1. Как вы можете объяснить бытовой термин «переизбыток информации»? Что имеется в виду: излишняя полнота данных; излишняя сложность методов; неадекватность поступающих данных и методов, имеющихся в наличии?

“Переизбыток информации” – это термин, который описывает ситуацию, когда имеется слишком много информации. Это может произойти, когда данные слишком полны или слишком сложны. Также это может означать, что поступающая информация не соответствует имеющимся методам ее обработки. Чтобы избежать переизбытка информации, необходимо уметь фильтровать и анализировать данные, выбирать только те, которые действительно необходимы для решения конкретной задачи.  
Например, если вы хотите купить новый телефон, но не знаете, какой выбрать, то вы можете начать искать информацию о различных моделях в интернете. Однако, если вы не знаете, какие характеристики важны для вас, то вы можете получить слишком много информации, которая будет вам не нужна. В этом случае, лучше сосредоточиться на определенных параметрах, которые важны для вас, таких как цена, размер экрана, камера и т.д.

1. Как вы понимаете термин «средство массовой информации»? Что это? Средство массовой поставки данных? Средство, обеспечивающее массовое распространение методов? Средство, обеспечивающее процесс информирования путем поставки данных гражданам, обладающим адекватными методами их потребления?

Это средства массовой информации, которые используются для распространения информации среди большого числа людей. Это может включать в себя телевидение, радио, газеты, журналы и другие формы коммуникации.

СМИ – это средство, обеспечивающее процесс информирования путем поставки данных гражданам, обладающим адекватными методами их потребления, потому что оно является средством обеспечения процесса информирования, так как предоставляют гражданам информацию о различных событиях, фактах и мнениях. Они делают это путем поставки данных, которые затем могут быть использованы гражданами для формирования своего мнения и принятия решений. Адекватные методы потребления данных могут включать чтение газет, просмотр телепередач, прослушивание радио и т.д.

1. Как вы полагаете, являются ли данные товаром? Могут ли методы быть товаром?

Да, я полагаю, что данные можно считать товаром. Они имеют ценность и могут быть проданы или куплены. Данные могут использоваться для различных целей, таких как анализ рынка, прогнозирование продаж и т.д. Также данные могут быть использованы для создания новых продуктов или услуг.

Примером того, когда данные являются товаром, может быть продажа данных о поведении пользователей веб-сайта компании. Компания может продавать эти данные другим компаниям для использования в маркетинговых целях.

Методы не могут быть товаром, так как они не имеют физической формы и не могут быть измерены или взвешены. Методы – это инструкции или процедуры, которые помогают людям выполнять задачи или достигать целей. Товары, наоборот, имеют физическую форму и могут быть куплены или проданы. Например, автомобиль является товаром, который можно купить или продать, в то время как инструкция по эксплуатации автомобиля не является товаром.

1. На примере коммерческих структур, обеспечивающих коммуникационные услуги, покажите, как взаимодействуют между собой маркетинг данных и маркетинг методов? Можете ли вы привести примеры лизинга данных и методов?

**Лизинг методов** – это процесс передачи прав на использование определенных методов на определенный срок. Например, компания может передать права на использование своего программного обеспечения на определенный период времени.

**Лизинг данных** – это предоставление доступа к определенным данным на определенный срок. Например, компания может предоставить доступ к своей базе данных для анализа.

**Маркетинг методов** – это продвижение и продажа определенных методов. Например, компания может продавать свое программное обеспечение для использования другими компаниями.

**Маркетинг данных** – это продажа определенных данных. Например, компания может продать свою базу данных для анализа.

1. *Интернет-провайдеры:* Маркетинг данных - предоставление доступа к интернету с определенной скоростью. Маркетинг методов - предоставление оборудования для подключения к интернету (роутеры, модемы). Лизинг данных - аренда выделенных серверов для хранения данных. Лизинг методов - аренда оборудования для доступа в интернет.  
2. *Операторы мобильной связи:* Маркетинг данных - предоставление пакетов трафика для интернета и звонков. Маркетинг методов - продажа мобильных устройств. Лизинг данных - предоставление услуги “облачного” хранения данных. Лизинг методов - предоставление услуг по аренде мобильных устройств.

Маркетинг данных и маркетинг методов взаимодействуют между собой через процесс продажи и покупки данных и методов.

1. Как вы понимаете диалектическое единство данных и методов? Можете ли вы привести примеры аналогичного единства двух понятий из других научных дисциплин: естественных, социальных, технических?

**Диалектическое единство данных и методов** – взаимосвязь и взаимодействие между данными и способами, с помощью которых эти данные собираются, анализируются и интерпретируются (методами).

Примеры из различных научных дисциплин:

1. **Естественные науки:** В биологии, данные о геноме организма и методы анализа генетической информации демонстрируют диалектическое единство.

2. **Социальные науки:** В социологии, данные о социальных тенденциях и методы сбора данных, такие как опросы или наблюдение, представляют диалектическое единство.

3.**Технические науки:** В информационных технологиях, данные о сетевой безопасности и методы обнаружения и предотвращения кибератак демонстрируют диалектическое единство.

1. Как вы понимаете динамический характер информации? Что происходит с ней по окончании информационного процесса?

Информация не является статичным объектом — она динамически меняется и существует только в момент взаимодействия данных и методов. Все прочее время она пребывает в состоянии данных. Таким образом, информация существует только в момент протекания информационного процесса. Все остальное время она содержится в виде данных.

1. Можем ли мы утверждать, что данные, полученные в результате информационного процесса, адекватны исходным? Почему? От каких свойств исходных данных и методов зависит адекватность результирующих данных?

Нет, не можем. Адекватность информации – это степень соответствия реальному состоянию дела. А так как эта степень может быть мала, то данные могут являться неадекватными. Такое может произойти, если новая информация создается на основе неполных или недостоверных данных. Хотя полные и достоверные данные также могут привести к созданию неадекватной информации, если к ним были применены неадекватные методы.

1. Что такое вектор данных? Является ли список номеров телефонов населенного пункта вектором данных? Является ли вектором данных текстовый документ, закодированный двоичным кодом, если он не содержит элементов оформления?

Вектор данных – это упорядоченный набор данных, размещенных в памяти на одинаковом "расстоянии" друг от друга (строки, столбцы, диагонали, массивы целиком)

Список номеров телефонов населенного пункта является вектором данных.

Да, текстовый документ, который был закодирован двоичным кодом и не содержит элементов оформления, по-прежнему является вектором данных (элементы информации – это символы, а закодированный двоичный код представляет собой способ перевода текста в числовую форму для компьютерного хранения и обработки).

1. Является ли цифровой код цветного фотоснимка вектором данных? Если нет, то чего ему не хватает?

Цифровой код цветного фотоснимка не является вектором данных, так как он представляет собой растровое изображение, а не набор математических векторов. Растровое изображение представляет собой сетку пикселей, каждый из которых содержит информацию о цвете и яркости. Вектор данных, напротив, представляет собой набор математических объектов (точек, линий, кривых), которые могут быть масштабированы без потери качества.

Для того чтобы цифровой код цветного фотоснимка стал вектором данных, ему необходимо добавить информацию о геометрических формах и объектах на изображении, таким образом превращая его из растрового в векторное изображение.

1. Как вы понимаете следующие термины: аппаратно-программный интерфейс, программный интерфейс, аппаратный интерфейс? Как бы вы назвали специальность людей, разрабатывающих аппаратные интерфейсы? Как называется специальность людей, разрабатывающих программные интерфейсы?

**Aппаратно-программный интерфейс (API/АПИ)** - представляет собой набор инструментов и правил, которые позволяют программному обеспечению взаимодействовать с аппаратным обеспечением или другим программным обеспечением. API определяет способы, по которым программы могут использовать функциональность или данные аппаратного обеспечения без необходимости знать подробности его внутренней реализации.

**Программный интерфейс** – это интерфейс, предоставляющий набор правил и инструментов для взаимодействия программного обеспечения. Программные интерфейсы обеспечивают структурированный способ обмена данными или командами между различными программами.

**Аппаратный интерфейс** – это механизм, обеспечивающий связь между аппаратными компонентами устройства или системы, или аппаратным и программным обеспечением. Аппаратные интерфейсы могут включать в себя разъемы, шины данных, протоколы связи и другие аспекты, обеспечивающие взаимодействие устройств.

**Люди, занимающиеся разработкой аппаратных интерфейсов –** *инженеры по разработке аппаратных интерфейсов*.

**Люди, специализирующиеся на разработке программных интерфейсов** – *разработчиками программных интерфейсов/разработчиками программного обеспечения* или *инженерами по программному проектированию*.

1. На основе личных наблюдений сделайте вывод о том, какими средствами может пользоваться преподаватель для обеспечения интерфейса с аудиторией. Можете ли вы рассмотреть отдельно методические и технические средства, имеющиеся в его распоряжении?

**Методические средства:**

**Лекции и презентации: п**реподаватели могут использовать структурированные лекции и презентации для передачи информации. Презентационные материалы, такие как слайды, могут визуально поддерживать устную речь.

**Интерактивные методы обучения:** задания, групповые обсуждения и другие интерактивные методы могут вовлечь студентов в учебный процесс.

**Электронные источники:** Поддержка материала с помощью электронных ресурсов, таких как статьи, книги, видео и веб-сайты, может обогатить учебный опыт.  
  
**Технические средства:**

**Проекторы и экраны:** использование проекторов и экранов позволяет преподавателям демонстрировать слайды, графику и другие визуальные материалы.

**Аудио- и видеозаписи:** записи лекций или видеоматериалов могут быть полезными для студентов, чтобы повторять материал или подготавливаться к экзаменам.

**Электронные системы оценивания:** использование электронных средств для оценки знаний студентов, таких как электронные тесты или системы онлайн-оценивания, может упростить процесс оценивания.

1. Может ли преподаватель рассматривать вашу тетрадь и авторучку как свое средство обеспечения интерфейса? Если да, то в какой мере? В чем вы видите диалектический характер связи между программным обеспечением и аппаратным?

Преподаватель может рассматривать вашу тетрадь и авторучку как средство обеспечения интерфейса, поскольку они являются физическими носителями информации и средством взаимодействия с обучаемым. Однако, использование этих предметов в качестве интерфейса является ограниченным и не может полностью заменить современные цифровые технологии.

Диалектический характер связи между программным и аппаратным обеспечением заключается в том, что они дополняют друг друга и образуют единое целое. Программное обеспечение определяет функциональность и логику работы системы, а аппаратное обеспечение обеспечивает физическую реализацию и выполнение этих функций. Без программного обеспечения аппаратное не имеет смысла, а без аппаратного обеспечения программное не может быть реализовано и использовано.

1. Назовите четыре основных уровня программного обеспечения. Каков порядок их взаимодействия? К какому классу относятся программные средства, встроенные в видеомагнитофон, программируемую стиральную машину, СВЧ-плиту?

1. Прикладное ПО

2. Служебное ПО

3. Системное ПО

4. Базовое ПО

Программы системного уровня опираются на программы базового уровня и обеспечивают взаимодействие пользователя с оборудованием, взаимодействие дополнительного оборудования с базовым, а также предоставляют возможность для установки и работы программ более высоких уровней.

Программы служебного уровня выполняют обслуживание компьютерной системы, обеспечивают ее контроль и настройку. В своей работе они опираются на  
программы базового и системного уровней.

Программы прикладного уровня используются человеком для исполнения практических задач с помощью компьютера. Эти программы опираются на программы  
нижележащих уровней.

Программные средства, встроенные в программируемую стиральную машину, СВЧ-плита, видеомагнитофон относятся к базовому уровню ПО.

1. В чем преимущества и недостатки выполнения офисных работ (например, копировально-множительных) аппаратными и программными средствами?

**Аппаратные средства:**

Преимущества:  
1. Высокая производительность.

2. Простота использования.

3. Надежность.

Недостатки:

1. Ограниченность функциональности.

2. Зависимость от оборудования.

**Программные средства:**

Преимущества:

1. Больший спектр функциональности.

2. Удобство хранения и обработки данных.

3. Обновления и модификации.

Недостатки:

1. Требуется обучение.

2. Зависимость от технологии.

3. Затраты на лицензии и обновления.

1. Какие категории программного обеспечения могут быть использованы в работе малого предприятия и для каких целей?

**Системное ПО** (хранит информацию, создаёт копии и т.д.)

**Прикладное ПО** (Системы общего назначения часто интегрируются в многокомпонентные пакеты для автоматизации офисной деятельности, например Microsoft Office, StarOffice и др.)

**Офисные пакеты:** склад/бухгалтерия/расчёт з.п./обеспечение техпроцесса (накладные, путевые листы и пр.) - планирование закупок, доставки товара, завоза сырья, графика работы персонала.

**Прикладные программы** обеспечивают выполнение необходимых пользователям работ; системные программы выполняют различные вспомогательные функции.

1. Какие виды работ, характерные для крупного промышленного предприятия (например, машиностроительного завода), могут быть автоматизированы с помощью компьютеров? Какие категории программных средств для этого необходимы?

Сварка, доставка на конвеер, тестирование работы автомобиля и его электрики.

**Системное ПО** (хранит информацию, создаёт копии и т.д.)

**Прикладное ПО** (Системы общего назначения часто интегрируются в многокомпонентные пакеты для автоматизации офисной деятельности – офисные пакеты – Microsoft Office, StarOffice и др.).

**Инструментально-программное обеспечение** (системы программирования, системы моделирования).

1. Назовите основные категории программного обеспечения, относящиеся к классу графических редакторов. В чем состоит принципиальная разница между этими категориями?

Существует две категории графических редакторов:

**Растровые редакторы**: используются для редактирования и ретуширования изображений, созданных с помощью цифровых камер или сканеров.

**Векторные редакторы:** эти программы используются для создания и редактирования векторных изображений, таких как логотипы, иконки и чертежи. Основное отличие между растровой и векторной графикой заключается в том, что растровая графика представляет изображения в виде пикселов, а векторная графика – с помощью геометрических фигур.

1. Что общего и в чем различие между понятиями программное обеспечение и информационное обеспечение средств вычислительной техники?

Из общего – это данные для компьютера для выполнения каких-либо действий, а из различий – одно является обновлением программ, а другое является предоставлением информации.

1. Что такое операционная система?

ОС – комплекс программ, обеспечивающих взаимодействие всех аппаратных и программных частей компьютера между собой и взаимодействие пользователя и компьютера.

1. Перечислите основные функции операционной системы.
2. Основная функция — посредническая. Она заключается в обеспечении нескольких видов интерфейса:

а) интерфейса между пользователем и программно-аппаратными средствами компьютера (интерфейс пользователя);

б) интерфейса между программным и аппаратным обеспечением;

в) интерфейса между разными видами программного.

1. Обеспечение автоматического запуска.
2. Обеспечение создания файловой системы и доступа к ней.
3. Обслуживание файловой структуры (создание файлов, каталогов и присвоение им имён; переименование/удаление файлов, каталогов; навигация по файловой структуре; копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска).
4. Управление установкой, исполнением и удалением приложений.
5. Взаимодействие с аппаратным обеспечением
6. Обслуживание компьютера (предоставление средств проверки дисков, средств «сжатия» дисков, средств управления виртуальной памятью, средств кэширования дисков, средств резервного копирования данных).
7. Расскажите о видах интерфейса пользователя, применяемых в разных операционных системах

Виды интерфейсов пользователя:

1. **Текстовый интерфейс** - предоставляет пользователю только текстовую информацию и простейшие элементы управления, такие как меню и кнопки. Используется в старых операционных системах, таких как MS-DOS и UNIX.
2. **Графический интерфейс** - позволяет использовать графические элементы для отображения информации и управления системой. Является стандартом для современных операционных систем, таких как Windows, macOS и Linux.
3. **Сенсорный интерфейс** - предназначен для использования на устройствах с сенсорными экранами, таких как смартфоны и планшеты. Позволяет управлять системой с помощью касаний и жестов на экране.
4. Перечислите функции операционной системы по обслуживанию файловой структуры.

1) создание файлов и присвоение им имен;   
2) создание каталогов (папок) и присвоение им имен;   
3) переименование файлов и каталогов (папок);   
4) копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;   
5) удаление файлов и каталогов (папок);   
6) навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);   
7) управление атрибутами файлов.

1. Объясните правила, по которым формируются короткое имя файла и длинное имя файла.

По соглашению 8.3 (короткое соглашение об именовании в стиле MS-DOS), имя файла состоит из двух частей: собственно имени и расширения имени. На имя файла отводится 8 символов, а на его расширение — 3 символа. Имя от расширения отделяется точкой. Как имя, так и расширение могут включать только алфавитно-цифровые символы латинского алфавита. Такое имя называется коротким именем файла.  
Длинным именем файла считается любое имя файла, которое превышает соглашение 8.3.

Правила формирования длинного имени файла:

1) Если «длинное» имя файла включает пробелы, то в служебных операциях его надо заключать в кавычки. Рекомендуется не использовать пробелы, а заменять их символами подчеркивания.

2) В корневом каталоге диска нежелательно хранить файлы с длинными именами. В файловых системах на основе FAT количество единиц хранения в этой папке ограничено. Чем длиннее имена, тем меньше файлов можно разместить в корневом каталоге.

3) Кроме ограничения на длину имени файла (256 символов) существует гораздо более жесткое ограничение на длину полного имени файла (в него входит путь доступа к файлу). Полное имя не может быть длиннее 260 символов.

4) В длинных именах файлов разрешается использовать символы любых алфавитов, в том числе и русского, но, если документ готовится для передачи, с заказчиком (потребителем документа) необходимо согласовать возможность воспроизведения файлов с такими именами на его оборудовании.

5) Прописные и строчные буквы в именах не различаются операционной системой. Для нее имена Письмо.txt и письмо.txt соответствуют одному и тому же файлу. Однако отображаются символы разных регистров операционной системой исправно.

1. В чем заключается операция установки приложения?

Установочный пакет программного обеспечения представляет собой полуфабрикат, из которого в процессе установки на компьютере формируется полноценное рабочее приложение. При этом осуществляется привязка приложения к существующей аппаратно-программной среде и его настройка на работу именно в этой среде.

Современные графические операционные системы берут на себя управление установкой приложений. Она включает в себя следующие процессы: распределение ресурсов вычислительной системы между приложениями; обеспечение доступа устанавливаемых приложений к драйверам устройств вычислительной системы; формирование общих ресурсов, которые могут использоваться разными приложениями; выполнение регистрации установленных приложений и выделенных им ресурсов.

1. Что такое язык программирования?

Язык программирования — это формальный язык, используемый для формулирования структур данных и алгоритмов (вычислительных правил), которые могут быть выполнены компьютером. Они обычно состоят из пошаговых инструкций, которые составлены из разрешенных (текстовых) шаблонов, называемых синтаксисом.

1. В чем различие компиляторов и интерпретаторов?

С помощью языка программирования создается только текст программы, 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).

Основные различия между компиляторами и интерпретаторами:

* Процесс перевода исходного кода:

Компилятор переводит весь исходный код в машинный код и сохраняет его в исполняемом файле. Затем этот файл запускается на выполнение. Интерпретатор переводит и выполняет исходный код построчно, то есть он выполняет каждую строку кода по очереди, а затем переходит к следующей строке.

* Скорость выполнения программы:

Поскольку компилятор создает исполняемый файл, программы, скомпилированные с его помощью, обычно выполняются быстрее, чем программы, обработанные интерпретатором. Интерпретируемые программы обычно выполняются медленнее, так как интерпретатору требуется время на перевод и выполнение каждой строки кода.

* Возможность использования более сложных конструкций языка:

Интерпретаторы обычно поддерживают более широкий спектр конструкций языка программирования, чем компиляторы, поскольку они должны быть способны выполнить эти конструкции сразу после их ввода. Компиляторам обычно требуется, чтобы код был написан строго определенным образом, прежде чем они смогут его обработать.

1. Объясните термины «язык низкого уровня» и «язык высокого уровня».

Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется языком программирования **низкого уровня**. В данном случае «низкий уровень» не значит «плохой». Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

Языки программирования **высокого уровня** значительно ближе и понятнее человеку, нежели компьютеру. Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка.

1. Расскажите о поколениях языков программирования. Какие языки программирования активно используются сегодня?

1) Первое поколение языков программирования: **машинно-ориентированные языки** (ассемблеры) - программы писались на языке, который был максимально приближен к машинному коду. К ним относятся, например, 68000 Assembly, COBOL, BASIC.

2) Второе поколение языков программирования: **процедурно-ориентированные языки** - программы пишутся с использованием процедур и функций, что позволяет структурировать код и упростить его понимание. Примеры таких языков включают C, C++, Java, Python.

3) Третье поколение языков программирования: **объектно-ориентированные языки** - они позволяют создавать программы с использованием объектов, которые обладают свойствами и методами. Примеры: C#, JavaScript, Swift, Java.

4) Четвертое поколение языков программирования – это языки, которые ориентированы на работу с большими данными и машинное обучение. Примеры: R, Python, Julia.

1. Что нужно для создания программы?

1. **Текстовый редактор**. Так как текст программы записывается с помощью ключевых слов, обычно происходящих от слов английского языка, и набора стандартных символов для записи всевозможных операций, то формировать этот текст можно в любом редакторе, получая в итоге текстовый файл с исходным текстом программы.  
2. Исходный текст с помощью **программы-компилятора** переводится в машинный  
код. Если обнаружены синтаксические ошибки, то результирующий код создан  
не будет. На этом этапе уже возможно получение готовой программы, но чаще всего  
в ней не хватает некоторых компонентов, поэтому компилятор обычно выдает  
промежуточный объектный код (двоичный файл, стандартное расширение .OBJ).  
3. Исходный текст большой программы состоит, как правило, из нескольких модулей (файлов с исходными текстами), потому что хранить все тексты в одном файле неудобно — в них сложно ориентироваться. Каждый модуль компилируется в отдельный файл с объектным кодом, которые затем надо объединить в одно целое.

4. Объектный код обрабатывается специальной программой — **редактором связей** или сборщиком, который выполняет связывание объектных модулей и машинного кода стандартных функций, находя их в библиотеках, и формирует на выходе работоспособное приложение — исполнимый код для конкретной платформы.  
5. **Исполнимый код** — это законченная программа, которую можно запустить на  
любом компьютере, где установлена операционная система, для которой эта  
программа создавалась. Как правило, итоговый файл имеет расширение .EXE.

1. Что такое среды быстрого проектирования?

**Среда быстрого проектирования (СБП)** – это программное обеспечение, предназначенное для разработки электронных схем и печатных плат. Она облегчает и ускоряет процесс проектирования, позволяя инженерам создавать, редактировать и анализировать схемы и платы на компьютере.

1. Объясните понятие «архитектура программной системы»

**Архитектура программной системы** – это общий план или структура системы, определяющий ее основные компоненты, их взаимодействие и принципы работы.

Архитектура программной системы определяет основные принципы, на которых строится система, и предоставляет общую картину ее компонентов и их взаимодействия. Она включает в себя такие аспекты, как структура данных, архитектурные шаблоны, выбор технологий, принципы разделения ответственности, управление данными, интерфейсы и многое другое.

Цель – обеспечить удобство разработки, поддержки, масштабируемости и эффективность системы. Хорошо спроектированная архитектура позволяет создавать гибкие, надежные и расширяемые программные системы, а также облегчает их изменение и модификацию по мере развития требований и потребностей пользователей.

1. Какие типы данных считаются базовыми?

1. **Целочисленные** типы данных (integers): представляют целые числа без десятичной части, например, 1, 2, -5 и т.д. В различных языках программирования могут быть разные размеры и диапазоны целочисленных типов.

2. **Вещественные** типы данных (floating-point): представляют числа с плавающей точкой, которые могут иметь десятичную часть. Например, 3.14, -0.5 и т.д. Вещественные типы данных обычно используются для представления чисел с плавающей точкой, таких как десятичные дроби или научные значения.

3. **Символьные** типы данных (characters): представляют отдельные символы или буквы. Например, 'a', 'b', 'c' и т.д. Символьные типы данных могут быть использованы для представления текстовых данных.

4. **Логические** типы данных (boolean): представляют логические значения true (истина) и false (ложь). Логические типы данных часто используются для выполнения логических операций и принятия решений в программе.

1. Приведите примеры арифметических и логических выражений.

Арифметические выражения:  
1) 5 + 3 - 2 \* 4 // результат: 5  
2) 2 \* (7 - 4) + 6 // результат: 14  
3) 10 / 2 + (8 - 3) \* 2 // результат: 15  
  
Логические выражения:  
1) 5 > 3, результат: true  
2) 10 < 5, результат: false  
3) (4 + 2) == 6, результат: true  
4) (8 - 3) != 5, результат: false

1. Зачем нужны комментарии?

1) Облегчают понимание кода: они поясняют логику и цель определенного блока кода, особенно если он сложный или неочевидный.

2) Улучшают сопровождаемость: Комментарии помогают при сопровождении кода. Если вы или другой программист возвращаетесь к коду после некоторого времени, комментарии помогут вспомнить, что именно делает каждая часть кода, и обновить его при необходимости.

3) Помогают при отладке: Комментарии могут быть полезны при отладке кода. Вы можете использовать их для временного отключения определенных частей кода или для пометки мест, где происходят ошибки, чтобы легче найти и исправить проблемы.

4) Документация: Комментарии можно использовать для автоматической генерации документации. Специальные инструменты могут анализировать комментарии и создавать документацию, которая поможет другим программистам использовать ваш код.

1. Из каких частей состоит оператор цикла?

Цикл в программировании состоит из трех частей: 1. Оператор – название цикла. 2. Условие – это то, при каких обстоятельствах цикл работает. 3. Тело – код цикла, который должен сработать при прохождении условия.

1. Назовите достоинства и недостатки параллельных вычислений

**Достоинства:**

1.возможность выполнения нескольких задач в один момент времени (Повышение производительности и ускорение времени работы);

2.Решение сложных задач

3.Масштабируемость (позволяет адаптировать систему под требования конкретной задачи и использовать доступные ресурсы наилучшим образом.);

4. Повышение надежности

**Недостатки:**

1.трудности для новичков, связанные с иным типом мышления при программировании;

2.неопределенность приоритета задач;

3.трудности взаимосвязи между задачами;

4.взаимоблокировка задач;

5.проблемы при отказе оборудования

1. Как организуется работа с файлами?

Файлы на компьютере можно создавать, удалять, редактировать, просматривать, хранить, копировать, перемещать. Работа с файлами в 3 этапа:

a) файл открывается в одном из выбранных режимов. Он может состоять из последовательности одинаковых блоков

b) выполняется считывание, обновление или удаление записей в файле

c) файл закрывается

1. С какой целью применяют подпрограммы?

* Избавляют от необходимости многократно повторять в тексте программы аналогичные фрагменты;
* улучшают структуру программы, облегчая ее понимание;
* повышают устойчивость к ошибкам программирования и непредвидимым последствиям при модификациях программы;
* позволяет разработчикам разбить сложную задачу на более простые подзадачи. Каждая подпрограмма может быть разработана и отлажена отдельно, что упрощает процесс разработки и уменьшает вероятность ошибок. Использование подпрограмм может улучшить производительность программы

1. Чем характеризуются процедурные языки программирования?

Процедурные языки программирования характеризуются тем, что программы в них пишутся в виде последовательности инструкций, которые выполняются одна за другой. В таких языках обычно используются операторы ветвления (if-else, switch), циклы (for, while, do-while), функции и процедуры для организации кода.

1. В чем состоит идея нисходящего проектирования?

Наличие подпрограмм позволяет вести проектирование и разработку приложения сверху вниз — такой подход называется нисходящим проектированием. Сначала выделяется несколько подпрограмм, решающих самые глобальные задачи (например, инициализация данных, главная часть и завершение), потом каждый из этих модулей детализируется на более низком уровне, разбиваясь, в свою очередь, на небольшое число других подпрограмм, и так происходит до тех пор, пока вся задача не окажется реализованной. Такой подход удобен тем, что позволяет человеку постоянно мыслить на предметном уровне, не опускаясь до конкретных операторов и переменных.

1. Что общего и в чем отличия процедуры и функции?

Процедуры и функции отличаются тем, что процедура просто выполняет группу операторов, а функция вдобавок вычисляет некоторое значение и возвращает его.

1. В чем различие между событийным и структурным программированием?

Событийное программирование является развитием идей нисходящего проектирования, когда постепенно определяются и детализируются реакции программы на различные события.

Идея структурного программирования заключается в том, что структура программы должна отражать структуру решаемой задачи, чтобы алгоритм решения был ясно виден из исходного текста.

1. Как организуется обработка программных событий?

Главная часть представляет собой один бесконечный цикл, который опрашивает систему, следя за тем, не появилось ли новое сообщение. При его обнаружении вызывается подпрограмма, ответственная за обработку соответствующего события, и подобный цикл опроса продолжается, пока не будет получено сообщение «Завершить работу».

1. Для чего в языки программирования было введено понятие класса? В чем различие между классом и объектом?

Объекты могут иметь идентичную структуру и отличаться только значениями свойств. В таких случаях в программе создается новый тип, основанный на единой структуре объекта (по аналогии с тем, как создаются новые типы для структур данных). Он называется классом, а каждый конкретный объект, имеющий структуру этого класса, называется экземпляром класса.

Класс — это некий шаблон, который предоставляет абстрактный вид любого объекта.  
Объект или Экземпляр класса — это уже конкретная сущность, которая сделана по шаблону класса.

1. Поясните понятие инкапсуляции на бытовых примерах

Представим на минутку, что мы оказались в конце позапрошлого века, когда Генри Форд ещё не придумал конвейер. Представим, что для управления первым паровым автомобилем необходимо было знать, как устроен паровой котёл, постоянно подбрасывать уголь, следить за температурой, уровнем воды. При этом для поворота колёс использовать два рычага, каждый из которых поворачивает одно колесо в отдельности. Думаю, можно согласиться с тем, что вождение автомобиля того времени было весьма неудобным и трудным занятием.

Теперь вернёмся в сегодняшний день к современным чудесам автопрома с коробкой-автоматом. На самом деле, по сути, ничего не изменилось. Бензонасос всё так же поставляет бензин в двигатель, дифференциалы обеспечивают поворот колёс на различающиеся углы, коленвал превращает поступательное движение поршня во вращательное движение колёс. Прогресс в другом. **Сейчас все эти действия скрыты от пользователя** и позволяют ему крутить руль и нажимать на педаль газа, **не задумываясь**, что в это время происходит с инжектором, дроссельной заслонкой и распредвалом. Именно сокрытие внутренних процессов, происходящих в автомобиле, позволяет эффективно его использовать даже тем, кто не является профессионалом-автомехаником с двадцатилетним стажем. То есть человек пользуется привычным интерфейсом (который ему доступен), но может не вникать в процессы, происходящие внутри. Это сокрытие в ООП носит название инкапсуляции.

Инкапсуляция – это свойство системы, позволяющее объединить данные и методы, работающие с ними в классе, и скрыть детали реализации от пользователя.

1. 1)Для чего применяется механизм наследования? 2)Как полиморфизм модифицирует принцип наследования?

**(ОТВ. вопр 1) Наследование позволяет создавать новые классы, повторно используя уже готовый исходный код и не тратя времени на его переписывание.  
(вопр 2)** Полиморфизм позволяет классам с общей иерархией иметь разную реализацию методов, в зависимости от конкретного класса. Это означает, что можно определить один метод в базовом классе, а затем переопределить его в потомках, чтобы реализовать разные поведения для разных типов объектов. Это позволяет избежать дублирования кода и улучшает модульность системы. Например, есть класс «живые организмы», в него входят объекты «кошка» и «амеба», но они по-разному выполняют действие «идти», следовательно у них разные методы исполнения одной и той же команды.

1. Опишите использование принципов объектно-ориентированного программирования в средах быстрого проектирования.

Технологии объектного, событийного и структурного программирования сегодня  
объединены в RAD-системах, которые содержат множество готовых классов, представленных в виде визуальных компонентов, которые добавляются в программу  
одним щелчком мыши. Программисту надо только спроектировать внешний вид  
окон своего приложения и определить обработку основных событий — какие операторы будут выполняться при нажатии на кнопки, при выборе пунктов меню или   
щелчках мышкой. Весь вспомогательный исходный код среда сгенерирует сама, позволяя программисту полностью сосредоточиться только на реализации алгоритма.

1. В чем трудности разработки крупных программных проектов?

Чем крупнее проект, тем больше в нем ошибок. При этом слишком затягивать этап   
тестирования нельзя — нарушаются сроки, растет недовольство потребителей, и на   
рынок выпускается «сырая» система с множеством ошибок, которые устраняются   
уже в процессе эксплуатации выпуском многочисленных «заплаток».

1. Опишите организацию работы над сложной программной системой

1. Формируются и анализируются требования к проекту

2. Обследование объекта автоматизации

3. Создается подробное ТЗ, план работ.

4. Выбирается методология разработки ПО и начинается разработка (кодирование).

5. Тестирование программы, поиск ошибок

6. После того как заказчик удовлетворен качеством продукта, начинается его внедрение — подготовка к окончательному запуску в эксплуатацию

7. После того как новая система готова к работе, сотрудников организации заказчика нужно обучить работе с этой системой

8. После того как заказчик подписывает акт приемки, проект считается завершенным, но связь с исполнителем не теряется. Особенно в первое время у пользователей системы постоянно будет возникать множество вопросов по работе с ней. Неизбежно и возникновение ошибок, которые требуется устранять. Кроме того, исполнитель может выпускать новые версии системы, и старая система ПО требует обновления. Сотрудничество с заказчиком по обслуживанию системы называется сопровождением. Оно бесплатно на определенный гарантийный срок (например, год).

1. Какой этап разработки проекта является наиболее ответственным?

Проектная стадия считается самым ответственным и длительным по времени этапом. На этом этапе разрабатывается архитектура системы, определяются требования и проектируются компоненты системы. Ошибки, допущенные на этом этапе, могут привести к проблемам в работе системы и увеличить стоимость разработки.

1. Какова роль собственно программирования в ходе работы над проектом?

В ходе работы над проектом программирование играет важную роль, так как без него невозможно создать работающее приложение или систему.

1. Опишите известные вам методы контроля качества программного обеспечения.

**Тестирование:** это процесс проверки программы на наличие ошибок и дефектов. Тестирование может быть ручным, когда тестировщик выполняет все действия вручную, или автоматизированным, когда используются специальные программы для тестирования.

**Инспекция кода:** это процесс анализа исходного кода программы с целью выявления возможных ошибок и несоответствий. Инспекция кода может проводиться как вручную, так и с использованием автоматизированных инструментов.

**Модульное тестирование:** это метод тестирования, при котором каждый модуль программы проверяется отдельно. Модульное тестирование позволяет выявить ошибки на ранних стадиях разработки и сократить время на их исправление.

**Запуск бето-версии ПО.**

1. Каковы основные методы распространения программного обеспечения?

**Коммерческое ПО.** При создании программного продукта издатель, выполнив анализ рынка, заказывает у исполнителя разработку такого ПО, которое должно пользоваться на рынке спросом, и выделяет на его создание деньги.

**Бесплатное ПО** (freeware, public domain). Такие программы не имеют никаких ограничений, однако автор может попросить заплатить ему некоторую сумму, не настаивая, впрочем, на этом (это метод freeware). Некоторые программы авторы называют «общественным достоянием» (public domain), ничего взамен не требуют и нередко распространяют такое ПО в исходных текстах.

**Подписка (возможно)**

1. CONVERT. Для чего предназначена данная программа?

**CONVERT** - команда DOS, предназначенная для конвертации файлов из одного формата в другой, преобразует тома FAT в NTFS.

1. ECHO. Для чего предназначена данная команда?

**ECHO** - команда Windows Command Prompt и Bash, предназначенная для вывода сообщений на экран.

1. При работе в командной строке Windows как можно локально или удаленно перезагрузить (выключить) компьютер?

**SHUTDOWN** - Позволяет локально или удаленно завершить работу компьютера. shutdown /s - завершение работы компьютера. shutdown /r - перезагрузка компьютера.

1. Работа в командной строке. Напишите команду для вывода имени каталога либо смены текущего каталога.

**CD** - вывод имени либо смена текущего каталога. Также при включении расширенной обработки команд можно использовать **CHDIR**. Их отличие в том, что при использовании второй команды не нужно заключать имя каталога с пробелами в кавычки: cd "\s 12" = chdir \s 12

1. Работа в командной строке. Напишите команду для вывода графического отображения структуры каталогов диска или каталога.

**TREE** - Графически отображает структуру каталогов диска или пути.

1. Работа в командной строке. Напишите команды для вывода сведений о системе и конфигурации компьютера.

**SYSTEMINFO** - Отображает сведения о свойствах и конфигурации определенного компьютера.

1. Работа в командной строке. Напишите команду для вывода Отображения всех выполняемых задач, включая службы и команды для прекращения или остановка процесса, или приложения.

**TASKLIST** - Отображает все выполняемые задачи, включая службы. **TASKKILL** - Прекращение или остановка процесса либо приложения. **PAUSE** - Приостановка выполнения командных файлов и вывода сообщения.

1. Работа в командной строке. Напишите команду для копирования файлов и деревьев (структуры) каталогов.

**XCOPY** - Копирует файлы и деревья каталогов.

1. Работа в командной строке. Напишите команды для вывода сведений о версии Windows и вывода сведений о системе и конфигурации компьютера (в графическом и текстовом режиме).

**VER** - Отображает сведения о версии Windows.

**SYSTEMINFO** - Отображает сведения о свойствах и конфигурации определенного компьютера, SYSTEMINFO /sbs – в текстовом виде.

**msinfo32** – это вывод сведений о системе и конфигурации в графическом режиме

1. Работа в командной строке. Напишите команду для создания или удаления каталога.

**md или mkdir** - создание директории. Команда создает при необходимости все промежуточные каталоги в пути.

**rd или rmdir** - удаление.

/S Удаление дерева каталогов, т. е. не только указанного каталога,  
но и всех содержащихся в нем файлов и подкаталогов.

/Q Отключение запроса подтверждения при удалении дерева каталогов  
с помощью ключа /S.

*Невозможно* удалить каталог, *содержащий файлы*, включая скрытые или системные файлы.   
Используйте команду dir /a для перечисления всех файлов (включая скрытые и системные файлы). Затем используйте команду attrib с параметром -h, чтобы удалить скрытые атрибуты файла, -s , чтобы удалить атрибуты системного файла или -h -s, чтобы удалить как скрытые, так и системные атрибуты файлов. После удаления скрытых и файлового атрибутов можно удалить файлы.

Команду rmdir нельзя использовать для удаления *текущего* каталога.

1. Работа в командной строке. Напишите команду вывода содержимого файла на экран в потоке или по частям.

**TYPE** [диск:][путь]имя\_файла - Вывод содержимого одного или нескольких текстовых файлов.

1. Работа в командной строке. Напишите команду для отображения статистики протоколов и текущих сетевых подключений TCP/IP.

Команда **NETSTAT** предназначена для получения сведений о состоянии сетевых соединений и слушаемых на данном компьютере портах TCP и UDP,   
а также, для отображения статистических данных по сетевым интерфейсам и протоколам.

1. Работа в командной строке. Напишите команду для изменения кодовой страницы.

**CHCP** – просмотр или изменение кодовой страницы в командной строке Windows.

1. Работа в командной строке. Напишите команду для изменения копирования и перемещения (переименования) файлов.

**move** c:\test\**file1.txt** D:\folder2\file2.txt - перенести файл **file1.txt** из каталога test диска C: в каталог folder2 диска D: под именем file2.txt

1. Работа в командной строке. Напишите команду для форматирования диска. Основное назначение данной команды в DOS и Windows.

Команда **FORMAT** позволяет выполнить форматирование диска /дискеты в командной строке Windows.

Пример format A: - форматировать гибкий диск в дисководе A: с параметрами по умолчанию.

1. Работа в командной строке. Напишите команду для проверки IP-уровня подключения к другому TCP/IP компьютера, запрос (ICMP) сообщения с эхо-запросом.

Команда **PING** с помощью отправки сообщений с эхо-запросом по протоколу ICMP проверяет соединение на уровне протокола IP с другим компьютером, поддерживающим TCP/IP.

Например: "ping ya.ru"

Эта команда предоставит нам следующую информацию:

1.IP-адрес компьютера, с кем осуществляется проверка связи;

2.Число отправленных байт – по умолчанию 32 байта;

3. Время отклика сервера в мс – вот это самый основной параметр, на который нужно

обратить максимальное внимание;

4. Количество промежуточных устройств в сети, через которые пакет должен пройти, по-другому называют «время жизни пакета» - Time to live (TTL).

1. Назначение программы cmd.exe

Cmd.exe, также известная как командная строка или интерпретатор командной строки Windows, является программой операционной системы Windows, выполняющей команды ввода пользователя в текстовом формате.  
Назначение программы cmd.exe включает:

**Выполнение команд и управление операционной системой Windows.** С помощью cmd.exe пользователь может выполнять различные команды, такие как создание, копирование, удаление или перемещение файлов и каталогов, изменение параметров системы, настройку сетевых подключений, запуск программ и скриптов и т.д.

**Отладка и диагностика проблем.**

**Автоматизация задач.** Пользователи могут создавать и запускать сценарии, известные как пакетные файлы (.bat), с помощью cmd.exe. Это позволяет автоматизировать и повторять определенные задачи, упрощая работу пользователя.  
**Доступ к системным информационным ресурсам.** Cmd.exe предоставляет пользователю доступ к различным системным файлам, службам и расширенным функциям, что позволяет выполнять дополнительные настройки и управление операционной системой. (REG – Запустить редактор реестра)  
**Запуск других программ и скриптов**. С помощью cmd.exe можно запускать другие программы и скрипты, включая программы на других языках программирования.

1. Операционные системы для современных компьютеров.

1**.Windows** - Является лидером среди операционных систем для компьютеров, ноутбуков и планшетов. Отличительные характеристики: Платная и закрытая ОС; Возможность установки большого количества аппаратного обеспечения; Высокое качество интерфейса; Простота управления; Управление различными устройствами ввода; Высокая скорость работы; Удобная установка ПО; Отличный уровень безопасности (но не в старых версиях); Разнообразие программ и игр.

2. **Mac OS** - Данная ОС встраивается в компьютеры и ноутбуки фирмы Apple. По типу лицензии является закрытой операционной системой, установка которой осуществляется только на устройства бренда.  
Отличительные характеристики: Платная и закрытая ОС. Не предназначена для игр. Устанавливается только на ПК и ноутбуки Apple. Высокая скорость работы. Простота использования. Высокий уровень безопасности. Надежность. Отлаженная координация ресурсов компьютера.

3. **Linux** - Отличительные характеристики: Открытая и бесплатная OC. Бесплатное ПО. Есть возможность выполнять программы от Windows при помощи эмулятора. Хороший уровень безопасности. Отличные показатели распределения ресурсов устройства. Трудности при настройке и использовании. Сложно найти информацию о решении проблем с системой. Ограниченная поддержка аппаратного обеспечения.

4.**Unix** - многопользовательская операционная система. Unix широко используется в веб-серверах, суперкомпьютерах и встроенных системах. Он отличается высокой степенью гибкости и расширяемости, а также поддержкой многозадачности и многопоточности. Unix имеет открытый исходный код, что позволяет пользователям модифицировать и адаптировать систему под свои нужды. В настоящее время на основе Unix существует множество различных дистрибутивов, таких как Linux, FreeBSD, OpenBSD и другие.

1. Файловые системы в Windows.

В Windows существуют несколько типов файловых систем, которые определяют способ хранения и организации данных на жестком диске или других устройствах хранения информации.   
FAT (File Allocation Table) – это одна из самых старых файловых систем в Windows. Она используется в старых версиях операционной системы, таких как Windows 95, Windows 98 и Windows ME. FAT имеет ограниченную поддержку длинных имен файлов, а также не предоставляет механизмов защиты данных.

FAT32 (File Allocation Table 32) – усовершенствованная версия FAT, используемая в более новых версиях Windows, таких как Windows XP, Windows Vista, Windows 7 и Windows 8. FAT32 обеспечивает поддержку более длинных имен файлов и больших томов, но по-прежнему не имеет надежной системы защиты данных.

NTFS (New Technology File System) – это наиболее распространенная файловая система в Windows NT и последующих версиях операционной системы. NTFS предлагает более высокую степень надежности и безопасности данных, поддерживает большие файлы и разделы, а также предоставляет дополнительные функции, такие как шифрование данных и управление доступом.

exFAT (Extended File Allocation Table) – это файловая система, разработанная Microsoft для обеспечения поддержки больших файлов и разделов, которая предназначена для использования съемных носителей, таких как флеш-накопители и внешние жесткие диски. exFAT поддерживает большинство функций NTFS, но не обеспечивает надежность и безопасность данных на том же уровне.

ReFS (Resilient File System) – это новая файловая система, разработанная Microsoft для больших организаций и корпораций. Она обладает высокой стабильностью и защитой данных, но доступна только в некоторых эдициях Windows, таких как Windows Server.

1. Файловые системы в DOS.

В DOS (Disk Operating System), существует несколько типов файловых систем, используемых для организации и управления файлами на диске. Некоторые из наиболее распространенных файловых систем в DOS включают:   
FAT12: это файловая система, используемая на 360 кб и 720 кб дискетах. FAT12 поддерживает имена файлов длиной до восьми символов и трехсимвольные расширения. Она также поддерживает длину имени файла до восьми символов и длину расширения файла до трех символов.   
FAT16: эта файловая система используется на жестких дисках и дискетах большей емкости, чем FAT12. FAT16 поддерживает имена файлов длиной до восьми символов и трехсимвольные расширения, но также может поддерживать длинные имена файлов путем использования длинных записей каталогов.   
FAT32: эта файловая система поддерживает еще большие диски и файлы, чем FAT16. FAT32 поддерживает имена файлов длиной до 255 символов и расширения до пяти символов.   
NTFS: начиная с версии DOS 7.0 и Windows NT, NTFS стала поддерживаться в DOS. NTFS обладает более высокими стандартами безопасности, восстановления после сбоев и управления файлами.

1. NTFS – дать расшифровку аббревиатуры. Основное назначение, причина использования.

NTFS (аббревиатура от англ. new technology file system — «файловая система новой технологии»). Основное назначение заключается в хранении данных внутри операционной системы Windows NT и выше. Данный формат пришел на смену формату FAT и превосходит его в следующих факторах:

* Поддержка больших объемов данных,
* Надежность и восстановление данных,
* Управление доступом и безопасность,
* Поддержка сжатия и шифрования.

1. TCP/IP–дать расшифровку аббревиатуры. Основное назначение, привести примеры использования.

Transmission Control Protocol/Internet Protocol. Для упорядочения обмена данными между компьютерами применяются наборы правил, или протоколы. В настоящее время наиболее широко распространен набор протоколов под общим названием TCP/IP. Основные функции семейства протоколов TCP/IP: Передача файлов в другую систему, Выполнение команд в удаленном режиме, Отправка сообщений удаленным пользователям, Вход в удаленную систему.

1. FAT32 и NTFS дать расшифровку аббревиатуры. Основное назначение, причина использования.

FAT32 (от англ. File Allocation Table — «таблица размещения файлов») — это файловая система, созданная Microsoft, разновидность FAT. FAT32 — предпоследняя версия файловой системы FAT. Она была создана, чтобы преодолеть ограничения на размер тома в FAT16, позволяя при этом использовать старый код программ MS-DOS и сохранив формат. FAT32 использует 32-разрядную адресацию кластеров.

1. Классы TCP/IP. Назначение и характеристики классов TCP/IP

IP-адреса делятся на 5 классов (A, B, C, D, E). A, B и C — это классы коммерческой адресации. D – для многоадресных рассылок, а класс E – для экспериментов.

Класс A - этот класс IP-адресов используется для крупных сетей с большим количеством узлов.

Класс B - этот класс используется для средних сетей и имеет диапазон адресов от 128 до 191. Он позволяет использовать больше адресов на узел, но меньше узлов в сети.

Класс C - этот класс предназначен для небольших сетей и имеет самый маленький диапазон адресов (от 192 до 223).

Класс D - этот класс зарезервирован для многоадресной рассылки и не может быть использован для обычных IP-адресов.

Класс E - этот класс также зарезервирован для будущего использования и не должен использоваться в настоящее время.

1. Linux и Windows. Основные отличия.

**Стоимость:** Linux обычно распространяется бесплатно, в то время как Windows требует оплаты.

**Открытость исходного кода:** Linux имеет открытый исходный код, что позволяет пользователям и разработчикам свободно изучать и изменять его код. Windows имеет закрытый исходный код.

**Многообразие:** существует множество дистрибутивов Linux, каждый из которых имеет свои особенности и подходит для разных целей. Microsoft предлагает только одну версию Windows.

**Удобство использования:** Windows имеет более удобный и интуитивно понятный интерфейс, чем многие дистрибутивы Linux. Однако многие дистрибутивы Linux становятся все более удобными для пользователя, особенно те, которые ориентированы на новичков.

**Совместимость:** Windows широко используется и имеет отличную совместимость с большинством приложений и устройств. Linux также имеет хорошую совместимость, но некоторые приложения и устройства могут не работать на нем без дополнительной настройки.

1. Microsoft Office. Назначение, совместимость и аналоги для OC Windows, Linux?

*Microsoft Office* – это набор приложений для работы с документами, включающий Word, Excel, PowerPoint и OneNote. Он был создан компанией Microsoft для операционной системы Windows и macOS.

*Назначение* Microsoft Office заключается в том, чтобы предоставить пользователям удобные инструменты для работы с текстовыми документами, электронными таблицами, презентациями и заметками.

*Для Linux существуют альтернативные пакеты офисных приложений, такие как:*

* LibreOffice
* WPS Office
* OnlyOffice
* Softmaker
* Freeoffice

1. СУБД. Назначение, типы данных в СУБД, основные операции?

СУБД – это система управления базами данных, которая предназначена для хранения, обработки и предоставления доступа к данным. Основные операции в СУБД включают создание, изменение, удаление записей, фильтрацию данных по заданным условиям и обновление данных.

*Типы данных в СУБД:*

Строка (CHAR, VARCHAR)

Числовой (INTEGER, SMALLINT, BIGINT, DECIMAL, NUMERIC, REAL, FLOAT

Дата и время (DATE, TIME, TIMESTAMP

Логический (BOOLEAN, BIT

Пользовательский тип (USER-DEFINED) - тип данных определяется пользователем, может быть основан на других типах данных или представлять собой составной тип данных (массив, структура, объединение).

Геометрический - типы данных для хранения географических координат и векторных объектов.

1. QBE – дать расшифровку аббревиатуры. Основное назначение. Примеры применения.

QBE расшифровывается как **“Query By Example”** (Запрос по Примеру) и является методом написания запросов к базам данных на основе примера желаемого результата. **Основное назначение QBE** - упростить процесс написания SQL-запросов для пользователей, которые могут быть менее знакомы с языком SQL.

*Примеры применения QBE включают:*

**Создание отчетов:** например, можно создать отчет, который показывает продажи по каждому товару за определенный период времени.

**Фильтрация данных:** например, можно использовать QBE для поиска всех записей, относящихся к определенному товару или клиенту.

**Изменение данных:** например, для обновления цен на товары или для удаления старых записей.

1. SQL – дать расшифровку аббревиатуры. Основное назначение, причина использования.

**SQL** (Structured Query Language/структурированный язык запросов/язык структурированных запросов) – это декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных.

В отличие от процедурных языков программирования, он определяет не последовательность процедур, а условие выполнения запроса.

Причина использования SQL заключается в его главных задачах — составлять запросы так, чтобы находить среди большого объёма информации ту, что нужна для конкретных целей, сортировать её, структурировать и представлять в наиболее простом и понятном виде.

1. DDL и DML – дать расшифровку аббревиатуры. Основное назначение, и область применения.

**DDL** (Data Definition Language/язык описания данных) – включает в себя операторы определения данных, которые позволяют создавать объекты базы данных (таблицы, представления, триггеры, хранимые процедуры, индексы, ключи, ограничения и т.д.). Эти операторы не могут быть отменены.

Операторы:  
• CREATE (*TABLE, VIEW, TRIGGER*…) **– создаёт** указанный объект БД.

• DROP (*TABLE, VIEW,**TRIGGER*…) – **уничтожает** указанный объект БД.

• ALTER (*TABLE, VIEW, TRIGGER*…) – **изменяет** структуру указанного объекта БД  
**DML** (Data Manipulation Language/язык манипулирования данными) – включает в себя операторы манипулирования данными, которые позволяют производить извлечение, добавление, изменение и удаление данных. Эти операторы могут быть отменены.  
Операторы:  **•** SELECT – оператор **извлечения** данных.

**•** INSERT – оператор **добавления** данных.

• UPDATE – оператор **изменения** данных.

• DELETE – оператор **удаления** данных.

Область применения DDL и DML – в различных базах данных.

1. Компьютеры в СССР, примеры применения (использование).

**МЭСМ** (Малая электронная счётная машина) производила сложные арифметические вычисления и применялась для:   
• Решения динамических задач теории упругости;  
• Расчёта тепловых напряжений строительных конструкций;  
• Обработки геодезических наблюдений.  
**БЭСМ-1** (Большая электронная счётная машина). Применение:   
• Рассчитывала траекторию полёта снаряда быстрее, чем снаряд долетал до цели.  
• Рассчитала траекторию полёта ракеты, доставившей на Луну вымпел СССР в 1959 году.

1. Процессоры. Назначение и использование для вычислительных систем. Основные производители (перечислить), отличия.

**Назначение процессора** – это обработка команд и данных, поступающих от других компонентов вычислительной системы. Процессор используется для выполнения различных операций, таких как сложение, умножение, деление, а также для управления другими компонентами системы.

**Основные производители процессоров:** Intel, AMD, Nvidia, Qualcomm, Apple. Каждый из них имеет свои особенности и преимущества. Например, процессоры Intel обычно считаются более стабильными и надежными, в то время как процессоры AMD могут предложить более высокую производительность за меньшую стоимость. Процессоры Nvidia используются в основном для графических приложений, а Qualcomm - для мобильных устройств.

Отличия между производителями процессоров заключаются в различных технологиях, используемых в производстве, таких как количество ядер, тактовая частота, объем кэш-памяти, энергопотребление и т.д.

1. Гипервизор. Что это? Назначение, пример использования.

Гипервизор (Hypervisor) – монитор виртуальных машин – программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких операционных систем на одном и том же компьютере. Гипервизор обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными OC и управление ресурсами.  
Пример использования гипервизора – это установка Microsoft Hyper-V для создания виртуальных серверов и рабочих станций. Гипервизоры также могут использоваться на серверах облачных платформ для управления виртуальными ресурсами и предоставления услуг виртуализации.

1. NetBEUI, IPX (IPX/SPX). Что это? Назначение и использование

NetBEUI (NetBIOS Extended User Interface) – расширенный пользовательский интерфейс дейтаграммной передачи NetBIOS.

*\*NetBIOS - Network Basic Input/Output System (Сетевая базовая система ввода/вывода).*

Назначение NetBEUI заключается в обеспечении сетевой связности между компьютерами в локальной сети. Он обеспечивает возможность обмена данными и ресурсами между компьютерами, подключенными к сети, и обеспечивает простую сетевую конфигурацию.

IPX (Internetwork Packet Exchange) и SPX (Sequenced Packet Exchange) – это протоколы, которые были разработаны компанией Novell и использовались в компьютерных сетях, особенно в локальных сетях, прежде чем протокол TCP/IP стал доминирующим стандартом для сетевой связности.

Назначение IPX/SPX заключается в обеспечении сетевой связности и коммуникации между компьютерами в сети Novell NetWare. Они обеспечивают маршрутизацию, управление потоками данных и обработку ошибок.

1. Что такое архиватор. Назначение. Пример применения.

**Архиватор** – это средство сжатия данных, программа, предназначенная для упаковки файлов в архивы и их последующей распаковки. Предназначены для создания архивов. Архивирование данных упрощает их хранение за счет того, что большие группы файлов и каталогов сводятся в один архивный файл.

Пример применения архиватора – это использование программы WinRAR или 7-Zip для сжатия файлов перед их отправкой по электронной почте или для создания архивов с несколькими файлами перед записью их на съемный носитель.

1. RISC и CISC – дать расшифровку аббревиатуры. Основное назначение, область применения (использования)

RISC (Reduced Instruction Set Computing) и CISC (Complex Instruction Set Computing) - это процессоры с расширенной системой команд.

Основное назначение RISC-процессоров - увеличение производительности за счет оптимизации выполнения инструкций и сокращения времени на их выполнение. RISC-процессоры используют в специализированных вычислительных системах или устройствах, ориентированных на выполнение единообразных операций.

Основное назначение CISC-процессоров - обеспечение универсальности и расширенного набора инструкций для обработки различных задач. CISC-процессоры используют в универсальных вычислительных системах.

1. Перевести число из шестнадцатеричной системы в двоичную, затем в десятичную.

Из шестнадцатеричной в двоичную:  
Каждая цифра в шестнадцатеричном числе представляет собой последовательность из четырех двоичных цифр. Например, число "A" в шестнадцатеричной системе равно "1010" в двоичной системе.  
Повторите этот процесс для каждой цифры в шестнадцатеричном числе.  
Пример: переведем число "1A3" из шестнадцатеричной системы в двоичную.  
"1" = 0001, "A" = 1010, "3" = 0011  
Таким образом, "1A3" в двоичной системе будет равно "000110100011".  
Из двоичной в десятичную:  
Для перевода числа из двоичной системы в десятичную систему умножьте каждую цифру двоичного числа на 2 в степени, соответствующей позиции числа, начиная с 0 справа.  
Пример: переведем число "1101" из двоичной системы в десятичную.  
12^3 + 12^2 + 02^1 + 12^0 = 8 + 4 + 0 + 1 = 13  
Таким образом, число "1A3" в шестнадцатеричной системе после перевода в двоичную будет равно "000110100011", а после перевода из двоичной в десятичную будет равно 419.