорах, то слой безопасности вставляется между протоколом и соединением.
SMTP	(Simple Mail Transfer Protocol - простой протокол передачи почты) Сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.
SSL	(Secure Sockets Layer) Криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. При его использовании создаётся защищённое соединение между клиентом и сервером. SSL изначально разработан компанией Netscape Communications. Впоследствии на основании протокола SSL 3.0 был разработан и принят стандарт RFC, получивший имя TSL. Использует шифрование с открытым ключом для подтверждения подлинности передатчика и получателя.
TELNET	(TELecommunication NETwork) - сетевой протокол для реализации текстового интерфейса по сети (в современной форме - при помощи транспорта TCP). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола.
URL	(Uniform Resource Locator) Единый указатель ресурсов - единообразный локатор (определитель местонахождения) ресурса. URL - это стандартизированный способ записи ад-

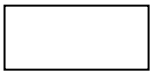
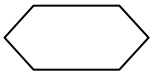
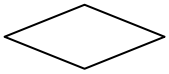
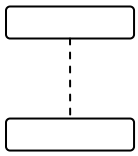
	реса ресурса в сети Интернет.
VPN	(Virtual Private Network - виртуальная частная сеть) Логическая сеть, создаваемая поверх другой сети, например Интернет. VPN позволяет объединить, например, несколько офисов организации в единую сеть с использованием для связи между ними неподконтрольных каналов.
Алгоритм	Конечная последовательность простейших формул и логических правил, чётко и недвусмысленно определяющих весь ход решения какой-либо задачи, который состоит в упорядоченном выполнении различных операций над данными с целью получения искомого ответа (результата) за конечное число шагов.
Псевдокод (ПСК)	Способ записи алгоритма на условном подмножестве естественного языка с элементами языка программирования и общепринятыми математическими обозначениями.
Программа	Последовательность недвусмысленных инструкций (операторов или команд), которую компьютер чётко выполняет одну за другой до тех пор, пока не дойдёт до оператора «конец».
Линейный алгоритм	Совокупность операций, выполняемых в одном направлении по единственному пути от начала к концу без повторений
Разветвлённый алгоритм	Алгоритм, имеющий несколько возможных путей решения задачи, выбор одного из которых зависит от выполнения или нарушения какого-либо условия
Циклический алгоритм	Алгоритм, операции в котором могут многократно повторяться в одном и том же порядке, но всякий раз с обновлёнными данными
VBA (Visual Basic for Application)	Современный язык программирования, поддерживаемый всеми приложениями пакета Microsoft Office 2003 - 07
Объект	Готовая программная конструкция интерфейса "человек - компьютер", которая наделена совокупностью свойств (параметров) и методов их обработки.
Свойства объекта	Характеристики его текущего состояния в приложении, его параметры. Их значения определяют уникальность объекта, его отличие или сходство по сравнению с другими объектами.
Метод	Команда или набор команд (подпрограмма), предназначенная для воздействия на свойства для их целенаправленных изменений.
События	Действия пользователя или других функционирующих в приложении программ
Класс	Понятие, объединяющее объект с ему подобными и определяющее его назначение, свойства и те действия, которые могут быть выполнены над ним
Макрос	Набор операций, производимых пользователем и автоматически зафиксированных в виде программы.
Разработчик	Режим работы в приложении для корректировки макросов и программ в среде редактора VBA
Оператор	Наименьшая единица VBA-кода, предназначенная для определения переменной, установки параметров или выполнения какого-либо действия в программе

Процедура	Отдельная единица программного кода VBA, которую можно вызывать по имени для выполнения; она может выполняться самостоятельно. Любая процедура содержит один или несколько операторов
Модуль	Именованная единица, состоящая из одной или нескольких процедур и раздела объявлений, в котором объявляются переменные, константы и пользовательские типы данных, а также устанавливаются параметры компилятора
Проект	Включает в себя все модули, формы и связанные с приложением объекты, относящиеся к конкретному документу, причем проект сохраняется вместе с самим этим документом
Переменная	Поименованное место в оперативной памяти компьютера
Литеральная константа	Константа, действительное значение которой (строка символов или число) записывается прямо в тексте программы
Символическая константа	Имеет своё имя, но, зато в отличие от переменной, значение такой константы никогда не меняется на всем протяжении выполнения программы
Массив	Набор элементов одинакового типа, имеющих общее имя
Одномерный массив	Массив, представляющий собой простой список данных, называется одномерным массивом, а число в скобках, стоящее рядом с именем массива, называется индексом элемента данного массива
Статический массив	Такой массив, размерность которого была указана непосредственно при его объявлении
Динамические массив	имеет переменное количество элементов, т.е., динамический массив может увеличиваться или сокращаться в зависимости от того, какое число элементов нужно в заданный момент исполнения участка программы

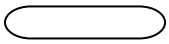
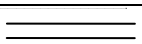
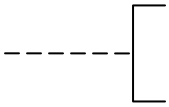
Приложение 1

Таблица 1

Описание символов, используемых в ГСА (Графическая Схема Алгоритмов)

Наименование	Обозначение	Фуикции
Данные		Символ отображает данные, носитель данных не определён. Используется для обозначения операций ввода и вывода данных
Процесс		Символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации). Используется для обозначения операций присваивания
Предопределенный процесс		Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов, которые определены в другом месте (в подпрограмме, модуле). Используется для обозначения неэлементарных блоков
Подготовка		Символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы). Может быть использован для обозначения заголовка цикла
Решение		Символ отображает решение или функ-цию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути. Используется для обозначения оператора условного перехода или оператора варианта
Граница цикла		Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т.д. помешаются внутри символа в начале или в конце в зависимости от типа цикла
Соединитель		Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение

Продолжение табл. 1

Наименование	Обозначение	Функции
Терминатор		Символ отображает выход во внешнюю среду и вход из внешней среды. Используется для обозначения начала или окончания алгоритма
Линия		Символ отображает поток данных или управления. Направления справа налево и снизу вверх обозначаются стрелками. Используется для соединения символов в алгоритме
Параллельные действия		Символ отображает синхронизацию двух или более параллельных операций
Пунктирная линия		Символ отображает альтернативную связь между двумя или более символами. Кроме того, символ используется для обведения аннотированного участка при записи комментариев
Комментарий		Символ используется для добавления описательных комментариев или пояснительных записей с целью объяснений или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев или примечаний должен быть помещен около о гран ичивающей фигуры

Символы могут быть вычерчены в любой ориентации, но предпочтительной является горизонтальная ориентация. Внутрь символа помещают обозначения или описания операций. Направления линий связи *слева направо* и *сверху вниз* считаются стандартными, и линии связи изображаются *без стрелок*, в противоположном случае — со стрелками. В операционные блоки (процесс, ввод-вывод и подпрограмма) входит и из них исходит только одна линия связи. В блок проверки условий (ветвление) входит одна, а выходит не менее двух линий, около которых записываются результаты проверки условий. Из начальной вершины исходит одна линия связи, в конечную вершину также входит одна линия связи. Линии могут *соединяться* одна с другой, но *не* могут *разветвляться*. Символы ГСА могут быть отмечены идентификаторами или порядковыми номерами (буква или буква с цифрой, располагающиеся слева над символом).

Внутри соединителей ставятся номера (координаты) блоков, к которым или от которых идут линии связи. Вместо номера (координат) может быть поставлен некоторый (один и тот же в обоих соединениях) идентификатор (рис. 5).

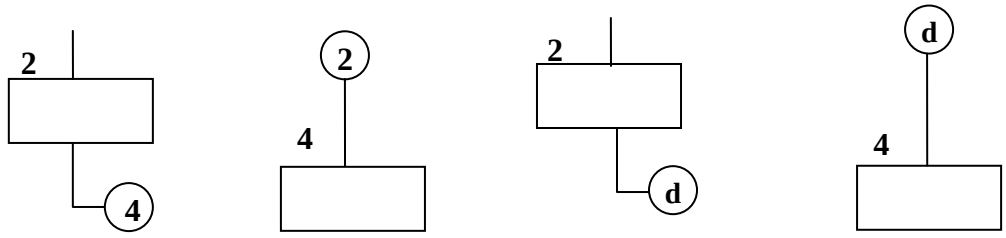


Рис. 5. Применение соединений

Недостатком графических схем алгоритмов является громоздкость. Для решения специальных задач, например проектирования вычислительных устройств, применяются другие способы задания алгоритмов, такие, как логические (операторные) и матричные схемы. Их достоинствами являются компактность и простота дальнейшей формализации, а недостатком — малая наглядность.

СПИСОК ЛИТЕРАТУРЫ

1. Visual Basic на практике / Под общ. ред. Г.И. Магданурова – СПб: БХВ – Петербург, 2008
2. *Аляев Ю.А., Козлов О.А.* Алгоритмизация и языки программирования Pascal, C++, Visual Basic. Учебно-справочное пособие. – М.: Финансы и Статистика, 2004
3. Аудит безопасности Intranet / *Петренко С.А., Петренко А.А.* – М.: ДМК Пресс, 2002.
4. *Бройдо В.Л., Ильина О.П.* Вычислительные системы, сети и телекоммуникации, 3-е изд. – СПб.: Питер, 2008.-766с.
5. Введение в правовую информатику: учебник для вузов / Под ред. *Новикова Д.Б., Камынина В.Л.* – М.:ООО НПО «Вычислительная математика и информатика», 2000.
6. Информатика для экономистов: Учебник / Под общ. Ред. *В.М. Матюшка.*-М.:ИНФРА-М, 2007.-880с.
7. Информатика для юристов и экономистов /Под редакцией *С.В.Симоновича* – СПб.:Питер, 2005.-688с.
8. Информатика: базовый курс: Учебник для студентов вузов, бакалавров, магистров, обучающихся по направлениям “Информатика и вычислительная техника” *Акулов О.А., Медведев Н.В.* – М:Омега-Л, 2004.-552с.
9. Информатика: Учебник / Под ред. проф. *Н.В. Макаровой*. – М.: Финансы и статистика, 2008.-768с.
10. Информационные системы в экономике: учебник для вузов / Под ред. *Титоренко Г.А*. - М.:ЮНИТИ-ДАНА, 2008.
11. Информационные технологии в профессиональной деятельности: Учеб. пособие для проф. образования /*Михеева Е.В.* – 2-е изд., стер. – М.: Издательский центр «Академия», 2005. – 384 с.
12. Информационные технологии управления: Учеб. пособие для вузов/ Под ред. Проф. *Г.А. Титоренко*. - М.:ЮНИТИ-ДАНА, 2004. -439с.

13. *Касперский Е.* Компьютерные вирусы / Касперский Е. – М.:Эдель, 2007. - 256 с.
14. *Кодолова И.А., Степанова Ю.В., Тартаковская Н.З.* Основы создания информационных систем в экономике, Казань, Изд-во: КГФЭИ, 2007.-272с.
15. *Козырев А.А.* Информационные технологии в экономике и управлении: Учебник. 3-е изд., перераб. и доп. – СПб.: Изд-во В.А. Михайлова, 2001. – 496 с.
16. *Коноплева И.А., Хохлова О.А., Денисов А.В.* Информационные технологии / под ред. И.А. Коноплевой. – М: Проспект, 2008. – 304с.
17. Культин Н.Б. Visual Basic. Освой самостоятельно. - СПб.; БХВ - Петербург, 2005
18. *Лесничная И.Г., Миссинг И.В., Романова Ю.Д., Шестаков В.И.* Информатика и информационные технологии. Учебное пособие / Под ред. Романовой Ю.Д.-М.:Изд-во Эксмо, 2005.-544с.
19. *Меняев М.Ф.* Информационные технологии управления: Учебное пособие. В 3 кн.: Книга 2: Информационные ресурсы. – М.: Омега-Л, 2003. – 432 с.
20. Основы защиты информации: учебное пособие / *Куприянов А.И., А.В. Сахаров, В.А. Шевцов* – М.:Академия, 2008.
21. Основы современных компьютерных технологий: Учебник /Под ред. проф. *А.Д. Хомоненко*. – СПб.: КОРОНА принт, 2005. – 672 с.
22. *Петренко С.А.* Политики информационной безопасности / Петренко С.А., Курбатов В.А. – М.:Компания АйТи, 2006.
23. Практикум по экономической информатике: Учебно-практическое пособие.–/Под ред. *В.П. Косарева* – М.: Финансы и Статистика, 2007.–461с.
24. *Романова Ю.Д.* Информатика и информационные технологии. Конспект лекций: учеб. пособие / Ю.Д. Романова, И.Г. Лесничная.-2-е изд., перераб. и доп.- М.: Эксмо, 2009.- 320с.

25. *Федорова Г.В.* Информационные технологии бухгалтерского учета, анализа и аудита. – М.: Омега-Л, 2004. – 304 с.
26. Экономическая информатика.: Учебник/ Под ред. *В.П. Косарева и Л.В. Еремина* – М.: Финансы и статистика, 2006. -655 с.

СОДЕРЖАНИЕ