



Backend-разработка.фип_БАК_н/с

- 1) Какое ключевое слово используется для определения функции в Python?
- 2) Какое ключевое слово используется для определения класса в Python?
- 3) Какой метод автоматически вызывается при создании нового объекта класса?
- 4) Какой из следующих примеров показывает правильное использование необязательных аргументов в функции?
- 5) Принцип единственной ответственности (SRP) требует, чтобы:
- 6) Принцип открытости/закрытости (OCP) предполагает, что:
- 7) Упорядочите следующие шаги для создания Singleton паттерна в Python:
- 8) Принцип инверсии зависимостей (DIP) предполагает, что:
- 9) Соотнесите принципы SOLID с их описаниями.
- 10) Какой из следующих примеров является реализацией паттерна Singleton в Python?
- 11) Абстрактный класс в Python:
- 12) Какой декоратор используется для обозначения абстрактного метода в Python?
- 13) Какой принцип SOLID отвечает за обеспечение легкости изменения кода без необходимости модификации существующих классов?
- 14) Принцип единственной ответственности говорит о том, что:
- 15) В каком случае вы не нарушаете принцип подстановки Барбары Лисков?
- 16) Какой из следующих методов демонстрирует правильное использование принципа инверсии зависимостей?
- 17) Какой принцип SOLID помогает избежать создания "толстых" интерфейсов?





- 18) Паттерн Singleton гарантирует:
- 19) Какой из следующих методов является частью реализации Singleton паттерна?
- 20) Какой метод в Python используется для создания единственного экземпляра класса в паттерне Singleton?
- 21) Что такое асинхронное программирование?
- 22) Какой метод из библиотеки `asyncio` используется для планирования выполнения функции в ближайшее время?
- 23) Чем отличается метод `call_later()` от метода `call_soon()`?
- 24) Что делает функция `asyncio.as_completed()`?
- 25) Какой метод из библиотеки `asyncio` позволяет добавить функцию в очередь выполнения через определенное время?
- 26) Какой метод позволяет добавлять колбэки к асинхронным задачам?
- 27) Какой ключевой элемент асинхронного программирования в `asyncio` позволяет приостанавливать и возобновлять выполнение функции?
- 28) Какой метод позволяет запустить асинхронный цикл событий в `asyncio`?
- 29) Что возвращает функция `asyncio.gather()`?
- 30) Упорядочите методы в библиотеке `asyncio`, которые используются для управления асинхронными задачами, от самого быстрого к медленному:
- 31) Упорядочите действия, связанные с выполнением асинхронной функции, от первого к последнему:
- 32) Соответствие функций и их описаний:
- 33) Что делает метод `asyncio.run()`?
- 34) Какой метод в `asyncio` используется для создания новых задач в цикле событий?
- 35) Как остановить асинхронный цикл событий в `asyncio`?





- 36 Какой из следующих методов используется для асинхронного ожидания выполнения нескольких задач?
- 37 Что возвращает функция `asyncio.create_task()`?
- 38 В каком порядке выполняются задачи, созданные с помощью `asyncio.create_task()`?
- 39 Какой метод позволяет завершить выполнение всех задач и выйти из цикла событий `asyncio`?
- 40 Какой метод в библиотеке `asyncio` используется для создания нового цикла событий?
- 41 Какой класс в модуле `concurrent.futures` представляет пул потоков?
- 42 Какой метод в `ThreadPoolExecutor` используется для планирования выполнения задачи?
- 43 Что возвращает метод `submit()` в `ThreadPoolExecutor`?
- 44 Какой метод используется для корректного завершения работы `ThreadPoolExecutor`?
- 45 Какой метод в `ThreadPoolExecutor` позволяет получить результат выполнения задачи?
- 46 Какая функция в `ThreadPoolExecutor` используется для ожидания завершения всех задач?
- 47 Какой метод в `ThreadPoolExecutor` позволяет добавить функцию обратного вызова к задаче?
- 48 Что делает метод `wait()` в `ThreadPoolExecutor`?
- 49 Задание на соответствие
- 50 Какой метод `asyncio` создает новый цикл событий?
- 51 Что делает метод `download_data()` в асинхронном контексте?
- 52 Какой метод используется для получения результата асинхронной задачи?
- 53 Что такое `ThreadPoolExecutor.map()`?
- 54 Упорядочите методы в соответствии с их выполнением:





- 55) Какой тип блокировки используется для рекурсивного захвата ресурса?
- 56) Какой класс в threading предоставляет возможность синхронизации потоков?
- 57) Какой метод блокировки Lock позволяет снять блокировку и освободить ресурс?
- 58) Что делает метод acquire() у RLock?
- 59) Какой метод используется для завершения работы пула потоков и получения результатов всех задач?
- 60) Какой метод класса Lock используется для захвата блокировки?
- 61) Какой метод в SQLAlchemy используется для создания объекта движка базы данных?
- 62) Какой класс предоставляет интерфейс для управления транзакциями и отслеживания изменений в SQLAlchemy?
- 63) Какой метод объекта MetaData используется для создания всех таблиц в базе данных?
- 64) Какой тип связи в SQLAlchemy устанавливается между двумя таблицами, где каждая запись в одной таблице связана с одной записью в другой таблице?
- 65) Какой аргумент используется для указания обратной ссылки в функции relationship()?
- 66) Какой метод используется для создания асинхронного движка базы данных в SQLAlchemy?
- 67) Какой метод в Pydantic используется для ручной проверки данных модели?
- 68) Какой метод SQLAlchemy позволяет выполнять SQL-запросы напрямую?
- 69) В каком объекте SQLAlchemy хранятся все определения таблиц и их метаданные?
- 70) Какой класс в SQLAlchemy используется для создания связующей таблицы в связи "Многие-ко-Многим"?
- 71) Какой аргумент функции relationship() используется для указания связующей таблицы в связи "Многие-ко-Многим"?





- 72) В каком формате Pydantic позволяет преобразовывать данные для сериализации?
- 73) Какой метод используется для создания асинхронной сессии в SQLAlchemy?
- 74) Какой метод используется для добавления нового объекта в базу данных через Session?
- 75) Какой метод в SQLAlchemy позволяет получить доступ к объектам модели в Session?
- 76) Какой метод SQLAlchemy используется для обновления существующих записей в таблице?
- 77) Какой аргумент функции relationship() указывает имя атрибута обратной ссылки?
- 78) Упорядочите следующие шаги для создания таблицы и работы с ней в SQLAlchemy:
- 79) Упорядочите следующие шаги для создания таблицы и работы с ней в SQLAlchemy:
- 80) Какое из следующих преимуществ предлагает FastAPI?
- 81) Какой архитектурный стиль используется в RESTful API?
- 82) Какой HTTP-метод обычно используется для обновления ресурса на сервере?
- 83) Какое преимущество асинхронного выполнения запросов в FastAPI?
- 84) Какой HTTP-метод используется для получения данных с сервера?
- 85) Что такое эндпоинт в контексте веб-сервисов?
- 86) Какой принцип проектирования API обеспечивает безопасность данных?
- 87) Какой принцип проектирования API предполагает использование единообразных имен для ресурсов и методов?
- 88) Какой декоратор используется для установки времени жизни кэша в Redis?
- 89) Какой модуль используется для работы с Redis в FastAPI?





- 90 Какой параметр указывает на использование брокера сообщений в Celery?
- 91 Какой метод используется для асинхронного вызова задачи в Celery?
- 92 Упорядочите шаги для создания фоновой задачи с использованием Celery и Redis:
- 93 Соотнесите принципы проектирования API с их описаниями:
- 94 Какой декоратор используется для определения эндпоинта в FastAPI, который обрабатывает запросы типа POST?
- 95 Какой параметр указывает на использование Redis в качестве бэкенда для хранения сессий в FastAPI?
- 96 Какой тип данных Redis подходит для хранения строк?
- 97 Какой из следующих методов Celery используется для определения задачи, которая должна быть выполнена асинхронно?
- 98 Какой ключевой компонент используется в Celery для организации распределенных задач?
- 99 Какой метод в FastAPI используется для определения эндпоинта, который обрабатывает запросы типа GET?
- 100 Упорядочите шаги для создания таблицы и работы с ней в SQLAlchemy:
- 101 Упорядочите следующие этапы создания и использования асинхронного метода с использованием asyncio:
- 102 Соответствие между типами фреймворков и их характеристиками
- 103 Соответствие между принципами RESTful API и их описаниями:
- 104 Какие методы HTTP используются в RESTful API для работы с ресурсами?
- 105 Какой из следующих фреймворков наиболее подходит для создания высоконагруженных приложений с большим количеством одновременных подключений?
- 106 Какой метод используется для создания асинхронных задач в Celery?
- 107 Какой из следующих принципов проектирования API помогает обеспечить обратную совместимость?





- 108) Что из следующего является правильным способом создания модели в SQLAlchemy?
- 109) Какая библиотека используется для создания фоновых задач в FastAPI с помощью Redis?
- 110) Что из перечисленного является характеристикой Redis?
- 111) Какой декоратор используется для кэширования функций в Python с использованием Redis?
- 112) Какой фреймворк предоставляет полный набор инструментов для создания веб-приложений любой сложности?
- 113) Какой из следующих методов используется для создания и запуска асинхронных задач с помощью asyncio?
- 114) Что такое эндпоинт в контексте веб-сервисов и API?
- 115) Какой из следующих методов не относится к HTTP методам в RESTful API?
- 116) В чем основное преимущество использования асинхронных методов с asyncio в Python?
- 117) Какой из следующих принципов API относится к разделению клиентской и серверной частей системы?
- 118) Какой декоратор используется для асинхронного выполнения задач в FastAPI с помощью Celery?
- 119) Что из перечисленного относится к принципам REST API?
- 120) Какой из следующих параметров можно использовать для установки TTL (time-to-live) в Redis?
- 121) Какой из следующих фреймворков поддерживает как синхронное, так и асинхронное программирование?
- 122) Какой метод HTTP используется для получения данных ресурса в REST API?
- 123) Какой из перечисленных вариантов является библиотекой для работы с асинхронными HTTP запросами в Python?
- 124) Какой HTTP метод используется для обновления существующего ресурса?
- 125) Что такое сессионный объект в SQLAlchemy?





- 126) Какое из следующих утверждений верно относительно Celery?
- 127) Какой тип данных Redis использует для хранения значений с ключами?
- 128) Какой из перечисленных методов запускает асинхронную задачу в FastAPI?
- 129) Какой из следующих методов в Celery используется для немедленного выполнения задачи, без добавления ее в очередь?

