



УТВЕРЖДАЮ

Генеральный директор

ООО «Мерион Нетворкс»

Тундайкина Т.Н.

«12» мая 2026 г.

Дополнительная профессиональная программа

повышения квалификации

«Основы Java-разработки»

г. Москва

2026

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Общие сведения о программе

Дополнительная профессиональная программа повышения квалификации «Основы Java-разработки» (далее — Программа) разработана в соответствии с требованиями Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» и приказа Министерства образования и науки Российской Федерации от 01 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам».

Программа ориентирована на начинающих разработчиков, а также специалистов смежных IT-направлений, стремящихся освоить backend-разработку на языке Java с использованием современного стека технологий: Spring Boot, Maven, REST API, MongoDB, PostgreSQL, Spring Security.

Наименование программы	Основы Java-разработки
Категория слушателей	Специалисты в области IT, имеющие высшее или среднее профессиональное образование
Форма обучения	Заочная с применением дистанционных образовательных технологий (ДОТ)
Объем программы	220 академических часов
Срок освоения программы	Не менее 2 месяцев (доступ к платформе — 2 года)
Итоговый документ	Удостоверение о повышении квалификации (для слушателей тарифа «Наставник»)
Разработчик	ООО «Мерион Нетворкс» (Merion Academy)

1.2. Цель программы

Цель Программы — формирование профессиональных компетенций в области backend-разработки на языке Java: от основ объектно-ориентированного программирования до создания полноценных REST API веб-приложений с аутентификацией и работой с реляционными и нереляционными базами данных.

1.3. Задачи программы

Для достижения цели Программы решаются следующие задачи:

- освоить синтаксис Java, основные типы данных, структуры данных и принципы объектно-ориентированного программирования (ООП);
- сформировать навыки работы с системой контроля версий Git и инструментом сборки Maven;

- изучить протокол HTTP, принципы REST архитектуры и научиться работать с API через Postman и Retrofit;
- освоить разработку веб-приложений с использованием Spring Boot и Spring MVC;
- получить практические навыки работы с базами данных MongoDB и PostgreSQL в контексте Spring Data;
- изучить методологию и инструменты тестирования (JUnit, Mockito, Spring Testing);
- научиться документировать REST API с помощью Swagger (OpenAPI);
- освоить защиту веб-приложений средствами Spring Security (аутентификация, авторизация);
- закрепить полученные знания в самостоятельном проекте — реализации API для движка Merion Academy.

1.4. Требования к слушателям

К освоению Программы допускаются лица, имеющие высшее или среднее профессиональное образование. Специальных требований к предварительным знаниям Java не предъявляется; рекомендуется базовое понимание принципов работы компьютера и операционных систем.

1.5. Материально-техническое обеспечение

Реализация Программы осуществляется с применением дистанционных образовательных технологий на платформе Merion Academy (lms.merionet.ru). Для обучения слушателю необходимы:

- персональный компьютер или ноутбук (ОС: Windows 10/11, Ubuntu 20.04+, macOS 12+) с процессором не ниже Intel Core i5 / AMD Ryzen 5 и оперативной памятью не менее 8 ГБ;
- программное обеспечение: IntelliJ IDEA (Community или Ultimate), JDK 17+, Apache Maven, Git, Docker (для запуска MongoDB и PostgreSQL), Postman;
- доступ к сети Интернет со скоростью не менее 10 Мбит/с;
- веб-браузер актуальной версии (Google Chrome, Mozilla Firefox).

2. УЧЕБНЫЙ ПЛАН

№	Наименование раздела	Всего ак. ч.	Лекции/ видео	Практика	Контроль
1	Java Base. Базовые типы и структуры данных, переменные. Условные операторы, работа с функциями	65	29	26	10
2	Основы работы с Git	9	4	4	1
3	Основы работы с Maven. Сборка проекта	10	5	4	1
4	Знакомство с HTTP, REST API, работа с запросами	16	7	6	3
5	Знакомство с Spring. Создание первого веб-приложения	12	5	5	2
6	Работа с базами данных в веб-приложении (MongoDB, PostgreSQL)	24	11	10	3
7	Тестирование (Unit-тесты, Spring Testing)	10	5	4	1
8	Документация REST API. Swagger	1	0	0	1
9	Защита веб-приложений. Spring Security	12	5	5	2
10	Самостоятельный проект. API для движка Merion Academy	21	9	8	4
	ИТОГО	220	80	72	68

Примечание: лекции проводятся в формате видеоуроков. Практические занятия включают выполнение кодовых заданий, лабораторных работ и самостоятельного проекта. Промежуточный контроль — тестирование в LMS по завершении каждого блока. Код практических работ размещается слушателем на GitHub.

3. РАБОЧАЯ ПРОГРАММА

Блок 1. Java Base. Основы языка и ООП (65 ак. ч.)

Содержание. Введение в язык Java: история, платформа JVM, JDK, JRE. Установка Java и IntelliJ IDEA, обзор инструментов разработчика. Базовые типы данных: примитивные типы и классы-обёртки. Переменные, операторы, выражения. Пакеты Java. Управляющие конструкции: условный оператор if-else, switch. Массивы и коллекции: Collection Framework (List, Set, Map), итераторы, циклы (for, while, for-each). Объектно-ориентированное программирование: классы, объекты, конструкторы, поля, методы (статические и экземплярные), уровни доступа. Геттеры и сеттеры. Наследование. Аннотации (в том числе @Override). Equals и hashCode. Неизменяемые объекты. Перечисления (Enum). Работа со строками (String, StringBuilder). Преобразование типов. Интерфейсы и абстрактные классы. Вложенные и анонимные классы. Исключения (checked/unchecked, try-catch-finally). Лямбда-выражения. Stream API. Многопоточность (базовый уровень).

Практические работы: решение задач на типы данных, ООП, коллекции и Stream API; публикация кода на GitHub. Промежуточный контроль: тест по блоку 1 (порог — 80%, попыток — без ограничений).

Блок 2. Основы работы с Git (9 ак. ч.)

Содержание. Системы контроля версий: назначение, виды (централизованные, распределённые). Обзор хостингов: GitHub, GitLab, Bitbucket. Установка и конфигурация Git. Базовые операции: git init, add, commit, push, pull, status, log. Отмена изменений: git checkout, reset, revert. Команда stash. Ветвление: создание веток, pull request, разрешение конфликтов, слияние (merge, rebase), удаление веток. Fetch и навигация по веткам. Работа с удалённым репозиторием на GitHub.

Практическая работа: создание репозитория, ведение истории коммитов, работа с ветками. Промежуточный контроль: тест по блоку 2 (порог — 80%).

Блок 3. Основы работы с Maven. Сборка проекта (10 ак. ч.)

Содержание. Системы сборки проектов: Maven, Gradle, Ant — сравнение и область применения. Жизненный цикл сборки Maven (validate, compile, test, package, install, deploy). Структура pom.xml: координаты проекта, зависимости, плагины, профили, свойства, лицензии, разработчики, SCM. Управление зависимостями: транзитивные зависимости, scopes (compile, test, provided). Мультимодульные проекты. Установка Maven и интеграция с IntelliJ IDEA.

Практическая работа: настройка Maven-проекта с зависимостями, сборка и публикация на GitHub. Промежуточный контроль: тест по блоку 3 (порог — 80%).

Блок 4. Знакомство с HTTP, REST API, работа с запросами (16 ак. ч.)

Содержание. Протокол HTTP: структура запроса и ответа, заголовки, методы (GET, POST, PUT, DELETE, PATCH), коды состояния (2xx, 3xx, 4xx, 5xx). Куки (Cookie): назначение, работа в HTTP. Форматы данных: XML и JSON — структура, сравнение, сериализация/десериализация. Принципы REST-архитектуры (6 ограничений REST). Инструмент Postman: установка, создание запросов, коллекции, тестирование API на примере jsonplaceholder.typicode.com. Библиотека Retrofit: реализация HTTP-клиента в Java-приложении.

Практическая работа: написание HTTP-клиента на Retrofit; тестирование API в Postman. Промежуточный контроль: тест по блоку 4 (порог — 80%).

Блок 5. Знакомство с Spring. Первое веб-приложение (12 ак. ч.)

Содержание. Экосистема Spring: Spring Framework, Spring Boot, Spring MVC, Spring Data, Spring Security. Spring Boot Initializr: создание проекта, структура, автоконфигурация. Spring MVC: архитектура DispatcherServlet, аннотации @Controller, @RestController, @RequestMapping. Реализация REST-эндпоинтов: GET-, POST-, PUT-, DELETE-, Path-запросы (@GetMapping, @PostMapping и др.), @PathVariable, @RequestParam, @RequestBody. Сериализация ответов в JSON (Jackson). Публикация кода на GitHub.

Практическая работа: создание REST API с базовыми CRUD-операциями. Промежуточный контроль: тест по блоку 5 (порог — 80%).

Блок 6. Работа с базами данных в веб-приложении (24 ак. ч.)

Содержание. Роль баз данных в веб-приложениях. SQL и NoSQL: сравнение, сценарии применения. Spring Data: абстракция репозитория, JPA, MongoRepository. MongoDB: установка (Docker), настройка подключения, основные операции (сохранение, поиск по ID, получение всех, фильтрация, обновление, удаление). Реализация API сущности User на MongoDB. PostgreSQL: установка (Docker), настройка DataSource и JPA/Hibernate, аналогичный набор операций. ORM-маппинг: @Entity, @Id, @Column, @OneToMany и др. Написание практических заданий и видеоразборы.

Практические работы: CRUD API на MongoDB; CRUD API на PostgreSQL. Промежуточный контроль: тест по блоку 6 (порог — 80%).

Блок 7. Тестирование (10 ак. ч.)

Содержание. Зачем нужны тесты. Виды тестирования: Unit-, Integration-, End-to-End. Unit-тесты: JUnit 5, Mockito — аннотации, моки, стабы, проверки. Тестирование в Spring: настройка тестового окружения (@SpringBootTest, @WebMvcTest), MockMvc. Покрывание тестами REST-контроллеров: тесты на каждый эндпоинт, позитивные и негативные сценарии.

Практическая работа: написание Unit- и интеграционных тестов для REST-контроллеров. Промежуточный контроль: тест по блоку 7 (порог — 80%).

Блок 8. Документация REST API. Swagger (1 ак. ч.)

Содержание. Стандарт OpenAPI (Swagger). Подключение springdoc-openapi к проекту. Аннотации Swagger: @Operation, @ApiResponse, @Parameter, @Schema. Генерация интерактивной документации (Swagger UI). Промежуточный контроль: тест по блоку 8 (порог — 80%).

Блок 9. Защита веб-приложений. Spring Security (12 ак. ч.)

Содержание. Основы информационной безопасности веб-приложений. Spring Security: архитектура (Filter Chain, SecurityContext, Authentication). Базовая HTTP-аутентификация. Хранение паролей: BCryptPasswordEncoder. Реализация регистрации и аутентификации по логину/паролю. Авторизация: роли, @PreAuthorize, @Secured. JWT (JSON Web Token): структура, генерация, валидация (опционально). Интеграция тестов со Spring Security: @WithMockUser, @WithUserDetails.

Практическая работа: реализация аутентификации и авторизации в REST API. Промежуточный контроль: тест по блоку 9 (порог — 80%).

Блок 10. Самостоятельный проект. API для движка Merion Academy (21 ак. ч.)

Содержание. Самостоятельный итоговый проект: разработка REST API для учебной платформы Merion Academy. Слушатель реализует три независимые группы эндпоинтов: API для сущности «Ученик» (CRUD), API для сущности «Курс» (CRUD), API для сущности «Урок» (CRUD с привязкой к курсу). Требования: Spring Boot, работа с базой данных на выбор (MongoDB или PostgreSQL), документирование через Swagger, базовая защита Spring Security, тесты на ключевые эндпоинты, публикация кода на GitHub.

Результат: рабочее приложение с документацией на GitHub, которое может быть использовано в качестве портфолио.

Итоговая аттестация (входит в общий объём программы)

Итоговая аттестация проводится в форме тестирования на платформе Merion Academy (lms.merionet.ru). Тест охватывает все разделы программы. Порог зачёта — 80% правильных ответов. Количество попыток — неограниченно. Оценивание: «зачёт» (80% и выше) / «незачёт» (до 79% включительно).

4. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ. ПОРЯДОК ПРОВЕДЕНИЯ АТТЕСТАЦИИ

Оценивание освоения Программы осуществляется в форме тестирования на образовательной платформе Merion Academy (lms.merionet.ru). Применяется система «зачёт / незачёт».

4.1. Промежуточный контроль

По завершении каждого из 9 содержательных блоков слушатель проходит тематический тест. Параметры:

Параметр	Значение
Форма контроля	Тестирование в LMS (автоматизированная проверка)
Количество попыток	Неограниченно
Порог зачёта	$\geq 80\%$ правильных ответов
Результат «незачёт»	До 79% включительно
Система оценивания	Зачёт / Незачёт

4.2. Итоговая аттестация

К итоговой аттестации допускаются слушатели, успешно сдавшие все промежуточные тесты.

Параметр	Значение
Форма итоговой аттестации	Итоговое тестирование
Количество попыток	Неограниченно
Порог зачёта	$\geq 80\%$ правильных ответов
Результат «незачёт»	До 79% включительно — незачёт, передача без ограничений
Документ по итогам аттестации	Удостоверение о повышении квалификации — выдаётся слушателям тарифа «Наставник» при результате «зачёт»

5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

5.1. Основная литература

18. Эккель Б. Философия Java. — 4-е изд. — М.: Питер, 2016. — 1168 с.
19. Блох Дж. Effective Java. Лучшие практики проектирования. — 3-е изд. — М.: Вильямс, 2019. — 464 с.
20. Уоллс К. Spring в действии. — 5-е изд. — М.: ДМК Пресс, 2020. — 540 с.
21. Хорстманн К. Java. Библиотека профессионала. Том 1: Основы. — 12-е изд. — М.: Вильямс, 2022. — 864 с.
22. Форсен Д. Spring Boot: Up and Running. — O'Reilly Media, 2021. — 320 p.
23. Чандра Д. MongoDB: The Definitive Guide. — 3rd ed. — O'Reilly Media, 2019. — 514 p.
24. Ахо А., Лам М., Сети Р., Ульман Д. Компиляторы: принципы, технологии и инструментарий. — М.: Вильямс, 2010. — 1184 с.

5.2. Нормативно-правовые акты и стандарты

25. ГОСТ Р 57580.1-2017. Безопасность финансовых операций. Защита информации. — М.: Стандартинформ, 2018.

5.3. Методические материалы

26. Merion Academy. Методические материалы курса «Основы Java-разработки». — М.: Merion Networks LLC, 2024. — <https://merionet.ru>

5.4. Интернет-ресурсы

27. Oracle. Java Documentation — <https://docs.oracle.com/en/java/>
28. Spring. Официальная документация — <https://spring.io/docs>
29. Maven. Официальная документация — <https://maven.apache.org/guides/>
30. MongoDB. Документация — <https://www.mongodb.com/docs/>
31. PostgreSQL. Документация — <https://www.postgresql.org/docs/>
32. Swagger / OpenAPI — <https://swagger.io/docs/>
33. Merion Academy — <https://merionet.ru>
34. LMS-платформа курса — <https://lms.merionet.ru>