***1.Понятие информации, виды и способы её представления***

Термин информация происходит от латинского informatio, что означает «сведения, разъяснения, изложение».

Информация - это отображение окружающего нас мира с помощью знаков и сигналов или иначе сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы.

Однако можно лишь утверждать, что понятие ИНФОРМАЦИЯ предполагает наличие материального носителя информации, источника информации, передатчика информации, приемника и канала связи между источником и приемником.

Виды информации.

* общественно-политическая,
* социально-экономическая,
* актуальность,
* ясность,
* краткость,
* убедительность.

Формы информации.

Существует множество способов передачи и обработки информации. Человек может передавать информацию, используя тот или иной язык, жесты, мимику, звуки и воспринимать информацию используя любые органы чувств. Иными словами, информация человеком передается, обрабатывается и принимается в форме знаков или сигналами. Сигнал может быть световым, звуковым (радиоволны), электромагнитным, биохимическим и т.д.

Знаками можно считать алфавит любого языка, знаки языка жестов, любые коды или шифры, нотные значки и т.д.

Процесс обработки информации предусматривает наличие носителя информации и средства передачи информации и обработки информации.

Информацию можно:

* создавать
* принимать
* комбинировать
* хранить
* передавать
* копировать
* обрабатывать
* искать
* воспринимать
* формализовать
* делить на части
* измерять
* использовать
* распространять
* упрощать
* разрушать
* запоминать
* преобразовывать
* собирать и т.д.

Все эти процессы, связанные с определенными операциями над информацией, называются информационными процессами.

***2.Задачи получения, передачи, преобразования и хранения информации***

Обработка информации — получение одних информационных объектов из других путем выполнения некоторых алгоритмов.

Обработка является одной из основных операций, выполняемых над информацией, и главным средством увеличения объема и разнообразия информации.

Средства обработки информации — это всевозможные устройства и системы, созданные человечеством, и в первую очередь компьютер — универсальная машина для обработки информации.

Компьютеры обрабатывают информацию путем выполнения некоторых алгоритмов. Живые организмы и растения обрабатывают информацию с помощью своих органов и систем.

Информационные ресурсы — это идеи человечества и указания по реализации этих идей, накопленные в форме, позволяющей их воспроизводство.

Информационные ресурсы (в отличие от всех видов ресурсов — трудовых, энергетических, минеральных) тем быстрее растут, чем больше их расходуют.

Информационные технологии — это совокупность методов и устройств, используемых людьми для обработки информации.

К способам представления информации можно отнести:

всевозможные символы, языки и коды (русский, английский, азбука Морзе, язык жестов, язык мимики), чертежи, картины, музыкальные произведения, кино- и видео фильмы, компьютерные игры и т.п.

Хранить информацию можно:

\* в форме знаков на бумаге, перфокартах, перфоленте, на камне, дереве, ткани, в рельефе ключа и т.п.;

\* в форме электромагнитных сигналов на дисках, дискетах, кассетах,

\* кинопленке и т.п.;

\* в форме структурных комбинаций в клетках и генах биологических объектов;

Передача информации происходит:

\* в форме сигналов, как в технических устройствах, так и в общении людей, в телекоммуникациях, в жизни общества, в живых существах;

\* механическим путем в общении людей и в технических устройствах.

К техническим средствам хранения, передачи и обработки информации относятся:

- различные диски,

- дискеты,

- аудио и видео кассеты,

- перфокарты,

- перфоленты,

- кинопленка,

- объекты телекоммуникации (радио, телевидение, телефон, телеграф, персональные компьютеры и компьютерные сети и т.п.)

ЭВМ — это универсальная вычислительная машина, которая обрабатывает информацию, используя электрические сигналы.

Если сигнал непрерывен во времени, то его называют АНАЛОГОВЫМ, и машины, использующие такие сигналы, называют АНАЛОГОВЫМИ.

Если сигнал имеет прерывистую (дискретную) форму, он называется ЦИФРОВЫМ (дискретным), и машины соответственно называются ЦИФРОВЫМИ. Современные ПК относятся к ЦВМ.

Всю информацию ПК может воспринимать в форме сигналов, которые имеют только два различных состояния: есть сигнал или нет сигнала.

Этим сигналам поставлен в соответствие разряд кода, который может принимать только два значения: 0 или 1.

Сигнал, который имеет только два различных значения, или соответствующий ему разряд кода, который может принимать только два значения - 0 или 1, называется бит. Бит — это минимальная единица информации (англ. Bit- binary digit — двоичная цифра).

На практике чаще применяется более крупная единица — байт, равная восьми битам. Именно восемь бит требуется для того, чтобы закодировать любой из 256 символов (256 = 2^8).

1 байта достаточно для кодирования одного знака текстовой, числовой, звуковой или графической информации.

Крупные производные единицы информации:

1 Килобайт (Кбайт) = 1024 байт

1 Мегабайт (Мбайт) = 1024 Кбайт

1 Гигабайт (Гбайт) = 1024 Мбайт

1 Терабайт(Тбайт) = 1024Гбайт

1 Петабайт(Пбайт)= 1024 Тбайт

1 Экзабайт (Эбайт)=1024 Пбайт

1 Зеттабайт (Збайт)= 1024 Эбайт

1 Йоттабайт (Йбайт)= 1024 Збайт

***3.Язык как способ представления и передачи информации.***

Знаковая форма восприятия, хранения и передачи информации означает использование какого-либо языка. Языки делятся на разговорные (естественные) и формальные. Естественные языки носят национальный характер. Формальные языки чаще всего относятся к специальной области человеческой деятельности (например, язык математики или язык флажков на флоте).

Информация во внешней среде выражается с помощью некоторых материальных объектов (носителей), ассортимент и способ расположения которых задает информацию. Человек воспринимает сообщение посредством органов чувств. Приемник информации в технике воспринимает сообщения с помощью различной измерительной и регистрирующей аппаратуры. Носителем информации в различных информационных процессах может быть, например, камень, бумага, электрический кабель, магнитный диск.

Отображение множества состояний источника во множество состояний носителя называется способом кодирования. Таким образом, при выбранном способе кодирования какое-либо состояние заменяется своим образом — кодом состояния (или кодом информации, задаваемой этим состоянием). Так, мысли источника-человека могут быть закодированы определенным набором звуков, которые в свою очередь можно закодировать какими-то символами. Конечный набор знаков (символов) любой природы, из которых конструируются сообщения, образует алфавит некоторого языка.

Итак, последовательность символов алфавита, кодирующая состояние

источника и воспринимаемая адресатом как сообщение, как информация,

образует слово на этом языке. На передачу и переработку информации

влияет то, сигналами какой природы отображается одна и та же информация,

то есть каким кодом задана одна и та же информация. Если говорить о

сигналах, дискретных по виду, то их множество конечно, поэтому их принято

кодировать буквами алфавита того или иного естественного языка или

цифрами той или иной системы счисления. Таким образом, дискретная

информация отождествляется с алфавитно-цифровой, а простейшим

алфавитом, достаточным для записи (представления) информации, является

алфавит из двух символов, допустим 0 и 1.

***4.Задачи, требующие автоматизированной обработки информации.***

Это задачи, в которых при обработке информации нельзя справится вручную, но одновременно нельзя обойтись без человека. К таким задачам относятся:

***5.Системы автоматизированной и автоматической обработки информации***

Системы автоматизированной обработки информации — это системы, в которых при обработке информации нельзя справится вручную, но одновременно нельзя обойтись без человека. К таким задачам относятся:

Системы автоматической обработки информации — это системы, в которых при обработке информации требуется непосредственное участие человека. К таким задачам относятся:

***6. Исторические этапы автоматизации обработки информации***

**Первый этап (1946, до конца1950-х).** Были созданы электронные цифровые компьютеры, которые строились на электронных лампах. Основной режим использования этих машин состоял в том, что математик, составивший программу, садился за пульт управления машиной и производил необходимые вычисления. В этот период началась интенсивная разработка средств автоматизации программирования, создание входных языков разных уровней, систем обслуживания программ, упрощающих работу на машине и увеличивающих эффективность её использования.

**Второй этап (до середины 1960-х).** С технической точки зрения период чётко очерчен переходом на полупроводники, с точки зрения ЭВМ второго поколения характеризуются расширенными возможностями по вводу-выводу, увеличенным объёмом запоминающих устройств, развитыми системами программирования. Первые ОС просто автоматизировали работу оператора ЭВМ, связанную с выполнением задания пользователя: ввод в ЭВМ текста программы, вызов нужного транслятора, вызов необходимых библиотечных программ и т.д. Теперь же вместе с программой и данными в ЭВМ вводится ещё и инструкция, где перечисляются этапы обработки и приводится ряд сведений о программе и её авторе. Затем в ЭВМ стали вводить сразу по нескольку заданий пользователей (пакет заданий), ОС стали распределять ресурсы ЭВМ между этими заданиями – появился мультипрограммный режим обработки.

**Третий этап (до начала 1970-х).** Базой ЭВМ третьего поколения являются интегральные схемы, состоящей из десятков электронных элементов, образованных в прямоугольной пластине кремня с длиной стороны не более 1 см, позволило увеличить быстродействие и надёжность ЭВМ на их основе, а также уменьшить габариты, потребляемую мощность и стоимость ЭВМ. Машины третьего поколения имеют развитые операционные системы, обладают возможностями мультипрограммирования, т.е. одновременного выполнения нескольких программ. Многие задачи управления памятью, устройствами и ресурсами стала брать на себя операционная система или же непосредственно сама машина. В период машин третьего поколения произошёл крупный сдвиг в области применения ЭВМ. Если раньше ЭВМ использовалась в основном для научно-технических расчётов, то в 1960-1970-е гг. всё больше места стала занимать обработка символьной информации.

**Четвёртый этап (по настоящее время).** На первый план вышла задача экономии человеческих, а не машинных ресурсов. Существовавшая концепция первых этапов информационной технологии постепенно была заменена на новую: «всё, что могут делать машины должны делать машины; люди выполняют лишь ту часть работы, которую нельзя автоматизировать». С точки зрения структуры машины этого поколения представляют собой многопроцессорные и многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Для этого периода характерно широкое применение систем управления базами данных, компьютерных сетей, систем

Последующие поколения ЭВМ будут представлять, по-видимому, оптоэлектронные ЭВМ с массовым параллелизмом и нейронной структурой – с распределённой сетью большого числа (десятки тысяч) несложных процессоров, моделирующих структуру нейронных биологических систем, произойдёт качественный переход от обработки данных к обработке знаний.

Примечание:  
БИС - большие интегральные схемы;  
АСУ - автоматизированная система управления;  
САПР - система автоматизированного проектирования;  
АРМ - автоматизированное рабочее место;  
БЭСМ и МЭСМ - соответственно большая и малая электронные счетные машины;  
ПЭВМ - персональная электронная вычислительная машина.

С каждым новым поколением ЭВМ увеличивались быстродействия и надежность их работы при уменьшении стоимости и размеров, совершенствовались устройства ввода и вывода информации. В соответствии с трактовкой компьютера - как технической модели информационной функции человека - устройства ввода информации (зрительному, звуковому) и, следовательно, операция по ее вводу в компьютер становится все более удобной для человека.

***7. Последовательность обработки прикладных программ***

***8. Жизненный цикл программного обеспечения***

**Системное программное обеспечение:** Комплекс аппаратно - программных средств, расширяющих функциональные возможности операционной системы.

Все программы можно разбить на два класса по характеру использования:

      утилитарные программы — предназначены для удовлетворения нужд их разработчиков, программы «для себя»;

     программные продукты — предназначены для удовлетворения потребностей пользователей, широкого распространения и продажи.

Программный продукт должен быть соответствующим образом подготовлен к эксплуатации, иметь необходимую техническую документацию, предоставлять сервис и гарантию надежной работы программы, иметь товарный знак изготовителя. Только при таких условиях созданный программный комплекс может быть назван **программным продуктом.**

Программный продукт имеет несколько качественных характеристик:

-         алгоритмическая сложность;

-         полнота функций обработки;

-         объём файлов программ;

-         требования к операционной системе и техническим средствам обработки со стороны программного средства;

-         объём дисковой памяти;

-         размер оперативной памяти.

Показатели качества должны содержать следующие аспекты:

-         насколько хорошо можно использовать программный продукт (просто, надёжно, эффективно);

-         насколько легко эксплуатировать программный продукт;

-         можно ли использовать программный продукт при изменении условия его применения.

В условиях существования рынка программных продуктов важными характеристиками являются стоимость, количество продаж, время нахождения на рынке, известность фирмы-производителя и самой программы, наличие на рынке программных продуктов аналогичного назначения.

Программный продукт любого вида характеризуется жизненным циклом, состоящим из отдельных этапов.

**Маркетинг** предназначен для изучения требований к создаваемому программному продукту (технических, программных, пользовательских). Изучаются также существующие аналоги и продукты-конкуренты. Оцениваются необходимые для разработки материальные, трудовые и финансовые ресурсы, а также устанавливаются примерные сроки разработки.

**Проектирование структуры** — алгоритмизация процесса обработки данных, детализация функций, разработка архитектурного проекта, выбор методов и средств создания программ.

**Программирование, тестирование и отладка** — основной этап работы по разработке программного средства. Часто отдельные работы этого этапа ведутся параллельно, что позволяет сократить общее время разработки.

**Документирование** — обязательный вид работы. Документация должна содержать необходимые сведения по установке, обеспечению надёжной работы продукта, справочное пособие для пользователя, демонстрационные версии, примеры документов, создаваемых при помощи данного программного продукта, обучающие программы.

**Выход программного продукта на рынок** связан с организацией продаж массовому пользователю. Здесь применяются стандартные методы — реклама, увеличение числа каналов реализации, создание дилерской и дистрибьюторской сети, гибкая ценовая политика.

**Эксплуатация и сопровождение** идут, как правило, параллельно. В процессе эксплуатации могут выявляться ошибки, и устранение этих ошибок ведётся в режиме сопровождения, то есть оказание сервисной помощи, обеспечение новыми версиями программ, организация «горячих телефонных линий» для консультаций.

**Снятие программного продукта с продажи** и отказ от его сопровождения происходит, как правило, в случае изменения технической политики фирмы-изготовителя, неэффективности работы программного продукта, наличия в нём неустранимых ошибок, отсутствие спроса.

Длительность жизненного цикла разных программных продуктов неодинакова. Для большинства современных программ его длительность составляет 2-3 года. Хотя часто встречаются на компьютерах и давно снятые с производства программные продукты.

***9. Задачи системного программного обеспечения***

Этапы решения задачи на ЭВМ

**Программирование** (programming) - теоретическая и практическая деятельность, связанная с созданием программ. Решение задач на компьютере включает в себя следующие основные этапы, часть из которых осуществляется без участия компьютера.

**1. Постановка задачи:**

• сбор информации о задаче;

• формулировка условия задачи;

• определение конечных целей решения задачи;

• определение формы выдачи результатов;

• описание данных (их типов, диапазонов величин, структуры и т. п.).

**2. Анализ и исследование задачи, модели:**

• анализ существующих аналогов;

• анализ технических и программных средств;

• разработка математической модели;

• разработка структур данных.

**3. Разработка алгоритма:**

• выбор метода проектирования алгоритма;

• выбор формы записи алгоритма (блок-схемы, псевдокод и др.);

• выбор тестов и метода тестирования;

• проектирование алгоритма.

**4. Программирование:**

• выбор языка программирования;

• уточнение способов организации данных;

• запись алгоритма на выбранном языке

программирования.

**5. Тестирование и отладка:**

• синтаксическая отладка;

• отладка семантики и логической структуры;

• тестовые расчеты и анализ результатов тестирования;

• совершенствование программы.

**6. Анализ результатов решения задачи и уточнение в случае необходимости математической модели с повторным выполнением этапов 2-5.**

**7. Сопровождение программы:**

• доработка программы для решения конкретных задач;

• составление документации к решенной задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию.

Основная категория специалистов, занятых разработкой программ, — это программисты. Программисты неоднородны по уровню квалификации, а также по характеру своей деятельности.

**Системный программист** - занимается разработкой, эксплуатацией и сопровождением системного программного обеспечения, поддерживающего работоспособность компьютера и создающего среду для выполнения программ.

**Прикладной программист** - осуществляет разработку и отладку программ для решения функциональных задач (т. е. задач по реализации функций управления в рамках информационной системы - управление деятельностью торгового предприятия, управление перевозкой грузов, планирование выпуска продукции).

**Программист-аналитик -** программист, анализирующий и проектирующий комплекс взаимосвязанных программ.

**Постановщик задач -** разработчик формальных постановок задач, требующих реализации на ЭВМ.

**Администратор базы данных -** человек, который обеспечивает организационную поддержку базы данных.

**Администратор сети -** человек, который обеспечивает организационную поддержку работы локальной сети.

Основным потребителем программ является **конечный пользователь,** который, как правило, не является специалистом в области программирования.

Для работы с ЭВМ существует группа специально обученных технических работников - **операторов ЭВМ**. Они не программируют, а используют готовые программы для обеспечения работы на ЭВМ конечных пользователей: набор текстов, печать документов, копирование информации, запись на внешние носители и др.

***10. Назначение и структура операционных систем***

Операционная система является программной надстройкой над архитектурой компьютера, которая обеспечивает удобный пользовательский интерфейс, берет на себя функции автоматического управления рядом его подсистем и предоставляет готовые процедуры управления внутренними и внешними ресурсами. То есть, операционная система является некоей автоматизированной системой управления работой и ресурсами компьютера, повышающая удобство и эффективность его использования.

**Операционная система** (ОС) - неотъемлемая часть большинства современных вычислительных систем, управляющая всеми его аппаратными компонентами. Другими словами, Операционная (супервизорная, мониторная, исполнительная) Система (ОС) - есть организованная совокупность программ (систем), которая действует как интерфейс между аппаратурой ЭВМ и пользователями. Она обеспечивает пользователей набором средств для облегчения проектирования, программирования, отладки и сопровождения программ и в то же время управляет распределением ресурсов для обеспечения эффективной работы.

Структуру ОС составляют следующие модули:

* *базовый модуль* (ядро ОС) - управляет работой программ и файловой системой, обеспечивает доступ к ней и обмен файлами между периферийными устройствами;
* *командный процессор* - расшифровывает и исполняет команды пользователя, поступающие прежде всего через клавиатуру;
* *драйверы периферийных устройств* - программно обеспечивают согласованность работы этих устройств с процессором (каждое периферийное устройство обрабатывает информацию по-разному и в различном темпе);
* *дополнительные сервисные программы* (утилиты) - делают удобным и многосторонним процесс общения пользователя с компьютером.

В общем случае операционная система выполняет следующие четыре функции:

1. предоставляет другим программам определенный вид обслуживания (посредством программ-утилит), например выделение и назначение памяти, синхронизацию процесса вычислений и организацию взаимосвязи между различными процессами в вычислительной системе;
2. обеспечивает защиту (в определенной мере) других программ от последствий различных особых ситуаций, возникающих при машинной реализации данной программы, таких, как прерывания и машинные сбои;
3. реализует с той или иной степенью сложности принцип “виртуальной машины”, что позволяет группе программ использовать общие вычислительные ресурсы, например процессор (процессоры) и основную память;
4. организует и следит за выполнением принципов управления при решении таких задач, как обеспечение защиты данных от несанкционированного доступа и реализация системы приоритетов доступа программ к вычислительным ресурсам.

**Графический интерфейс пользователя**

Графический интерфейс пользователя изначально был несвойственен неигровым программам, однако, будучи призванным облегчить общение пользователя с компьютером и программой, хорошо прижился на IBM PC и стал неотъемлемой частью любой уважающей себя операционной системы.

Оболочка Microsoft Windows не была изначально операционной системой, она возникла в виде стандарт затора графического интерфейса и прижилась исключительно потому, что пользователь хотел видеть программу, с которой ему часто приходится работать красивой, практичной, удобной и легкой в освоении и использовании.

***11.Назначение и последовательность функционирования компиляторов***

Компьютерные программы, или *программное обеспечение,* представляют собой файлы, содержащие инструкции, которые указывают компьютеру, что следует делать.

*Программирование* представляет собой процесс определения последовательности инструкций, которые должен выполнить компьютер для решения определенной задачи. Для указания этих инструкций мы используем *язык программирования* C++.

Система должна выделять место в памяти для написания текста. С помощью текстового редактора мы вносим программные операторы в *исходный файл.* Потом мы запускаем программу, различающую текст, т.е. используем — *компилятор —* для преобразования операторов из формата, который мы можем читать и понимать, в единицы и нули, которые понимает компьютер. Переводом текста в машинные команды занимаются программы-трансляторы. Функциями трансляторов являются либо ликвидация неполадок, либо предупреждение о совершенных ошибках. Основной задачей является оптимизация текста.

Современные трансляторы являются компиляторами, следовательно, мы можем сформулировать последовательность функционирования компилятора:

-сначала компилятор проверяет программу на наличие ошибок и сообщает о них.

-создает объектный файл, который работает с относительными адресами (неизвестна точка старта).

-после загрузки работает с абсолютными адресами (.exe). Для этого требуется загрузчик, который является частью операционной системы и компилятора.

После компиляции (экзекуции) информация приходит к драйверам- вывод на экран, принтер, остальную периферию.

Компиляторы бывают однопроходными, т.е. выполняют compile и linking за один раз (присущи огромные размеры), и с разделенными функциями.

Для маломощных машин и программ, не зависящих от аппаратуры, созданы интерпретаторы. Для работы этих программ требуется небольшой объем памяти, т.к. интерпретатор выполняет инструкции последовательно, одна за другой, переводит текст и находит ошибки поштучно(первые версии Basic).

Современные Java – интерпретаторы, поэтому программы, написанные на этом языке, идут на любой маломощной машине, но работают они довольно медленно.

***12. Требования к языкам программирования и их классификация***

**Машинно – ориентированные языки** – это языки, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.). Машинно –ориентированные языки позволяют использовать все возможности и особенности Машинно – зависимых языков:

- высокое качество создаваемых программ (компактность и скорость выполнения);

- возможность использования конкретных аппаратных ресурсов;

- предсказуемость объектного кода и заказов памяти;

- для составления эффективных программ необходимо знать систему команд и особенности функционирования данной ЭВМ;

- трудоемкость процесса составления программ (особенно на машинных языках и ЯСК), плохо защищенного от появления ошибок;

- низкая скорость программирования;

- невозможность непосредственного использования программ, составленных на этих языках, на ЭВМ других типов.

Машинно-ориентированные языки по степени автоматического программирования подразделяются на классы.

- Машинный язык

Отдельный компьютер имеет свой определенный Машинный язык (далее МЯ), ему предписывают выполнение указываемых операций над определяемыми ими операндами, поэтому МЯ является командным.

- Языки Символического Кодирования

Языки Символического Кодирования (далее ЯСК), так же, как и МЯ, являются командными. Однако коды операций и адреса в машинных командах, представляющие собой последовательность двоичных (во внутреннем коде) или восьмеричных (часто используемых при написании программ) цифр, в ЯСК заменены на символы (идентификаторы), форма написания которых помогает программисту легче запоминать смысловое содержание операции. Это обеспечивает существенное уменьшение числа ошибок при составлении программ. Использование символических адресов – первый шаг к созданию ЯСК.

- Автокоды

Есть также языки, включающие в себя все возможности ЯСК, посредством расширенного введения макрокоманд - они называются Автокоды.

Макрокоманды обеспечивают передачу фактических параметров, которые в процессе трансляции вставляются в «остов» программы, превращая её в реальную машинную программу.

Развитые автокоды получили название Ассемблеры. Сервисные программы и пр., как правило, составлены на языках типа Ассемблер.

- Макрос

Язык, являющийся средством для замены последовательности символов описывающих выполнение требуемых действий ЭВМ наиболее сжатую форму - называется Макрос (средство замены).

В основном, Макрос предназначен для того, чтобы сократить запись исходной программы. Компонент программного обеспечения, обеспечивающий функционирование макросов, называется макропроцессором. На макропроцессор поступает макро определяющий и исходный текст. Реакция макропроцессора на вызов-выдачу выходного текста.

Макрос одинаково может работать, как с программами, так и с данными.

**Машинно – независимые языки** – это средство описания алгоритмов решения задач и информации, подлежащей обработке. Они удобны в использовании для широкого круга пользователей и не требуют от них знания особенностей организации функционирования ЭВМ и ВС.

Подобные языки получили название высокоуровневых языков программирования. Программы, составляемые на таких языках, представляют собой последовательности операторов, структурированные согласно правилам рассматривания языка (задачи, сегменты, блоки и т.д.). Операторы языка описывают действия, которые должна выполнять система после трансляции программы на МЯ.

**Проблемно – ориентированные языки**

С расширением областей применения вычислительной техники возникла необходимость формализовать представление постановки и решение новых классов задач. Необходимо было создать такие языки программирования, которые, используя в данной области обозначения и терминологию, позволили бы описывать требуемые алгоритмы решения для поставленных задач, ими стали проблемно – ориентированные языки. Эти языки, языки, ориентированные на решение определенных проблем, должны обеспечить программиста средствами, позволяющими коротко и четко формулировать задачу и получать результаты в требуемой форме.

Фортран, Алгол – языки, созданные для решения математических задач;

**Универсальные языки**

Универсальные языки были созданы для широкого круга задач: коммерческих, научных, моделирования и т.д.

**Диалоговые языки**

Появление новых технических возможностей поставило задачу перед системными программистами – создать программные средства, обеспечивающие оперативное взаимодействие человека с ЭВМ, их назвали диалоговыми языками.

Задачи: управление и описание алгоритмов решения задач...

Одним из примеров диалоговых языков является Бэйсик.

Бэйсик использует обозначения подобные обычным математическим выражениям. Многие операторы являются упрощенными вариантами операторов языка Фортран. Поэтому этот язык позволяет решать достаточно широкий круг задач.

**Непроцедурные языки**

Непроцедурные языки составляют группу языков, описывающих организацию данных, обрабатываемых по фиксированным алгоритмам (табличные языки и генераторы отчетов), и языков связи с операционными системами.

Программы, составленные на табличном языке, удобно описывают сложные ситуации, возникающие при системном анализе

***13.Понятие алгоритма, его основные свойства и способы записи***