ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол №2

Рабочая программа дисциплины (модуля)
«Разработка серверных приложений с использованием Java и Spring»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Дизайн и разработка информационных продуктов*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2026*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2026

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 5
3. **Тематический план** 5
4. **Содержание дисциплины (модуля)** 10
5. **Учебно-методическое обеспечение** 12
6. **Материально-техническое обеспечение** 12
7. **Методические и оценочные материалы** 14

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка серверных приложений с использованием Java и Spring» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Дизайн и разработка информационных продуктов, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Разработка серверных приложений с использованием Java и Spring» позволяет сформировать фундаментальные и практико-ориентированные компетенции, необходимые для разработки современных промышленный backend-сервисов на платформе Java. Освоение Spring-экосистемы позволяет выпускнику уверенно разрабатывать масштабируемые, тестируемые и готовые к эксплуатации корпоративные приложения.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Дизайн и разработка информационных продуктов и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре.

Цель изучения дисциплины (модуля): сформировать у обучающихся системное понимание принципов построения backend-сервисов на Java и Spring и развить практические компетенции по проектированию, реализации, тестированию и подготовке к эксплуатации полнофункциональных Spring Boot приложений.

Задачи изучения дисциплины (модуля):

- изучить принципы IoC/DI, устройство Spring Context, жизненный цикл бинов и способы конфигурирования приложений для осознанного управления зависимостями и архитектурой;
- освоить механизмы AOP, проксирования и расширения контейнера Spring для реализации сквозной функциональности и диагностики типовых проблем;
- научиться проектировать и реализовывать слой доступа к данным с использованием Spring JDBC, JPA/Hibernate и Spring Data с корректной настройкой транзакций и обработки исключений;
- приобрести умения разработки RESTful веб сервисов на Spring MVC с валидацией, централизованной обработкой ошибок и покрытием модульными и интеграционными тестами;
- освоить интеграционные и эксплуатационные аспекты разработки backend сервисов, включая работу с HTTP клиентами, средствами безопасности, документирования API, наблюдаемости и базовыми подходами к деплою.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- принципы IoC/DI и устройство DI-контейнера (Spring Context), виды конфигураций Spring;
- основы конфигурирования Spring-приложений: свойства, профили, SpEL, conditional-конфигурация;
- AOP в Spring: прокси, pointcut/advice, ограничения Spring AOP и типовые ловушки (внутренние вызовы);
- жизненный цикл бинов и scope-ы, расширение контейнера через post-processors и события;

— доступ к данным: паттерны DAO/Repository/Service, Spring JDBC, транзакции, Hibernate/JPA, Spring Data;
— веб-слой: Servlet API, Spring MVC, валидация, централизованная обработка ошибок, REST/Ajax, WebSocket (обзорно);
— интеграции и эксплуатационные аспекты: HTTP-клиенты (RestTemplate/WebClient/RestClient), OpenAPI, сообщения (RabbitMQ/Kafka — обзорно), Security, наблюдаемость (Actuator/метрики/трейсинг), вводно про деплой в K8S;

уметь:

— проектировать компоненты приложения с соблюдением DIP и проверяемой архитектурой (тестируемые зависимости);
— конфигурировать Spring-приложение несколькими способами и осознанно выбирать подход (XML/Java/annotations, profiles/conditionals);
— писать и отлаживать аспекты (AOP) и конфигурационные цепочки (lifecycle, scope, proxy);
— реализовывать доступ к данным с корректной работой транзакций и обработкой исключений на слоях приложения;
— реализовывать веб-эндпоинты и REST API с валидацией и единообразной ошибочной моделью;
— писать модульные и интеграционные тесты для Spring Boot приложения (включая работу с контейнерами тестовых зависимостей — обзорно);

владеть:

— навыками сборки backend-сервиса на Spring Boot от конфигурации до готовых API и доступа к данным;
— навыками диагностики проблем конфигурации, проксирования и транзакций (пониманием “как это работает внутри”);
— навыками подготовки проекта к эксплуатации: безопасность, документация API, базовая наблюдаемость.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области искусственного интеллекта, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
УК-2.	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1.	Знает действующие правовые нормы, регулирующие деятельность в области решения задач, основные методы и подходы к определению круга задач
		УК-2.2.	Умеет определять круг задач в рамках поставленной цели, выбирать оптимальные способы решения задач, учитывая имеющиеся ресурсы и ограничения
		УК-2.3.	Имеет практический опыт применения знаний о правовых нормах и ресурсах в реальных ситуациях, разработки и

			реализации решений в соответствии с установленными ограничениями
УК-3.	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1	Знает основные принципы социального взаимодействия и командной работы, роли и ответственности в команде
		УК-3.2	Умеет эффективно взаимодействовать с членами команды, разрешать конфликты и достигать согласия в группе
		УК-3.3	Имеет практический опыт участия в командной работе, реализации своей роли в команде для достижения общих целей
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализация математических моделей в профессиональной деятельности
ОПК-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности

		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ОПК-5.	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-5.1.	Знает технологии, необходимые для прикладного программирования, включая современные функциональные языки программирования, а также основные принципы и понятия, применяемые при использовании компьютерных сетей
		ОПК-5.2.	Умеет пользоваться технологиями прикладного программирования, включая среды высокоуровневого программирования
		ОПК-5.3.	Имеет практический опыт использования технологий прикладного программирования
ОПК-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики в области профессиональной деятельности
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования в области профессиональной деятельности

		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных Электронный документ информационных систем для визуализации результатов исследований в области профессиональной деятельности
ПК-2.	Способен решать типовые задачи профессиональной деятельности, опираясь на информационную и библиографическую культуру, используя информационно-коммуникационные технологии и учитывая основные требования информационной безопасности	ПК-2.1.	Знает основы информационной и библиографической культуры, а также принципы информационной безопасности и применения информационно-коммуникационных технологий в профессиональной деятельности
		ПК-2.2.	Умеет эффективно использовать информационно-коммуникационные технологии для решения стандартных задач профессиональной деятельности, учитывая требования информационной безопасности
		ПК-2.3.	Имеет опыт работы с информационными ресурсами и технологиями в области профессиональной деятельности, включая соблюдение норм информационной безопасности
ПК-3.	Способен применять методы математического и алгоритмического моделирования для решения как теоретических, так и практических задач в рамках профессиональной деятельности	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет применять методы математического и алгоритмического моделирования для анализа и решения различных задач в области математики и

			компьютерных наук
		ПК-3.3.	Имеет опыт использования методов математического и алгоритмического моделирования при решении теоретических и прикладных задач в профессиональной деятельности
ПК-4.	Способен разрабатывать программное обеспечение для решения прикладных задач в области профессиональной деятельности под руководством более опытного специалиста	ПК-4.1.	Знает основные принципы разработки программного обеспечения и методы решения прикладных задач в сфере профессиональной деятельности
		ПК-4.2.	Умеет разрабатывать программное обеспечение под руководством специалистов более высокой категории, используя современные технологии и инструменты
		ПК-4.3.	Имеет опыт участия в разработке программного обеспечения для решения прикладных задач в команде под руководством более опытных специалистов

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы					ТКУ (текущий контроль успеваемости)
		Очная форма					
		Контактная работа			Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)	Консультации			
1	Введение + IoC/DI	2	2	1		9	Домашние задания
2	Spring Context	2	2	1		9	Домашние задания
3	Properties + SpEL	2	2	1		9	Домашние задания
4	AOP	2	2	1		9	Домашние задания
5	Lifecycle + Scope	2	2	1		10	Домашние задания
6	Spring Boot	2	2	1		10	Домашние задания
7	Data access I	2	2	1	6	10	Домашние задания Коллоквиум
8	Транзакции	2	2	1		10	Домашние задания
9	JPA/Hibernate	2	2	1		10	Домашние задания
10	Spring Data	2	2	1		10	Домашние задания
11	Web	3	3	2		10	Домашние задания
12	Интеграции и эксплуатация	3	3	2		10	Домашние задания Проект
	Экзамен				2		
	Итого:	***26***	***26***	***14***	***8***	***116***	
Объем дисциплины (модуля) (в ак. ч.)		***190***					
Объем дисциплины (модуля) (в зач. ед.)		***5***					

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Введение + IoC/DI	Настройка IDE/дебаг. IoC/DI, DIP, тестируемость, мотивация DI-контейнера
2	Spring Context	Spring как экосистема. ApplicationContext. XML/Java/Annotation конфиг, @Autowired, @Qualifier/@Primary, варианты внедрения
3	Properties + SpEL	Property sources, Environment, @Value. SpEL (возможности/сценарии/инъекции). ConversionService и конвертеры
4	AOP	Proxy vs Decorator. JDK/CGLib proxy. AspectJ pointcut. Spring AOP (@Aspect), внутренние вызовы и ограничения

5	Lifecycle + Scope	BeanFactory vs ApplicationContext. Scope-ы, proxy для короткоживущих бинов. ЖЦ бина, post-processors, события Spring
6	Spring Boot	Boot как платформа: starters/автоконфигурация. Conditional vs profiles. Локализация (i18n)
7	Data access I	DAO/Repository/Service. Spring JDBC. Исключения на слоях приложения и перевод исключений через @Repository
8	Транзакции	@Transactional, propagation/isolation, типовые ошибки, отладка транзакционного поведения
9	JPA/Hibernate	Введение в Hibernate + JPA. Закон протекания абстракций. Маппинги, fetching, типовые проблемы производительности
10	Spring Data	Spring ORM (JPA/Hibernate) и обработка исключений. Spring Data JPA. Обзор Spring Data JDBC и NoSQL (Mongo/Redis/KeyValue)
11	Web	Servlet API. Spring MVC, Thymeleaf (обзорно), валидация, централизованная обработка исключений. REST/Ajax, WebSocket (обзорно)
12	Интеграции и эксплуатация	HTTP-клиенты, OpenAPI. RabbitMQ/Kafka/Spring Integration (обзорно). Spring Security (URL/method/ACL). Actuator/метрики/трейсинг. Введение в K8S

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Уоллс, К. Spring в действии : практическое руководство / К. Уоллс ; пер. с англ. А. Н. Киселева. - 6-е изд. - Москва : ДМК Пресс, 2022. - 544 с. - ISBN 978-5-93700-112-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/2110012.

2. Хеклер, М. Spring Boot по-быстрому : практическое руководство / М. Хеклер. - Санкт-Петербург : Питер, 2022. - 352 с. - (Бестселлеры O'Reilly). - ISBN 978-5-4461-3942-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/2122953.

Дополнительная литература:

1. Гутьеррес, Ф. Spring Boot 2: лучшие практики для профессионалов : практическое руководство / Ф. Гутьеррес. - Санкт-Петербург : Питер, 2021. - 464 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-1587-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1733697.

2. Карнелл, Д. Микросервисы Spring в действии : практическое руководство / Д. Карнелл, И. Уайлупо Санчес ; пер. с англ. А. Н. Киселева. - Москва : ДМК Пресс, 2022. - 490 с. - ISBN 978-5-97060-971-2. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2155906.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		

Электронный документ

Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка серверных приложений с использованием Java и Spring» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, коллоквиумы, домашние задания, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Консультации – структурированные встречи, на которых преподаватели предоставляют индивидуальную или групповую помощь в освоении учебного материала, обсуждении вопросов и решении проблем, возникающих в процессе обучения.

Консультации могут включать разъяснение сложных тем, подготовку к экзаменам и помощь в выполнении проектных работ, что способствует более глубокому пониманию предмета и улучшению академической успеваемости.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Коллоквиум – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки к коллоквиуму необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Разработка серверных приложений с использованием Java и Spring»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***экзамена***, при этом проводится оценка компетенций, сформированных по дисциплине. Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Разработка серверных приложений с использованием Java и Spring» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	20%	14	Набор задач по темам недели
Коллоквиум	20%	1	Письменная работа с набором задач, которые нужно решить за ограниченное время
Проект	30%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Экзамен	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка серверных приложений с использованием Java и Spring»: «0,2 × среднее за домашние задания + 0,2 × коллоквиум + 0,3 × проект + 0,3 × экзамен».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание по теме «Введение + IoC/DI».

1. Создать простой Java-проект без Spring, реализующий сервис с двумя зависимостями (например, **OrderService → PaymentService** + **NotificationService**), сначала с жесткими зависимостями, затем с внедрением через интерфейсы (DIP).
2. Написать модульные тесты и продемонстрировать, как DI упрощает тестирование (использовать mock-реализации).
3. Подключить Spring (без Boot), настроить IoC-контейнер и перевести создание объектов под управление контейнера.
4. Отладить запуск в IDE, показать точки создания бинов и внедрения зависимостей (debug walkthrough — краткий отчет).

Домашнее задание по теме «Spring Context».

1. Реализовать конфигурацию приложения тремя способами: XML, Java Config и аннотациями (для разных компонентов).
2. Продемонстрировать использование @Autowired, @Qualifier, @Primary на примере нескольких реализаций одного интерфейса.
3. Сравнить внедрение через конструктор, сеттер и поле — описать плюсы/минусы в отчете.
4. Добавить кастомный бин в контекст и получить его программно через ApplicationContext.

Домашнее задание по теме «Data Access I (Spring JDBC)».

1. Спроектировать слои DAO/Repository/Service для работы с сущностью (например, Order).
2. Реализовать CRUD-операции через Spring JDBC (JdbcTemplate).
3. Продемонстрировать обработку исключений и механизм перевода исключений через @Repository.
4. Разделить ответственность слоев и описать поток обработки запроса.
5. Написать интеграционный тест с использованием встроенной БД (H2).

Домашнее задание по теме «Интеграции и эксплуатация».

1. Реализовать вызов внешнего HTTP-сервиса через RestClient или WebClient.
2. Подключить OpenAPI (Swagger) и задокументировать API.
3. Настроить базовую безопасность (Spring Security) с разграничением доступа по ролям.
4. Подключить Spring Boot Actuator, включить метрики и health-checks.
5. Подготовить Dockerfile и описать (текстово) базовый процесс деплоя в Kubernetes.
6. (Обзорно) Настроить отправку сообщения в RabbitMQ или Kafka (можно без реального брокера — через mock или docker).

Примерные вопросы для коллоквиума.

1. В чем заключается принцип Inversion of Control и Dependency Inversion Principle, и каким образом DI-контейнер Spring способствует повышению тестируемости и гибкости архитектуры?
2. Чем отличается BeanFactory от ApplicationContext, и какие ключевые возможности предоставляет ApplicationContext в корпоративных приложениях?
3. Какие способы конфигурирования Spring-приложения существуют (XML, Java Config, аннотации), в чем их различия и когда оправдано использование каждого из них?
4. Как работают механизмы внедрения зависимостей (@Autowired, конструкторная инъекция, @Qualifier, @Primary), и какие проблемы они помогают решать?
5. Как в Spring реализован механизм AOP (JDK vs CGLIB proxy), в чем ограничения Spring AOP и почему внутренние вызовы (self-invocation) могут приводить к неожиданному поведению?
6. Что такое жизненный цикл бина в Spring, какие этапы он включает, и как можно расширить поведение контейнера с помощью BeanPostProcessor и событий?
7. Как устроена автоконфигурация в Spring Boot, что такое starters и conditional-конфигурация, и чем профили отличаются от @Conditional?
8. Как организуется доступ к данным в Spring (DAO/Repository/Service), как работает JdbcTemplate, и каким образом происходит перевод исключений через @Repository?
9. Как работает декларативное управление транзакциями с помощью @Transactional (propagation, isolation), и какие типовые ошибки возникают при его использовании?
10. В чем особенности работы JPA/Hibernate и Spring Data JPA (fetching, N+1, производительность), и как эти технологии интегрируются с веб-слоем, безопасностью и средствами наблюдаемости (Spring MVC, валидация, Security, Actuator)?

Примеры заданий и критерии оценивания проекта.

Цель проекта

Спроектировать и реализовать полнофункциональный backend-сервис на основе Spring Boot, демонстрирующий понимание архитектурных принципов (DIP, слоистость), механизмов Spring (IoC, AOP, транзакции, контекст, жизненный цикл), работы с данными (JPA/Spring Data), веб-слоя, безопасности и базовой подготовки приложения к эксплуатации.

Задачи и этапы выполнения проекта

Проект выполняется в виде законченного backend-приложения (например, система управления заказами, задачами, бронированиями, пользователями и т.п.) и должен включать следующие шаги:

— Проектирование архитектуры

Определить предметную область и описать функциональные требования.

Спроектировать слоистую архитектуру (Controller → Service → Repository/DAO → DB).

Обосновать применение DIP и способ внедрения зависимостей (конструкторная инъекция как основной подход).

— Настройка Spring Boot приложения

Создать проект с использованием starter’ов.

Настроить профили (dev/test/prod) и конфигурационные свойства (application.yml/properties).

Использовать @Configuration, @Conditional или профили для условной конфигурации.

Реализовать локализацию (i18n) для сообщений об ошибках или ответов API.

— Работа с контекстом и жизненным циклом

Продемонстрировать понимание жизненного цикла бина (@PostConstruct, @PreDestroy или аналог).

Использовать разные scope (например, singleton + prototype или request).

Реализовать простой BeanPostProcessor или обработчик события Spring.

— Реализация доступа к данным

Настроить подключение к БД (H2/PostgreSQL/MySQL).

Реализовать сущности JPA с корректными маппингами (включая связи).

Использовать Spring Data JPA (репозитории, производные запросы, @Query).

Продемонстрировать понимание lazy/eager fetching и проблемы N+1 (описание или пример).

Обеспечить корректную обработку и перевод исключений.

— Транзакционное управление

Настроить декларативные транзакции через @Transactional.

Продемонстрировать rollback при ошибке.

Привести пример различий propagation (например, REQUIRED и REQUIRES_NEW) или описать их поведение.

— Реализация веб-слоя

Разработать REST API с CRUD-операциями.

Использовать DTO и валидацию (@Valid, аннотации Bean Validation).

Реализовать централизованную обработку ошибок (@ControllerAdvice) с единым форматом ответа.

Обеспечить версионирование API (например, /api/v1/...).

— AOP и сквозная функциональность

Реализовать аспект логирования или аудита (например, логирование времени выполнения сервисных методов).

Продемонстрировать понимание проксирования и ограничений Spring AOP (описание в отчете).

— Интеграции

Реализовать вызов внешнего HTTP-сервиса через RestClient или WebClient.

Подключить OpenAPI (Swagger) и задокументировать API.

(Опционально) Реализовать отправку сообщения в брокер (RabbitMQ/Kafka) либо описать архитектурную интеграцию.

— Безопасность и эксплуатация

Настроить Spring Security (аутентификация, разграничение доступа по ролям).

Подключить Spring Boot Actuator (health, info, метрики).

Подготовить Dockerfile для сборки образа приложения.

Описать (текстово) предполагаемый процесс деплоя в Kubernetes.

— Тестирование

Реализовать модульные тесты сервисного слоя.

Реализовать интеграционные тесты (например, с использованием SpringBootTest и тестовой БД).

Продемонстрировать тестируемость архитектуры.

Критерии оценивания проекта

1. Архитектурная корректность и применение принципов DI/DIP — логичность слоистой структуры, слабая связанность компонентов, корректное использование внедрения зависимостей.
2. Полнота и корректность использования технологий Spring — обоснованное применение Spring Boot, Data JPA, транзакций, AOP, валидации, безопасности и других изученных механизмов.
3. Качество реализации и обработки ошибок — единая модель ошибок, корректная работа транзакций, безопасность и устойчивость к некорректным данным.
4. Понимание внутренних механизмов — способность объяснить проксирование, жизненный цикл бинов, работу транзакций, fetching и другие ключевые аспекты.
5. Готовность к эксплуатации и качество оформления — наличие документации (README, OpenAPI), Dockerfile, понятная структура проекта, читаемость кода и наличие тестов.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Как называется принцип, при котором управление созданием и связыванием объектов передается контейнеру Spring?	IoC / Inversion of Control	УК-1
2	Как называется принцип, согласно которому модули верхнего уровня не зависят от модулей нижнего уровня, а зависят от абстракций?	DIP / Dependency Inversion Principle	УК-1
3	Сколько основных уровней изоляции транзакций определено в стандарте SQL?	4	УК-2
4	Как называется аннотация Spring, позволяющая задать условную загрузку бина?	@Conditional	УК-2
5	Сколько основных типов внедрения зависимостей используется в Spring?	3	УК-3
6	Как одним словом называется совместная разработка кода несколькими разработчиками с использованием общей системы версий?	коллаборация / collaboration	УК-3
7	Как называется математическая структура, лежащая в основе реляционной модели данных?	отношение	ОПК-1
8	Сколько нормальных форм обычно рассматривается в базовом курсе проектирования БД?	3	ОПК-1
9	Как называется алгоритмическая проблема избыточных запросов при использовании ORM и ленивой загрузки связей?	N+1	ОПК-4
10	Как называется механизм преобразования типов в Spring, используемый через ConversionService?	конвертация / преобразование типов	ОПК-4

11	Как называется модуль Spring Boot, предоставляющий production-ready метрики и health-check?	Actuator	ОПК-5
12	Как называется спецификация, используемая для документирования REST API и поддерживаемая в Spring через springdoc-openapi?	OpenAPI / Swagger	ОПК-5
13	Как называется декларативный механизм управления транзакциями в Spring через аннотацию?	@Transactional	ОПК-6
14	Сколько типов прокси использует Spring AOP?	2	ОПК-6
15	Как называется аннотация, обеспечивающая перевод исключений доступа к данным в иерархию DataAccessException?	@Repository	ПК-2
16	Как называется механизм аутентификации и авторизации в Spring Boot приложении?	Spring Security	ПК-2
17	Как называется шаблон проектирования, реализующий разделение Controller–Service–Repository?	слоистая архитектура / layered architecture	ПК-3
18	Сколько основных стадий жизненного цикла бина включает инициализация и уничтожение?	2	ПК-3
19	Как называется основной интерфейс контекста Spring, расширяющий BeanFactory?	ApplicationContext	ПК-4
20	Как называется технология взаимодействия клиента и сервера в режиме постоянного двустороннего соединения, поддерживаемая в Spring?	WebSocket	ПК-4