

ПРАКТИЧЕСКАЯ ПРОГРАММА ОБУЧЕНИЯ

Микросервисная архитектура на Java

Архитектурные решения, интеграции и проверка отказов

Java 21 • Spring Boot 4 • REST • gRPC • Kafka • RabbitMQ • Docker • Testcontainers

Объём курса	Модули	Проектные этапы
24 ак. ч.	12 × 2 ак. ч.	12

Проект после каждого модуля

OrderPlatform: учебная система заказов с готовыми каркасами сервисов. Участник последовательно реализует 12 небольших проектных этапов: границы, устойчивость, безопасность, контейнеры, REST/gRPC, брокеры и тесты.

Обучение лично у основателя ТЕРАКОР

Сергей Валерьевич Герасименко

Практикующий full-stack разработчик, технический директор и основатель ТЕРАКОР и ТЕРАКОР Digital, ведущий преподаватель и практикующий эксперт.

Лицензированный учебный центр • Живые онлайн-занятия

Редакция программы: 28 сентября 2026 года

01 / КОМУ ПОДОЙДЁТ КУРС

Вход и результаты обучения

Для кого

Разработчикам с опытом Spring Boot, которым нужна системная практика распределённых приложений.

Что нужно знать до старта

Уверенный Spring Boot REST, JPA, Spring Security, Docker и основы брокеров. До старта нужно запустить предоставленный репозиторий; курс не заменяет базовое изучение этих технологий.

После обучения вы сможете

- Определять границы сервисов и фиксировать архитектурные компромиссы.
- Выбирать синхронный или событийный обмен.
- Настраивать защиту и ограничивать каскадные сбои.
- Проверять повторную доставку и компенсацию операций.
- Диагностировать и тестировать распределённый бизнес-сценарий.

Как организована практика

Каждый модуль завершается проектным этапом с отдельным результатом и критериями приёмки. Этапы развивают единое приложение; при отдельном входе используется стартовый репозиторий.

Все обязательные этапы выполняются в пределах 24 академических часов на подготовленном стенде. Модуль: 1 ак. ч. разбор решения + 1 ак. ч. проектная практика и проверка. Дополнительная самостоятельная нагрузка не назначается.

02 / УЧЕБНЫЙ ПЛАН

Распределение часов

Модуль	С преподавателем	Самостоятельно
1. Границы сервисов и декомпозиция	2	Не назначено
2. Паттерны устойчивости	2	Не назначено
3. Идентичность и права доступа	2	Не назначено
4. Контейнерный стенд	2	Не назначено
5. Синхронный и асинхронный обмен	2	Не назначено
6. EDA, идемпотентность и компенсации	2	Не назначено
7. Контракты REST API	2	Не назначено
8. HTTP-семантика и валидация	2	Не назначено
9. gRPC и Protobuf	2	Не назначено
10. RabbitMQ и обработка задач	2	Не назначено
11. Kafka в микросервисной системе	2	Не назначено
12. Тестирование и итоговая защита	2	Не назначено
ИТОГО	24	Не назначено

Все значения приведены в академических часах. Один академический час - 45 минут. Проекты и защита входят в указанный объём; дополнительные часы сверху не добавляются.

03 / МОДУЛЬ 01

Границы сервисов и декомпозиция

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Монолит, модульный монолит и микросервисы: критерии выбора.
- Bounded context и ответственность за данные.
- Автономность и зависимости; запрет общей записи в чужую БД.
- Стоимость сетевых вызовов и эксплуатации.

Проект после модуля 1

Domain Map

Разделить заказы, каталог и уведомления; зафиксировать владение данными и одно ADR.

Что сдаёт участник

Карта контекстов и архитектурное решение.

Критерии приёмки

- У каждого сервиса есть ответственность и данные.
- Обоснована хотя бы одна граница и альтернатива.

03 / МОДУЛЬ 02

Паттерны устойчивости

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Gateway и discovery: назначение и ограничения.
- Timeout, retry, Circuit Breaker и bulkhead.
- Saga, Outbox и CQRS: обзор по бизнес-задачам.
- Resilience4j на подготовленной интеграции.

Проект после модуля 2

Failure Lab

Настроить timeout и circuit breaker для обращения к каталогу, воспроизвести задержку.

Что сдаёт участник

Конфигурация и сценарий отказа.

Критерии приёмки

- Время ожидания ограничено.
- Fallback не сообщает об успешном заказе при отказе.

03 / МОДУЛЬ 03

Идентичность и права доступа

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- OAuth2 и OIDC: роли участников.
- Resource Server, JWT issuer/audience и scopes.
- Права на ресурс; 401 и 403.
- Service-to-service доступ; mTLS - обзор.

Проект после модуля 3

Access Lab

Подключить готовый IdP, закрыть чтение чужого заказа и служебный endpoint.

Что сдаёт участник

Матрица доступа и security-тесты.

Критерии приёмки

- Невалидный токен отклоняется.
- Пользователь не получает чужие данные.

03 / МОДУЛЬ 04

Контейнерный стенд

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Dockerfile и многоэтапная сборка.
- Docker Compose: сеть, переменные и volumes.
- Health, readiness и порядок готовности зависимостей.
- Лимиты ресурсов и запуск без root.

Проект после модуля 4

Compose Lab

Упаковать один сервис и подключить к готовому стенду с БД.

Что сдаёт участник

Dockerfile и дополнение Compose.

Критерии приёмки

- Сервис запускается по инструкции.
- Пароли не встроены в образ.

03 / МОДУЛЬ 05

Синхронный и асинхронный обмен

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- REST client и HTTP Service Clients.
- Latency budget и цепочки вызовов.
- Команда, событие и фоновая задача.
- Correlation ID и распределённая трассировка.

Проект после модуля 5

Integration Map

Реализовать один HTTP-вызов и составить схему альтернативного событийного сценария.

Что сдаёт участник

Рабочий вызов и решение о выборе транспорта.

Критерии приёмки

- Обоснован выбор синхронности.
- Идентификатор запроса проходит между сервисами.

03 / МОДУЛЬ 06

EDA, идемпотентность и компенсации

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Choreography и orchestration.
- At-least-once и повторное событие.
- Outbox и идемпотентный обработчик.
- Saga и состояния; Event Sourcing - обзор.

Проект после модуля 6

Duplicate Lab

Добавить дедупликацию на готовом обработчике и показать компенсацию отказа резервирования.

Что сдаёт участник

Обработчик и диаграмма состояний.

Критерии приёмки

- Повтор не повторяет бизнес-эффект.
- Компенсация явно отражена в состоянии заказа.

03 / МОДУЛЬ 07

Контракты REST API

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Ресурсы и границы внешнего API.
- OpenAPI и совместимость изменений.
- Problem Details: RFC 9457.
- Версионирование; HATEOAS - обзор.

Проект после модуля 7

Contract Lab

Описать API заказа и унифицировать ответы об ошибках.

Что сдаёт участник

OpenAPI и примеры ответов.

Критерии приёмки

- Контракт содержит успешный и ошибочные ответы.
- Изменение не ломает согласованный клиентский сценарий.

03 / МОДУЛЬ 08

HTTP-семантика и валидация

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- POST/PUT/PATCH/DELETE и идемпотентность.
- DTO, Jakarta Validation и обработка ошибок.
- Пагинация и фильтры.
- Idempotency key для создания заказа.

Проект после модуля 8

Resource Lab

Доработать создание заказа и повтор запроса с одинаковым ключом.

Что сдаёт участник

Endpoint и проверки повторного запроса.

Критерии приёмки

- Одинаковый запрос не создаёт второй заказ.
- Неверный DTO даёт предсказуемый 400-ответ.

03 / МОДУЛЬ 09

gRPC и Protobuf

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Контракт .proto и генерация Java-кода.
- Unary call и deadline.
- Статусы ошибок и совместимость полей.
- Streaming и gateway - обзор.

Проект после модуля 9

gRPC Lab

По готовому каркасу реализовать unary-вызов проверки наличия товара.

Что сдаёт участник

Proto-контракт, клиент и серверный метод.

Критерии приёмки

- Deadline ограничивает ожидание.
- Отсутствующий товар и сетевой сбой различаются.

03 / МОДУЛЬ 10

RabbitMQ и обработка задач

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Exchange, queue и binding.
- Spring AMQP и RabbitTemplate.
- Acknowledgment, prefetch и dead-letter queue.
- Повторная доставка и ограниченный retry.

Проект после модуля 10

Queue Lab

Добавить очередь уведомлений, воспроизвести неуспешную обработку и dead-letter маршрут.

Что сдаёт участник

Обработчик и конфигурация очередей.

Критерии приёмки

- Подтверждение следует за обработкой.
- Плохое сообщение не повторяется бесконечно.

03 / МОДУЛЬ 11

Kafka в микросервисной системе

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Topic, key, partition и consumer group.
- KafkaTemplate и KafkaListener.
- Offsets, дубликаты и порядок.
- Границы exactly-once; Connect и Schema Registry - обзор.

Проект после модуля 11

Event Bus Lab

Опубликовать событие заказа и обработать его в проекции, проверить повтор.

Что сдаёт участник

Событийный маршрут и тестовые сообщения.

Критерии приёмки

- Ключ обеспечивает нужное распределение.
- Повтор не искажает состояние проекции.

03 / МОДУЛЬ 12

Тестирование и итоговая защита

2 ак. ч. с преподавателем • Практика включена

Содержание модуля

- Unit, integration, contract и e2e: выбор уровня.
- Testcontainers и WireMock на готовом стенде.
- Проверка отказов и диагностические логи.
- Runbook и демонстрация архитектурного решения.

Проект после модуля 12

OrderPlatform Release

Запустить готовый набор тестов, добавить проверку одного сбоя и показать полный путь заказа.

Что сдаёт участник

Итоговый репозиторий, тест и runbook.

Критерии приёмки

- Проверены штатный путь и выбранный отказ.
- Автор объясняет границы гарантий и следующий шаг к эксплуатации.

04 / ИТОГОВЫЙ РЕЗУЛЬТАТ

Проект для портфолио

OrderPlatform: учебная система заказов с готовыми каркасами сервисов. Участник последовательно реализует 12 небольших проектных этапов: границы, устойчивость, безопасность, контейнеры, REST/gRPC, брокеры и тесты.

Комплект сдачи

- Репозиторий с исходным кодом и понятной структурой.
- README: требования к окружению, команды сборки и запуска, пример использования.
- Конфигурация и данные для воспроизведения проверок без секретов.
- Результаты тестов или проверочные сценарии для каждого проектного этапа.
- Краткое описание архитектуры, ограничений и принятых решений.

Порядок защиты

Участник запускает приложение, показывает основной сценарий и обработку ошибки, объясняет ключевые решения. Проверка опирается на критерии каждого модуля. Невыполненный критерий фиксируется как конкретная доработка.

Условие завершения

Все обязательные проектные этапы приняты; итоговый сценарий воспроизводится по README. Итоговый проект объединяет ранее выполненные этапы и не является дополнительной работой сверх программы.

Результат обучения

Работающий учебный проект и подтверждённые практикой навыки. Прохождение отдельного курса не означает автоматического присвоения грейда или гарантии трудоустройства.