Экзаменационные вопросы пересмотрены и одобрены на заседании кафедры ИСиТ (протокол №10 от 09 апреля 2024 г.)

И.о. заведующего кафедрой

информационных систем и технологий,

старший преподаватель Е. А. Блинова

**Экзаменационные 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, параметры, коды состояний. Протокол HTTPS. 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). Форматы модулей.
4. Асинхронное программирование. Функция обратного вызова. Проблема «Callback hell» и способы решения.
5. Асинхронное программирование. Механизм Promises. Механизм async/await.
6. Класс EventEmitter, назначение, применение.
7. Функции setTimeout, setInterval, nextTick, ref, unref, назначение, применение.
8. 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.
9. Модули Node.js. Форматы модулей. Модули ES6: экспорт (по умолчанию, именованный, до/после объявления), импорт, динамический импорт.
10. Пакетный менеджер NPM, глобальное хранилище, просмотр установленных пакетов, скачивание пакетов, назначение файла package.json, локальные хранилища пакетов, удаление пакетов, публикация пакета, SemVer.
11. Websocket: основные свойства, процедура установки соединения, обмен данными, применение. WebSockets API.
12. Порядок разработки Websocket-сервера и клиента: использование потоков, ping/pong-сообщения, обработка json-сообщений.
13. Понятие RPC. Пакет rpc-websockets: порядок разработки RPC-Websockets-сервера и клиента, работа с процедурами, генерация событий и обработка уведомлений.
14. Файловая система. Порядок работы с файловой системой в Node.js: создание, копирование, чтение, запись, синхронные и асинхронные операции.
15. Файловая система. Порядок работы с файловой системой в Node.js: создание, удаление, переименование, запись в конец, слежение за файлом/директорией, синхронные и асинхронные операции.
16. 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.
17. Потоки данных, их виды, примеры. Readable поток: режимы работы, порядок создания и использования Readable потока.
18. Потоки данных, их виды, примеры. Writable и Duplex потоки: порядок создания и использования.
19. TCP: основные свойства, установка и закрытие соединения. Разработка простейшего TCP-сервера и TCP-клиента на Node.js.
20. UDP: основные свойства, отличия от TCP. Разработка простейшего UDP-сервера и UDP-клиента на Node.js.
21. HTTP-аутентификация, виды. Общий принцип работы каждого из них.
22. Протокол HTTPS. Протокол TSL. Сертификаты. Взаимодействие центра сертификации и владельца защищенного ресурса.
23. Redis. Основные характеристики Redis. Применение и основные принципы работы.
24. Применение пакета Sequelize. Основные принципы работы (подключение, объявление моделей, конфигурация, подход Code First, поисковые методы, фильтрация, пагинация, сортировка, группировка).
25. Применение пакета Sequelize. Основные принципы работы (ассоциации, добавление, изменение, удаление строк в таблице, raw query).
26. Применение пакета Sequelize. Основные принципы работы (scopes, хуки, соединения, ассоциации, транзакции).
27. Пакет Prisma. Основные характеристики. Способы создания моделей. Основные принципы работы (подключение, схема Prisma (поля, типы, атрибуты, модификаторы, функции атрибутов), поисковые методы (с различными опциями)).
28. Пакет Prisma. Основные характеристики. Способы создания моделей. Основные принципы работы (подключение, выборка связанных записей, фильтры на связи -ко-многим, -к-одному, Fluent API, пагинация, подсчет записей, агрегирование, группировка, distinct).
29. Пакет Prisma. Основные характеристики. Способы создания моделей. Основные принципы работы (подключение, создание записей исходной модели и связанной, присоединение существующих записей, изменение записей исходной модели и связанной).
30. Пакет Prisma. Основные характеристики. Способы создания моделей. Основные принципы работы (подключение, схема Prisma (поля, типы, атрибуты, модификаторы, функции атрибутов), удаление записей исходной модели и связанной, транзакции, миграции).
31. Пакет Express. Основные принципы работы. Маршрутизация, промежуточные обработчики, обработка ошибок.
32. Пакет Express. Основные принципы работы. Статические файлы. Отдача статики. Кэширование на стороне клиента.
33. Пакет Express. Основные принципы работы. Объекты запроса и ответа. Обработка query- и path-параметров GET-запроса. Методы download и attachment, переадресация.
34. Пакет Express. Основные принципы работы. Обработка тела (form, json) POST-запроса. Методы accepts и format.
35. Handlebars. Назначение и порядок использования шаблонов, макетов, хэлперов и частичных представлений.
36. Пакет Express. Основные принципы работы. Порядок работы с cookie и signed cookie.
37. Пакет Express. Основные принципы работы. Порядок работы с session.
38. Аутентификация. Разработка приложения с Basic-аутентификацией.
39. Аутентификация. Разработка приложения с Digest-аутентификацией.
40. Аутентификация. Разработка приложения с Forms-аутентификацией.
41. Токен-аутентификация.
42. OAuth 2.0. Общая схема работы OAuth. Разрешение на авторизацию, виды. OpenID.
43. Авторизация: виды, реализация. Библиотека CASL.
44. Протокол WebDav. Назначение, применение, стандарты, методы. Разработка приложения с применением WebDav.
45. Telegram Bot. Порядок разработки telegram bot’а (pooling). Polling, long pooling, webhook, принцип работы.
46. Telegram Bot. Порядок разработки telegram bot’а (webhooks). Polling, long polling, webhook, принцип работы.
47. REST: определение, ограничения. Open API, Swagger. Порядок разработки сервера с поддержкой Swagger.

Старший преподаватель кафедры ИСиТ М.В. Дубовик

Секретарь кафедры ИСиТ Е.В. Барковский

**Перечень лабораторных работ**

1. Лабораторная 1 (ВВЕДЕНИЕ, HELLO WORLD)
2. Лабораторная 2 (ASYNC REQUESTS)
3. Лабораторная 3 (GLOBALS, ASYNC)
4. Лабораторная 4 (EVENT EMITTER)
5. Лабораторная 5 (NPM)
6. Лабораторная 6 (HTTP SERVER)
7. Лабораторная 7 (HTTP CLIENT)
8. Лабораторная 8 (WS)
9. Лабораторная 9 (WS RPC)
10. Лабораторная 10 (TCP, UDP)
11. Лабораторная 11 (MSSQL)
12. Лабораторная 12 (REDIS).
13. Лабораторная 13 (SEQUELIZE).
14. Лабораторная 14 (PRISMA)
15. Лабораторная 15 (EXPRESS\_HBS)
16. Лабораторная 16 (Basic, Digest).
17. Лабораторная 17 (Forms).
18. Лабораторная 18 (OAuth).
19. Лабораторная 19 (CASL).
20. Лабораторная 20 (HTTPS).
21. Лабораторная 21 (WEBDAV).
22. Лабораторная 22 (TLGBOT).
23. Лабораторная 23 (SWAGGER).