**9. Дайте определение понятию «Интернет».**

Глобальная система взаимосвязанных компьютерных сетей;

1) сеть на основе TCP/IP;

2) стандарты Internet (RFC, STD);

3) службы Интернет (DNS, SMTP/POP3/IMAP, WWW, FTP, Telnet, SSH,…);

4) организации, управляющие сетью Internet (ISOC, IETF, ICANN, IANA, W3C, …).

**10. Дайте определение понятию «Служба Интернет».**

Онлайн-сервис, предоставляемый через Интернет, который может предоставлять различные функциональные возможности

**11. Дайте определение понятию «Узел сети Интернет».**

устройство, имеющее IP-адрес и подключенное к сети Интернет (обычно к сети Интернет-провайдера). Каждый узел характеризуется своей программно-аппаратной платформой – аппаратурой и операционной системой

**12. Дайте определение понятию «клиент-серверное приложение».**

Архитектурный шаблон в разработке программного обеспечения, где одни компоненты (клиенты) запрашивают и получают услуги или ресурсы от других компонентов (серверов). Клиенты и серверы взаимодействуют через сеть, часто посредством сетевых протоколов.

**13. Дайте определение понятию «сетевой протокол».**

Набор правил и соглашений, которые определяют формат и последовательность обмена данными между устройствами в сети. Протоколы определяют, как информация упаковывается, передается и распаковывается, чтобы обеспечить надежную и эффективную связь.

**14. Перечислите основные свойства протокола HTTP.**

* версии HTTP/1.1 – действующий (текстовый), HTTP/2 – черновой (не распространен, бинарный), HTTP/3 (раньше HTTP-over-QUIC, стадия черновика);
* два типа абонентов: клиент и сервер;
* два типа сообщений: request и response;
* от клиента к серверу – request;
* от сервера к клиенту – response;
* на один request всегда один response, иначе ошибка;
* одному response всегда один request, иначе ошибка;
* stateless (каждый запрос как независимая транзакция, которая не связана с предыдущим запросом);
* TCP-порты: 80, 443;
* для адресации используется URI или URL;
* поддерживается W3C, описан в нескольких RFC.

**15. Перечислите состав информации, пересылаемой в HTTP-запросе.**

* метод;
* URI;
* версия протокола (HTTP/1.1);
* заголовки (пары: имя/заголовок);
* *параметры (пары: имя/заголовок);*
* расширение.
* 

**16. Перечислите состав информации, пересылаемой в HTTP-ответе.**

* версия протокола (HTTP/1.1);
* код состояния (1xx, 2xx, 3xx, 4xx, 5xx);
* пояснение к коду состояния;
* заголовки (пары: имя/заголовок);
* расширение.

**17. Перечислите группы заголовков HTTP.**

* **Общие заголовки(General):** применяются к обоим запросам и ответам.



* **Заголовки запроса(Request):** предоставляют информацию о запросе.



* **Заголовки ответа(Response):** предоставляют информацию о ответе.



* **Заголовки сущности(Entity):** предоставляют информацию о сущности, передаваемой или возвращаемой.
* Пользовательский?



**18. Дайте определение понятию «web-приложение».**

клиент-серверное приложение, компоненты которого взаимодействуют по протоколу HTTP

**19. Дайте определение понятиям «frontend» и «backend».**

* **Frontend** - это часть веб-приложения, с которой взаимодействует пользователь. Это включает в себя пользовательский интерфейс, который отображается в браузере, и клиентскую логику, которая выполняется на стороне клиента (например, JavaScript).
* **Backend** - это серверная часть веб-приложения, которая обрабатывает запросы от клиентов, выполняет бизнес-логику, взаимодействует с базами данных и отправляет данные клиентам. Backend обычно работает на сервере.

**20. Дайте определение понятию «кроссплатформенное приложение».**

Приложение, которое может работать на разных операционных системах или платформах без значительных изменений. Кроссплатформенные приложения обычно разрабатываются с использованием специальных технологий и инструментов, которые обеспечивают совместимость с разными средами выполнения.

**21. Изобразите и поясните общую схему web-приложения.**



**22. Назовите основные технологии разработки серверных кроссплатформенных приложений.**

- PHP/Apache, LAMP;

- Java/JVM/Application Server;

- C#/ASP.NET CORE;

- Python/Django;

- Ruby on Rails;

- JS/Node.js, ….

**23. Поясните понятие «асинхронная операция».**

Асинхронная операция (или асинхронная задача) - это операция в программировании, которая выполняется в фоновом режиме, без блокировки выполнения других операций в приложении

**24. Поясните основное назначение NODE.JS.**

Среда выполнения JavaScript, построенная на движке Chrome V8, которая позволяет выполнять JavaScript на сервере. Основным назначением Node.js является создание масштабируемых и высокопроизводительных сетевых приложений, таких как веб-серверы. Node.js поддерживает асинхронное программирование, что делает его идеальным для обработки множества одновременных запросов.

**25. Перечислите основные свойства NODE.JS.**

* основан на **Chrome V8**;
* **среда (контейнер) исполнения** приложений на JavaScript;
* поддерживает механизм **асинхронности**;
* ориентирован на **события**;
* **однопоточный** (код приложения исполняется только в одном потоке, один стек вызовов); обычно в серверах для каждого соединения создается свой поток, в Node.js все соединения обрабатываются в одном JS-потоке;
* **не блокирует** выполнение кода при вводе/выводе (в файловой системе до 4х одновременно);
* в состав Node.js входят инструменты: **npm** – пакетный менеджер; **gyp** - Python-генератор проектов; **gtest** – Google фреймворк для тестирования С++ приложений;
* использует библиотеки: **V8** – библиотека V8 Engine, **libuv** – библиотека для абстрагирования неблокирующих операций ввода/вывода (представляет собой обертку над epoll, kqueue, IOCP); **llhttp** – легковесный парсер http-сообщений (написан на C и не выполняет никаких системных вызовов); **c-ares** -библиотека для работы с DNS; **OpenSSL** – библиотека для криптографии; **zlib** – сжатие и распаковка.