УТВЕРЖДАЮ

Генеральный директор

ООО «ЗИЛ ДИДЖИТАЛ»

_____ /Т. А. Ровнягина/

«22» ноября 2026 г.

Приказ ООО «ЗИЛ ДИДЖИТАЛ» от 22.11.2026 № 1-ДПО

Редакция 1.0 от 22.11.2026

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

«Проектирование и разработка высоконагруженных вычислительных систем»

Обозначение: ПК-01

Объём: 24 академических часа

ООО «ЗИЛ ДИДЖИТАЛ»

Москва, 2026

1. Общие сведения о программе

Параметр	Положение
Полное наименование	Дополнительная профессиональная программа повышения квалификации «Проектирование и разработка высоконагруженных вычислительных систем»
Вид программы	Дополнительное профессиональное образование
Подвид программы	Повышение квалификации
Форма обучения	Очная с применением электронного обучения и дистанционных образовательных технологий (ЭО/ДОТ)
Способ реализации	Исключительно с применением ЭО/ДОТ. Компоненты с обязательным физическим присутствием слушателей и посещение помещений ООО «ЗИЛ ДИДЖИТАЛ» не предусмотрены
Общая трудоёмкость	24 академических часа
Структура трудоёмкости	23 ак. ч. онлайн-занятий с преподавателем и 1 ак. ч. итогового тестирования
Продолжительность	6 учебных дней по 4 ак. ч.; конкретные даты устанавливаются расписанием учебной группы
Формат занятий	Онлайн-занятия с преподавателем в реальном времени: лекции с демонстрацией практических примеров на сквозном приложении
Самостоятельная работа	По выбору: повторение показанных преподавателем реализаций на сходных бизнес-постановках вне 24 ак. ч., без балльной оценки
Язык обучения	Русский
Итоговая аттестация	Тестирование по материалам онлайн-занятий и продемонстрированных практических примеров, 1 ак. ч., 35 вопросов, 45 минут; успешный результат — не менее 60 % правильных ответов (21 балл из 35)
Документ по итогам	При успешном прохождении итоговой аттестации — удостоверение о повышении квалификации установленного образца, оформляемое в установленном порядке
Разработчик и преподаватель	Ровнягин Михаил Михайлович, кандидат технических наук

Базовый технологический стек: Java 25 (LTS), Spring Boot, PostgreSQL, Docker, Apache Kafka, Kubernetes (k3s), Redis, Prometheus + Grafana, k6.

1.1. Сведения об организации, реализующей программу

Параметр	Значение
Полное наименование	Общество с ограниченной ответственностью «ЗИЛ ДИДЖИТАЛ»
Сокращённое наименование	ООО «ЗИЛ ДИДЖИТАЛ» (ZIL.digital)
ИНН	9725198820
ОГРН	1257700566848
Сайт	https://zil.digital/

Область деятельности организации в части реализуемых программ — высокопроизводительные вычисления, Big Data, машинное обучение и архитектуры распределённых систем.

1.2. Модель и формат обучения

Программа реализуется исключительно с применением ЭО/ДОТ. Занятия с преподавателем проходят онлайн в реальном времени и сочетают изложение лекционного материала с практическими демонстрациями: преподаватель показывает реализацию на сквозном приложении, разбирает конфигурации, исходный код, сценарии нагрузочного тестирования и результаты наблюдения за работой распределённой системы.

Практические демонстрации LAB0–LAB18 входят в 23 ак. ч. онлайн-занятий и не образуют отдельные лабораторные занятия или дополнительную трудоёмкость. Отдельные оцениваемые практические занятия с посещением помещений Общества не предусматриваются. Участие из помещений Общества не требуется ни для онлайн-занятий, ни для самостоятельной работы, ни для итогового тестирования.

На занятиях преподаватель последовательно развивает сквозной пример приложения от monolith-CRUD до распределённой highload-системы. Параллельно слушателю предоставляется возможность самостоятельно повторить показанную технологическую реализацию на сходной бизнес-постановке: онлайн-магазине, системе записи к врачу, кинотеатре, авиакассе, доставке еды или другом варианте из учебного комплекта. Такое повторение закрепляет именно продемонстрированный преподавателем порядок реализации, а не заменяет его самостоятельным изучением материала без показа.

Самостоятельные тренировки:

- выполняются по выбору слушателя вне общей трудоёмкости 24 ак. ч.;
- носят рекомендованный и тренировочный характер;
- не подлежат балльной оценке и обязательной защите;
- не являются условием допуска к итоговой аттестации;
- не являются условием выдачи документа о квалификации.

Основанием для выдачи документа о квалификации является успешное прохождение итогового тестирования при соблюдении установленного порядка его оформления.

2. Цель и задачи программы

2.1. Цель

Сформировать у слушателей компетенции по проектированию, разработке, развёртыванию, нагрузочному тестированию и эксплуатации высоконагруженных (highload) распределённых backend-систем на стеке Java/Spring с применением современных практик контейнеризации, оркестрации, потоковой обработки и наблюдаемости.

2.2. Задачи

В рамках программы предусматривается решение следующих задач:

1. Освоить разработку backend-приложений на Java и экосистеме Spring (Core, MVC/REST, Boot, Data JPA).
2. Научиться работать с реляционными БД и стратегиями доступа к данным под нагрузкой.
3. Систематизировать базовые навыки Git, семантического версионирования и сборки проектов Maven/Gradle как вспомогательный инструментарий разработки.
4. Освоить контейнеризацию приложений (Docker) и развёртывание стендов.
5. Освоить методику нагрузочного тестирования (k6) и анализ зависимости «отклик ↔ ресурсы/нагрузка».
6. Освоить горизонтальное масштабирование, выделение микросервисов и межсервисное взаимодействие.
7. Освоить потоковую обработку на Apache Kafka и асинхронную архитектуру.
8. Освоить оркестрацию контейнеров в Kubernetes (k3s): манифесты, ресурсы, liveness/readiness probes; обзорно рассмотреть HPA.
9. Освоить паттерны отказоустойчивости (retry, circuit breaker) и кеширование (in-memory, Redis).
10. Получить обзорное представление о наблюдаемости систем: метрики, логи, трейсы, SLA/SLO/SLI и дашборды без формирования обязательной компетенции по настройке всего стека наблюдаемости.

3. Планируемые результаты обучения

В результате освоения программы слушатель должен приобрести следующие обязательные знания, умения и навыки. Обзорные темы, не образующие обязательные компетенции, отдельно перечислены в разделе 3.4.

3.1. Знать

- принципы highload-архитектур (data-intensive / compute-intensive), метрики производительности (время отклика, пропускная способность), виды масштабирования;
- принципы ООП и SOLID, особенности языка Java и рекомендации по чистому коду;
- архитектуру и философию Spring (IoC/DI), модель Spring Boot, Spring Data JPA / Hibernate;
- модель работы HTTP/REST, сервлетов, сервера приложений;
- модели данных и жизненный цикл сущностей в ORM;
- архитектуру Docker, Kafka и Kubernetes, назначение и базовые принципы применения Redis;
- паттерны масштабирования и отказоустойчивости, стратегии кеширования;
- методику нагрузочного тестирования, измерения и интерпретации показателей производительности.

3.2. Уметь

- проектировать и реализовывать REST-сервисы на Spring Boot для заданной предметной области;
- работать с PostgreSQL через Spring Data JPA, проектировать схему и миграции;
- контейнеризировать приложения и разворачивать стенды в docker-compose и k3s;
- проводить нагрузочное тестирование в k6 и строить графики зависимости показателей;
- выделять микросервисы и организовывать межсервисное взаимодействие;
- настраивать продюсеров/консьюмеров Kafka и параллелизм обработки;
- настраивать ресурсы и liveness/readiness probes в Kubernetes;
- внедрять resilience4j (retry, circuit breaker) и кеширование на Redis.

3.3. Владеть

- основами работы с Docker, k6 и kubectl применительно к содержанию практических примеров;
- навыками эксплуатации распределённого стенда: виртуальные машины и SSH-туннелирование;
- методикой измерения и интерпретации показателей производительности.

3.4. Обзорные и выравнивающие темы вне обязательных компетенций

К обзорным и выравнивающим темам относятся:

- JSP/JSF, WireMock, Git, семантическое версионирование, устройство Maven/Gradle и репозитории артефактов; Git, Maven/Gradle и IntelliJ IDEA могут применяться как рабочие инструменты практических демонстраций без формирования отдельной обязательной компетенции;
- Apache Ignite, Docker Swarm, автоскейлинг HPA, Prometheus и Grafana, построение дашбордов, SLA/SLO/SLI, Golden Signals, RED, USE и реестры образов;
- Key User Journeys, Error Budget и burn rate, статические и динамические базлайны, Z-score, скользящие средние, обнаружение точек изменения, multi-window alerting, architectural fitness functions, DORA, SPACE, KPI/OKR, LTV/CAC, когортный и прогнозный анализ;
- OpenMetrics, StatsD, InfluxDB, VictoriaMetrics, Telegraf, Actuator/Micrometer, JFR/JDK Mission Control, VisualVM/JProfiler, OpenTelemetry, Jaeger, Zipkin, JMeter/Gatling и другие инструменты, перечисляемые в лекциях для сравнения или обзорного знакомства.

Перечисленные темы и инструменты не образуют самостоятельные обязательные результаты обучения и не требуют прямых вопросов в итоговом тесте. Их изучение входит в соответствующие часы онлайн-занятий, но отсутствие по ним отдельных итоговых заданий не является дефектом оценочного комплекта. Включение инструмента в обзорный список не означает его обязательное развёртывание на учебном стенде.

Проверка обязательных результатов, включая результаты, связанные с практическими демонстрациями, обеспечивается итоговым тестированием на основании матрицы покрытия комплекта оценочных материалов ОМ-ПК-01, утверждённого приказом ООО «ЗИЛ ДИДЖИТАЛ» от 22.11.2026 № 1-ДПО. Обзорные и выравнивающие темы раздела 3.4 обязательному покрытию итоговыми вопросами не подлежат.

4. Категория слушателей и входные требования

Категория слушателей: студенты старших курсов и специалисты в области разработки программного обеспечения, желающие повысить квалификацию в области highload-систем.

Рекомендуемые входные требования:

- базовые навыки программирования, желательно знакомство с Java или другим объектно-ориентированным языком;
- базовое понимание реляционных БД и SQL;
- базовые навыки работы в командной строке Linux/macOS;
- владение системой контроля версий Git: клонирование, коммиты, ветки;
- знакомство с системами сборки проектов Maven / Gradle.

Условия приёма, подтверждение уровня образования и документальное оформление образовательных отношений определяются правилами ООО «ЗИЛ ДИДЖИТАЛ» и договором об образовании.

5. Учебно-тематический план

План отражает вводное занятие и фактическую последовательность 16 лекций. Теоретический материал и соответствующие ему практические показы преподавателя проводятся параллельно внутри часов каждой темы. Самостоятельное повторение показанных реализаций на сходных бизнес-постановках рекомендуется между занятиями, выполняется вне 24 ак. ч. и в сумму часов не входит.

№	Лекция / вид аттестации	Онлайн-занятия с преподавателем, ак. ч.	Итоговое тестирование, ак. ч.	Самостоятельное повторение вне 24 ак. ч.
В	Введение в highload: метрики, масштабирование, «highload-конструктор»	1	0	Не предусмотрено
1	Основы языка Java. SOLID и чистый код	1	0	Не предусмотрено
2	Spring Core: IoC/DI, конфигурация, аннотации	1	0	Не предусмотрено
3	Путь к Spring Boot: HTTP, сетевые модели, сервлеты	1	0	Не предусмотрено
4	Разработка HTTP API на Spring MVC	1	0	Предусмотрено
5	Spring Boot и тестирование приложения	1	0	Предусмотрено
6	Работа с базами данных: JDBC, Hibernate, Spring Data JPA	2	0	Предусмотрено
7	Testcontainers, WireMock и RestTemplate	0,5	0	Предусмотрено
8	Системы сборки и версионирования кода: Git, SemVer, Maven, Gradle	0,5	0	Предусмотрено

№	Лекция / вид аттестации	Онлайн-занятия с преподавателем, ак. ч.	Итоговое тестирование, ак. ч.	Самостоятельное повторение вне 24 ак. ч.
9	Apache Ignite: in-memory data grid	1	0	Не предусмотрено
10	Брокер сообщений Apache Kafka	2	0	Предусмотрено
11	Технология контейнеризации Docker	2	0	Предусмотрено
12	Паттерны взаимодействия и масштабирования highload-приложений	2	0	Предусмотрено
13	Kubernetes / k3s	2	0	Предусмотрено
14	Метрики высоконагруженных приложений и нагрузочное тестирование	2	0	Предусмотрено
15	Сбор и анализ метрик	1	0	Предусмотрено
16	Кеширование: стратегии и Redis	2	0	Предусмотрено
ИА	Итоговая аттестация — тестирование	0	1	Не предусмотрено
Итого	**Вводное занятие, 16 лекций и итоговое тестирование — 24 ак. ч.**	**23**	**1**	**По выбору, без балльной оценки**

Контрольная сумма по занятиям: 1 + (5 × 1) + 2 + 0,5 + 0,5 + 1 + 2 + (4 × 2) + 1 + 2 = **23 ак. ч.**; с итоговым тестированием: 23 + 1 = **24 ак. ч.** Коды LAB0–LAB18 в разделе 7 обозначают практические показы преподавателя и связанные с ними возможности самостоятельного повторения, а не отдельные лабораторные занятия.

6. Содержание программы по темам

Содержание синхронизировано с вводной презентацией и лекциями `HL_lec1`—`HL_lec16`. Перечисленные практические показы проводятся преподавателем параллельно с соответствующей теорией; их детализация приведена в разделе 7.

Вводное занятие. Введение в highload — 1 ак. ч.

Понятие высоконагруженных приложений: data-intensive и compute-intensive. Ключевые свойства: производительность, масштабируемость, надёжность, удобство эксплуатации. Метрики: время отклика и пропускная способность. Вертикальное и горизонтальное масштабирование. Типовая

архитектура highload-системы («конструктор»): балансировка, серверы приложений, кеширование, очереди, потоковая и пакетная обработка, системы хранения.

Лекция 1. Основы языка Java. SOLID и чистый код — 1 ак. ч.

Особенности Java и сравнение с C++: управление памятью, JVM, generics, аннотации, встроенная многопоточность. Принципы SOLID на примерах разделения ответственности, расширения без модификации, подстановки подтипов, разделения интерфейсов и инверсии зависимостей. Осмысленные имена, декомпозиция методов, устранение дублирования и корректное выделение констант.

Лекция 2. Spring Core — 1 ак. ч.

Философия Spring, IoC/DI и autowiring. XML- и Java-конфигурация бинов, `@Bean`, `@Configuration`, стереотипные аннотации. `@Value`, значения по умолчанию, коллекции и SpEL. `@ConditionalOnProperty`. Планирование через `@Scheduled` и `@EnableScheduling`, fixedDelay/fixedRate/cron. Асинхронность через `@Async`, `@EnableAsync`, ThreadPoolTaskExecutor и ограничения self-invocation.

Лекция 3. Путь к Spring Boot: HTTP, сетевые модели, сервлеты — 1 ак. ч.

Модели OSI и TCP/IP, структура HTTP-запроса и ответа, URI/URL/URN, методы, заголовки и коды ответов. Примеры HTTP-серверов на Java-сокетах, JDK HttpServer и фреймворке. Блокирующие, многопоточные, неблокирующие и асинхронные модели исполнения. Application server и servlet container, жизненный цикл сервлета, HttpSession. JSP/JSF как архитектурный контекст backend UI.

Лекция 4. Разработка HTTP API на Spring MVC — 1 ак. ч.

MVC, REST и CRUD-маршруты. DispatcherServlet, HandlerMapping, Controller и ViewResolver. Трёхзвенная архитектура Controller–Service–Repository. `@Controller`, `@RestController`, `@RequestMapping` и специализированные mapping-аннотации, `@RequestParam`, `@PathVariable`, `@RequestBody`, `@ResponseBody`. Сериализация Jackson и её аннотации. Swagger, api-doc и Swagger UI. Практический показ Spring MVC и перехода к Spring Boot.

Лекция 5. Spring Boot и тестирование приложения — 1 ак. ч.

Назначение Spring Boot, стартеры, управление зависимостями, автоконфигурация и встроенные серверы. `@SpringBootApplication`. Виды автотестов и запуск Spring-контекста через `@SpringBootTest`; JUnit 5, `@MockBean`, TestRestTemplate, MockMvc, `@DirtiesContext`. Практические показы LAB0–LAB1: подготовка окружения и REST-сервис с репозиториями на Java-коллекциях для сквозной бизнес-постановки.

Лекция 6. Работа с базами данных: JDBC, Hibernate, Spring Data JPA — 2 ак. ч.

Объектная и реляционная модели, отображение классов и связей на таблицы. JDBC, PreparedStatement и ResultSet; JdbcTemplate, MyBatis и ORM. JPA/Hibernate: разметка сущностей и связей, Persistence Context, EntityManager, жизненный цикл Entity, fetch-стратегии и риски избыточной загрузки. JPQL, Spring Data JPA, CrudRepository и derived query methods. Proxy/AOP и `@Transactional`. Практический показ LAB2: PostgreSQL, перевод репозиториев на Spring Data JPA и наполнение БД.

Лекция 7. Testcontainers, WireMock и RestTemplate — 0,5 ак. ч.

Синхронный HTTP-вызов через RestTemplate, проверка статуса и разбор JSON; WebClient как рекомендуемая альтернатива. Практические показы использования WireMock для имитации

внешнего HTTP-сервиса и Testcontainers для интеграционных тестов с реальными зависимостями в контейнерах.

Лекция 8. Системы сборки и версионирования кода — 0,5 ак. ч.

Git: репозиторий, рабочий каталог и индекс, состояния файлов, init/clone/add/commit/push/pull, назначение ветвления и удалённых репозиториев. Семантическое версионирование и JAR-артефакты. Maven: POM, координаты и фазы жизненного цикла. Gradle: tasks, DAG, multi-project build, initialization/configuration/execution, `settings.gradle`, `doFirst`/`doLast`, типизированные задачи и `@TaskAction`. Инструменты применяются в практических показах LAB1–LAB18.

Лекция 9. Apache Ignite — 1 ак. ч.

In-memory computing и data grid. Durable memory и Native Persistence, off-heap, страницы и сегменты памяти, B+ индексы, WAL и checkpointing. Partitioned/replicated режимы и синхронизация резервных копий. Клиентские и серверные узлы, discovery и варианты кластерного развёртывания. Распределённые SQL-запросы и join, JDBC, DDL/DML, индексы и SqlFieldsQuery. Обзор Compute Grid, ML Grid и интеграции со Spark.

Лекция 10. Брокер сообщений Apache Kafka — 2 ак. ч.

Назначение и архитектура Kafka: producer, consumer, broker, topic, partition. Key-value сообщения, offset и позиция читателя. Log compaction, leader/follower, репликация и ISR. Partitioner, балансировка потребителей, at most once / at least once / exactly once, batching и подтверждения. Практические показы LAB11–LAB13: Kafka в KRaft и Docker Swarm, топики и Kafka UI через SSH-туннель, Spring Kafka `@KafkaListener`, партиции и concurrency, сообщения `{entity, operation, payload}`, batch listener и нагрузочный профиль k6/xk6-kafka.

Лекция 11. Технология контейнеризации Docker — 2 ак. ч.

XaaS, IaaS/PaaS/SaaS и Cloud Native. Контейнеры и ВМ, namespaces/cgroups, Union File System. Docker client/daemon, образы, контейнеры и registry. Многоступенчатый Dockerfile, build context, build/run/pull/push, exec/cp, жизненный цикл контейнера. Docker Compose и healthcheck. Сети bridge/host/overlay/macvlan/none, подключение сетей и проброс портов. Docker Swarm: manager/worker, init/join и ingress routing mesh. Практические показы LAB3 и инфраструктурная часть LAB6: контейнеризация приложения, DDL/DML-миграции, Compose-стенд, ВМ, SSH-туннелирование, CPU/MEM и ENV-параметры; нагрузочные замеры выполняются после введения методики k6 в лекции 14.

Лекция 12. Паттерны взаимодействия highload-приложений — 2 ак. ч.

Сквозное monolith-CRUD-приложение пользователей и статистики. Настройка потоков Tomcat и вертикальное масштабирование. Вынос БД на отдельный узел и сетевая задержка. Stateless-реплики, функциональное выделение микросервисов и объединение данных на стороне сервиса. Асинхронное отделение потока статистики через Kafka и масштабирование consumer group. Retry через ServiceMesh/sidecar; circuit breaker Closed/Open/Half Open и resilience4j. Практические показы LAB7–LAB9; их нагрузочные замеры выполняются после введения методики k6 в лекции 14, а показ отказоустойчивости LAB16 — после необходимых основ Kubernetes.

Лекция 13. Kubernetes / k3s — 2 ак. ч.

Кластер, master/worker-узлы и компоненты управления; Minikube и k3s. Pod, жизненный цикл, labels/selectors, ReplicaSet, Deployment и rolling update. Service, NodePort, LoadBalancer и Ingress. ConfigMap/Secret, env/аргументы и файлы конфигурации. Тома, PV/PVC. RBAC, RoleBinding/ClusterRoleBinding. `securityContext`, requests/limits, QoS и LimitRange. Команды

kubectl: apply/get/logs/describe/exec/port-forward/cp/explain. Практические показы LAB14–LAB15: манифесты, namespace, ConfigMap/Secret, ресурсы, ResourceQuota, HPA, нагрузочные профили и liveness/readiness probes.

Лекция 14. Метрики высоконагруженных приложений — 2 ак. ч.

Мониторинг и observability, MTTD/MTTR, SLA/SLO/SLI, Key User Journeys и Error Budget. Метрики, логи, трейсы, алерты и дашборды; Golden Signals, RED и USE. Метрики балансировщиков, приложений, БД, очередей, кешей и CDN. Capacity planning, HPA/VPA, DORA и обзор бизнес-/продуктовых метрик. Статические и динамические пороги, Z-score, базлайны, change-point detection, percentiles и multi-window alerting. Виды нагрузочных тестов: load/stress/spike/soak/scalability/volume; baseline, ramp-up, плато, корреляция с ресурсами и профилирование. Actuator/Micrometer, PromQL, Performance Gates. Практические показы LAB4–LAB5, LAB9 и нагрузочные части последующих LAB: k6, генерация данных, измерение отклика и интерпретация графиков.

Лекция 15. Сбор и анализ метрик — 1 ак. ч.

Временные ряды, timestamp/value/labels, Counter/Gauge/Histogram/Summary и агрегируемость квантилей. OpenMetrics и унификация форматов. Push/StatsD и pull/Prometheus, клиентская и серверная агрегация, сэмплирование и sliding window. TSDB: InfluxDB и VictoriaMetrics, retention, HA и backup/restore. Принципы визуализации и компоновки дашбордов. Разбор аварий Facebook/Meta и AWS US-EAST-1, оптимизация YouTube CDN. Практический показ LAB18: Prometheus–Loki–Grafana–Promtail в k3s и персональный дашборд по Pod, ResourceQuota и аллокации ресурсов.

Лекция 16. Кеширование — 2 ак. ч.

Задержки CPU, памяти, дисков и сети как обоснование кеширования. In-memory, distributed, client-side и server-side cache. Cache-Aside (lazy loading), Read-Through, Write-Through и Write-Behind: пути чтения/записи, преимущества, недостатки и применимость. Stale data, eventual consistency, cold start/warming, гонки заполнения, batching и риски потери данных. Метрики hit/miss/eviction/expiration и memoization. Redis как распределённый кеш. Практические показы LAB10 и LAB17: HashMap-кеш, статистика его наполнения, нагрузочное сравнение и замена на Redis с отдельным Deployment и Service NodePort.

7. Практические показы и самостоятельное повторение

На онлайн-занятиях преподаватель параллельно с теоретическим материалом выполняет практические показы LAB0–LAB18 на сквозном приложении, последовательно развивая его от monolith-CRUD до распределённой highload-системы. Показы входят в часы соответствующих лекций, не выделяются в отдельные лабораторные занятия и не требуют посещения помещений Общества.

Одновременно с показом слушатель получает возможность самостоятельно повторить ту же технологическую реализацию на сходной по структуре, но иной по предметной области бизнес-постановке. Учебный комплект содержит 18 вариантов: онлайн-магазин, запись к врачу, кинотеатр, авиакасса, доставка еды и другие. Самостоятельное повторение выполняется по выбору вне 24 ак. ч., не оценивается баллами, не подлежит обязательной защите и не влияет на допуск к итоговому тестированию или выдачу документа.

Перечень практических показов преподавателя и связанных с ними самостоятельных тренировок:

Код	Тема показа	Содержание
LAB0	Подготовка окружения	JDK 25 (Temurin), IntelliJ IDEA Community Edition, Docker, гипервизор виртуальных машин
LAB1	HashMap-репозитории	REST-сервис для бизнес-постановки, репозитории на статических Java-коллекциях, сохранение изменений в Git
LAB2	Spring Data JPA	PostgreSQL, перевод репозиториев на работу с таблицами БД, наполнение данными
LAB3	Docker	Dockerfile, инициализация схемы и данных (DDL/DML), стенд приложения и БД в Docker Compose
LAB4	Нагрузочное тестирование k6	Профиль ramping-vus, настраиваемое соотношение POST/GET 50/50, график «средний отклик ↔ VUS» по 4–5 точкам
LAB5	Генерация данных	Python-скрипт с requests и Faker, параметры `--count` / `--endpoint` / `--clear`, эндпоинт `/clear`, отделение генерации данных от миграций схемы
LAB6	Виртуальные машины и нагрузка	ВМ, SSH-туннелирование, внешний реестр образов, CPU/MEM, ENV-параметры приложения и БД, графики «отклик ↔ CPU» для профилей 5/95, 50/50 и 95/5
LAB7	Вынос БД на отдельный узел	Отдельный узел PostgreSQL и pgAdmin в Compose, `max_connections`, подключение приложения по ENV
LAB8	Additional service	Выделение микросервиса статистики, отдельный образ, вызов через RestTemplate/WebClient/Feign, join на стороне Java, сравнение при CPU 0,5 / 1,0
LAB9	ObservabilityService	Scheduled-bean с интервалами 10 с / 30 с / 1 мин, тайминги контроллеров, БД, межсервисных вызовов и расчёта статистики, нагрузочные графики
LAB10	Кеширование в сервисе	Кеш на HashMap в Additional Service, периодический вывод размера кеша, графики для CPU 0,5 / 1,0
LAB11	Kafka: знакомство	Kafka в KRaft и Docker Swarm, создание топика, Kafka UI через SSH-туннель, запись сообщений

Код	Тема показа	Содержание
LAB12	Kafka-консьюмер	`@KafkaListener`, concurrency и соответствующее число партиций, обработка сообщений `{entity, operation, payload}`, тестовый producer
LAB13	Kafka: нагрузка	Запись через xk6-kafka или прокси, Batch Listener, графики для concurrency 1 / 2 и двух партиций
LAB14	Kubernetes (k3s)	Подключение узла, манифесты Deployment/Service/ConfigMap/Secret, namespace, ресурсы Guaranteed, kubectl apply/port-forward и NodePort
LAB15	Kubernetes: нагрузка и HPA	ResourceQuota, HPA до 3 реплик, профили 95/5–50/50–5/95, healthcheck и liveness/readiness probes
LAB16	Отказоустойчивость	Crash-эндпоинт и killer-метод, resilience4j retry / circuit breaker, сравнение параметров и результатов k6
LAB17	Redis-кеш	Замена HashMap на Redis, отдельный Deployment, Service NodePort, k6-профили 95/5–50/50–5/95
LAB18	Grafana-дашборд	Стек Prometheus–Loki–Grafana–Promtail в k3s, персональный дашборд: Pod, ResourceQuota, requests/limits

Практические показы являются частью содержания онлайн-занятий. Самостоятельное повторение не образует отдельную оцениваемую практическую аттестацию: результаты программы, связанные с показанными реализациями, оцениваются через итоговый тест на основании матрицы покрытия.

8. Оценочные материалы и формы аттестации

8.1. Форма итоговой аттестации

Итоговая аттестация проводится только в форме тестирования и занимает 1 ак. ч. Тест включает 35 вопросов и рассчитан на 45 минут. Итоговый банк содержит вопросы по Java/Spring, работе с данными, контейнеризации, нагрузочному тестированию, масштабированию, Kafka, Kubernetes, кешированию и отказоустойчивости.

Комплект оценочных материалов ОМ-ПК-01 включает отдельные промежуточный и итоговый тесты по 35 вопросов с закрытыми ключами и матрицей покрытия обязательных результатов обучения. Комплект утверждён приказом ООО «ЗИЛ ДИДЖИТАЛ» от 22.11.2026 № 1-ДПО. Вопросы и правильные ответы размещаются в закрытой части ZIL LMS и не публикуются.

8.2. Критерий успешного прохождения

Итоговая аттестация считается успешно пройденной при доле правильных ответов **не менее 60 %**, то есть при получении **не менее 21 балла из 35**.

Слушателям, успешно прошедшим итоговое тестирование, оформляется удостоверение о повышении квалификации установленного образца в порядке, предусмотренном применимыми требованиями и локальными актами Общества. Самостоятельное повторение практических показов и результат формирующего промежуточного контроля не являются дополнительными условиями выдачи документа.

8.3. Параметры тестирования и пересдачи

Параметр	Установленное правило
Количество заданий	35 вопросов итогового контроля ОМ-ПК-01
Продолжительность	45 минут, что соответствует выделенному 1 ак. ч.
Балльная шкала	Каждый вопрос — 1 балл; максимум — 35 баллов
Типы заданий	13 вопросов одиночного и 22 вопроса множественного выбора
Несколько правильных вариантов	Балл начисляется только при выборе всех правильных вариантов и отсутствии ошибочных; частичный балл не начисляется
Проходной порог	Не менее 21 балла из 35, то есть не менее 60 %
Банк и порядок заданий	Один итоговый банк для первой попытки и пересдачи; LMS перемешивает вопросы и варианты ответов с сохранением привязки ключей к содержимому
Пересдача	После неуспешной первой попытки ответственный активирует одну дополнительную бесплатную попытку по тому же итоговому банку с новым порядком вопросов и вариантов в течение 10 рабочих дней после сообщения результата
Технический сбой	Подтверждённый технический сбой не считается использованной попыткой или пересдачей; слушатель продолжает сохранённую попытку либо получает новый запуск после повторной идентификации
Материалы и помощь при тестировании	Учебные материалы, справочная документация, внешние сервисы, инструменты искусственного интеллекта и помощь третьих лиц не разрешаются; слушатель использует только интерфейс итогового теста в ZIL LMS
Вопросы и ключи	Промежуточный и итоговый банки входят в закрытый комплект ОМ-ПК-01; правильные ответы и разбор вопросов слушателям не раскрываются
Протокол итоговой аттестации	Оформляется ответственным лицом по данным ZIL LMS и содержит сведения о слушателе, группе, программе и редакции, дате, попытке, идентификации, результате, техническом сбое при наличии и проверяющем

8.4. Текущая работа

Промежуточный контроль проводится по 35 вопросам комплекта ОМ-ПК-01 как формирующее тестирование. Его результат справочный: не добавляет баллы к итоговой аттестации, не заменяет её и не влияет на допуск или выдачу документа о квалификации. Отдельные часы сверх учебно-тематического плана не добавляются.

Практические показы из раздела 7 проводятся преподавателем внутри часов соответствующих лекций. Их самостоятельное повторение слушателем носит рекомендованный тренировочный характер, выполняется вне 24 ак. ч. и не является предметом обязательной защиты или балльного оценивания. Такое повторение рекомендуется для качественной подготовки к итоговому тестированию, но не является условием допуска к нему или выдачи документа о квалификации.

8.5. Идентификация слушателя

Применяется следующий порядок:

1. Каждому слушателю предоставляется персональная учётная запись для доступа к электронной информационно-образовательной среде и итоговому тестированию.
2. Перед итоговым тестом проводится видеосверка личности слушателя с документом, удостоверяющим личность, в объёме, необходимом для идентификации.
3. Результат идентификации фиксируется в протоколе.
4. Обязательное сохранение изображения документа, удостоверяющего личность, не предусматривается.
5. Автоматизированный прокторинг не применяется.

Ответственные лица, порядок идентификации, действия при технических проблемах и оформление результатов определяются Положением о применении электронного обучения и дистанционных образовательных технологий ООО «ЗИЛ ДИДЖИТАЛ».

9. Организационно-педагогические условия

9.1. Программное обеспечение и технологический стек

Перечень программного обеспечения и инструментов, используемых в лекциях и практических показах:

- JDK 25 (Eclipse Temurin), Java 25 (LTS);
- IntelliJ IDEA Community Edition;
- Git;
- Apache Maven и Gradle;
- Docker / Docker Desktop, Docker Compose, Docker Swarm;
- гипервизор виртуальных машин VMware / VirtualBox;
- PostgreSQL и pgAdmin;
- Apache Ignite;
- Apache Kafka в режиме KRaft, Kafka UI;
- Kubernetes / k3s, kubectl, Minikube;
- Redis;
- k6 и xk6-kafka;
- Python 3 с библиотеками requests и Faker;

- Prometheus, Loki, Grafana, Promtail;
- resilience4j;
- Spring Boot, Spring Framework, Spring Data JPA, Hibernate, Spring MVC/REST, Spring Kafka;
- Spring Test / Spring Boot Test, TestRestTemplate, MockMvc, JUnit 5, Testcontainers, WireMock;
- Spring Boot Actuator и Micrometer для разбора метрик;
- Jackson, Swagger UI / OpenAPI;
- RestTemplate, WebClient, Feign;
- реестр образов Harbor / DockerHub.

Перечень отражает содержание лекций и практических показов. Обзорные сравнения технологий из раздела 3.4 не означают обязательное развёртывание каждого названного продукта. Использование программного обеспечения осуществляется в пределах прав, предоставленных соответствующими лицензиями и условиями использования.

9.2. Учебно-лабораторная инфраструктура

Практические демонстрации LAB0–LAB18 проводятся преподавателем на подготовленном учебном стенде в рамках онлайн-занятий. Преподаватель демонстрирует экран, исходный код, конфигурации, команды и результаты работы распределённой системы. Обязательное подключение каждого слушателя к отдельной облачной виртуальной машине или самостоятельное развёртывание учебного стенда не требуется.

Демонстрационный код, инструкции и материалы практических примеров предоставляются слушателям через ZIL LMS. Добровольное самостоятельное повторение выполняется на оборудовании слушателя или в доступной ему среде вне общей трудоёмкости 24 ак. ч. и не является условием допуска к итоговой аттестации.

9.3. Электронная информационно-образовательная среда

Для реализации программы используются следующие компоненты электронной информационно-образовательной среды:

- LMS / ЭИОС с персональными учётными записями слушателей;
- размещение материалов программы, расписания и инструкций по подключению;
- проведение онлайн-занятий с преподавателем в реальном времени;
- доступ к демонстрационному коду и материалам практических примеров;
- проведение итогового тестирования с фиксацией результата;
- ведение протоколов идентификации и итоговой аттестации;
- разграничение прав доступа слушателей, преподавателя и администраторов;
- хранение и резервное копирование учебных результатов в соответствии с утверждёнными правилами.

Используется ZIL LMS по адресу https://lms.zil.digital. Электронный курс «Highload (24 ак. ч.)» включает вводное занятие, 16 лекций, практические демонстрации LAB0–LAB18, учебные презентации, промежуточный и итоговый тесты. ZIL LMS обеспечивает персональный доступ, назначение курса и группы, фиксацию попыток, ответов, баллов и результатов. Правильные ответы и разбор вопросов слушателям не отображаются.

9.4. Кадровое обеспечение и методическое сопровождение

Разработчик, преподаватель и лицо, осуществляющее методическое сопровождение программы, — Ровнягин Михаил Михайлович, преподаватель по договору гражданско-правового характера.

Параметр	Сведения
Ф. И. О.	Ровнягин Михаил Михайлович
Высшее образование	НИЯУ МИФИ, 2012; специальность «Вычислительные машины, комплексы, системы и сети», квалификация инженер
Учёная степень	Кандидат технических наук, 2016
Основное место работы	АО «Райффайзенбанк»
Функция по программе	Преподаватель и методическое сопровождение ПК-01 по договору гражданско-правового характера

10. Календарный учебный график

Обучение проводится в интенсивном режиме: **6 учебных дней по 4 академических часа, всего 24 ак. ч.** Конкретные даты, время начала и окончания занятий, перерывы и идентификатор учебной группы устанавливаются расписанием. Обязательное посещение помещений Общества расписанием не предусматривается.

Темы изучаются в последовательности фактических материалов: введение в highload → Java/SOLID → Spring Core → HTTP и сервлеты → Spring MVC → Spring Boot → Hibernate/Spring Data JPA → Testcontainers/WireMock → Git и системы сборки → Apache Ignite → Apache Kafka → Docker → паттерны масштабирования → Kubernetes/k3s → метрики highload-приложений → сбор и анализ метрик → кеширование. Практические показы выполняются преподавателем параллельно с соответствующей теорией. Самостоятельное повторение на сходных бизнес-постановках выполняется по выбору между занятиями вне 24 ак. ч. Итоговое тестирование проводится на заключительном занятии.

Распределение часов по учебным дням; конкретные даты и время занятий устанавливаются расписанием учебной группы:

Учебный день	Содержание	Онлайн-занятия, ак. ч.	Итоговое тестирование, ак. ч.	Всего, ак. ч.
1	Вводное занятие — 1 ак. ч.; лекции 1–3 — по 1 ак. ч.	4	0	4
2	Лекции 4–5 — по 1 ак. ч.; лекция 6 — 2 ак. ч.	4	0	4
3	Лекции 7–8 — по 0,5 ак. ч.; лекция 9 — 1 ак. ч.; лекция 10 — 2 ак. ч.	4	0	4

Учебный день	Содержание	Онлайн-занятия, ак. ч.	Итоговое тестирование, ак. ч.	Всего, ак. ч.
4	Лекции 11–12 — по 2 ак. ч.	4	0	4
5	Лекции 13–14 — по 2 ак. ч.	4	0	4
6	Лекция 15 — 1 ак. ч.; лекция 16 — 2 ак. ч.; итоговое тестирование — 1 ак. ч.	3	1	4
Итого	**Вводное занятие, 16 лекций и итоговое тестирование**	**23**	**1**	**24**

Расписание учебной группы может детализировать время и перерывы, но должно сохранять общую трудоёмкость 24 ак. ч., объём 23 ак. ч. онлайн-занятий и 1 ак. ч. итогового тестирования, а также исключительное применение ЭО/ДОТ.

11. Список литературы и источников

Для самостоятельного изучения тем и получения дополнительной справочной информации используются следующие литература и электронные ресурсы.

11.1. Основная литература

1. Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман ; пер. с англ. — Санкт-Петербург : Питер, 2019. — 640 с.
2. Уоллс, К. Spring в действии / К. Уоллс ; пер. с англ. — 6-е изд. — Москва : ДМК Пресс, 2022. — 520 с.
3. Блох, Дж. Java. Эффективное программирование / Дж. Блох ; пер. с англ. — 3-е изд. — Москва : Диалектика, 2019. — 464 с.
4. Лукша, М. Kubernetes в действии / М. Лукша ; пер. с англ. — Москва : ДМК Пресс, 2019. — 672 с.
5. Ньюмен, С. Создание микросервисов / С. Ньюмен ; пер. с англ. — 2-е изд. — Санкт-Петербург : Питер, 2023. — 624 с.

11.2. Дополнительная литература

6. Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин ; пер. с англ. — Санкт-Петербург : Питер, 2019. — 464 с.
7. Нархид, Н. Apache Kafka. Потоковая обработка и анализ данных / Н. Нархид, Г. Шапира, Т. Палино ; пер. с англ. — Санкт-Петербург : Питер, 2019. — 320 с.
8. Бауэр, К. Java Persistence API и Hibernate / К. Бауэр, Г. Кинг, Г. Грегори ; пер. с англ. — Москва : ДМК Пресс, 2017. — 632 с.
9. Beyer, B. Site Reliability Engineering: How Google Runs Production Systems / B. Beyer, C. Jones, J. Petoff, N. R. Murphy. — Sebastopol : O'Reilly Media, 2016. — 552 p.

10. Brazil, B. Prometheus: Up & Running: Infrastructure and Application Performance Monitoring / B. Brazil. — Sebastopol : O'Reilly Media, 2018. — 386 p.

11.3. Электронные ресурсы и документация

11. Spring Boot Reference Documentation [Электронный ресурс]. — URL: https://docs.spring.io/spring-boot/reference/ (дата обращения: 30.06.2026).
12. Spring Framework Documentation [Электронный ресурс]. — URL: https://docs.spring.io/spring-framework/reference/ (дата обращения: 30.06.2026).
13. Spring Data JPA — Reference Documentation [Электронный ресурс]. — URL: https://docs.spring.io/spring-data/jpa/reference/ (дата обращения: 30.06.2026).
14. Hibernate ORM User Guide [Электронный ресурс]. — URL: https://hibernate.org/orm/documentation/ (дата обращения: 30.06.2026).
15. Apache Kafka Documentation [Электронный ресурс]. — URL: https://kafka.apache.org/documentation/ (дата обращения: 30.06.2026).
16. Apache Ignite Documentation [Электронный ресурс]. — URL: https://ignite.apache.org/docs/latest/ (дата обращения: 30.06.2026).
17. Docker Documentation [Электронный ресурс]. — URL: https://docs.docker.com/ (дата обращения: 30.06.2026).
18. Kubernetes Documentation [Электронный ресурс]. — URL: https://kubernetes.io/docs/home/ (дата обращения: 30.06.2026).
19. K3s — Lightweight Kubernetes. Documentation [Электронный ресурс]. — URL: https://docs.k3s.io/ (дата обращения: 30.06.2026).
20. Redis Documentation [Электронный ресурс]. — URL: https://redis.io/docs/latest/ (дата обращения: 30.06.2026).
21. Resilience4j User Guide [Электронный ресурс]. — URL: https://resilience4j.readme.io/docs/ (дата обращения: 30.06.2026).
22. Prometheus — Documentation [Электронный ресурс]. — URL: https://prometheus.io/docs/ (дата обращения: 30.06.2026).
23. Grafana Documentation [Электронный ресурс]. — URL: https://grafana.com/docs/grafana/latest/ (дата обращения: 30.06.2026).
24. Grafana k6 Documentation [Электронный ресурс]. — URL: https://grafana.com/docs/k6/latest/ (дата обращения: 30.06.2026).
25. Развёртывание Apache Kafka (KRaft) в Docker Swarm на Debian 12 [Электронный ресурс] : руководство ZIL.digital. — URL: https://zil.digital/blog/kafka-kraft-swarm-debian12 (дата обращения: 30.06.2026).
26. Исходный код микросервиса-примера, используемого в практических показах и самостоятельных тренировках [Электронный ресурс] : репозиторий hl-module1. — URL: https://bitbucket.org/zil-courses/hl-module1/src/main/ (дата обращения: 30.06.2026).