

Ответы колоквиума по предмету "Информатика"

Тема 1. Предмет и содержание курса

1. Понятие информатики - разные подходы к определению.

Термин «информатика» происходит от французского слова *Informatique*, полученного комбинацией двух слов: *Information* (информация) и *Automatique* (автоматика). Французское словосочетание вкладывает в термин «информатика» смысл автоматической обработки информации.

Информатика — это наука об информации, способах ее сбора, хранения, обработки и представления с помощью электронных средств. Таким образом, информатика имеет дело с информационными процессами, как обобщенно называют получение, хранение, обработку и представление информации, происходящими в объектах произвольной природы: технических, биологических, социальных. Экономическая информатика — это наука об информационных системах, применяющихся для подготовки и принятия решений в управлении, экономике и бизнесе, а также об экономике этих систем.

Весьма авторитетный специалист в области информатики Кристен Нюгард (Kristen Nygaard) дает такое определение: информатика — это наука об информационных процессах и связанных с ними явлениях в обществе, природе и человеческой деятельности¹. В соответствии с этим определением поле деятельности информатики гораздо шире, чем то, что обычно изучает наука о компьютерах (*computer science*). Занимаясь определенным классом явлений, информатика не может претендовать на то, чтобы быть строгой наукой подобно математике. Явления, возникающие в связи с информационными процессами, могут быть самой различной природы и должны изучаться с позиций соответствующей науки, как это делается в физике, экономике, социологии, биологии и т.д. Известно, что далеко не все в этих науках описывается строгими математическими законами и наука не заканчивается там, где исчерпывают себя математические методы. К таким наукам относится и информатика.

Экономическая информатика — это наука об информационных системах, применяющихся для подготовки и принятия решений в управлении, экономике и бизнесе, а также об экономике этих систем.

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.

таким образом информатика изучает:

- информацию и ее свойства.
- информационные процессы:

- хранение;
- обработка;
- передача;

с помощью компьютерной технологии.

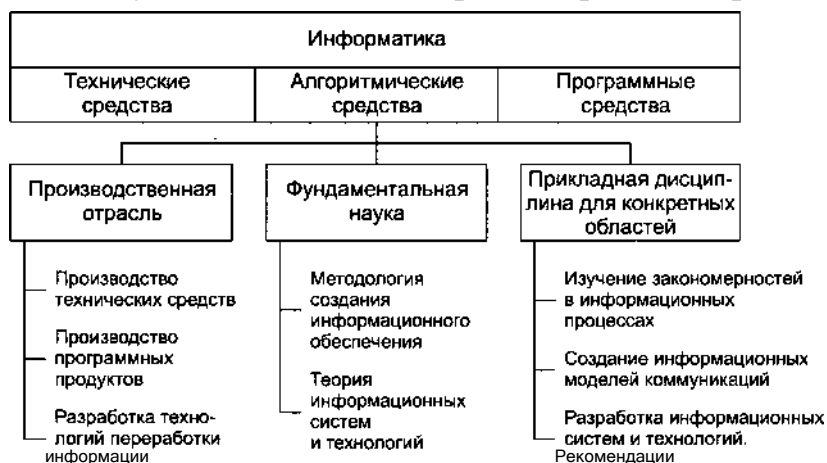
2. Информатика и ее структура.

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.

3. 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 интересуется, от отправителя — к получателю.

Передовая практика управления утверждает, что основная роль ИТ в организации — информационное обслуживание ее подразделений. Поскольку термин «информационное обслуживание» весьма 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 процесс выполняется средствами ИТ. Например, если юрист отыскивает и твердой копии некоторого кодекса необходимую для данного случая 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.

4. 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.

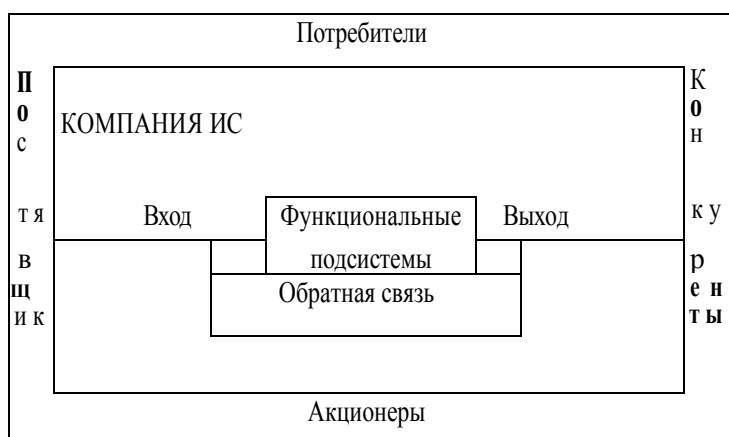


Схема функционирования ИС.

5. „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, метакапитализм.

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 на главную компанию — носитель торговой марки или бренда

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 больший стартовый капитал и серьезное конкурентное преимущество на современном рынке труда.

Тема II: «Данные, информация и знания.».

1. 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 последнего 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 описываемые объекты с заданной по условиям решаемой задачи точностью.

Свойства информации.

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 дерево, как обработать древесину, как сделать лодку и весла, как плыть на лодке по реке и т.д.);
- концептуальный, или обобщающий, использующий информацию из многих областей и определяющий, как извлекать знания из информации, — это методологии (например, принципы кораблестроения, земледелия, управления компаниями 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.

3. Формы существования и адекватности информации

Данные могут существовать в виде таблиц, публикуемых в газетах или на сайтах соответствующих компаний. Примером могут служить ежедневные сведения о котировках акций. В этом случае говорят о том, что данные *структурированы*.

Кроме того, данные могут получаться с изображений, из текстов газетных статей или программ телевидения. В этом случае они не могут быть непосредственно представлены в виде упорядоченной последовательности результатов измерений конкретных параметров, и такие данные обычно называют *неструктурированными*.

Знания появляются в результате многократного использования информации для принятия решения и могут существовать в двух видах: формальном (*tangible*) и неформальном (*intangible*). *Формальные* знания существуют, например, в виде документов, регламентирующих принятие решения, методов и методик подготовки и принятия решения, стандартов и нормативов.

Неформальные знания — это прежде всего знания и опыт квалифицированных специалистов, их интуиция, умение работать в группе, корпоративная культура.

Таким образом, формальные знания существуют в виде томов документов, учебников и монографий с описанием методов решения задач. Неформальные знания — менее конкретная категория, но реально имеющая ничуть не меньшую силу, влияющую на успех подготовки и принятия решения.

Адекватность информации — это уровень соответствия образа, создаваемого с помощью информации, реальному объекту, процессу, явлению. От степени адекватности информации зависит правильность принятия решения.

Выделяют три вида адекватности: семантическая, синтаксическая и прагматическая.

1. Синтаксический аспект отражает физические характеристики информации: способ представления, скорость передачи, тип носителя, способ кодирования, используемые каналы, надежность и безопасность передачи. Информация, рассматриваемая только с точки зрения синтаксиса, может считаться данными, так как в этом аспекте не рассматривается ее содержание.

2. Содержательная сторона характеризует семантический аспект информации, когда рассматриваются состав содержащихся сведений и связь между ними.

3. Прагматический аспект информации связан с ценностью информации для пользователя при принятии им решения.

4. Информация как предмет экономического анализа

5.

Профессор Е.З. Майминас писал: «Главным, коренным отличием является ее неуничтожаемость в процессе потребления (использования), возможность многократного потребления и не одним, а многими потребителями (пользователями). Информацию не теряет, не лишается ее и производитель — в процессе передачи потребителю она сохраняется и у производителя.

Отметим основные черты информации с экономической точки зрения, основываясь на списке Е.З. Майминаса.

Трудность однозначной фиксации потребителя при свободном {public} распределении информации. Это касается прежде всего данных, публикуемых в открытой печати (например, ежедневные сведения о котировках биржевых акций, публикуемые финансовыми изданиями). Общая аналитическая информация, основанная на этих данных, также может быть найдена в средствах массовой информации. Мировыми лидерами сегодня являются агентства *Reuter* и *Bloomberg*, превратившие представление информации в выгоднейший бизнес.

Невозможность точной стоимостной оценки полученной информации в общем случае. «Априорная» оценка в данном случае обозначает оценку информации до ее использования. На информационном рынке, как и на всяком другом, товар стоит столько, сколько за него платят. Но существует и бурно развивается рынок контрафактной, нелегальной продукции, где программное обеспечение, компьютерные игры, фильмы, музыкальные произведения стоят гораздо меньше, чем в официальных торговых точках. Здесь продавцы отказываются платить авторам или владельцам авторских прав их долю в цене продаваемого продукта.

6. 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.

Тема III: «Меры информации»

1.Понятие энтропии системы

Для определения меры информации необходимо ввести понятие меры неопределенности. Неопределенность — неперенное 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 β получено. Интересно, что именно разность $H\beta(\alpha) - H(\alpha)$ оказывается важной характеристикой полученного сообщения β .

Этой важной характеристикой сообщения β 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 β не изменилась, и количество информации в β 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 n 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.

2. 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 α и β (т.е. таких два опыта, что любые сведения об исходе первого никак не меняют вероятностей исходов второго). Если опыт α имеет p равновероятных исходов, а опыт β — q равновероятных исходов, то сложный опыт $\alpha\beta$, состоящий в одновременном

выполнении опытов аир, очевидно, обладает большей неопределенностью, чем каждый опыт а или р в отдельности.

Клод Шеннон в 1950 г. предложил в качестве меры неопределенности системы а с к состояниями энтропию $H(a)$:

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

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.

3.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.

4.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

$$C = I_c / V_d$$

5. 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 :

$$I_p = \log P_1 / P_0$$

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: $p_x = p_a$ 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: /?, стала практически равной единице — $p_i = 0,99$. 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.).

Тема IV: «Основы автоматизированной обработки информации»

1. 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 следу

ющей 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.

2. 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 komponуется 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.

3. 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.

4. 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 любом 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 представляет ся 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.

5.Выполнение компьютером машинной программы. Классификация ЭВМ.

В компьютере хранение данных и их обработка пространственно разделены: устройство компьютера, которое предназначено для хранения данных, называется памятью компьютера; устройство, которое производит различные вычисления и управляет работой компьютера, — центральным процессором. Работа компьютера полностью определяется той информацией, которая хранится в его рабочей памяти.

Память компьютера принято делить на три основных раздела: адресуемая память, регистры процессора и ячейки ввода-вывода, используемые портами компьютера. Задача последнего раздела отображать внешнюю среду при работе компьютера. Адресуемая память логически представляет собой последовательность ячеек, предназначенных для хранения информации. В современных компьютерах размер ячейки составляет 1 байт. Все байты оперативной памяти последовательно пронумерованы. *Адресом байта* называется его номер.

Центральный процессор — главное устройство компьютера. На него возложены две основные функции: во-первых, производить все вычисления и, во-вторых, управлять работой всех узлов компьютера. Регистры процессора представляют собой наиболее быстродействующую часть памяти компьютера. Регистров процессора несколько десятков, и они выполняют различные функции. Работа компьютера представляет собой последовательность элементарных операций. Каждая элементарная операция есть результат выполнения определенной машинной команды. Процесс выполнения машинной команды состоит из следующих трех шагов: выборка очередной команды, выполнение команды и вычисление адреса следующей команды.

Классификация ЭВМ

По принципу действия вычислительные машины делятся на три больших класса: аналоговые (АВМ), цифровые (ЦВМ) и гибридные (ГВМ).

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

- ЦВМ – вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее, в цифровой форме.
- АВМ - вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).

- ГВМ – вычислительные машины комбинированного действия
- Классификация ЭВМ по этапам создания.

По этапам создания и используемой элементной базе ЭВМ условно делятся на поколения:

- Первое поколение, 50-е годы; ЭВМ на электронных вакуумных лампах.
- Второе поколение, 60-е годы; ЭВМ на дискретных полупроводниковых приборах (транзисторах).
- Третье поколение, 70-е годы; ЭВМ на полупроводниковых интегральных схемах с малой и средней степенью интеграции (сотни – тысячи транзисторов в одном корпусе).
- Четвертое поколение, 80-е годы; ЭВМ на больших и сверхбольших интегральных схемах – микропроцессорах (десятки тысяч – миллионы транзисторов в одном
- Пятое поколение, 90-е годы; ЭВМ с многими десятками параллельно работающих микропроцессоров, позволяющих строить эффективные системы обработки знаний; ЭВМ на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных команд программы;
- Шестое и последующие поколения; оптоэлектронные ЭВМ с массовым параллелизмом и нейтронной структурой – с распределенной сетью большого числа (десятки тысяч) несложных микропроцессоров, моделирующих архитектуру нейтронных биологических систем.

Тема V. Основы алгоритмизации.

1. Этапы разработки алгоритма.

Для решений задач с помощью компьютера разрабатывается 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.

2. 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):

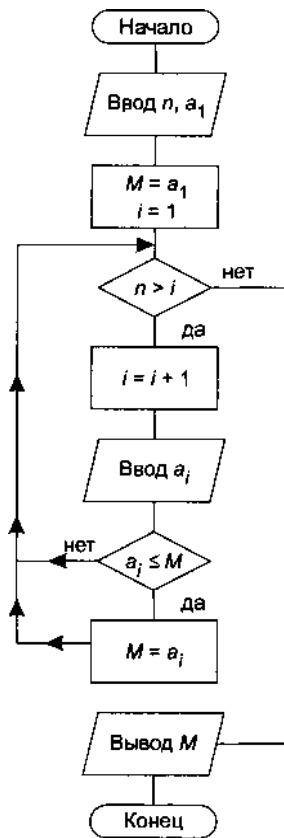
1. Присваиваем переменной s значение, равное единице, и переходим к следующему шагу.
2. Присваиваем переменной g значение, равное единице, и переходим к следующему шагу.
3. Вычисляем выражение $g \times s$, полученное значение присваиваем переменной s и переходим к следующему шагу
4. Проверяем, равно 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 $a_1, a_2, \dots, a_n$ ($n \geq 0$).

¹



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**.

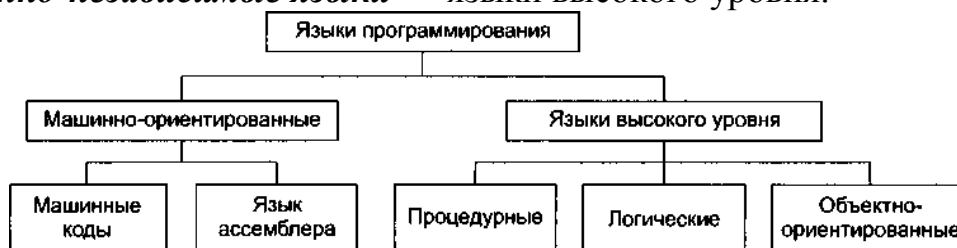
Дополнительные ключевые слова: **запись**, **истина**, **лог** (логический), **ложь**, **массив**, **множество**, **функция**, **дано**, **надо**, **ввод**, **вывод**, **утв** (утверждение).

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 (Вабю, РаБсал, 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, C++, 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.

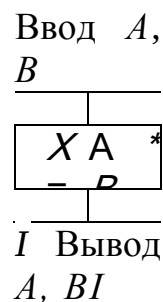
3. Линейные алгоритмы.

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 алгоритм- 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).

4. Разветвляющие алгоритмы.

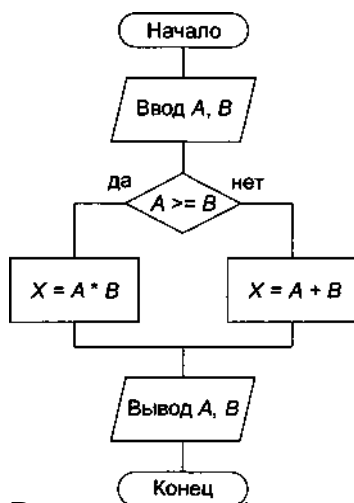
Для выполнения типичных последовательностей действий в алгоритме разработаны базовые алгоритмические конструкции в виде определенного набора блоков и стандартных средств их соединения. К базовым алгоритмическим конструкциям относятся *линейные, разветвляющиеся и циклические*.

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 ветвляющегося 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.



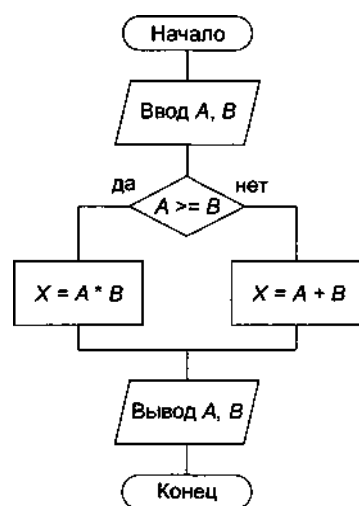
Разветвляющийся алгоритм

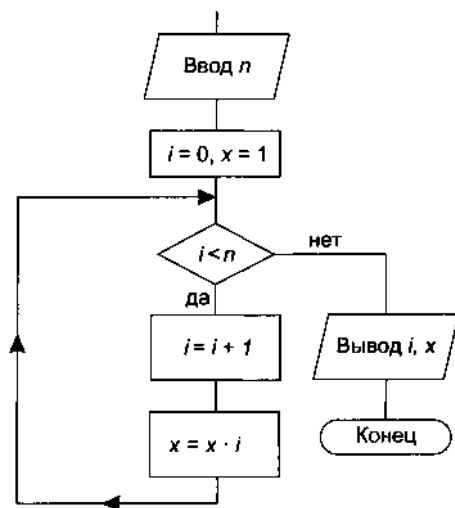
5. Циклические алгоритмы.

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





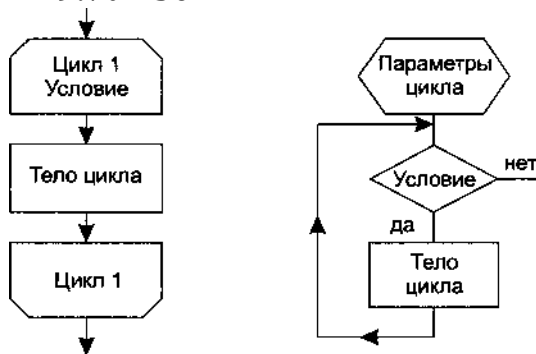
Алгоритм с циклическим элементом

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*.

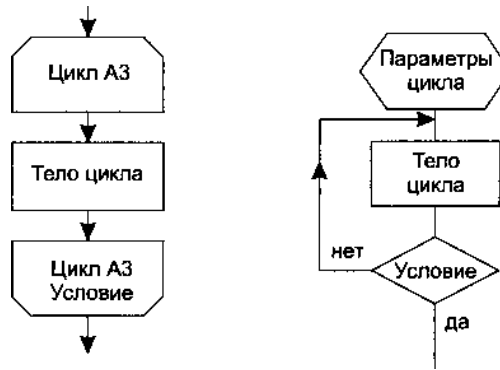
а) Общее обозначение б) Реализация
по ГОСТ 19.701-90



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 по ГОСТ 19.701-90 б) Реализация



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.