

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косычук Сергей Михайлович
Должность: ректор
Дата подписания: 01.07.2026 12:59:30
Уникальный программный ключ:
e3a68f34c1c62674154f499808947c6bdfcf836

Оценочные материалы для промежуточной аттестации по дисциплине

Микросервисные архитектуры, 2 семестр

Код, направление	11.04.02. Инфокоммуникационные технологии и системы связи
подготовки	
Направленность (профиль)	Корпоративные инфокоммуникационные системы и сети
Форма обучения	Очная
Кафедра-разработчик	Радиоэлектроники и электроэнергетики
Выпускающая кафедра	Радиоэлектроники и электроэнергетики

Задание для контрольной работы

Разработка многостраничного сайта в формате микросервисной архитектуры

Цель задания — научиться работать с базой данных, внешними источниками и автодействиями.

Задача:

Создать многостраничный сайт с использованием базы данных и внешних данных.

Требования:

- Работа с базой данных – создать таблицу в БД и загрузить в неё данные из Excel-файла (любые тестовые данные: новости, товары, события и т.п.);
- Подключение внешних данных – настроить HTTP GET-запрос к внешнему источнику данных (например, курс валют или прогноз погоды);
- сохранить полученные данные в таблицу.

Настройка автодействий:

- создать автодействие, которое срабатывает при изменении данных (например, при изменении курса валюты);
- настроить отправку уведомления во внешний сервис (например, в Telegram через HTTP POST-запрос).

Структура сайта:

- главная страница;
- страница со списком данных из БД (в виде карточек или таблицы);
- типовая страница элемента (детальная страница записи).
- настроить переходы между страницами и вывод данных из базы.

Рекомендации:

Можно повторить пример с занятия или создать собственный проект;

Вопросы к зачету

1. Дайте определение понятию «микросервис» и опишите его отличие от традиционных сервисов.
2. Что такое концепция скрытия информации (по Д. Парнасу) и как она применяется в микросервисах?
3. Какова роль принципа независимого развертывания в архитектуре микросервисов?
4. В чем заключается суть моделирования микросервисов вокруг предметной области бизнеса?
5. Почему совместное использование баз данных (Shared Database) считается антипаттерном в микросервисной архитектуре?
6. Как соотносятся понятия слабой связанности (loose coupling) и сильной связности (high cohesion) в контексте микросервисов?
7. Сформулируйте закон Конвея и объясните его влияние на проектирование ИТ-систем.
8. В чем разница между горизонтальными (техническими) и вертикальными (бизнес) срезами архитектуры программного обеспечения?
9. Что такое «потокосовая команда» (Stream-aligned team) согласно методологии «Топологии команд»?
10. Сервис-ориентированная архитектура (SOA) и микросервисы: в чем их сходства, ключевые различия и причины исторических неудач ранних SOA-реализаций?
11. Дайте характеристику трем основным типам монолитных систем: однопроцессному, модульному и распределенному монолиту.
12. Каковы ключевые преимущества и недостатки модульного монолита по сравнению с микросервисной архитектурой?
13. Что понимается под термином «конфликт доставки» (delivery contention) в монолитных системах?
14. Каковы основные технологические и организационные сложности (слабые места), возникающие при переходе на микросервисы?
15. Каковы основные типы связанности (предметная, сквозная, общая, по содержимому) и как они влияют на стабильность системы?
16. Понятие Ограниченного контекста (Bounded Context) и Агрегата (Aggregate) в предметно-ориентированном проектировании (DDD) и их связь с микросервисами.
17. Метод Event Storming: цели проведения и роль в определении границ будущих сервисов.
18. В чем заключается опасность преждевременной декомпозиции монолитной системы на микросервисы?
19. Опишите стратегии декомпозиции монолита: шаблон «Душитель» (Strangler Fig), параллельное выполнение и переключаемые функции.
20. Проблемы декомпозиции данных при разделении монолита: производительность, транзакции и целостность.
21. Сравните шаблон синхронной блокирующей связи и шаблон асинхронного неблокирующего взаимодействия (плюсы, минусы, области применения).
22. Особенности и риски реализации шаблона взаимодействия микросервисов через общие данные.
23. Событийное взаимодействие (Event-driven) микросервисов: реализация, преимущества и что должно входить в структуру события.
24. Технологии межпроцессного взаимодействия: сравнительный анализ подходов RPC, REST и GraphQL.
25. Форматы сериализации данных (текстовые и двоичные): компромиссы между читаемостью и производительностью в микросервисах.

26. Каковы стратегии управления критическими (обратно несовместимыми) изменениями API в микросервисной среде?
27. В чем заключается опасность повторного использования кода через общие библиотеки (принцип DRY) в мире микросервисов?
28. Механизмы обнаружения сервисов (Service Discovery): DNS против динамических реестров.
29. Архитектурные паттерны API-шлюз (API Gateway) и Сервисная сеть (Service Mesh): назначение и различия.
30. Проблема распределенных транзакций: почему стоит избегать двухфазной фиксации (2PC) и в чем суть паттерна Saga (Saga).
31. Концепция непрерывной интеграции (CI) и непрерывной доставки (CD) применительно к микросервисам.
32. Сравнительный анализ стратегий организации репозиторий: мультирепозиторий против монорепозитория для микросервисов.
33. Варианты развертывания микросервисов: физические машины, виртуальные машины, контейнеры, PaaS и FaaS (бессерверный подход).
34. Базовые концепции Kubernetes и его роль в оркестрации контейнеризированных микросервисов.
35. Поэтапная доставка: стратегии сине-зеленого развертывания, канареечных релизов и использование переключателей функций (Feature Toggles).
36. Пирамида тестирования в микросервисах: компромиссы между модульными, сервисными и сквозными (E2E) тестами.
37. Контрактное тестирование (Consumer-Driven Contract Testing) как альтернатива хрупким сквозным тестам.
38. От мониторинга к наблюдаемости (Observability): три столпа (логи, метрики, распределенная трассировка) и роль идентификаторов корреляции.
39. Шаблоны обеспечения отказоустойчивости систем: тайм-ауты, повторные попытки (retries), переборки (bulkheads), автоматические выключатели (circuit breakers) и идиоматичность.
40. Паттерны проектирования пользовательских интерфейсов для микросервисов: Монолитный фронтенд, Микрофронтенды и Бэкенд для фронтенда (BFF).