Вопросы к зачетурассмотрены и одобрены на заседании кафедры ИСиТ (протокол № 4 от 17 ноября 2023 г.)

И.о. заведующего кафедрой

информационных систем и технологий,

старший преподаватель Е. А. Блинова

**Вопросы по дисциплине**

**«Программирование серверных кроссплатформенных приложений»**

**для студентов 3-го курса специальности ПОИТ**

1. Протокол HTTP, основные свойства HTTP, структура запроса и ответа, методы, заголовки, коды состояний. Понятие web-приложения, структура и принципы работы web-приложения. Понятие асинхронности.
2. Web-сервер. Ресурсы, потребляемые web-сервером. Блокирующие и неблокирующие операции ввода/вывода. Решение проблемы блокирующего ввода/вывода. Понятия конкурентность и параллельность. Закон Амдала.
3. Платформа Node.js, версии, назначение, основные свойства, структура, принципы работы, основные встроенные модули и их назначение, применение внешних модулей (пакетов).
4. Глобальные объекты Node.js (global, process) и их применение. Системные (стандартные потоки) Node.js и их применение. Модуль console: функции log, error, dir, time, timeEnd, trace. Примеры.
5. Асинхронное программирование. Функция обратного вызова. Проблема «Callback hell» и способы решения. Примеры.
6. Асинхронное программирование. Механизм Promises. Механизм async/await. Примеры.
7. Класс EventEmitter, назначение, применение. Пример.
8. Функции setTimeout, setInterval, nextTick, ref, unref, назначение, применение. Примеры.
9. Модули и пакеты Node.js, CommonJS, функция require, кэширование модуля, область видимости в пакете, экспорт объектов, функций, конструкторов. Применение require для работы с json-файлами. Параметризируемый модуль. Пример.
10. Модули Node.js. Форматы модулей. Модули ES6: экспорт (по умолчанию, именованный, до/после объявления), импорт, динамический импорт. Примеры.
11. Пакетный менеджер NPM, глобальное хранилище, просмотр установленных пакетов, скачивание пакетов, назначение файла package.json, локальные хранилища пакетов, удаление пакетов, публикация пакета, SemVer. Примеры.
12. HTTP-сервер. Порядок разработки простейшего HTTP-сервера в Node.js. События, генерируемые на сервере. Извлечение данных из HTTP-запроса (параметры, тело, заголовки…), формирование данных HTTP-ответа.
13. HTTP-сервер. Порядок разработки простейшего HTTP-сервера в Node.js. Объекты request и response, их свойства, методы, события. Порционная отправка и получение данных.
14. HTTP-сервер. Типичный цикл работы http-сервера, маршрутизация. Статические ресурсы, MIME, отдача статики.
15. JSON, XML: порядок работы с json- и xml-сообщениями (формирование и разбор). Загрузка файла на сервер (пакет multiparty).
16. HTTP-клиент. Порядок разработки простейшего HTTP-клиента в Node.js (модуль http). Отправка GET- и POST-запросов с параметрами, добавление заголовков и обработка ответа. Порционная отправка данных.
17. HTTP-клиент. Порядок разработки HTTP-клиента в Node.js (axios). Отправка GET- и POST-запросов с параметрами, параллельных запросов, добавление заголовков и обработка ответа.
18. Пакет Express. Основные принципы работы. Маршрутизация, промежуточные обработчики, обработка ошибок.
19. Пакет Express. Основные принципы работы. Статические файлы. Отдача статики. Кэширование на стороне клиента.
20. Пакет Express. Основные принципы работы. Объекты запроса и ответа. Обработка query- и path-параметров GET-запроса. Методы download и attachment, переадресация.
21. Пакет Express. Основные принципы работы. Обработка тела (form, json) POST-запроса. Методы accepts и format.
22. Пакет Express. Основные принципы работы. Порядок работы с cookie и signed cookie.
23. Пакет Express. Основные принципы работы. Порядок работы с session.
24. Websocket: основные свойства, процедура установки соединения, обмен данными, применение. WebSockets API.
25. Порядок разработки Websocket-сервера и клиента: использование потоков, ping/pong-сообщения, обработка json-сообщений.
26. Понятие RPC. Пакет rpc-websockets: порядок разработки RPC-Websockets-сервера и клиента, работа с процедурами, генерация событий и обработка уведомлений.
27. Файловая система. Порядок работы с файловой системой в Node.js: создание, копирование, чтение, запись, синхронные и асинхронные операции.
28. Файловая система. Порядок работы с файловой системой в Node.js: создание, удаление, переименование, запись в конец, слежение за файлом/директорией, синхронные и асинхронные операции.
29. Файловая система. Порядок работы с файловой системой в Node.js: работа с директориями, проверка наличия файла/директории, синхронные и асинхронные операции. Объект Buffer и работа с ним.
30. Потоки данных, их виды, примеры. Readable поток: режимы работы, порядок создания и использования Readable потока.
31. Потоки данных, их виды, примеры. Writable и Duplex потоки: порядок создания и использования.
32. TCP: основные свойства, установка и закрытие соединения. Разработка простейшего TCP-сервера и TCP-клиента на Node.js.
33. UDP: основные свойства, отличия от TCP. Разработка простейшего UDP-сервера и UDP-клиента на Node.js.
34. MSSQL: пакет mssql, драйверы, подключение, параметры подключения, поддерживаемые типы данных, обычные и подготовленные запросы, обработка результата запроса*,* создание неявной транзакции.
35. MSSQL: пакет mssql, драйверы, подключение, параметры подключения, статические и динамические запросы, вызов процедуры, создание явной транзакции.
36. MSSQL: пакет mssql, потоковый режим обработки данных, пул соединений.

ст. препод. каф. ИСиТ М.В. Дубовик