

**УТВЕРЖДЕНА**

Решением Ученого совета  
АНО ВО «Центральный университет»  
«24» июня 2025 г.  
Протокол №2

**Рабочая программа дисциплины (модуля)  
«Разработка backend-сервисов»**

**Направление подготовки:** 02.03.01 Математика и компьютерные науки

**Направленность (профиль) подготовки:** Искусственный интеллект

**Квалификация (степень) выпускника:** бакалавр

**Форма обучения:** очная

**Срок освоения программы:** 4 года

**Год набора:** 2025

**Москва  
2025**

## Содержание

<b>1. Краткая характеристика дисциплины (модуля)</b> .....	<b>3</b>
<b>2. Перечень планируемых результатов обучения</b> .....	<b>4</b>
<b>3. Тематический план</b> .....	<b>6</b>
<b>4. Содержание дисциплины (модуля)</b> .....	<b>6</b>
<b>5. Учебно-методическое обеспечение</b> .....	<b>7</b>
<b>6. Материально-техническое обеспечение</b> .....	<b>7</b>
<b>7. Методические и оценочные материалы</b> .....	<b>9</b>

## 1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка backend-сервисов» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Искусственный интеллект, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Разработка backend-сервисов» позволяет создавать надежные и масштабируемые серверные приложения, обеспечивающие эффективное взаимодействие между клиентом и сервером. Это фундаментальный навык для разработки современных веб-приложений и сервисов, поддерживающих бизнес-логику и обработку данных.

### **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 Искусственный интеллект и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 2, 3 или 4 курсе в 4, 5, 6, 7, 8 семестрах на выбор.

**Цель изучения дисциплины (модуля):** формирование навыков проектирования, создания и поддержки серверной части приложений для обеспечения надежного и эффективного обмена данными и бизнес-логики.

### **Задачи изучения дисциплины (модуля):**

- изучить архитектурные принципы backend-разработки, включая RESTful API, микросервисы и паттерны проектирования для создания надежных систем;
- освоить инструменты и технологии для работы с базами данных, аутентификацией и обработкой запросов, такие как Node.js, Spring Boot;
- развить навыки тестирования, развертывания и мониторинга backend-сервисов для обеспечения их безопасности, производительности и отказоустойчивости.

### **В результате освоения дисциплины (модуля) обучающийся должен:**

#### ***знать:***

- фреймворки Spring и технологии вспомогательных проектов;

#### ***уметь:***

- создавать Web-приложения на микросервисной архитектуре;
- встраивать сервисы в существующую инфраструктуру;

#### ***владеть:***

- навыком решения высокоуровневых задач по разработке.

## 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 разработки, 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
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 реализации математических моделей в профессиональной деятельности
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 искусственного интеллекта
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных

			технологий программирования в области искусственного интеллекта
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований в области искусственного интеллекта
ПК-1.	Способен формулировать задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Обладает базовыми знаниями, полученными в области математических наук, программирования и информационных технологий
		ПК-1.2.	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике
		ПК-1.3.	Имеет опыт работы с задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач

### 3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Введение в Spring Framework	4	4		20	Кейс
2	Работа с БД	4	4		20	Кейс
3	Разработка Web-приложений	4	4		20	Кейс
4	Продвинутые возможности Spring	16	16		70	Кейс Коллоквиум
	<i>Зачет с оценкой</i>			4		Проект
	<b>Итого:</b>	<b>28</b>	<b>28</b>	<b>4</b>	<b>130</b>	
	<b>Объем дисциплины (модуля) (в ак. ч.)</b>	<b>190</b>				
	<b>Объем дисциплины (модуля) (в зач. ед.)</b>	<b>5</b>				

### 4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Введение в Spring Framework	Введение в Spring Framework и Spring Boot. Spring context, бины и Dependency Injection
2	Работа с БД	JDBC и Spring JDBC для доступа к данным. JPA, JOOQ, JDBI и миграции с Liquibase/Flyway
3	Разработка Web-приложений	REST API на Spring Web MVC. Межсервисная коммуникация: REST clients и gRPC
4	Продвинутые возможности Spring	Сообщения и JMS: ActiveMQ и RabbitMQ. Поточная обработка данных с Apache Kafka. Безопасность приложений на Spring Security. Мониторинг и Spring Actuator. Событийно-ориентированная архитектура и Event Listeners. Aspect-Oriented Programming и кросс-срезочные задачи. Шаблоны микросервисов и отказоустойчивость. Интеграция и деплой: контейнеры, CI/CD, облака. Производительность, кэширование (Redis) и профилирование

## 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. 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>.

2. 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>.

3. Спилкэ, Л. Spring быстро : практическое руководство / Л. Спилкэ. - Санкт-Петербург : Питер, 2023. - 448 с. - ISBN 978-5-4461-1969-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2122935>.

4. 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>.

### *Дополнительная литература:*

1. Полуэктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567610>.

2. Зараменских, Е. П. Разработка информационных систем : учебник и практикум для вузов / Е. П. Зараменских. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 78 с. — (Высшее образование). — ISBN 978-5-534-21420-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/571333>.

## 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	<a href="https://elibrary.ru/defaultx.asp">https://elibrary.ru/defaultx.asp</a>
2.	База данных для IT-специалистов	<a href="https://habr.com">https://habr.com</a>
3.	База данных ScienceDirect	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	<a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>
5.	Федеральный портал «Российское образование»	<a href="https://www.edu.ru/">https://www.edu.ru/</a>
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	<a href="http://window.edu.ru/">http://window.edu.ru/</a>
7.	Единая коллекция цифровых образовательных ресурсов	<a href="http://school-collection.edu.ru/">http://school-collection.edu.ru/</a>
8.	Федеральный центр информационно - образовательных ресурсов	<a href="http://fcior.edu.ru/">http://fcior.edu.ru/</a>

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
<b>Операционные системы:</b>		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
<b>Браузеры:</b>		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
<b>Офисные приложения:</b>		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
<b>Программное обеспечение для планирования и учета времени:</b>		
Toggle app	зарубежное	свободно распространяемое
<b>Системы управления проектами:</b>		
Microsoft Imagine (Project)	зарубежное	лицензионное
<b>Системы управления базами данных:</b>		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
<b>Системы резервного копирования (backup):</b>		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное

<b>Справочно-правовые системы:</b>		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
<b>Средства антивирусной защиты:</b>		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
<b>Среды разработки:</b>		
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
<b>Пакеты программных средств и библиотек:</b>		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
<b>Системы управления библиографической информацией:</b>		
Zotero	зарубежное	свободно распространяемое
<b>Сервисы и службы:</b>		
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 backend-сервисов» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, кейсы, коллоквиум, проект, 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 удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

*Участие в семинаре (аудиторная работа)* – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

*Кейс* – практическая работа студентов над реальными или смоделированными задачами, что позволяет студенту применять теоретические знания на практике.

Студент самостоятельно разрабатывает стратегию решения поставленной задачи, что

способствует развитию навыков критического мышления и самостоятельного принятия решений. Такой подход помогает подготовить будущих специалистов к реальным вызовам в их профессиональной деятельности.

*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 курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

*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 backend-сервисов»**

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 *зачета с оценкой*, 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	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	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
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			ответы, уверенно интерпретирует результаты 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	Зачтено	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	Зачтено	
5	Удовлетворительно	Зачтено	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	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Разработка backend-сервисов» оценивается следующим образом:

Активность	Вес	Описание
Кейс	40%	Практическая работа студентов над реальными или смоделированными задачами, что позволяет студенту применять теоретические знания на практике
Коллоквиумы	30%	Устные ответы на вопросы, список которых известен студенту заранее
Проект	30%	Защита итогового проекта

**Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка backend-сервисов»:** « $0,4 \times$  среднее за кейсы +  $0,3 \times$  среднее за коллоквиумы +  $0,3 \times$  проект».

### Текущий контроль успеваемости обучающихся по дисциплине (модулю)

#### Примерные задания для кейсов Введение в Spring Framework

1. Создайте новый проект на Spring Boot с использованием Spring Initializr, настройте базовый контекст приложения и зарегистрируйте простой бин (например, сервис для вывода приветствия) с аннотацией `@Component`, проверив его работу через тестовый метод.
2. Реализуйте Dependency Injection в Spring-приложении, создав интерфейс и две реализации (например, для отправки email и SMS), инжектировав их в контроллер с помощью `@Autowired` и протестировав переключение через конфигурацию.
3. Настройте Spring Context с использованием XML-конфигурации, определив бины для репозитория и сервиса, и продемонстрируйте их взаимодействие в консольном приложении без Spring Boot.
4. Разработайте приложение с несколькими профилями (dev и prod), где в каждом профиле инжектируется разная реализация бина (например, база данных в памяти vs. реальная), используя `@Profile` и проверив переключение через `application.properties`.
5. Создайте простой REST-контроллер в Spring Boot, инжектировав сервис через конструктор, и объясните преимущества Dependency Injection над прямым созданием объектов, протестировав endpoint с Postman.

#### Работа с БД

1. Реализуйте доступ к базе данных H2 с помощью JDBC в Spring-приложении, создав класс для выполнения простого SELECT-запроса и вывода результатов в консоль, используя DataSource из конфигурации.
2. Настройте Spring JDBC Template для выполнения CRUD-операций с базой данных PostgreSQL, создав DAO-класс для управления сущностью "Пользователь" (добавление, обновление, удаление и чтение записей).
3. Внедрите JPA с Hibernate в Spring Boot-проект, определив сущность и репозиторий для работы с базой данных MySQL, реализовав запросы с JPQL для поиска пользователей по критериям.
4. Используйте JOOQ для генерации кода из схемы базы данных и выполнения типизированных запросов, создав сервис для извлечения данных о заказах из таблицы и обработки их в приложении.
5. Настройте миграции базы данных с помощью Flyway в Spring-проекте, создав SQL-скрипты для создания таблиц и вставки тестовых данных, и протестируйте их применение при запуске приложения.

## Разработка Web-приложений

1. Разработайте REST API на Spring Web MVC с контроллером для управления ресурсами "Продукты", реализовав endpoints для GET, POST, PUT и DELETE с валидацией входных данных через @Valid.
2. Создайте клиентскую часть для REST-коммуникации между сервисами, используя RestTemplate в Spring Boot для отправки запросов к внешнему API (например, получение данных о погоде) и обработки ответов.
3. Настройте gRPC-сервис в Spring-приложении с помощью protobuf, определив сервис для обмена сообщениями между микросервисами (например, передача информации о заказе) и протестируйте его с клиентом.
4. Реализуйте асинхронную обработку запросов в REST API с использованием @Async, создав endpoint для отправки email-уведомлений и проверив неблокирующее поведение с помощью многопоточной нагрузки.
5. Интегрируйте WebSocket в Spring Web MVC для реального времени, создав чат-приложение с сервером, обрабатывающим сообщения от клиентов и транслирующим их другим подключенным пользователям.

## Продвинутые возможности Spring

1. Настройте JMS-сообщения с ActiveMQ в Spring-приложении, создав продюсера и консьюмера для отправки и обработки сообщений в очереди, симулируя асинхронную обработку заказов.
2. Реализуйте потоковую обработку данных с Apache Kafka в Spring Boot, настроив топик для публикации событий (например, логи действий пользователей) и консьюмера для их обработки в реальном времени.
3. Внедрите безопасность в приложение с помощью Spring Security, настроив аутентификацию по JWT-токенам для REST API, включая роли пользователей и защиту endpoints от несанкционированного доступа.
4. Используйте Spring Actuator для мониторинга приложения, добавив endpoints для health-check, метрик и логов, и протестируйте их в браузере или с помощью инструментов вроде Prometheus.
5. Настройте кэширование с Redis в Spring-приложении, аннотируйте методы сервиса @Cacheable для ускорения запросов к базе данных, и измерьте производительность с помощью JMeter перед и после внедрения.

## Примерные задания для коллоквиума

№ п/п	Задание	Ответ
1	Укажите основную аннотацию в Spring Security для защиты методов контроллеров.	@PreAuthorize
2	Укажите название механизма аутентификации в Spring Security, основанного на токенах.	JWT
3	Укажите количество стандартных ролей в Spring Security (например, USER, ADMIN).	2
4	Укажите название эндпоинта в Spring Actuator для проверки здоровья приложения.	/actuator/health

№ п/п	Задание	Ответ
5	Укажите основной инструмент в Spring Actuator для мониторинга метрик приложений.	Micrometer
6	Укажите название свойства для включения всех эндпоинтов Actuator в конфигурации.	management.endpoints.web.exposure.include=*
7	Укажите основной интерфейс для создания слушателей событий в Spring.	ApplicationListener
8	Укажите название аннотации для публикации событий в событийно-ориентированной архитектуре.	@EventListener
9	Укажите количество основных компонентов в модели публикации-подписки (например, publisher, subscriber).	2
10	Укажите название аннотации для создания аспектов в Aspect-Oriented Programming.	@Aspect
11	Укажите основной тип совета (advice) в AOP для выполнения кода после метода.	@AfterReturning
12	Укажите название понятия в AOP для кросс-срезочных задач, таких как логирование.	Concern
13	Укажите название шаблона микросервисов для обработки сбоев и повторных попыток.	Circuit Breaker
14	Укажите основной инструмент для реализации отказоустойчивости в микросервисах.	Resilience4j
15	Укажите название паттерна для разделения сервисов на домены в микросервисах.	Bounded Context
16	Укажите название инструмента для контейнеризации приложений в интеграции и деплое.	Docker
17	Укажите основной этап в CI/CD для автоматического тестирования кода.	Build
18	Укажите название облачной платформы для деплоя Spring-приложений (например, AWS, Azure).	AWS
19	Укажите название инструмента для кэширования данных в Redis в Spring.	@Cacheable
20	Укажите основной инструмент для профилирования производительности Java-приложений.	VisualVM

### Примерное описание и критерии оценивания к проекту

#### Описание:

Разработайте backend-приложение на Spring Boot с REST API для управления предметной областью (например, система управления библиотекой, заказами или блогот). Приложение должно включать:

- Конфигурацию и использование Spring Beans с внедрением зависимостей (IoC).

- Подключение и работу с реляционной базой данных через JPA и Spring Data.
- Реализацию основных CRUD-операций через REST-контроллеры с корректной обработкой HTTP-запросов и ошибок.
- Безопасность приложения: аутентификацию и авторизацию пользователей с разграничением доступа по ролям.
- Кэширование наиболее часто запрашиваемых данных для повышения производительности.
- Логирование вызовов сервисных методов с помощью аспектно-ориентированного программирования (AOP).
- Управление транзакциями при работе с базой данных для обеспечения целостности данных.
- Интеграцию с внешним REST-сервисом (например, получение актуальной информации или данных для расширения функционала).
- Реализацию WebSocket для отправки уведомлений пользователям в реальном времени.
- Наличие набора юнит- и интеграционных тестов, покрывающих ключевые компоненты приложения.
- Документацию проекта с описанием функционала и инструкциями по запуску.

### **Критерии оценивания:**

#### **1. Функциональность и полнота реализации**

Проект должен полностью соответствовать техническому заданию: реализованы все ключевые функции (CRUD, безопасность, интеграция, WebSocket и т.д.). Недостающие или частично реализованные функции снижают оценку.

#### **2. Качество архитектуры и использование Spring**

Корректное применение принципов IoC и DI, правильное разделение слоёв (контроллеры, сервисы, репозитории), использование Spring Boot и сопутствующих технологий согласно best practices.

#### **3. Работа с базой данных и управление транзакциями**

Корректное моделирование данных с помощью JPA, эффективное использование Spring Data, правильная настройка и применение транзакций для предотвращения ошибок и потери данных.

#### **4. Безопасность приложения**

Надёжная аутентификация и авторизация с разграничением прав доступа, обработка ошибок безопасности, предотвращение типичных уязвимостей.

#### **5. Производительность и расширяемость**

Использование кэширования для оптимизации, логирование через AOP для удобства поддержки и диагностики, а также гибкая архитектура, позволяющая легко расширять функционал.

#### **6. Интеграция и коммуникация**

Корректная и надёжная интеграция с внешними REST-сервисами, обработка ошибок при взаимодействии, а также работа WebSocket для обмена сообщениями в реальном времени.

#### **7. Тестирование и документация**

Наличие покрывающих тестов (юнит и интеграционных) с хорошим покрытием ключевых компонентов, понятная и полная документация, позволяющая запустить и использовать приложение без дополнительных пояснений.

### **Задания для промежуточной аттестации по дисциплине (модулю)**

<b>№ п/п</b>	<b>Задание</b>	<b>Ответ</b>	<b>Компетенция</b>
1	Укажите основной контейнер для управления компонентами в Spring Framework.	ApplicationContext	УК-1
2	Укажите название аннотации для внедрения	@Autowired	УК-1

	зависимостей в Spring.		
3	Укажите количество основных модулей в тематическом планировании курса "Разработка backend-сервисов".	4	УК-1
4	Укажите название инструмента для миграции баз данных в Spring (например, Liquibase или Flyway).	Liquibase/Flyway	УК-1
5	Укажите основной протокол для межсервисной коммуникации в REST API.	HTTP	УК-1
6	Укажите название математической концепции, используемой для оптимизации запросов к базам данных (например, теория множеств).	Теория множеств	ОПК-1
7	Укажите количество основных операций в дискретной математике, применяемых для моделирования связей в JPA (например, транзитивное замыкание).	1	ОПК-1
8	Укажите вероятность события в теории вероятностей, используемой для оценки отказоустойчивости микросервисов (например, вероятность сбоя).	0.1	ОПК-1
9	Укажите название численного метода, применяемого для аппроксимации данных в профилировании производительности (например, метод наименьших квадратов).	Метод наименьших квадратов	ОПК-1
10	Укажите количество шагов в алгоритме сортировки данных при работе с JDBC (например, быстрая сортировка).	3	ОПК-1
11	Укажите название алгоритма для обработки сообщений в Apache Kafka.	Consumer Group	ОПК-6
12	Укажите основной инструмент для тестирования REST API в Spring.	Spring Test	ОПК-6
13	Укажите название паттерна архитектуры для микросервисов (например, Circuit Breaker).	Circuit Breaker	ОПК-6
14	Укажите количество основных типов советов в Aspect-Oriented Programming (например, Before, After).	5	ОПК-6
15	Укажите название метода для кэширования данных в Redis в Spring.	@Cacheable	ОПК-6
16	Укажите основной язык для формулировки задач в области компьютерных наук.	Алгоритмический	ПК-1
17	Укажите название структуры данных, используемой для хранения сессий в web-приложениях.	Map	ПК-1
18	Укажите количество шагов в стандартном алгоритме решения задачи аутентификации (например, в Spring Security).	4	ПК-1
19	Укажите название метода анализа результатов в математической статистике для оценки производительности приложения.	Регрессия	ПК-1
20	Укажите основной инструмент для модульного тестирования в Spring.	JUnit	ПК-1