

Ответы экзамена по предмету “Информатика”

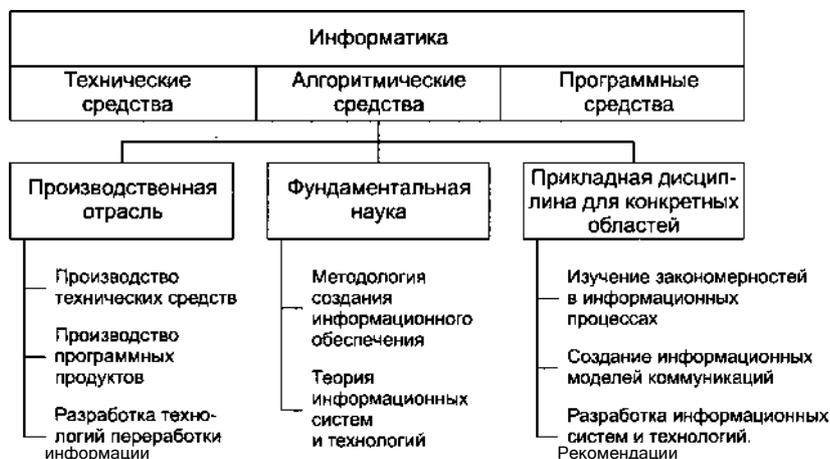
1. Информатика и ее структура.

Термин *информатика* возник в 60-х гг. во Франции для названия области, занимающейся автоматизированной обработкой информации с помощью электронных вычислительных машин. Французский термин *informatique* (информатика) образован путем слияния слов *information* (информация) и *automatique* (автоматика) и означает буквально информационную автоматику, или автоматизированную переработку информации. В англоязычных странах этому термину соответствует синоним *computer science* (наука о компьютерной технике).

Информатика занимается изучением процессов преобразования и создания новой информации более широко, практически не решая, в отличие от кибернетики, задачи управления различными объектами. Поэтому может сложиться впечатление об информатике как о более емкой дисциплине, чем кибернетика. Однако информатика не занимается решением проблем, не связанных с использованием компьютерной техники, что, несомненно, сужает ее, казалось бы, обобщающий характер. Между этими двумя дисциплинами провести четкую границу не представляется возможным в связи с ее размытостью и неопределенностью, хотя существует довольно распространенное мнение о том, что информатика является одним из направлений кибернетики.

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработкой информации главным образом с помощью компьютеров и телекоммуникационных средств связи во всех сферах человеческой деятельности.

Информатику в узком смысле можно представить как состоящую из трех взаимосвязанных частей — технических (*hardware*), программных (*software*) и алгоритмических средств (*brainware*). В свою очередь, информатику в целом и каждую ее часть обычно рассматривают с разных позиций (рис. 2.3):



Структура информатики как отрасли, науки, прикладной дисциплины, как производственную отрасль, как фундаментальную науку, как прикладную дисциплину.

Информатика как *производственная отрасль* состоит из однородной совокупности предприятий разных форм хозяйствования, где занимаются производством компьютерной техники, программных продуктов и разработкой современной технологии переработки информации. Специфика и значение информатики как отрасли производства состоят в том, что от нее во многом зависит рост производительности труда в других отраслях народного хозяйства. Более того, для нормального развития этих отраслей производительность труда в самой информатике должна возрастать более высокими темпами, так как в современном обществе информация все чаще выступает как предмет конечного потребления: людям необходима информация о событиях, происходящих в мире, о предметах и явлениях, относящихся к их профессиональной деятельности, о развитии науки и самого общества. Дальнейший рост производительности труда и уровня благосостояния возможен лишь на основе использования новых интеллектуальных средств и человеко-машинных интерфейсов, ориентированных на прием и обработку больших объемов мультимедийной информации (текст, графика, видеоизображение, звук, анимация). При отсутствии достаточных темпов увеличения производительности труда в информатике может произойти существенное замедление роста производительности труда во всем народном хозяйстве. В настоящее время около 50 % всех рабочих мест в мире поддерживается средствами обработки информации.

Информатика как *фундаментальная наука* занимается разработкой методологии создания информационного обеспечения процессов управления любыми объектами на базе компьютерных информационных систем. Существует мнение о том, что одна из главных задач этой науки — выяснение, что такое информационные системы, какое место они занимают, какую должны иметь структуру, как функционировать, какие общие закономерности им свойственны. В Европе можно выделить следующие основные научные направления в области информатики: разработка сетевой структуры, компьютерно-интегрированные производства, экономическая и медицинская информатика, информатика социального страхования и окружающей среды, профессиональные информационные системы.

Цель фундаментальных исследований в информатике — получение обобщенных знаний о любых информационных системах, выявление общих закономерностей их построения и функционирования.

Информатика как *прикладная дисциплина* занимается:

- изучением закономерностей в информационных процессах (накопление, переработка, распространение информации);
- созданием информационных моделей коммуникаций в различных областях человеческой деятельности;
- разработкой информационных систем и технологий в конкретных

областях и выработкой рекомендаций относительно их жизненного цикла: этапов проектирования и разработки, производства, функционирования и т. д.

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

Задачи информатики состоят в следующем:

- исследование информационных процессов любой природы;
- разработка информационной техники и создание новейшей технологии переработки информации на базе полученных результатов исследования информационных процессов;
- решение научных и инженерных проблем создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах общественной жизни.

2. Новые информационные технологии.

Информационные технологии (ИТ) — инфраструктура, обеспечивающая реализацию информационных процессов. Ее формируют: каналы связи, по которым передается информация; программы, управляющие сбором, хранением, обработкой и представлением информации; компьютеры и аппаратура, обеспечивающие выполнение этих программ. Информация на технологическом уровне выступает в виде данных, поэтому сами данные также относятся к ИТ. К информационным технологиям также относятся инструменты, с помощью которых реализуются информационные процессы: языки, используемые для написания программ, протоколы, применяющиеся при передаче данных, модели хранения и передачи данных и т.д.

Итак, инфраструктура ИТ — это, с одной стороны, аппаратно-программная среда и телекоммуникации, а с другой — инструментарий осуществления информационных процессов. В центре инфраструктуры — данные.

ИТ иногда рассматривают как «компьютерную инфраструктуру» организации. Составляющие ИТ имеют «твердую основу» {hardware) в виде компьютеров, сетей, телекоммуникационного оборудования и каналов связи, а также «мягкую основу» {software) в виде программного обеспечения, управляющего работой аппаратуры. Для нас важнейшей составляющей ИТ, своеобразным ядром являются данные, организованные с помощью специальных программ в базы данных.

База данных (БД) — это организованная в соответствии с определенными правилами и поддерживаемая в памяти компьютера совокупность данных, используемая для удовлетворения информационных потребностей пользователей. БД — это своеобразная среда, в которой данные превращаются в информацию. Базы данных, по существу, представляют собой важнейшее связующее звено, мост, соединяющий техническую мощь информационных систем с реальными задачами конкретных ФП и бизнес-приложений.

Информационные технологии обеспечивают информационные процессы вне зависимости от их содержания, они одинаково работают с бухгалтерскими сведениями и сводками погоды — обеспечивают хранение, получение информации и передачу ее средствами телекоммуникаций от отправителя по конкретному адресу в заданном виде. Результат работы ИТ — обработка и доставка сообщения, содержание которого ИТ не интересует, от отправителя — к получателю.

Передовая практика управления утверждает, что основная роль ИТ в организации — информационное обслуживание ее подразделений. Поскольку термин «информационное обслуживание» весь-ми широк, его следует уточнить. Под *информационным сервисом* мы **будем** понимать информационный процесс, функционирующий в режиме, определяемом одним или несколькими бизнес-процессами или проектами. Информационный сервис, выполняемый средствами ИТ, мы будем далее именовать *сервисом ИТ*. Попросту говоря, информационный сервис — это информационный процесс, параметры которого значимы для протекания бизнес-процессов или проектов в организации. При этом не каждый информационный сервис выполняется средствами ИТ. Например, если юрист отыски-ii.li-т и твердой копии некоторого кодекса необходимую для данного Влучая статью закона, речь идет об информационном сервисе, который не является сервисом ИТ, — информационные технологии в этом случае не используются. Напротив, расчет баланса в 1С или любой другой бухгалтерской системе является сервисом ИТ.

3. Информационная система.

Термин «информационные системы» вбирает в себя два важных понятия — «информация» и «системы». Каждое из них настолько важно, что для их описания существуют целые теории: теория информации и теория систем. Мы лишь познакомимся с базовыми понятиями, необходимыми для дальнейшего изложения.

Информационная система (ИС) — связанный набор аппаратных и программных средств, информационных ресурсов, а также управленческого сервиса, осуществляющих информационные процессы для обеспечения подготовки и принятия решений. Таким образом, понятие ИС — весьма широкое и включает в себя как технические системы, управляющие конкретными технологическими процессами (например, автопилоты самолетов), так и управленческие системы, применяемые компаниями для планирования и управления собственными ресурсами — материальными, финансовыми и интеллектуальными. Такие управленческие системы, используемые в экономике, иногда называют еще интегрированными системами управления (ИСУ), корпоративными информационными системами (КИС).

Основная цель разработки и применения ИС — создание современной информационной инфраструктуры для управления компанией. В управлении

компанией принято выделять три уровня: стратегический, тактический (его иногда называют управленческим) и оперативный или операциональный.

Основные задачи ИС:

- на стратегическом уровне — обеспечение высшего руководства информацией о долгосрочных тенденциях в развитии бизнеса, лучших технологиях, продуктах, методах управления и способах изменения бизнеса для выработки долгосрочной стратегии компании, обеспечивающей ее конкурентное преимущество;

- на тактическом уровне — максимально быстрое и качественное обеспечение специалистов среднего и высшего звеньев руководства оперативной и аналитической информацией, способствующей подготовке и принятию наилучших решений;

- на оперативном уровне — качественное и быстрое выполнение рутинных, часто повторяющихся операций, связанных с обеспечением информационных процессов, прежде всего по вводу и обработке первичной информации и представлению необходимых документов.

- Чтобы завершить обсуждение данного нами определения экономической информатики, осталось уточнить, что мы понимаем под принятием решения. Принятие решения — это выбор наилучшего в каком-то смысле варианта решения из множества допустимых на основании имеющейся информации.

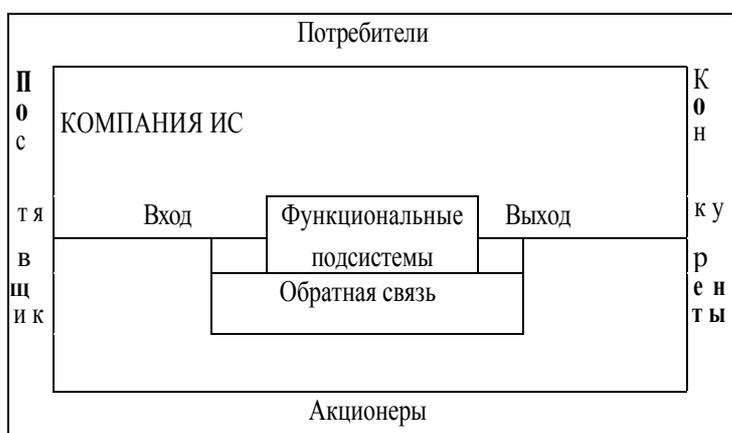


Схема функционирования ИС.

4. „ Новая экономика ” - экономика информации.

Информационные системы изменили облик мировой экономики. Появились новые термины, определяющие экономику, работа которой характеризуется глубоким проникновением информационных систем: интернет-экономика, новая экономика, метакapитализм.

Так что будем говорить об экономике, базирующейся на современных достижениях информационных технологий и информационных систем, как об «экономике современного информационного общества». Рассмотрим несколько характерных черт такой экономики'.

1. Продукция некоторых отраслей экономики в динамике характеризуется повышением качества и производительности при относительном снижении цены.

2. Происходит декапитализация компаний: крупные компании дробятся и превращаются в конгломерат тесно связанных, но до статочно независимых небольших фирм-партнеров, работающих пи главную компанию — носитель торговой марки или брэнда

3. Развитие прямых связей с потребителями, работа по их индивидуальным заказам.

В заключение очертим ближайшую перспективу развития экономики и общества, к которой приведет изменяющаяся роль информации. Используем и разовьем для этого некоторые тезисы одного из первых исследователей постиндустриального, или информационного, общества Д. Белла', которые были сформулированы им уже давно, но не утратили актуальности, а также П. Дракера²:

1) центральное место в информационном обществе принадлежит знаниям;

2) на основе знаний быстрыми темпами создаются и применяются в деятельности компаний новые интеллектуальные технологии;

3) отвечая на рост интеллектуальных технологий, быстро растет класс людей, обладающих необходимыми знаниями;

4) власть постепенно переходит к меритократии, т.е. высокообразованным профессионалам;

5) основным ресурсным ограничением является нехватка информации;

6) экономика информационного общества характеризуется как экономика информации;

7) изменяется характер труда: все чаще становится трудно определить — работает ли сотрудник в данный момент. Исчезает необходимость примитивного контроля занятости — остается контроль по результатам. Вместе с тем отпадает нужда в постоянном присутствии сотрудника в офисе в рабочее время. Иногда отпадает необходимость и в физическом офисе — достаточно логического или виртуального;

8) существенно меняется роль женщин и мужчин в организации семейной жизни: возможность организовать работу вне заранее определенного «присутственного места» разрушает привычную схему распределения ролей в семье. Виртуальные организации с удаленным доступом к трудовым ресурсам позволяют женщине уделять больше внимания работе, а мужчине — семье.

Гораздо большее количество специалистов получают возможность заниматься творческим трудом и непосредственно участвовать в создании стоимости продуктов.

9) Очевидна главная мысль — знания становятся основным ресурсом прогрессивного развития не только индивида, но и бизнеса, экономики компаний, отрасли. Для будущих молодых специалистов это означает, что обладающий глубокими знаниями выпускник университета имеет большой стартовый капитал и серьезное конкурентное преимущество на современном рынке труда.

5. Данные – информация – знания

Данные — это фиксированные сведения о событиях и явлениях. Будучи полученными, данные могут храниться произвольное время в неизменном виде. *Информация* — это обработанные данные, представленные в виде, пригодном для принятия решений получателем. Информация появляется при необходимости решить конкретную задачу или ответить на запрос на основании имеющихся данных. Примерами данных могут служить: линии, оставляемые на бумажной ленте самописцами (например, кардиограмма или кривая данных сейсмографа); рентгеновский снимок части тела больного; колонки чисел, представляющие котировки акций на бирже.

Информация — это обработанные данные, которые представлены в виде, пригодном для принятия получателем решений или проведения аналитических исследований.

Знания — это обработанная информация, использованная и используемая для принятия решений и решения задач, а также сведения о способах обработки информации для приведения ее к виду, пригодному для принятия решений.

В информационных системах можно найти примеры информации всех видов. Данные, как информация с первичных документов (счетов, накладных и т.д.), заносятся в базу данных и хранятся тоже как данные. Сформированный запрос инициирует поиск необходимых данных в базе, их обработку и представление в заданном виде получателю уже в качестве информации.

2. Свойства данных информации и знаний

Свойства данных. Репрезентативность данных означает способность собранных данных адекватно отобразить свойства описываемого ими явления. Важное значение здесь имеют:

- правильный отбор объектов для сбора данных;
- определение набора существенных признаков для измерения;
- достаточное количество объектов;
- соответствие данных формулировке задачи, для решения которой осуществляется сбор данных.

Точность данных. Принято выделять:

- формальную точность, измеряемую значением единицы по следнего разряда числа;
- реальную точность, измеряемую значением единицы послед него разряда, достоверность которого гарантируется;
- максимальную, или достижимую, точность, которая может быть получена при конкретных условиях сбора данных;
- необходимую точность, определяемую требованиями задачи, для решения которой данные собираются. Достоверность данных — это способность представлять опи сываемые объекты с заданной по условиям решаемой задачи точ ностью.

6. Свойства данных информации и знаний

Актуальность информации определяется степенью сохранения ее пригодности для принятого решения и зависит от того, в течение какого периода времени сохраняется репрезентативность данных, использованных для получения информации. Если данные устаревают или теряют способность представлять описываемый ими процесс, говорят о необходимости «актуализации» данных, т.е. их обновления.

Своевременность информации означает, что она получена в нужный момент принятия решения, без опоздания. Информация, поступившая после принятия решения, скорее всего, не нужна. С другой стороны, не всякая заблаговременность предоставления информации хороша: может быть утрачена ее актуальность.

Свойства знаний. Как мы знаем, знания формируются из обработанной информации, используемой и уже использовавшейся для принятия решений, так что положительные свойства и данных, и информации в знаниях молчаливо предполагаются. Поэтому знания характеризуются свойствами несколько иного типа.

Знания могут существовать в следующих видах:

- предметный или конкретный, использующий информацию из конкретной области, — это методики принятия решений для конкретно поставленной задачи (например, как спилить де рево, как обработать древесину, как сделать лодку и весла, как плыть на лодке по реке и т.д.);
- концептуальный, или обобщающий, использующий инфор мацию из многих областей и определяющий, как извлекать знания из информации, — это методологии (например, прин ципы кораблестроения, землеведения, управления компания ми и т.д.);
- метазнания (знания о знаниях) — генерируют новые знания. Примеры метазнаний в конкретной области:

химия — таблица Менделеева, предсказавшая появление новых химических элементов; генетика — теория гомологических рядов Н.И. Вавилова, предсказывающая наличие растений с данными свойствами в конкретном районе.

7. Экономическая информация и ее виды

Основу экономической информации составляют транзакционные данные. Транзакция — это определенный факт хозяйственной жизни, фиксированные сведения об этом факте и составляют транзакционные данные. Обработанные транзакционные данные порождают транзакционную и аналитическую информацию. Примером транзакции может служить сделка по купле-продаже готовой продукции.

По отношению к функциям управления экономическая информация подразделяется: а) на нормативно-справочную; б) плановую; в) учетную; г) аналитическую.

Нормативно-справочная информация представляет собой экономические, технологические, материальные и трудовые нормативы, такие как тарифные ставки, оклады, ставки налогообложения, которые используются при решении задач учета. Нормативно-справочная информация поступает извне и является исходным материалом для последующей обработки.

Плановая информация отражает явления и события, которые должны быть совершены в будущем (стратегическое планирование, тактическое планирование), в системе управления реализуется прямой связью. Плановая информация в зависимости от функционального уровня структуры управления предприятием подразделяется: а) на прогнозную; б) перспективную; в) технико-экономическую; г) оперативно-календарную.

Учетная информация отражает уже совершившиеся события в хозяйственной деятельности предприятия, в системе управления реализуется обратной связью..

Аналитическая информация возникает в процессе обработки плановой, учетной и нормативно-справочной информации при оценке производственно-хозяйственной и финансовой деятельности предприятия за определенный период времени.

Управление предприятием и все бизнес-процессы основываются на информации.

В систему управления поступает информация, которая разделяется на два информационных потока: а) обрабатываемую информацию; б) необрабатываемую информацию.

По отношению к системе управления информация подразделяется на входную и выходную. Входная информация поступает в орган управления извне. По отношению к процессу управления входная информация подразделяется на внутреннюю и внешнюю.

Внутренняя информация включает совокупность данных, возникающих в экономическом объекте в форме учетно-статистической отчетности и оперативной информации (отчеты, договора, заявки, заказы и т. д.). Внутренняя информация состоит из первичной информации (информации, поступающей непосредственно от объекта управления) и вторичной информации (информации, полученной в результате обработки первичной информации).

Внешняя информация объединяет сведения о состоянии внешней среды окружения экономического объекта (информация о рынке, конкурентах, поставщиках, информация от вышестоящего руководства и пр.). Это различные директивы вышестоящих организаций, информация от поставщиков и покупателей, информация банка, информация о ценах на продукцию на рынках сбыта и пр.

8. Понятие энтропии системы

Для определения меры информации необходимо ввести понятие меры неопределенности. Неопределенность — неперемное свойство любого хозяйственного или управленческого решения: такие решения — это выбор из нескольких возможных вариантов, и полной уверенности, что выбран действительно лучший, практически никогда не бывает. Даже в простой ситуации, выходя утром из дома, мы принимаем решения о том, как лучше одеться и взять ли зонт: существует опасность промокнуть, если будет обещанный по прогнозу дождь. Уменьшение неопределенности выбора лучшего решения возможно благодаря получению новых сведений или дополнительной информации.

Принятой мерой неопределенности системы α является энтропия, обозначаемая $H(\alpha)$. При получении сообщения β энтропия системы — $H\beta(\alpha)$. Как мы заметили, может быть, что $H\beta(\alpha) < H(\alpha)$, $H\beta(\alpha) > H(\alpha)$ и $H\beta(\alpha) = H(\alpha)$, — все зависит от того, что за сообщение p получено. Интересно, что именно разность $H\beta(\alpha) - H(\alpha)$ оказывается важной характеристикой полученного сообщения p .

Этой важной характеристикой сообщения p о системе α является количество информации $I\beta(\alpha)$, содержащееся в сообщении β о системе α :

$$I\beta(\alpha) = H(\alpha) - H\beta(\alpha)$$

Понятно, что величина $I\beta(\alpha)$ может быть положительной (когда сообщение уменьшает неопределенность), отрицательной (когда неопределенность растет) и нулевой (когда сообщение не несет информации, полезной для принятия решения). В последнем случае $H\beta(\alpha) = H(\alpha)$, т.е. неопределенность системы по получению сообщения p не изменилась, и количество информации в P равно нулю. Другим крайним случаем является ситуация, когда сообщение β полностью снимает неопределенность и $H\beta(\alpha) = 0$. Тогда сообщение β содержит полную информацию о системе α и $I\beta(\alpha) = H(\alpha)$. Теперь для определения количества информации нам надо понять, как оценивать

энтропию системы. В общем случае энтропия системы, имеющей η возможных состояний ($H(\alpha)$), согласно формуле Шеннона равна

$$H(\alpha) = -\sum P_i \log P_i$$

где P_i — вероятность того, что система находится в i -и состоянии.

9. Количественная мера информации. Формула Шеннона

Нам необходимо научиться оценивать степень неопределенности различных ситуаций, опытов. Для самых простых опытов, имеющих k равновероятных исходов, степень неопределенности измеряется с помощью самого числа k : при $k = 1$ никакой неопределенности нет, так как исход предопределен, но не случаен. При росте числа возможных исходов предсказание результата опыта становится все более затруднительным, так что естественно предположить, что мера степени неопределенности является функцией k — $j(k)$, причем $j(1) = 0$ и $j(k)$ монотонно растет с ростом k .

Кроме того, надо научиться оценивать неопределенность нескольких опытов. Рассмотрим два независимых опыта a и b (т.е. таких два опыта, что любые сведения об исходе первого никак не меняют вероятностей исходов второго). Если опыт a имеет p равновероятных исходов, а опыт b — q равновероятных исходов, то сложный опыт ab , состоящий в одновременном выполнении опытов a и b , очевидно, обладает большей неопределенностью, чем каждый опыт a или b в отдельности.

Клод Шеннон в 1950 г. предложил в качестве меры неопределенности системы α с k состояниями энтропию $H(\alpha)$:

$$H(\alpha) = -\sum P_i \log P_i$$

где P_i — вероятность того, что система находится в i -и состоянии.

Энтропия равна нулю только в одном случае, когда все вероятности P_i равны нулю, кроме одной, которая равна единице. Это точно описывает отсутствие неопределенности: система находится всегда в одном и том же состоянии.

Например, энтропия нашего алфавита из 32 букв: $H = \log 32 = 5$ бит. Энтропия десятичного набора цифр: $H = \log 10 = 3,32$ бита. Энтропия системы, в которой отдельно хранятся 32 буквы и 10 цифр: $H = \log (32 \cdot 10) = 5 + 3,32 = 8,32$ бита.

10. Синтаксическая мера информации

Синтаксическая мера информации отображает структурные характеристики информации и не затрагивает его смыслового содержания. На синтаксическом уровне учитываются тип носителя, способ представления информации, скорость передачи и т.д.

В качестве синтаксической меры количество информации представляет объем данных.

Объем данных V_d в сообщении (измеряется количеством символов (разрядов) в этом сообщении. Как мы упоминали, в двоичной системе счисления единица измерения — бит. На практике наряду с этой «самой

мелкой» единицей измерения данных чаще применяется более крупная единица — байт, равная 8 бит. Для удобства в качестве измерителей используются кило- (10^3), мега- (10^6), гига- (10^9) и тера- (10^{12}) байты и т.д. В знакомых всем байтах измеряется объем кратких письменных сообщений, толстых книг, музыкальных произведений, изображений, а также программных продуктов. Понятно, что эта мера никак не может характеризовать того, что и зачем несут эти единицы информации.

Синтаксическая мера информации определяет отношения информации и технологии, семантическая — информации и получателя.

11. Семантическая мера информации.

Итак, одной синтаксической меры информации явно недостаточно для характеристики сообщения: в нашем примере с погодой в последнем случае сообщение приятеля содержало ненулевой объем данных, но в нем не было нужной нам информации. Заключение о полезности информации следует из рассмотрения содержания сообщения. Для измерения смыслового содержания информации, т.е. ее количества на семантическом уровне, введем понятие «тезаурус получателя информации».

Тезаурус — это совокупность сведений и связей между ними, которыми располагает получатель информации. Можно сказать, что тезаурус — это накопленные знания получателя.

В очень простом случае, когда получателем является техническое устройство — персональный компьютер, тезаурус формируется «вооружением» компьютера — заложенными в него программами и устройствами, позволяющими принимать, обрабатывать и представлять текстовые сообщения на разных языках, использующих разные алфавиты, шрифты, а также аудио- и видеоинформацию из локальной или всемирной сети. Если компьютер не снабжен сетевой картой, нельзя ожидать получения на него сообщений от других пользователей сети ни в каком виде. Отсутствие драйверов с русскими шрифтами не позволит работать с сообщениями на русском языке и т.д. Если получателем является человек, его тезаурус — это тоже своеобразное интеллектуальное вооружение человека, арсенал его знаний.

Относительной мерой количества семантической информации служит коэффициент содержательности C , который определяется как отношение количества семантической информации к ее объему данных V_d , содержащихся в сообщении β

$$C = I_c / V_d$$

12. Прагматическая мера информации.

Теперь рассмотрим отношение информации и бизнеса с помощью прагматической меры информации. Эта мера определяет полезность информации (ценность) для достижения получателем поставленной цели и является величиной относительной, субъективной, обусловленной особенностями использования этой информации для принятия решения.

Приведем два показателя, оценивающих прагматическую меру информации.

1. Приращение вероятности достижения цели. Если до получения сообщения p вероятность достижения цели была p_0 , а после получения — p_1 , то ценность информации, полученной из сообщения (3, можно оценить с помощью показателя I_p :

$$I_p = \log P1/P0 \bullet$$

Если сообщение не изменило вероятность достижения цели и $p_x = p_0$, ценность полученной с ним информации нулевая.

В центре города вы спрашиваете у прохожего, который выглядит как местный житель, о том, как пройти к искомому вами Старокоромысловскому переулку. Самостоятельно его найти вы не можете, и до получения ответа вероятность достижения цели ненулевая, но близка к нулю $p_0 = 0,01$. Может быть, что из полученного ответа вы ровно ничего не поняли и, поблагодарив, пошли дальше, имея прежнюю вероятность достижения цели: $p_x = p_0$ и $L = 0$. Позже вам вдруг повезло, и следующий встречный так хорошо объяснил путь до Старокоромысловского, что вы даже поняли, что доберетесь за 5 минут: I_p стала практически равной единице — $p_1 = 0,99$. Так что прагматическая мера информации, ее ценность в данной модели измерения равна $\log (0,99/0,01) = \log 99 = 6,63$ бита, или почти 2 дита.

2. Прагматическая мера (ценность) информации оценивается величиной изменения целевой функции, обусловленной получением информации. Измеряется в тех же самых единицах, в которых измеряется целевая функция. Целевая функция служит для определения экономического результата принятия решения (экономического эффекта) или, проще, для количественной оценки конкретного варианта решения. Она может оценивать величину прибыли (в рублях, долларах, евро и т.д.), получаемой в случае принятия данного решения, или измерять величину соответствующих данному решению расходов имеющегося набора ресурсов (в килограммах, метрах, штуках и т.д.).

13. Понятие алгоритма

Понятие алгоритма является центральным во всем курсе информатики. Слово «алгоритм» принято связывать с именем арабского ученого Аль-Хорезми.

Алгоритмом называют упорядоченную совокупность точных(формализованных) и полных команд исполнителю алгоритма,

задающих порядок и содержание действий, которые он должен выполнить для нахождения решения любой задачи из рассматриваемого класса задач.

Отметим основные свойства алгоритмов:

- 1) *привязка к языку.* Каждый конкретный алгоритм формулируется в рамках некоторой теории и оформляется с использованием средств, определенных в этой теории. Обычно это некоторый алгоритмический язык, например язык формул, язык блок-схем или язык программирования. Алгоритмический язык представляет собой систему обозначений и правил для записи алгоритмов и особенностей их выполнения;
- 2) *дискретность.* Алгоритм представляет собой структурированное конечное множество элементарных действий (инструкций, команд, предписаний, директив, операторов — используются разные термины). Все типы команд задаются заранее списком, результат выполнения одной команды предопределяет выбор следующей команды;
- 3) *детерминированность.* В каждый момент времени работы алгоритма не должно быть неопределенности в выборе следующей команды и данных, над которыми эта команда должна работать. Другими словами, алгоритм должен быть полностью формальным и определенным;
- 4) *массовость.* Алгоритм должен быть применим к целому классу задач, а не к одной задаче. Обычно это достигается за счет разнообразия информации на входе алгоритма;
- 5) *повторяемость.* Алгоритм должен давать один и тот же результат при исполнении с одними и теми же входными данными;
- 6) *конечность.* Алгоритм должен давать решение задачи или вывод, что решения не существует, за конечное число шагов.

14. Принципы Фон Неймана. ЭВМ и его структура

С 1943 г. группа ученых в США начала конструировать вычислительную машину на основе электронных ламп. В 1945 г. к работе был привлечен знаменитый математик Джон фон Нейман. В результате был подготовлен доклад о принципах построения этой машины. Он был опубликован фон Нейманом, и поэтому общие принципы функционирования компьютеров получили название принципов фон Неймана. Первый компьютер, в котором были воплощены принципы фон Неймана, был построен в 1949 г. английским ученым Морисом Уилксом.

Принципы фон Неймана представляют собой ряд положений, выполнение которых необходимо для эффективной работы вычислительной машины:

- компьютер компонуется из нескольких основных устройств;
- для хранения информации используется специальное запоминающее устройство;

- данные представлены в запоминающем устройстве в форме двоичных чисел;
- арифметические и логические операции выполняются арифметико-логическим устройством;
- выполнение программ в вычислительной машине контролируется устройством управления;
 - программа, задающая работу компьютера, хранится в том же запоминающем устройстве, в котором хранятся данные (принцип хранимой программы);
 - для ввода и вывода информации используются отдельные устройства ввода-вывода.

Принцип хранимой программы позволяет обрабатывать команды программы так, как если бы они были данными, организуя наиболее эффективное выполнение программ. Большинство современных компьютеров в основных чертах соответствуют принципам, предложенным фон Нейманом.

Структура компьютера – это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства – от основных логических узлов компьютера до простейших схем. Кратко сформулируем классические принципы устройства ЭВМ.

Использование двоичной системы счисления для представления чисел. В докладе Неймана были продемонстрированы преимущества двоичной системы для технической реализации узлов компьютера, удобство и простота выполнения в ней арифметических и логических операций. В дальнейшем ЭВМ стали обрабатывать текстовую, графическую, звуковую и другие виды информации, но по-прежнему двоичное кодирование данных составляет информационную основу любого современного компьютера.

Принцип программного управления. Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Принцип однородности памяти. Программа также должна храниться в виде набора нулей и единиц, причем в той же самой памяти, что и обрабатываемые ей числа. С точки зрения хранения и способов обработки принципиальная разница между программой и данными отсутствует.

Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Адресом ячейки фактически является её номер; таким образом, местонахождение информации в ОЗУ также кодируется в виде чисел.

15. Принцип работы компьютера. Система счисления

Система счисления — способ записи чисел с помощью заданного набора специальных символов (цифр) и сопоставления этим записям реальных значений. Все системы счисления можно разделить на непозиционные и позиционные. В непозиционных системах счисления, которые появились значительно раньше позиционных, смысл каждого символа не зависит от того места, на котором он стоит. Примером такой системы счисления является римская, в которой для записи чисел используются буквы латинского алфавита. При этом буква I всегда означает единицу, буква V — пять, X — десять, L — пятьдесят, C — сто, D — пятьсот, M — тысячу и т.д. Например, число 264 записывается в виде CCLXIV. Недостатком непозиционных систем является отсутствие формальных правил записи чисел и, соответственно, арифметических действий с многозначными числами. Правила выполнения вычислений с многозначными числами в позиционной системе счисления были разработаны средневековым математиком Мухамедом аль-Хорезми и в Европе были названы алгоритмами (от латинского написания имени аль-Хорезми – Algorithmi).

В вычислительной технике применяются позиционные системы счисления. Позиционных систем счисления существует множество и отличаются они друг от друга алфавитом — множеством используемых цифр. Размер алфавита (число цифр в нем) называется основанием системы счисления. Последовательная запись символов алфавита (цифр) изображает число. Позиция символа в изображении числа называется разрядом. Разряду с номером 0 соответствует младший разряд целой части числа.

Двоичная система счисления имеет набор цифр $\{0, 1\}$, $p=2$. В общем виде, используя формулу (1), двоичное число можно представить выражением:

(3)

Например, число 101101_2 можно записать так:

$$101101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Двоичная система счисления имеет особую значимость в информатике: внутреннее представление любой информации в компьютере является двоичным, т.е. описывается набором символов только из двух знаков 0 и 1.

Шестнадцатеричная система счисления имеет набор цифр $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$, $p = 16$. Для изображения чисел в шестнадцатеричной системе счисления требуются 16 цифр. Для обозначения первых десяти цифр используются цифры десятичной системы счисления, шесть остальных — первых шесть прописных букв латинского алфавита.

16. Представление информации в памяти компьютера.

Вся информация в компьютере представляется в цифровой форме. Для чисел это представление является естественным. Для нечисловой информации (например, текста) используется стандартный прием: все возможные значения нумеруются и вместо самих значений хранятся их номера (которые играют роль кодов). Так, для представления текстовой информации используется таблица символов, содержащая все символы алфавита, которые могут встретиться в тексте, а текст, хранящийся в памяти компьютера, заменяется списком номеров символов в этой таблице. Аналогично кодируется информация других видов. В лю-0ом случае содержание представляемых нечисловых данных, хранящихся в компьютере, зависит от таблиц нумерации (называемых таблицами кодирования).

Различают две основные формы представления числа в памяти компьютера. При первом способе все разряды выделенных для хранения числа байтов памяти последовательно нумеруются и двоичные цифры числа непосредственно записываются в соответствующие биты памяти. Один бит выделяется для представления знака числа (0 — плюс, 1 — минус). При втором способе число представляется в так называемой нормализованной (или экспоненциальной) форме: $X = M * 10^n$, где число M (называемое мантиссой) заключено от 1 до 10, число n (называемое порядком) — целое.

В память компьютера заносятся цифры числа, но при этом считается, что на определенном месте этой записи стоит десятичная точка. Такая форма записи называется представлением числа с фиксированной точкой.

Представление нормализованных чисел называется представлением числа с плавающей точкой. Оно используется для хранения величин, которые могут принимать значения в очень большом диапазоне. В памяти компьютера порядок и мантисса хранятся отдельно в форме двоичных целых чисел со знаком.

Для хранения в памяти компьютера более сложных объектов, таких, как видеоизображения или звуки, описания этих объектов преобразуются в числовую форму. Существует достаточно много способов кодирования такого вида информации, но в итоге изображение или звук представляются в виде последовательности нулей и единиц, которые размещаются в битах памяти компьютера и при необходимости извлекаются оттуда и интерпретируются определенным образом.

17. Этапы разработки алгоритма.

Для решений задач с помощью компьютера разрабатываются этапы:

- постановка задачи;
- составление алгоритма;
- определение структуры данных;
- выбор языка программирования и составление программы;

- трансляция программ и отладка;
- реализация рабочей программы и анализ результатов.

В зависимости характера решаемых задач и уровню сложности, некоторые этапы вышеперечисленных может быть не использован.

Эффективная и корректная постановка задач предопределяет успешное решение задач.

Для создания алгоритма решений, разрабатываются отдельные блоки решения и создают блок-схемы алгоритма. На дальнейшем этапе анализируются исходные данные, определяются типы данных таких, как числовые, текстовые, массив, запись, стек, список, структурные и т.д.

Следующим этапом решения задач на компьютере является программирование.

Компьютерные программы создают *программисты* — люди, обученные процессу их составления (*программированию*). Мы знаем, что программа — это логически упорядоченная последовательность команд, необходимых для управления компьютером (выполнения им конкретных операций), поэтому программирование сводится к созданию последовательности команд, необходимой для решения определенной задачи

Управлять компьютером нужно по определенному *алгоритму*. Алгоритм — это точно определенное описание способа решения задачи в виде конечной (по времени) последовательности действий. Такое описание еще называется *формальным*. Для представления алгоритма в виде, понятном компьютеру, служат *языки программирования*. Сначала всегда разрабатывается алгоритм действий, а потом он записывается на одном из таких языков. В итоге получается текст программы — полное, законченное и детальное описание алгоритма на языке программирования. Затем этот текст программы специальными служебными приложениями, которые называются *трансляторами*, либо переводится в машинный код, либо исполняется.

Самому написать программу в машинном коде весьма сложно, причем эта сложность резко возрастает с увеличением размера программы и трудоемкости решения нужной задачи. Условно можно считать, что машинный код приемлем, если размер программы не превышает нескольких десятков байтов и нет потребности в операциях ручного ввода/вывода данных.

Поэтому сегодня практически все программы создаются с помощью языков программирования. Теоретически программу можно написать и средствами обычного человеческого (естественного) языка — это называется программированием на *метаязыке* (подобный подход обычно используется на этапе составления алгоритма), но автоматически перевести такую программу в машинный код пока невозможно из-за высокой неоднозначности естественного языка.

Языки программирования — искусственные языки. От естественных они отличаются ограниченным числом «слов», значение которых понятно транслятору, и очень строгими правилами записи команд (*операторов*). Совокупность подобных требований образует *синтаксис* языка

программирования, а *смысл* каждой команды и других конструкций языка — его *семантику*. Нарушение формы записи программы приводит к тому, что транслятор не может понять назначение оператора и выдает сообщение о синтаксической ошибке, а правильно написанное, но не отвечающее алгоритму использование команд языка приводит к семантическим ошибкам (называемым еще логическими ошибками или ошибками времени выполнения).

Процесс поиска ошибок в программе называется *тестированием*, процесс устранения ошибок — *отладкой*.

С помощью языка программирования создается не готовая программа, а только ее текст, описывающий ранее разработанный алгоритм. Чтобы получить работающую программу, надо этот текст либо автоматически перевести в машинный код (для этого служат программы-*компиляторы*) и затем использовать отдельно от исходного текста, либо сразу выполнять команды языка, указанные в тексте программы (этим занимаются программы-*интерпретаторы*).

Интерпретатор берет очередной оператор языка из текста программы, анализирует его структуру и затем сразу исполняет (обычно после анализа оператор транслируется в некоторое промежуточное представление или даже машинный код для более эффективного дальнейшего исполнения). Только после того, как текущий оператор успешно выполнен, интерпретатор перейдет к следующему. При этом, если один и тот же оператор должен выполняться в программе многократно, интерпретатор всякий раз будет выполнять его так, как будто встретил впервые. Вследствие этого, программы, в которых требуется осуществить большой объем повторяющихся вычислений, могут работать медленно. Кроме того, для выполнения такой программы на другом компьютере там также должен быть установлен интерпретатор — ведь без него текст программы является просто набором символов

По-другому можно сказать, что интерпретатор моделирует некую виртуальную вычислительную машину, для которой базовыми инструкциями служат не элементарные команды процессора, а операторы языка программирования.

Компиляторы полностью обрабатывают весь текст программы (он иногда называется *исходный код*). Они просматривают его в поисках синтаксических ошибок (иногда несколько раз), выполняют определенный смысловой анализ и затем автоматически переводят (*транслируют*) на машинный язык — генерируют машинный код. Нередко при этом выполняется *оптимизация* с помощью набора методов, позволяющих повысить быстродействие программы (например, с помощью инструкций, ориентированных на конкретный процессор, путем исключения ненужных команд, промежуточных вычислений и т. д.). В результате законченная программа получается компактной и эффективной, работает в сотни раз быстрее программы, выполняемой с помощью интерпретатора, и может быть перенесена на другие компьютеры с процессором, поддерживающим

соответствующий машинный код.

Основной недостаток компиляторов — трудоемкость трансляции языков программирования, ориентированных на обработку данных сложной структуры, часто заранее неизвестной или динамически меняющейся во время работы программы. Тогда в машинный код приходится вставлять множество дополнительных проверок, анализировать наличие ресурсов операционной системы, динамически их захватывать и освобождать, формировать и обрабатывать в памяти компьютера сложные объекты, что на уровне жестко заданных машинных инструкций осуществить довольно трудно, а для ряда задач практически невозможно

С помощью интерпретатора, наоборот, допустимо в любой момент остановить работу программы, исследовать содержимое памяти, организовать диалог с пользователем, выполнить сколь угодно сложные преобразования данных и при этом постоянно контролировать состояние окружающей программно-аппаратной среды, благодаря чему достигается высокая надежность работы. Интерпретатор при выполнении каждого оператора проверяет множество характеристик операционной системы и при необходимости максимально подробно информирует разработчика о возникающих проблемах. Кроме того, интерпретатор очень удобен для использования в качестве инструмента изучения программирования, так как позволяет понять принципы работы любого отдельного оператора языка.

В реальных системах программирования перемешаны технологии и компиляции, и интерпретации. В процессе отладки программа может выполняться по шагам, а результирующий код не обязательно будет машинным — он даже может быть исходным кодом, написанным на другом языке программирования (это существенно упрощает процесс трансляции, но требует компилятора для конечного языка), или промежуточным машинно-независимым кодом абстрактного процессора, который в различных компьютерных архитектурах станет выполняться с помощью интерпретатора или компилироваться в соответствующий машинный код.

18. Методы представления алгоритмом.

Для строгого задания различных структур данных и алгоритмов их обработки требуется иметь такую систему формальных обозначений и правил, чтобы смысл всякого используемого предписания трактовался точно и однозначно. Соответствующие системы правил называют языками описаний. К средствам описания алгоритмов относятся следующие основные способы их представления: словесный, графический (блок-схема), псевдокоды, программный, диаграмма Несси—Шнейдермана

Словесный способ записи алгоритмов представляет собой последовательное описание основных этапов обработки данных, заданное в произвольном изложении на естественном языке. Этот способ записи основан на общепринятых средствах общения между людьми. Его удобно

использовать на начальном этапе алгоритмизации задачи.

Рассмотрим понятие словесного представления алгоритма на примере нахождения произведения n натуральных чисел — факториала числа n ($c = n!$), то есть вычисления по формуле $c = 1 \times 2 \times 3 \times \dots \times n$.

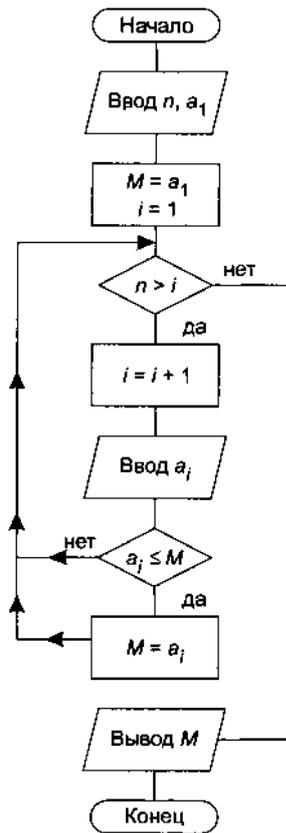
Этот процесс может быть записан в виде следующей последовательности шагов (инструкций, пунктов, указаний):

1. Присваиваем переменной c значение, равное единице, и переходим к следующему шагу.
2. Присваиваем переменной g значение, равное единице, и переходим к следующему шагу.
3. Вычисляем выражение $g \times c$, полученное значение присваиваем переменной c и переходим к следующему шагу.
4. Проверяем, равно ли значение переменной g значению переменной gg . Если $g = gg$, то вычисления прекращаем. Если $g < gg$, то увеличиваем g на единицу и переходим к шагу 3.

5. Запись алгоритма с помощью графических объектов в виде блок-схемы (ГОСТ 19.701-90, ИСО 5807-85) применяется довольно широко для представления простых алгоритмов небольшого размера. Однако по мере роста сложности отображаемого фрагмента алгоритма (программы) его логическая структура перегружается деталями и связями («спагетти») и схема становится нечитабельной. По этой причине в настоящее время блок-схемы алгоритмов используются в основном для иллюстрации программ¹.

6. При графическом представлении алгоритма с помощью схемы каждый пункт алгоритма отображается на схеме некоторой геометрической фигурой (блоком) и дополняется элементом словесного описания. Блоки на схемах соединяются линиями потоков информации. Основное направление потока информации идет сверху вниз и слева направо (стрелки могут не указываться), снизу вверх и справа налево (стрелки обязательны). Количество входящих линий для блока не ограничено. Выходящая линия должна быть одна (исключение составляет блок решения).

В виде блок-схемы представлен алгоритм нахождения минимального числа M в последовательности из n чисел a_1, a_2, \dots, a_n ($n \neq 0$).



Блок-схема алгоритма нахождения минимума в последовательности чисел

Широкое распространение получил способ записи алгоритма при помощи псевдокода. Этот способ записи является промежуточным — к нему прибегают перед записью алгоритма в терминах выбранного языка программирования. Псевдокод представляет собой удобный для практического применения промежуточный язык. Это симбиоз естественного языка и языка программирования. Псевдокод похож на язык программирования тем, что в нем, с одной стороны, присутствуют некоторые инструкции и конструкции языка программирования, а с другой — допускается словесная или формульная запись там, где сложно использовать средства языка программирования.

Перечислим основные ключевые слова алгоритмического языка: **алг** (алгоритм), **рез** (результат), **нач** (начало), **кон** (конец), **арг** (аргумент), **знач** (значение), **тип**, **вещ** (вещественный), **цел** (целый), **лит** (литерный), **таб** (табличный), **сим** (символьный), **не**, **то**, **если**, **и**, **все**, **или**, **выбор**, **иначе**, **нц** (начало цикла), **кц** (конец цикла), **от**, **до**, **шаг**, **для**, **пока**, := (оператор присваивания), **при**, **да**, **нет**.

Дополнительные ключевые слова: **запись**, **истина**, **лог** (логический), **ложь**, **массив**, **множество**, **функция**, **дано**, **надо**, **ввод**, **вывод**, **утв** (утверждение).

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм. Однако на практике в качестве

исполнителей алгоритмов используются компьютеры или иные вычислительные устройства (однокристальные микроЭВМ, промышленные компьютеры, технологические контроллеры и др.), поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем. Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке — программой для компьютера.

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования — чем язык менее детален, тем выше его уровень. По этому критерию можно выделить следующие уровни языков программирования .

- **машинно-ориентированные языки** — машинные языки и языки ассемблера;
- **машинно-независимые языки** — языки высокого уровня.
- **машинно-независимые языки** — языки высокого уровня.



Классификация языков программирования

Машинно-ориентированные языки — это языки низкого уровня, требующие указания мелких деталей процесса обработки данных. Языки же высокого уровня имитируют естественные языки, используя некоторые слова естественного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на процедурные, логические и объектно-ориентированные.

- **Процедурные**, или **алгоритмические**, языки (ВaБю, РаБса1, С и др.) предназначены для однозначного описания алгоритмов в виде некоторой последовательности операторов языка.
- **Логические** языки (Prolog, Lisp и др.) ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи, из которого должно следовать решение.
- **Объектно-ориентированные** языки (Object Pascal, С++, Java, С# и др.), в основе которых лежит понятие объекта, сочетают в себе данные и действия над ними. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути, описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы

взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

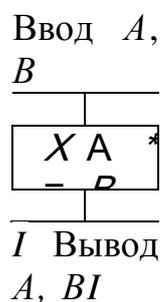
19. Линейные алгоритмы.

Для выполнения типичных последовательностей действий в алгоритме разработаны базовые алгоритмические конструкции в виде определенного набора блоков и стандартных средств их соединения. К базовым алгоритмическим конструкциям относятся *линейные, разветвляющиеся и циклические*.

Линейные алгоритмы - это алгоритм в котором действия осуществляются последовательно друг за другом.

Пример линейного алгоритма:

(Начало)



(Конец)

Линейный алгоритм

В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в разветвляющиеся алгоритмы входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (действий).

20. Разветвляющие алгоритмы.

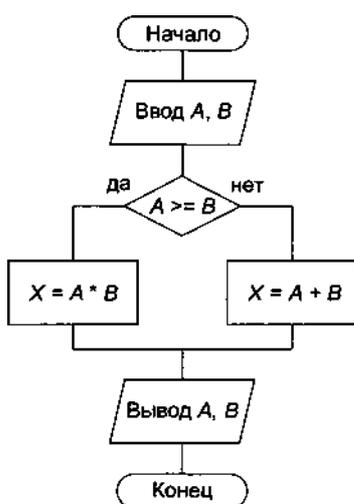
Для выполнения типичных последовательностей действий в алгоритме разработаны базовые алгоритмические конструкции в виде определенного набора блоков и стандартных средств их соединения. К базовым алгоритмическим конструкциям относятся *линейные, разветвляющиеся и циклические*.

В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в разветвляющиеся алгоритмы входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (действий). Пример ветвящегося алгоритма приведен на

Разветвляющие алгоритм - это алгоритм в котором действия выполняются по одной из возможных ветвей решения заданий в зависимости от выполнения условий.

В качестве условия в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может соблюдаться (быть истинным) или не соблюдаться (быть ложным). Такое утверждение может быть выражено как словами, так и формулой. Таким образом, алгоритм ветвления состоит из условия и двух последовательностей команд.

Многократные повторения одних и тех же действий обеспечиваются при помощи циклических алгоритмов. В цикл входят в качестве составляющих элементов блок проверки условия и блок, называемый телом цикла. Перед началом цикла осуществляются операции присваивания начальных значений тем объектам, которые используются в теле цикла.



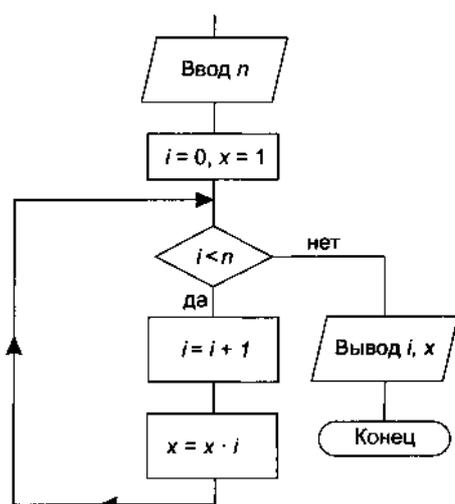
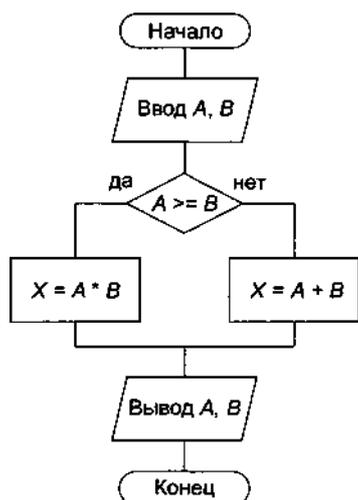
Разветвляющийся алгоритм

21. Циклические алгоритмы.

Для выполнения типичных последовательностей действий в алгоритме разработаны базовые алгоритмические конструкции в виде определенного набора блоков и стандартных средств их соединения. К базовым алгоритмическим конструкциям относятся *линейные, разветвляющиеся и циклические*.

Многократные повторения одних и тех же действий обеспечиваются при помощи циклических алгоритмов. В цикл входят в качестве составляющих элементов блок проверки условия и блок, называемый телом цикла. Перед началом цикла осуществляются операции присваивания начальных значений тем объектам, которые используются в теле цикла.

Разветвляющийся алгоритм



Алгоритм с циклическим элементом

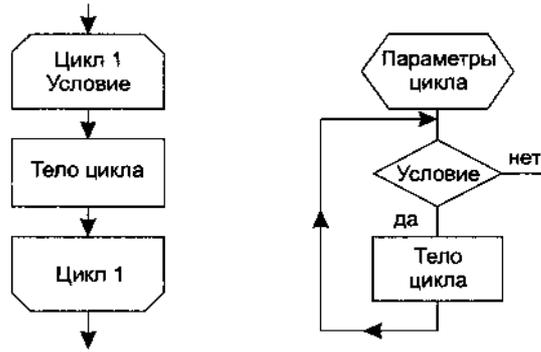
Циклический алгоритм - это алгоритм в котором некоторая часть операций (тело цикла) выполняется многократно.

Слово «многократно» в определении циклического алгоритма не значит «до бесконечности». Организация циклов, не приводящих к остановке в выполнении алгоритма, является нарушением требования его результативности — получения результата за конечное число шагов.

Если тело цикла расположено после проверки условий (цикл с предусловием), то при определенных условиях тело цикла не выполнится ни разу. Такой вариант организации цикла, управляемый предусловием, называется *циклом с предусловием*.

а) Общее обозначение б) Реализация

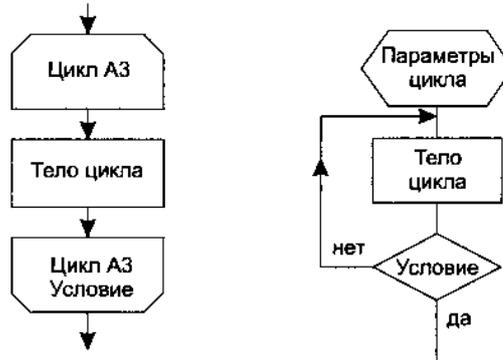
по ГОСТ 19.701-90



Цикл с предусловием

Другой вариант построения цикла состоит в том, что тело цикла выполняется по крайней мере один раз, и будет повторяться до тех пор, пока условие не станет истинным. Такая организация цикла, когда его тело расположено перед проверкой условия, носит название *цикла с постусловием*. Истинность условия в этом случае является условием выхода из цикла.

а) Общее обозначение б) Реализация по ГОСТ 19.701-90



Цикл с постусловием

Таким образом, цикл с предусловием завершается, когда условие ложно, а цикл с постусловием — когда условие становится истинным.

Третий тип циклов, в которых тело цикла **выполняется заданное число раз** (то есть еще до начала выполнения цикла точно известно, сколько раз он будет выполнен), реализуется с помощью счетчика. В цикле со счетчиком значения счетчика в теле цикла итеративно увеличиваются.

Процесс решения сложной задачи довольно часто сводится к решению нескольких более простых задач методом последовательной детализации. Соответственно, процесс разработки сложного алгоритма может разбиваться на этапы составления отдельных алгоритмов, которые называются вспомогательными. Каждый такой вспомогательный алгоритм описывает решение какой-либо подзадачи.

Процесс построения алгоритма методом последовательной детализации

состоит в следующем. Сначала алгоритм формулируется в «обобщающих» блоках, которые могут быть непонятны исполнителю и записываются, как вызовы вспомогательных алгоритмов. Затем происходит детализация, и все вспомогательные алгоритмы детализируются до уровня команд, понятных исполнителю.

22. Структура персонального компьютера. Внутри-машинный системный интерфейс.

Архитектурой называют прежде всего систему составляющих компьютер устройств и взаимосвязей между ними, а также совокупность правил, по которым происходит это взаимодействие.

Главными устройствами являются процессор и память. Именно взаимодействием этих компонентов определяется возможность компьютера производить вычисления. Линии связи, по которым данные передаются из процессора в память и обратно, называются *шиной*. Обычно это электрический провод (сейчас появились оптоволоконные провода). Линий связи в компьютере много, и они выполняют множество разных функций. Принято делить линии связи всей шины на шину данных, шину адреса, шину управления и шину питания. Кроме того, процессор и память должны быть связаны проводами со многими другими устройствами компьютера. В современных компьютерах одна и та же шина используется для обмена данными как между процессором и памятью, так и между процессором и всеми портами ввода-вывода. Такая шина называется *общей шиной*.

Реально часть устройств подключается к общей шине не непосредственно, а через одну из вспомогательных шин, которая, в свою очередь, присоединяется к общей шине. Такие шины называются *локальными шинами*. *Внутримашинным системным интерфейсом* называется вся система связей и сопряжений узлов и блоков компьютера между собой. Интерфейс включает совокупность электрических параметров, электронных микросхем сопряжения с компонентами компьютера, соглашений о передаче и преобразовании сигналов. Интерфейс с общей шиной называется *односвязным*.

23. Микро процессор и его функции

Процессор состоит из огромного количества электронных микросхем, сосредоточенных в микроскопическом объеме. Быть может, процессор является самым сложным устройством в мире.

Регистры процессора, представляющие собой наиболее быстродействующую часть памяти компьютера, конструктивно расположены внутри процессора, и время доступа к данным в регистрах значительно меньше, чем к данным в оперативной памяти.

Центральный процессор (ЦП) — это устройство, которое выполняет обработку информации в соответствии с выполняемой компьютером программой, находящейся в оперативной памяти и состоящей из отдельных команд, понятных для процессора. В каждой команде содержатся сведения о том, откуда взять исходные данные, какую операцию над ними выполнять и куда поместить результат. Процессор выполняет следующие функции:

- чтение команд из оперативной памяти и их дешифрация;
- чтение данных из оперативной памяти и портов ввода-вывода;
- запись данных в оперативную память или их пересылка в порты ввода-вывода;
- прием и обработка запросов и команд от адаптеров внешних устройств;
- выработка управляющих сигналов для всех прочих устройств компьютера.

Функционально процессор состоит из двух компонентов -операционной и интерфейсной частей. Операционная часть включает устройство управления, арифметико-логическое устройство и процессорную память (регистры общего назначения — РОН). Интерфейсная часть включает микросхемы управления шиной и портами, а также адресный и командный регистры. Устройство управления является наиболее сложной частью процессора. Оно вырабатывает сигналы, которые управляют все ми устройствами компьютера, и процессором в частности.

- Арифметико-логическое устройство (АЛУ) предназначено для выполнения арифметических и логических операций. Операнды операции перед этим должны быть размещены в регистрах общего назначения. Результат также помещается в регистр общего назначения.

24. Оперативная память. ОЗУ.ПЗУ

То, что до сих пор для простоты называлось оперативной памятью компьютера, следует назвать его внутренней адресуемой памятью. Адресуемой она называется потому, что доступ к ней осуществляется на основе единой адресации ячеек памяти. Основную часть адресуемой памяти составляет оперативное запоминающее устройство (ОЗУ, или оперативная память). Иногда ее называют памятью прямого доступа (*Random Access Memory — RAM*). Большинство типов компьютеров содержит также постоянное запоминающее устройство (ПЗУ), или память только для чтения (*Read-Only Memory — ROM*).

Оперативное запоминающее устройство является основным средством хранения информации при работе компьютера. Все дан ные, которые должны быть обработаны, сначала должны быть по мещены в оперативную память. В оперативной памяти также хра нятся промежуточные и конечные результаты

работы компьютера. Кроме того, там же располагаются все работающие программы.

От быстродействия ОЗУ, которое измеряется временем про цессов перемещения информации из оперативной памяти в про цессор и обратно, напрямую зависит быстродействие всего ком пьютера, поэтому одним из основных направлений в развитии вычислительной техники является повышение быстродействия ОЗУ. Технически ОЗУ устроено таким образом, что при выключе нии электрического питания информация в его ячейках стирает ся. В выключенном компьютере информация хранится на внеш них магнитных носителях информации, а при включении необ ходимые программы и данные пересылаются с этих внешних носителей в оперативную память.

Постоянное запоминающее устройство содержит ту началь ную программу, которая начинает работать при включении ком пьютера, а также некоторые служебные программы операцион ной системы. В частности, в ПЗУ хранятся программные модули так называемой базовой системы ввода-вывода (*Base Input Output System — BIOS*), а также диагностические программы, предназна ченные для проверки исправности и обслуживания аппаратуры самого компьютера. В постоянной памяти хранится также про грамма первоначальной загрузки главной обслуживающей про граммы компьютера — операционной системы. Содержимое ПЗУ сохраняется при выключении компьютера.

Для ускорения работы ОЗУ используется так называемая кэш-память процессора, которая является промежуточной между ОЗУ и процессором, имеет меньший объем, чем ОЗУ, но зато большее быстродействие. При наличии кэш-памяти данные из ОЗУ сначала переписываются туда и лишь затем в регистры процессора. При повторном обращении к памяти те данные, которые уже находятся в кэш-памяти, сразу переносятся в регистры процессора, за счет чего экономится время.

25. Системная шина. Шина данных, адреса, управления. Шина расширений. Локальные шины.

Системная шина представляет собой совокупность линий для передачи сигналов, объединенных по их назначению (данные, пдреса, управление). Основной функцией системной шины является обмен информацией между процессором и остальными электронными компонентами компьютера. По системным шинам осуществляется передача информации (по шине данных), адресация устройств (по шине адреса) и обмен специальными служебными сигналами (по шине управления).

Перемещение информации между оперативной памятью и процессором и между оперативной памятью и портами происходит по системе соединений, которые называются *шиной данных*. Для увеличения скорости передачи биты информации передаются одновременно по нескольким линиям шины. Количество линий называется *разрядностью* шины. В современных

персональных компьютерах используется 64-разрядная шина данных, по ней можно передавать 8 байтов данных одновременно.

Для правильной организации работы компьютера процессор и память должны обмениваться не только данными, но и управляющими сигналами. Для этого в компьютере предусмотрены, кроме шины данных, еще две шины: шина адреса и шина управления (на самом деле есть еще шина питания, по которой на весь компьютер подается питание).

При прохождении адреса по шине адреса активизируется именно байт памяти с указанным адресом. Остальные управляющие сигналы, необходимые для правильного выполнения операции, посылаются по шине управления.

Передачей информации по системной шине управляет одно из подключенных устройств или специально предназначенное для этого устройство, называемое *арбитром шины*.

Компоненты устройства, назначением которых является взаимодействие с другими устройствами, называются его *интерфейсом*, а правила, которым интерфейс обязан удовлетворять, — *протоколами*

26. Устройства ввода информации.

Устройства ввода информации в компьютер очень разнообразны: клавиатура, мышь, трекбол, джойстик, сканер, диджитайзер, устройство распознавания речи и т.д. Для того чтобы увеличить количество возможных комбинаций клавиш для передачи в компьютер, в клавиатуре аппаратно реализована возможность менять номер клавиши в том случае, если при этом будут нажаты другие специально выделенные клавиши, такие, как Shift, Ctrl или Alt. *Мыши* и трекболы являются координатными устройствами ввода информации в компьютер. *Трекбол* представляет собой «перевернутую» мышь: он закреплен, а крутится только его шар. Это позволяет существенно повысить точность управления курсором. *Джойстик* — это рычаг, закрепленный на подставке, способный перемещаться на шарнирах в двух направлениях. *Сканер* предназначен для ввода в компьютер текстовой и графической информации непосредственно с изображения. Изображение разбивается на точки, которые считываются с помощью оборудования, аналогичного тому, которое употребляется при ксерокопировании. Если сканируется текст, то он обрабатывается специальными, очень сложными программами распознавания текста.

Электронный планшет (или диджитайзер) является координатным преобразователем, который используется в основном для задач САПР.

27. Устройства вывода информации

Устройства вывода информации предназначены для представления результатов работы компьютера в «человеческом» виде: кроме видеомонитора, о котором шла речь выше, это принтер, предназначенный для бумажной печати текстовой и графической информации, плоттер (или графопостроитель), предназначенный для печати графиков и чертежей, и звуковые колонки.

Управление работой большинства устройств ввода-вывода компьютера осуществляется при помощи портов. Напомним, что портом называется виртуальная ячейка, соответствующая внешнему входу (или выходу) в компьютере.

Принтер является основным средством бумажного вывода.

Плоттер — это фактически большой принтер, предназначенный для построения чертежей. Он ориентирован на работу со специальными программами, например при печати чертежей и системах проектирования (системах автоматизации проектирования — САПР). Принципы его действия те же, что и у принтера.

Монитор (дисплей) компьютера предназначен для вывода текстовой и графической информации. Мониторы персональных компьютеров могут работать в двух режимах — текстовом или графическом. В графическом режиме экран состоит из точек, полученных разбиением экрана на большое количество строк и столбцов. Эти точки называются пикселями. Количество пикселей на экране называется разрешающей способностью монитора в данном режиме. Сейчас мониторы персональных компьютеров могут работать в режимах 480 x 640, 600 x 800, 768 x 1024, 864 x 1152 или 1024 x 1280 пикселей. В графическом режиме экран состоит из точек, полученных разбиением экрана на большое количество строк и столбцов. Количество пикселей на экране называется разрешающей способностью монитора. Сейчас мониторы персональных компьютеров могут работать в режимах 480 x 640, 600 x 800, 768 x 1024, 864 x 1152 или 1024 x 1280 пикселей.

28. Основные понятия программного обеспечения.

Программа — это упорядоченная последовательность команд для компьютера, записанная на специальном языке и помещенная в файл.

Совокупность программ и данных, предназначенных для их обработки, называется программным обеспечением ПК.

Все программы можно разбить на три группы:

1. прикладные программы. Т.е. каждая из этих программ служит для решения конкретной задачи (например, для создания текста существует

программа текстовый редактор LEXICON, MS Word; табличные процессоры - MS Excell, 1С; для создания рисунка – графический редактор, Paint, PhotoShop, CorelDraw).

2. инструментальные программы. Эти программы служат для создания новых программ, написанных на различных языках программирования. Visual Basic, Delphi, Pascal, C++.

3. системные программы. Они служат для управления работой ПК и выполняют различные операции по обслуживанию пользователя. Таких программ очень мало, но они очень важны, т.к. без них не сможет работать ни одна другая программа. Системные программы связывают все устройства ПК в единую систему, поэтому называются системными.

Инструментарий программирования – это средства, предназначенные для создания ПО, т.е. того же системного и прикладного ПО. Его составляют разнообразные языки и среды программирования.

29. Системное программное обеспечение

Системное ПО – является основным ПО, неотъемлемой частью компьютера. Без него невозможно взаимодействовать ни с одним устройством ЭВМ. Именно системное ПО руководит слаженной работой всех элементов компьютерной системы, как на аппаратном уровне, так и на программном.

Программы	Пояснение	Примеры
Операционные системы	Комплекс программ, распределяющих ресурсы компьютерной системы и организующих работу других программ	MS-DOS Windows Unix
Файловые менеджеры	Программы, обеспечивающие более комфортное общение пользователя с командами ОС	Windows Total FAR Commander Commander
Программы диагностики	Проверяют работу основных устройств компьютера	
Антивирусные программы	Программы обнаружения компьютерных вирусов и их уничтожения	DrWeb Nod32 Антивирус Касперского
Программы обслуживания дисков	Программы проверки целостности логической и физической структуры дисков, дефрагментация	
Архиваторы	Программы упаковки файлов и группы файлов для уменьшения занимаемого ими места на диске	WinRar WinZip

30. Пакеты прикладных программ.

Это комплекс взаимосвязанных программ для решения задач определенного класса. Выделяются следующие виды ППП:

1. проблемно-ориентированные. Используются для тех проблемных областей, в которых возможна типизация функций управления, структур данных и алгоритмов обработки. Например, это ППП автоматизации бухучета, финансовой деятельности, управления персоналом и т.д.;
2. автоматизации проектирования (или САПР). Используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм;
3. общего назначения. Поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры, графические редакторы, системы управления базами данных (СУБД);
4. офисные. Обеспечивают организационное управление деятельностью офиса. Включают органайзеры (записные и телефонные книжки, календари, презентации и т.д.), средства распознавания текста;
5. настольные издательские системы – более функционально мощные текстовые процессоры;
6. системы искусственного интеллекта. Используют в работе некоторые принципы обработки информации, свойственные человеку. Включают информационные системы, поддерживающие диалог на естественном языке; экспертные системы, позволяющие давать рекомендации пользователю в различных ситуациях; интеллектуальные пакеты прикладных программ, позволяющие решать прикладные задачи без программирования.

Пакеты общего назначения

Поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры (редакторы), графические редакторы, системы управления базами данных (СУБД), пакеты программ мультимедиа, пакеты демонстрационной графики.

31. Понятие операционной системы.

Операционной системой называется комплекс программных средств, обеспечивающих функционирование отдельных устройств компьютера и их взаимодействие, а также взаимодействие устройств компьютера и прикладных программ. Операционная система обеспечивает работоспособность компьютера и создает ту среду, в которой функционируют компьютерные программы. Кроме того, при включении компьютера в локальную сеть операционная система может выполнять часть сетевых функций.

Современные операционные системы обеспечивают:

- управление выполнением программ;
- управление памятью;
- управление вводом-выводом;
- управление файловой системой;
- обработку прерываний;
- управление работой аппаратных устройств компьютера;
- взаимодействие с ОС пользователей и программ (пользовательский интерфейс ОС);
- многозадачный режим;
- многопользовательский режим;
- автоматическое определение текущей конфигурации устройств компьютера (*Plug & Play*);
- тестирование устройств компьютера и функциональных подсистем ОС и исправление возможных нарушений;
- разделение ресурсов компьютера между программами (оперативной памяти, времени процессора, внешней памяти, периферийных устройств и др.);
- безопасность функционирования компьютера — защиту памяти, используемой одной программой или пользователем, от использования другой программой или пользователем;
- корректность совместного владения данными несколькими программами или пользователями.

32. Функции операционной системы.

Типовой функцией при работе на компьютере служит управление различными внешними устройствами (винчестерами, гибкими дисками, мониторами, клавиатурой и т.д.). Эти устройства выпускаются самыми различными фирмами. Одинаковые по назначению, они могут иметь различные характеристики и управляться совершенно разным способом. Для управления внешним устройством марки определенной фирмы требуется своя специальная программа, которая называется *драйвером устройства*. Современные операционные системы включают большой набор разнообразных типовых драйверов. Более того, они умеют при включении

компьютера автоматически определять тип используемого устройства и подключать к работе соответствующий драйвер (эта услуга называется *Plug & Play*). Все современные операционные системы (например, *UNIX* и *Windows NT*) обеспечивают многозадачный (одновременное выполнение нескольких программ) и многопользовательский (одновременная работа нескольких пользователей) режимы работы компьютера. *Многозадачный режим* означает одновременную работу на компьютере нескольких программ. Конечно, если компьютер содержит один центральный процессор, он может выполнить только одну программу, однако операционная система так организует работу компьютера, что создается иллюзия одновременной работы нескольких программ.

Режимом разделения времени называется такая организация многозадачной и многопользовательской работы программ на одном компьютере, при которой каждой программе в зависимости от приоритета выделяется квант времени, в течение которого работает только эта программа.

33. Понятие файла. Атрибуты файлов. Файловая система и ее функции..

Для организации доступа программ к данным, размещенным на внешних носителях, современные ОС используют сложную файловую систему. *Файлом* называется набор данных, хранящийся в виде единого логического объекта на внешнем носителе. Для операций операционной системы файл представляет собой структурную единицу, с которой пользователи программы могут совершать типовые действия, такие, как:

- создание, удаление, чтение в оперативную память, перезапись измененных данных;
- доступ к файлам по символьным именам;
- возможность доступа к файлам других пользователей; ¹ управление доступом к собственным файлам;
- восстановление файлов в случае повреждения.

Файловая система скрывает от пользователя картину фактического размещения данных во внешней памяти, предоставляя ему возможность идентификации файла путем описания его имени и местоположения в файловой системе и для указания операции над файлом.

Для установления соответствия между именем файла и его физическим размещением во внешней памяти используются специальные таблицы размещения файлои в памяти {File Allocation Table — FAT), с помощью которых ОС по имени файла находит диски внешней памяти и сектора на дисках, в которых размещен файл.

В системе *Windows 2000* используется файловая система *NFTS*, поддерживающая диски и файлы больших объемов, чем системы *FAT 16* и *FAT32*, используемые в *Windows 98*. Таблица *NFTS* более безопасна и обладает большими возможностями восстановления данных в случае повреждения данных в файлах или повреждения *FAT*.

34. Операционная система Windows основные элементы.

Операционная система *Windows* для IBM PC-совместимых персональных компьютеров также разработана фирмой *Microsoft* и обеспечивает большее количество возможностей и удобств для пользователей и программистов по сравнению с *MS DOS*. Она основана на принципах, придуманных создателями машин фирмы *Apple*. Наиболее важная отличительная особенность *Windows*, из которой она и получила свое название («Окна»), — это взаимно однозначная связь программ, работающих в текущий момент и операционной системе, и специальных графических объектов на экране монитора, называемых окнами. Это позволяет называть *Windows* многозадачной системой. Одно из основных достижений *Windows* — удобный графический пользовательский интерфейс. Пользователю не необходимо вводить команды в виде текстовых строк. Достаточно ориентироваться в деталях картинки, представленной в текущий момент на экране, с помощью мыши выбрать нужный элемент и щелчком мыши на нем осуществить нужное действие. С помощью мыши можно перемещать объекты (*Drag & Drop* — «переместить и положить»), менять их размер, открывать и закрывать окна на экране. *Windows* — интегрированная система. В нее входит целый комплекс подпрограмм специального назначения: текстовые- и графические редакторы, программы настройки, программы печати и т.д., которые могут взять на себя часть функций, необходимых для работы приложений.

Система *Windows* поддерживает стандарт *Plug & Play* (англ. «включил и работай»). Основная его идея заключается в том, что каждое устройство, соответствующее этому стандарту, сообщает операционной системе при подключении устройства определенную информацию о себе, благодаря которой операционная система выполняет автоматическую конфигурацию периферийных устройств и разрешает аппаратные конфликты.

35. Назначение и основные понятия электронных таблиц.

Первые электронные таблицы (табличные процессоры) и сам термин «электронная таблица» появились в начале 1980-х годов. Наиболее известна программа *Microsoft Excel*, входящая в пакет *Microsoft Office*. *Excel* — это программное обеспечение, с помощью которого можно создавать таблицы, производить вычисления и анализировать данные. Программы такого типа называются электронными таблицами. В приложении *Excel* можно создавать таблицы, в которых автоматические

вычисляются итоговые значения для введенных числовых данных, печатать красиво оформленные таблицы и создавать простые графики.

Приложение Excel входит в состав пакета Office, представляющего собой набор программных продуктов для создания документов, электронных таблиц и презентаций, а также для работы с электронной почтой.

Электронная таблица – это электронная матрица, разделенная на строки и столбцы, на пересечении которых образуются ячейки с уникальными именами. Ячейки являются основным элементом электронной таблицы, в которые могут вводиться данные и на которые можно ссылаться по именам ячеек. К данным относятся: числа, даты, время суток, текст или символьные данные и формулы.

Программа Excel работает с матрицей, которая может включать до $216 = 65\,536$ строк и $28 = 256$ столбцов. Эта матрица называется листом. Один документ Excel называется книгой и может состоять из нескольких листов. Область применения Excel: планово – финансовые и бухгалтерские расчеты, учет материальных ценностей, системы поддержки принятия решений (СППР) и другие области применения.

36. Компьютерные сети и их назначение.

Компьютерной сетью называется группа компьютеров, объединенных линиями передачи данных и способных обмениваться информацией. Компьютеры могут располагаться в одной или в различных фирмах либо в различных географических точках. Компьютерные сети следует отличать от многомашинных комплексов. Многомашинным вычислительным комплексом называется несколько компьютеров, соединенных средствами сопряжения и играющих разную роль в едином вычислительном процессе.

Всемирная тенденция к объединению компьютеров в сети обусловлена рядом важных объективных причин, таких, как глобализация экономики, повышение уровня управления предпринимательскими и государственными структурами, появление новых видов информационных услуг. Пользователи, подключенные к компьютерной сети, могут получать и передавать сообщения по электронной почте, имеют доступ к информации вне зависимости от ее географического расположения, а также возможность пользоваться программным обеспечением различных фирм.

Без использования компьютерных сетей невозможно создать информационную систему, эффективно управляющую предприятием.

37. Классификация вычислительных сетей.

Все вычислительные сети можно классифицировать по ряду признаков. В зависимости от расстояний между ПК различают следующие вычислительные сети:

1. локальные вычислительные сети – ЛВС (LAN – Local Area Networks) – компьютерные сети, расположенные в пределах небольшой ограниченной территории (здании или в соседних зданиях) не более 10 – 15 км;
2. территориальные вычислительные сети, которые охватывают значительное географическое пространство. К территориальным сетям можно отнести городские (MAN - Metropolitan Area Network),
3. региональные (Regional computer network), национальные (National computer network) и глобальные (WAN - Wide Area Network) сети. Городские и региональные сети связывают абонентов района, города или области. Глобальные сети объединяют абонентов, удаленных между собой на значительное расстояние, находящихся в различных странах или континентах.

38. Эталонная модель сетевых взаимодействий открытых систем (модель ISO OSI).

Проблема стандартизации, характерная для современного развития общества в целом, в сфере распространения информации особенно остра из-за возникновения глобального информационного пространства. Представьте, что вы посылаете сообщение с одного конца земного шара на другой. Тогда необходимо, чтобы программа, формирующая сообщение, и программа, принимающая его, делали это по одним правилам: или это должна быть одна программа, что нереально, или сетевые программы различных фирм должны удовлетворять кем-либо установленным соглашениям — протоколам.

Для процедур межсетевого взаимодействия за основу берутся стандарты, разработанные Международной организацией по стандартизации (*International Standard Organization — ISO*), которые называются «Стандарты взаимодействия открытых систем» (*Open Systems Interconnection — Reference Model {OSI Ref. Model}*) или «Семиуровневая эталонная модель сетевого обмена». Уровни взаимодействия определяют все: от стандартов физического соединения компьютеров до протоколов обмена прикладного программного обеспечения.

Модель *ISO* разбивает все задачи передачи данных в сети на семь уровней:

- 1) физический (*Physical Layer*);
- 2) канальный (*Data Layer*);
- 3) сетевой (*Network Layer*);
- 4) транспортный (*Transport Layer*);
- 5) сеансовый (*Session Layer*);

- б) уровень представления данных (*Presentation Layer*);
- 7) прикладной (*Application Layer*).

39. Передача данных в сети.

Канальный уровень, или уровень управления линией передачи данных (иногда он называется линейным уровнем), управляет передачей пакетов в локальной сети. Заголовок протокола канального уровня описывает характеристики канала передачи данных между двумя узлами сети. На основе данных этого протокола происходит взаимодействие между драйверами устройств и устройствами, а также между операционной системой и драйверами устройства. Примерами протоколов канального уровня являются стандарты SLIP (Serial Line Internet Protocol) или PPP (Point to Point Protocol) из семейства протоколов TCP/IP.

Пакеты, непосредственно передаваемые по линиям связи, принято называть кадрами. Канальный уровень обеспечивает поддержку логической линии связи. Это означает, что он берет на себя все служебные функции, позволяющие безошибочно передавать данные в сети. При возникновении ошибок автоматически выполняется повторная посылка кадра. Кроме того, на уровне управления линией передачи данных обычно обеспечивается правильная последовательность передаваемых и принимаемых кадров: если один компьютер передает другому несколько блоков данных, то принимающий компьютер получит эти блоки именно в той последовательности, в какой они были переданы. При помощи канальных протоколов проверяется также наличие соединения между компьютерами. При наличии физического соединения необходимо подтверждение готовности сети к передаче информации, которое касается передающего и принимающего узлов, а также канала связи.

За канальный уровень отвечают обеспечение сетевого адаптера, сетевые драйверы и сетевая ОС.

40. Сеть моноканальной топологии (Технология Ethernet).

При организации взаимодействия узлов в локальных сетях основную роль играет используемый метод доступа к каналу связи. По приведенной классификации доступ к каналу связи обеспечивает канальный уровень. Большинство локальных сетей содержат несколько десятков компьютеров и территориально ограничены. Главное требование к ним — надежность, простота управления и приемлемая цена. Еще при возникновении локальных сетей перед разработчиками встала задача нахождения простого и дешевого решения для объединения компьютеров в вычислительную сеть. В результате было предложено несколько схем кабельных соединений между компьютерами. Три разные идеи были использованы в наиболее известных технологиях Token Ring, Arcnet и Ethernet.

Спецификацию Ethernet в конце 1970-х годов предложила компания Xerox Corporation. Сообщение, отправляемое одним узлом, принимается одновременно всеми остальными узлами, подключенными к общей шине. Но сообщение предназначено только для одного из них (оно включает в себя адрес узла назначения и адрес отправителя). Тот узел, которому предназначено сообщение, примет его, остальные его пропустят.

Метод доступа Ethernet является методом множественного доступа с прослушиванием канала связи. Перед началом передачи рабочая станция определяет, свободен канал или занят. Если канал свободен, станция начинает передачу. Ethernet исключает возможности одновременной передачи сообщений двумя или несколькими станциями.

Данная топология применяется в локальных сетях с архитектурой Ethernet (классы 10Base-5 и 10Base-2 для толстого и тонкого коаксиального кабеля соответственно).

Преимущества сетей шинной топологии:

- отказ одного из узлов не влияет на работу сети в целом;
- сеть легко настраивать и конфигурировать;
- сеть устойчива к неисправностям отдельных узлов.

Недостатки сетей шинной топологии:

- разрыв кабеля может повлиять на работу всей сети;
- ограниченная длина кабеля и количество рабочих станций;
- трудно определить дефекты соединений

41. Сеть кольцевой топологии. (Технология Токен Ring).

При топологии «кольцо» компьютеры подключаются к кабелю, замкнутому в кольцо. Сигналы передаются по кольцу в одном направлении и проходят через каждый компьютер. В отличие от пассивной топологии «шина», здесь каждый компьютер выступает в роли повторителя, усиливая сигналы и передавая их следующему компьютеру. Поэтому, если выйдет из строя один компьютер, прекращает функционировать вся сеть. Следовательно, трудно локализовать проблемы, а изменение конфигурации требует остановки всей сети. Оборудование для сетей с топологией кольцо более дорогостоящее.

К преимуществам можно отнести: устойчивость сети к перегрузкам (нет коллизий, отсутствует центральный узел) и возможность охвата большой территории. Кроме того, количество пользователей не оказывает большого влияния на производительность сети.

Метод доступа, используемый в технологии Token Ring, предусматривает следующие правила:

- все устройства, подключенные к сети, могут передавать данные, только получив разрешение на передачу (маркер);
- в любой момент времени только одна станция в сети обладает таким правом;
- данные, передаваемые одной станцией, доступны всем станциям сети.

42. Сеть звездообразной топологии (Технология Arcnet).

Технология Arcnet (*Attached Resource Computer Network*) — простая, недорогая, надежная и гибкая архитектура локальной сети, разработанная корпорацией Datapoints в 1977 г. По технологии Arcnet один из компьютеров создает специальный маркер (сообщение специального вида), который последовательно передается от одного компьютера к другому. Если станция желает передать сообщение другой станции, она дожидается маркера, добавляет к нему сообщение, дополненное адресами отправителя и получателя сообщения, и посылает его дальше; следующий узел также может присоединить к группе свое сообщение и т.д. В результате по ней и проходит поток из нескольких сообщений, возглавляемых кольцевым маркером. Компьютер, которому адресовано одно из сообщений потока, отцепляет его. Технология Arcnet может использоваться при любой топологии.

Метод доступа, используемый в технологии Arcnet, предусматривает следующие правила:

- устройства, подключенные к сети, могут передавать данные по очереди, получив разрешение на передачу (маркер специального вида);
- в любой момент времени передавать данные может только одна станция в сети;
- данные, передаваемые одной станцией, доступны всем станциям сети.

Передача каждого байта в Arcnet выполняется специальной но-сылкой, состоящей из 3 служебных старт/стоповых битов и 8 битов данных. В начале каждого пакета передается начальный разделитель, который состоит из 6 служебных битов. Прежде чем посылать информационное сообщение, отправитель посылает запрос о готовности к приему данных. После приема сообщения посылается извещение о факте получения и об отсутствии ошибок в пакете.

43. Структура сети Интернет.

Глобальные сети предназначены для максимально широкого обмена и распространения информации, так как они связывают абонентов в пределах целой страны, континента или всего земного шара. Фактически, глобальная сеть - это множество компьютеров, обменивающихся между собой информацией преимущественно посредством телефонных линий или спутников связи.

Узлы и магистрали сети Интернет - это ее инфраструктура, а в сети Интернет существует несколько сервисов или служб (E-mail, USENET, TELNET, WWW, FTP и др.), одним из первых сервисов является электронная почта E-mail. В настоящее время большая часть трафика в Интернет приходится на службу WorldWideWeb(всемирная паутина).

Принцип работы сервиса WWW был разработан физиками Тимом Бернес-Ли и Робертом Кайо в европейском исследовательском центре CERN (Женева) в 1989 году. В настоящее время Web – служба Интернет содержит миллионы страниц информации с различными видами документов. Компоненты структуры сети Интернет объединяются в общую иерархию. Интернет объединяет множество различных компьютерных сетей и отдельных компьютеров, которые обмениваются между собой информацией. Вся информация в Интернет хранится на Web-серверах. Обмен информацией между Web-серверами осуществляется по высокоскоростным магистралям. К таким магистралям относятся: выделенные телефонные аналоговые и цифровые линии, оптические каналы связи и радиоканалы, в том числе спутниковые линии связи. Серверы, объединенные высокоскоростными магистралями, составляют базовую часть Интернет. Пользователи подключаются к сети через маршрутизаторы местных поставщиков услуг Интернета или провайдеров (ISP), которые имеют постоянное подключение к Интернет через региональных провайдеров. Региональный провайдер, подключается к более крупному провайдеру национального масштаба, имеющего узлы в различных городах страны.

44. Адресация ресурсов в интернете. Доменная система имен.

Структура Интернет напоминает паутину, в узлах которой находятся компьютеры, связанные между собой линиями связи. Узлы Интернет, связанные высокоскоростными линиями связи, составляют базис Интернет. Как правило, это поставщики услуг (провайдеры). Оцифрованные данные пересылаются через маршрутизаторы, которые соединяют сети с помощью сложных алгоритмов, выбирая маршруты для информационных потоков.

Каждый компьютер в Интернет имеет свой уникальный адрес. В протоколе TCP/IP каждый компьютер адресуется четырьмя отделяемыми друг от друга точками десятичными числами, каждое из которых может иметь значение от 1 до 255. Адрес компьютера выглядит следующим образом:

19.226.192.108

Такой адрес называется IP-адресом. Пользователю неудобно запоминать такие адреса, которые к тому же могут изменяться. Поэтому в Интернет существует Доменная Служба Имен (DNS - Domain Name System), которая позволяет каждый компьютер назвать по имени. В сети существуют миллионы компьютеров, и чтобы имена не повторялись, они разделены по независимым доменам.

Таким образом адрес компьютера выглядит как несколько доменов, разделенных точкой:

<сегмент n>. ... <сегмент 3>.<сегмент 2>.<сегмент 1>.

Здесь сегмент 1 – домен 1 уровня, сегмент 2 – домен 2 уровня и т.д.

Доменное имя - это уникальное имя, которое данный поставщик услуг избрал себе для идентификации, например: ic.vrn.ru или yahoo.com

Домен 1 уровня обычно определяет страну местоположения сервера (ru – Россия; ua – Украина; uk – Великобритания; de – Германия) или вид организации (com – коммерческие организации; edu - научные и учебные организации; gov - правительственные учреждения; org – некоммерческие организации).

Когда вводится доменное имя, например, www.mrsu.ru, компьютер должен преобразовать его в адрес. Чтобы это сделать, компьютер посылает запрос серверу DNS, начиная с правой части доменного имени и двигаясь влево. Его программное обеспечение знает, как связаться с корневым сервером, на котором хранятся адреса серверов имён домена первого уровня (крайней правой части имени, например, ru). Таким образом, сервер запрашивает у корневого сервера адрес компьютера, отвечающего за домен ru. Получив информацию, он связывается с этим компьютером и запрашивает у него адрес сервера mrsu. После этого от сервера mrsu он получает адрес www компьютера, который и был целью данной прикладной программы.

45. Сервисы Интернет.

Существуют разнообразные сервисы Интернета. Основными являются электронная почта, телеконференции, списки рассылки, удаленный доступ к файлам, работа в режиме удаленного терминала, всемирная паутина.

Принято делить сервисы Интернета на сервисы интерактивные, прямые и отложенного чтения. В сервисах отложенного чтения запрос и получение информации разделены по времени, как, например, в электронной почте. Сервисы прямого обращения характерны тем, что информация по запросу возвращается немедленно. Однако от получателя информации не требуется немедленной реакции. К такому виду сервисов можно отнести передачу файлов. Сервисы, где требуется немедленная реакция на полученную информацию, относятся к интерактивным сервисам. Таким сервисом является Всемирная паутина.

Сетевые новости *Usenet*, или телеконференции, — еще один распространенный сервис Интернета. Узел сети, получивший сообщение, транслирует его всем узлам, находящимся в его списке рассылки. Таким образом, посланное кем-то сообщение распространяется, многократно дублируясь, по сети, достигая щ довольно короткие сроки всех участников телеконференций *Usenet* во всем мире. При этом в обсуждении интересующей вас темы может участвовать множество людей независимо от того, где они находятся физически.

Сервис *FTP* обеспечивает доступ к файлам в файловых архивах, хранящихся на специальных серверах. Название «*FTP*» совпадает с названием протокола, предназначенного для передачи файлов между разными компьютерами. Протокол *FTP* оптимизирован для передачи

файлов. Сервер *FTP* можно настраивать таким образом, что соединиться с ним можно не только под своим именем, но и под условным именем — *anonymous* (аноним).

46. Всемирная паутина (www).

Сеть WWW образуют миллионы *веб-серверов*, расположенных по всему миру. *Веб-сервер* является программой, запускаемой на подключенном к сети компьютере и передающей данные по протоколу HTTP.

Для идентификации ресурсов (зачастую файлов или их частей) в WWW используются идентификаторы ресурсов *URI* (Uniform Resource Identifier). Для определения местонахождения ресурсов в этой сети используются локаторы ресурсов *URL* (Uniform Resource Locator). Такие URL-локаторы представляют собой комбинацию URI и системы DNS.

Доменное имя (или IP-адрес) входит в состав URL для обозначения компьютера (его сетевого интерфейса), на котором работает программа веб-сервер.

На клиентском компьютере для просмотра информации, полученной от веб-сервера, применяется специальная программа — *веб-браузер*.

Основная функция веб-браузера - отображение гипертекстовых страниц (веб-страниц). Для создания гипертекстовых страниц в WWW изначально использовался язык HTML. Множество веб-страниц образуют *веб-сайт*.

Разработкой стандартов для сети Веб, начиная с 1994 года, занимается Консорциум W3C (World Wide Web Consortium), основанный и до сих пор возглавляемый Тимом Бернерсом-Ли.

Консорциум W3C — организация, разрабатывающая и внедряющая технологические стандарты для Интернета и WWW. Миссия W3C формулируется следующим образом: "Полностью раскрыть потенциал Всемирной паутины путем создания протоколов и принципов, гарантирующих долгосрочное развитие Сети". Две другие важнейшие задачи Консорциума — обеспечить полную "интернационализацию Сети" и сделать ее доступной для людей с ограниченными возможностями.

W3C разрабатывает для WWW единые принципы и стандарты, называемые "*Рекомендациями*", которые затем внедряются разработчиками программ и оборудования. Благодаря *Рекомендациям* достигается совместимость между программными продуктами и оборудованием различных компаний, что делает сеть WWW более совершенной, универсальной и удобной в использовании.

47. База данных. Классификация и элементы базы данных.

База данных(БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области

СУБД – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организация поиска в них необходимой информации

По технологии обработки БД подразделяются на централизованные и распределенные. Централизованные БД хранятся в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе.

Распределенная БД состоит из нескольких, возможных пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ.

Работа с такой базой осуществляется с помощью системы управления распределенной базой данных(СУРБД)

Понятие БД связано с такими понятиями структурных элементов, как поле , запись, файл.

Поле – это элементарная единица логической организации данных, которая соответствует неделимой единице информации- реквизиту.

Для описания поля используются следующие характеристики

Имя , например, Фамилия , Имя, Отчество. Дата рождения

Тип, например. 15 байт. Причем будет определяться максимально возможным количеством символов.

Запись – это совокупность логически связанных полей.

Экземпляр записи – это отдельная реализация записи, содержащая конкретные значения ее полей

Файл(таблица) – это совокупность экземпляров записей одной структуры

48. Сетевая и иерархические модели данных.

Ядром любой базы данных является модель данных.

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

Модель данных- совокупность структур данных и операций их обработки.

Иерархическая модель данных.

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево) .

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. Узел-это совокупность атрибутов данных, описывающих некоторые объекты. На схеме дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в БД определяется числом корневых записей.

Сетевая модель данных.

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с другим элементом.

Примером сложной сетевой структуры может служить структура БД, содержащей сведения о студентах, участвующих в научных проектах. Возможно участие одного студента в нескольких научных проектах, а также участие нескольких студентов в одном научном проекте.

49. Реляционная модель данных. Структуры данных реляционной модели

Понятие реляционный (англ. Relation- отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда. Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

4. Каждый элемент таблицы-один элемент данных ;
5. Все столбцы в таблицах однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
6. Каждый столбец имеет уникальное имя;
7. Одинаковые строки в таблице отсутствуют;
8. Порядок следования строк и столбцов может быть произвольным.

50. Система управление базами данных. Определение и основные понятия . Свойства и технология использования.

СУБД – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организация поиска в них необходимой информации.

СУБД предназначена для централизованного управления БД в интересах всех работающих в этой системе. По системе универсальности различают два класса СУБД:

Система общего назначения

Специализированные системы

СУБД общего назначения- это сложные программные комплексы предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией БД.

Специализированные СУБД создаются в редких случаях при невозможности использования СУБД общего назначения.

Современные СУБД обладают средствами обеспечения целостности данных и надёжной безопасности, что даёт возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил.

В качестве примера программных продуктов СУБД можно назвать:

dBASE 4 2.0, компании Borland International

Microsoft Access 2.0

Microsoft FoxPro 2.6 for DOS

Paradox for Windows.

Производительность СУБД оценивается:

Временем выполнения запросов;

Скоростью поиска информации в неиндексированных полях;

Максимальным числом параллельных обращений к данным в многопользовательском режиме;