Бауманский учебный центр «Специалист»

Программа курса Python: микросервисная архитектура

Код курса: **МИКПИТ**

Тема	Ак. часов
Модуль 1. Введение в микросервисную архитектуру • Монолит vs Микросервисы: преимущества и недостатки. • Принципы проектирования микросервисов. • Проблемы распределенных систем и способы их решения. • **Практический пример:** Обзор стека технологий для микросервисов. *Изучение и сравнение ключевых технологий.* • **Практический пример:** Обзор Docker. Основные команды. *Освоение базовых команд управления контейнерами.* • **Практический пример:** Настройка рабочего окружения: Python, Docker, VS Code (PyCharm). *Подготовка локальной среды разработки.* • Обзор возможностей фреймворка Flask. • **Практический пример:** Создание простейшего HTTP-сервиса на Flask. *Разработка минимального веб-сервиса.* • Знакомство с docker-compose для оркестрации контейнеров.	**4**
Модуль 2. Разработка RESTful микросервисов на Flask • Принципы WEB API. • Структура микросервиса: маршрутизация, бизнес-логика, данные. • **Практический пример:** Работа с базами данных в микросервисах (PostgreSQL). *Интеграция сервиса с базой данных.* • **Практический пример:** Валидация данных и обработка ошибок. *Реализация механизмов проверки данных.* • **Практический пример:** Регистрация и аутентификация пользователей. *Создание системы аутентификации.* • **Практический пример:** JWT-токены для авторизации. *Внедрение токен-базированной авторизации.* • **Практический пример:** CRUD операции для управления пользователями. *Реализация операций управления данными.* • **Практический пример:** Логирование и мониторинг здоровья сервиса. *Настройка мониторинга и логирования.*	**8**
Модуль 3. Синхронное взаимодействие между микросервисами • **Практический пример:** HTTP-коммуникация между сервисами. *Организация межсервисного взаимодействия* • **Практический пример:** Паттерн API Gateway. *Создание единой точки входа* • **Практический пример:** Обработка таймаутов и retry-логика. *Реализация механизмов устойчивости* • **Практический пример:** Circuit Breaker для устойчивости системы. *Внедрение паттерна "Предохранитель"* • **Практический пример:** Паттерн Saga. *Организация распределенных транзакций* • **Практический пример:** Интеграционные тесты. *Тестирование взаимодействия сервисов*	**6**

Модуль 4. Асинхронная коммуникация с очередями сообщений **Часть 1: RabbitMQ** • Очереди сообщений: когда и зачем использовать. • **Практический пример:** Сравнение RabbitMQ и Apache Kafka. *Анализ технологий сообщений* • **Практический пример:** Паттерны: Pub/Sub, Message Queue, Event Sourcing. *Изучение паттернов коммуникации* • **Практический пример:** Гарантии доставки сообщений. *Обеспечение надежности доставки* • **Практический пример:** Работа с RabbitMQ через pika. *Интеграция с брокером сообщений* • **Практический пример:** Реализация паттерна Pub/Sub. *Создание системы публикации-подписки* **Часть 2: Apache Kafka** • **Практический пример:** Интеграция с Kafka через kafka-python. *Работа с потоковой платформой* • **Практический пример:** Обработка потоков данных. *Обработка потоковых данных*	**10**
Модуль 5. Финальная практика • **Практический пример:** Написание интеграционных тестов с использованием pytest. *Создание комплексных тестов* • **Практический пример:** Создание Dockerfile для каждого микросервиса. *Контейнеризация сервисов* • **Практический пример:** Настройка docker-compose для всего приложения. *Оркестрация многокомпонентного приложения* • **Практический пример:** Взаимодействие между микросервисами. *Интеграция всех компонентов системы* • **Практический пример:** Реализация health checks и метрик. *Настройка мониторинга приложения*	**12**
Аудиторная нагрузка в классе с преподавателем	**40 ак.ч. + 40 ак.ч. самостоятельно**

По окончании обучения на курсе проводится итоговая аттестация. Аттестация проводится в виде теста на последнем занятии или на основании оценок практических работ, выполняемых во время обучения на курсе.

Контакты центра

Для частных лиц
+7 (495) 232-32-16

Для организаций
+7 (495) 780-48-44

E-mail
info@specialist.ru