

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: Разработка

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2025

**Москва
2025**

Содержание

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

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 Разработка, 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 Разработка и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 3 или 4 курсе в 5, 6, 7 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-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-6. | Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ОПК-6.1. | Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики |
| | | ОПК-6.2. | Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования |
| | | ОПК-6.3. | Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований |
| ПК-3. | Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач | ПК-3.1. | Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач |
| | | ПК-3.2. | Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты |
| | | ПК-3.3. | Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях |

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 |
| | Итого: | 28 | 28 | 4 | 130 | |
| | 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 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. Поточковая 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 | 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: | | |
| 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 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)

| № п/п | Задание | Ответ | Компетенция |
|-------|---|--------------------------|-------------|
| 1 | Укажите термин для метода поиска информации о Spring Framework в области разработки. | Ключевое слово | УК-1 |
| 2 | Укажите принцип критической оценки источников по Dependency Injection в Spring. | Релевантность | УК-1 |
| 3 | Укажите метод синтеза данных из источников о JDBC для работы с базами данных. | Анализ релевантности | УК-1 |
| 4 | Укажите системный подход к структурированию задач по REST API. | Модульное проектирование | УК-1 |
| 5 | Укажите инструмент для обработки информации о микросервисах. | Анализатор зависимостей | УК-1 |
| 6 | Укажите концепцию из дискретной математики для моделирования связей в JPA. | Теория графов | ОПК-1 |
| 7 | Укажите применение теории вероятностей в анализе отказоустойчивости систем. | Надёжность | ОПК-1 |
| 8 | Укажите опыт разработки математической модели для миграций баз данных. | Схема зависимостей | ОПК-1 |
| 9 | Укажите математический алгоритм для сортировки данных в JDBC. | Быстрая сортировка | ОПК-4 |
| 10 | Укажите метод реализации алгоритма поиска в базах данных с JPA. | Индексация | ОПК-4 |
| 11 | Укажите практический опыт применения алгоритма кэширования в Redis. | LRU | ОПК-4 |
| 12 | Укажите алгоритм для разработки REST-клиентов в Spring. | HTTP-запрос | ОПК-6 |
| 13 | Укажите метод разработки программ для gRPC-коммуникации. | Протокол буферов | ОПК-6 |
| 14 | Укажите опыт создания систем визуализации для мониторинга Actuator. | Графический дашборд | ОПК-6 |
| 15 | Укажите алгоритм для обработки сообщений в RabbitMQ. | Очередь | ОПК-6 |
| 16 | Укажите метод математического моделирования потоков данных в Kafka. | Граф потоков | ПК-3 |
| 17 | Укажите алгоритмическое моделирование процесса деплой в контейнерах. | CI/CD-пайплайн | ПК-3 |
| 18 | Укажите практический опыт применения моделирования в профилировании производительности. | Бенчмаркинг | ПК-3 |
| 19 | Укажите модель для анализа безопасности в Spring Security. | Ролевая модель | ПК-3 |
| 20 | Укажите алгоритм для интеграции AOP в микросервисы. | Аспект-ориентированный | ПК-3 |