

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Java Spring (Разработка веб-приложений на Java с использованием Spring)»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Backend-разработка

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2024

**Москва
2024**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	4
3. Тематический план.....	5
4. Содержание дисциплины (модуля).....	5
5. Учебно-методическое обеспечение	6
6. Материально-техническое обеспечение	6
7. Методические и оценочные материалы	8

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Java Spring (Разработка веб-приложений на Java с использованием Spring)» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по специальности 02.04.01 Математика и компьютерные науки, профиль Backend-разработка, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Java Spring (Разработка веб-приложений на Java с использованием Spring)» имеет важное значение для студентов, поскольку оно позволяет им создавать сложные веб-приложения с высокими требованиями к производительности, масштабируемости и безопасности. Кроме того, знания Spring и Java являются одними из наиболее востребованных навыков на рынке труда, что делает их изучение перспективным и актуальным для будущей карьеры в области разработки веб-приложений.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Backend-разработка и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе во 2 семестре.

Цель изучения дисциплины (модуля): формирование у студентов навыков и знаний по разработке веб-приложений на Java с использованием фреймворка Spring, работа с базами данных, управление зависимостями и конфигурацией приложения.

Задачи изучения дисциплины (модуля):

- формирование знания основных возможностей фреймворка Spring и технологии вспомогательных проектов;
- формирование знания принципов внутреннего устройства виртуальной машины Java и сборщика мусора;
- формирование знания классических паттернов объектно-ориентированного программирования и их реализацию в Java;
- формирование умения реализовывать взаимодействие с сетью, с базами данных и с очередями сообщений на языке Java;
- формирование умения решать задачи промышленной разработки, включая проектирование архитектуры и интеграцию сложных систем;
- формирование навыка решать высокоуровневые задачи по разработке на языке Java.

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)
ОПК-3.	Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства	ОПК-3.1.	Знает основные принципы программирования, архитектуры программного обеспечения и современные языки программирования, а также особенности отечественных информационных технологий и сетевых ресурсов
		ОПК-3.2.	Умеет разрабатывать прикладные программные средства, используя современные инструменты и технологии, а также интегрировать их с сетевыми ресурсами для решения конкретных задач
		ОПК-3.3.	Имеет практический опыт разработки программных средств, используемых при построении математических моделей в естественных науках
ПК-6.	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере профессиональной деятельности	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в конкретной предметной области
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контр оль	Самосто ятельна я работа	
Лекции	Семинар (Практическ ое занятие)					
1	Основы многопоточности и работы с сетью	5	5		27	Домашнее задание
2	Введение в Spring	5	5		27	Домашнее задание
3	Взаимодействие с сетью	5	5		27	Домашнее задание
4	Интеграция с базами данных	5	5		28	Домашнее задание
5	Взаимодействие с очередями сообщений	5	5		28	Домашнее задание
6	Паттерны проектирования	5	5		27	Домашнее задание
	<i>Экзамен</i>			<i>4</i>		
	Итого:	30	30	4	164	
	Объем дисциплины (модуля) (в ак. ч.)	228				
	Объем дисциплины (модуля) (в зач. ед.)	6				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы многопоточности и работы с сетью	Клиент-серверная архитектура. Реализация простейшего сервера. Поток выполнения (threads). Многопоточный сервер. Thread pool и повторное использование потоков
2	Введение в Spring	Dependency Injection. Понятие фреймворка, фреймворк Spring. Внутреннее устройство Spring. Spring context. Понятие бина, типы бинов
3	Взаимодействие с сетью	Spring Web. REST controller. Spring Boot. Основные протоколы интеграции по сети: REST client, gRPC (protobuf)
4	Интеграция с базами данных	JDBC, Spring JDBC. JOOQ, JDBI, Spring JPA. Liquibase и Flyway. Интеграция с redis
5	Взаимодействие с очередями сообщений	JMS, Active MQ, RabbitMQ. Kafka
6	Паттерны проектирования	Паттерны объектно-ориентированного программирования. Примеры реализации паттернов ООП в языке Java

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. Кубенский, А. А. Функциональное программирование : учебник и практикум для вузов / А. А. Кубенский. — Москва : Издательство Юрайт, 2025. — 348 с. — (Высшее образование). — ISBN 978-5-9916-9242-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561074>.

2. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561176>.

Дополнительная литература:

1. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>.

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;

— 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 к 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	Катастрофы, стихийные бедствия, аварии, эпидемии. Солнечная и геомагнитная активность. /ежедневный обзор	http://www.disasters.chat.ru
2	Каталог по безопасности жизнедеятельности	http://www.eun.chat.ru
3	Научная электронная библиотека eLibrary.ru библиотека	https://elibrary.ru/defaultx.asp
4	База данных для IT-специалистов	https://habr.com
5	База данных ScienceDirect	https://www.sciencedirect.com
6	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
7	Федеральный портал «Российское образование»	https://www.edu.ru/
8	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
9	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
10	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/
11	Сайт различных плагинов	https://maven.apache.org/plugins/
12	Maven central repository - хранилище библиотек и фреймворков	https://mvnrepository.com/repos/central

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
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) «Java Spring (Разработка веб-приложений на Java с использованием Spring)» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, 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 удастся, 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.

Домашнее задание – набор заданий 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.

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) «Java Spring (Разработка веб-приложений на Java с использованием Spring)».

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

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 (модуль). 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 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.
9	Отлично	
8	Отлично	
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
4	Удовлетворительно	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		только самые простые задачи и владеет лишь минимальным набором методов исследования.
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) «Java Spring (Разработка веб-приложений на Java с использованием Spring)» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	70%	Оцениваются по критериям. Можно набрать максимум 10 баллов за каждое из заданий.
Экзамен	30%	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Java Spring (Разработка веб-приложений на Java с использованием Spring)»: « $0,7 \times$ среднее за домашние задания + $0,3 \times$ экзамен».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание

Чат

Напишите два консольных приложения: сервер и клиент, реализующие простой чат.

Условие

Серверное приложение в аргументах требует порт, через который будет происходить взаимодействие с клиентами по протоколу TCP. Клиентское -- адрес и порт сервера, имя клиента.

Клиент после подключения печатает в стандартный вывод сообщения о подключении и отключении к чату других клиентов, новые сообщения этих клиентов, а также ожидает в стандартном вводе сообщения пользователя. После получения сразу отправляет его на сервер.

Формат новых сообщений в терминале: `[mm:ss.SSS] client_name: message`, каждое с новой строки, где `mm:ss.SSS` -- серверное время сообщения (минуты, секунды, миллисекунды), `client_name` -- имя отправителя, `message` -- его сообщение.

Формат сообщений о подключении и отключении:

- `client_name joined the chat`
- `client_name left the chat`

Поддержите только однострочные сообщения. Уникальность имен клиентов поддерживать не нужно, могут совпадать.

Поддержка нескольких чатов

Поддержите возможность подключения к разным чатам. Четвертым опциональным аргументом в параметрах клиентского приложения можно передать номер чата -- положительное число. Если клиент подключается к чату `n`, то только клиентам этого чата отправляются сообщения о новом клиенте и сами сообщения клиентов. Если в параметрах приложения не указан номер чата, это означает, что клиент подключается к чату по умолчанию. Чат по умолчанию -- это отдельный чат, сообщения в котором видны только клиентам этого чата.

Смену чата у запущенного клиента поддерживать не нужно.

Критерии оценивания

- 4 балла: реализован сервер
- +3 балла: реализован клиент
- +3 балла: поддержана возможность подключения к разным чатам

Если вы реализуете только сервер, добавьте файл `README.md`, в котором укажите формат взаимодействия сервера и клиента.

Формат решения

Мультимодульный `gradle` проект, расположенный в директории `week_03/hometask`. Внутри два модуля: `server`, `client`, `gradle wrapper`, `settings.gradle` (или `settings.gradle.kts`) с подключение дочерних модулей.

В зависимостях проекта не должно быть сторонних библиотек кроме тестовых (написание самих тестов на ваше усмотрение).

Для поддержки запуска приложения через `gradle` в корневых модулях подключите плагин `application` https://docs.gradle.org/current/userguide/application_plugin.html.

В результате `cd week_03/hometask && ./gradlew server:run --args='port'` должно запуститься серверное приложение, слушающее порт `port`.

В результате `cd week_03/hometask && ./gradlew client:run --args='host:port client_name chat_id'` должно запуститься клиентское приложение, которое будет подключаться к серверу с адресом `host`, портом `port`. После успешного подключения будут приходить сообщения участников чата `chat_id`. У других участников этого чата сообщения клиента будут отображаться с именем `client_name`.

Домашнее задание

Spring Cached Proxy

Напишите библиотеку для кэширования результатов вызова методов бина.

Требования к решению

Ваша библиотека должна:

- Обеспечивать автоматическое кэширование в оперативной памяти результатов методов, аннотированных `@Cached`.
- При первом вызове метода выполнять реальный вызов.
- При последующих вызовах с теми же аргументами (равными по `Object#equals`) возвращать ранее сохранённый результат.

Обязательные компоненты

1. Аннотация `hometask.cached.annotations.Cached`
 - Поддерживает необязательные параметры:
 - `long ttl` — время жизни кэша (по умолчанию бесконечно).
 - `java.time.temporal.ChronoUnit unit` — единица измерения TTL (по умолчанию `ChronoUnit.SECONDS`).
2. Конфигурационный класс `hometask.cached.configuration.CachedConfiguration`

○ При подключении с помощью `@Import(CachedConfiguration.class)` в Spring Context должен автоматически включать механизм кэширования.

Ограничения

- Аннотация `@Cached` будет применяться только к публичным нефинальным методам с возвращаемым значением (не `void`).
- В одном бине может быть несколько `@Cached` методов. Считайте, что `@Cached` методы одного бина не вызывают друг друга.

Формат решения

- Gradle-модуль `hometask.spring:spring-cached:1.0` в директории `week_06/hometask/spring_cached`.
- MR в своем приватном форке с названием `[hometask] cached`
- Для создания прокси-классов используйте `org.springframework.cglib.proxy.Enhancer` и `org.springframework.cglib.proxy.MethodInterceptor`.

Дополнительные задачи

1. Реализация TTL (время жизни кэша) [+3 балла]

Добавьте возможность автоматического удаления устаревших кэшированных данных после заданного времени:

- При очередном вызове метода результат должен браться из кэша, если срок его хранения (TTL) не истёк, иначе должен происходить вызов реального метода с последующим сохранением в кэш.

2. Реализация @CacheEvict [+2 балла]

Добавьте аннотацию `hometask.cached.annotations.CacheEvict` и ее поддержку:

- Перед вызовом метода с `@CacheEvict` необходимо очистить кэш для всех `@Cached` методов в этом же бине.
- Считайте, что аннотация будет применяться только к публичным нефинальным `void` методам без параметров.

3. Покрыть код контекстными тестами

Напишите интеграционные тесты с поднятием Spring-контекста, проверяющие, что кэширование работает корректно

- Добавьте минимум 2 теста на основной функционал `@Cached` и минимум по 2 на дополнительные задачи при условии их реализации.

Критерии оценивания

- 4 баллов — реализованы `@Cached` и `CachedConfiguration`.
- +2 балла — реализована логика TTL.
- +2 балла — реализованы `@CacheEvict` и его поддержка.
- +2 балла — код покрыт контекстными тестами

Домашнее задание

Сервис для организации встреч

Разработайте REST API для управления встречами между пользователями и реализуйте его с помощью Spring Boot приложения. Все данные должны храниться в памяти в подходящих коллекциях.

Требования к API

API должно соответствовать принципам REST:

- Используйте HTTP-методы по назначению
- Возвращайте корректные HTTP-коды в ответах
- Используйте понятные URL-энпоинты
- Работайте с JSON в запросах и ответах.
- Подробнее: <https://restfulapi.net/resource-naming>

Создание встречи

- Встреча имеет название, организатора, список участников, дату и время начала, дата и время окончания, продолжительность.
- Поддержите формат [ISO-8601](#) для даты и времени с оффсетом (2020-01-02T03:04:05Z, 2020-01-02T03:04:05+06:00). Используйте классы из пакета `java.time`.
- Единственная информация про пользователей, с которой вы работаете, — это их `id`. Сами пользователи в приложении не хранятся и никак не валидируются.
- Поддержите валидацию параметров:
 - нельзя одновременно задать дату и время окончания встречи и ее продолжительность
 - дата и время окончания должны быть позже начала
 - встреча не должна пересекаться по времени с уже существующими встречами любого из участников
 - нельзя создавать встречи в прошлом
 - нельзя в списке участников передать организатора
- Если встреча нарушает одно из этих правил, приложение возвращает ошибку с подходящим HTTP кодом.

Получение полной информации по встрече

- в ответе должны быть название, организатор, участники, дата и время начала, дата и время окончания.

Получение списка встреч

- Можно получить список всех встреч пользователя.
- Можно получить список всех встреч пользователя, где он является организатором.
- Можно получить список всех встреч пользователя за указанный период.
- Можно получить только предстоящие встречи пользователя (начиная с текущего момента).
- Поддержите возможность использования любых вышеперечисленных комбинаций, кроме противоречащих (предстоящие встречи, но за прошедшие даты). В этом случае приложение должно возвращать ошибку с подходящим HTTP кодом.
- Во всех случаях возвращается полная информация о встречах.

Обновление встречи

- Можно изменить название встречи.
- Можно изменить дату и время встречи и ее продолжительность, если новый таймслот не конфликтует с другими встречами участников.
- Можно изменить список участников, если у новых в это время нет других встреч.
- Можно в одном запросе изменять любые вышеперечисленные комбинации.
- Встречу можно изменить, если она еще не началась.
- Организатора менять нельзя.
- При попытке некорректного изменения встречи приложение должно возвращать ошибку с подходящим HTTP кодом.

Удаление встречи

- Если удаляется встреча, она исчезает у организатора и всех участников.
- Встречу можно удалить, если она еще не началась, иначе приложение должно возвращать подходящую ошибку.

Дополнительные задачи

1. Поддержка `properties` (+2 балла)

Поддержите возможность задавать параметры приложения:

- максимальное количество встреч для одного пользователя в день.
- максимальное количество участников в одной встрече.
- минимальная продолжительность встречи в минутах.
- максимальная продолжительность встречи в минутах.

Добавьте файл `application.properties/application.yaml` с настройками по умолчанию.

Параметры должны загружаться в класс `@ConfigurationProperties`.

Если вы не реализуете этот пункт, считайте, что у приложения нет перечисленных выше ограничений.

2. Тестирование (+3 балла)

Протестируйте основные сценарии с помощью Spring Boot Test, JUnit и MockMvc.

Минимально должны быть покрыты тестами:

- успешное создание встречи
- получение полной информации по встрече
- получение списка встреч с разными комбинациями параметров
- обновление встречи разными вариантами
- успешное удаление
- все ошибочные сценарии
- нарушение ограничений из предыдущего пункта при условии поддержки `properties`

3. Логирование (+1 балла)

- Покройте логами `info` ключевые действия (создание, обновление, удаление).
- Добавьте логи `error` при ошибочных сценариях.
- Используйте SLF4J и стандартную имплементацию логгера Spring Boot.

Формат решения

- Spring Boot Application в директории `week_08/hometask/meeting`. В результате `cd week_08/hometask/meeting && ./gradlew bootRun` должно запуститься приложение, способное принимать HTTP запросы по порту 8088
- MR в своем приватном форке с названием `[hometask] meeting`

Критерии оценивания

- 4 балла: реализация API
- +2 балла: поддержка `properties`
- +3 балла: тестирование
- +1 балл: покрытие логами

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Выберите верное утверждение о Spring IoC контейнере. А) Контейнер управляет созданием и жизненным циклом Java-объектов в приложении В) IoC контейнер работает только с классами, явно объявленными как <code>@Bean</code> С) Контейнер создает бины только по запросу во время выполнения D) Контейнер не поддерживает внедрение зависимостей через конструктор.	А	ПК-6
2.	Какой аннотацией помечается класс, содержащий настройки Spring-приложения? Напишите название аннотации без пакета и символа <code>`@`</code> .	Configuration / configuration	ОПК-3
3.	Соотнесите аннотации и их назначение. Ответ напишите в виде пар цифра-буква без пробелов и других символов, например, 1A2B3C4D 1) <code>@Component</code> 2) <code>@Value</code> 3) <code>@Qualifier</code> 4) <code>@PostConstruct</code>	1C2A3B4D / 1c2a3b4d	ОПК-3

	<p>A) внедрение значения из конфигурации B) уточнение, какой именно бин внедрять C) пометка класса как управляемого Spring-бина D) метод вызывается после создания и инициализации бина</p>		
4.	<p>Что произойдет, если метод Spring-бина, помеченного @Transactional, выбросит unchecked-исключение?</p> <p>A) Транзакция зафиксируется B) Транзакция откатится C) Поведение зависит от настроек уровня изоляции D) Контейнер проигнорирует исключение</p>	В	ОПК-3
5.	<p>Какая аннотация в Spring используется для автоматического внедрения зависимостей в поля, методы или конструкторы класса? Напишите название аннотации без пакета и символа '@'.</p>	Autowired / autowired	ОПК-3
6.	<p>Упорядочите этапы обработки HTTP-запроса в Spring MVC. Напишите ответ в виде последовательности цифр, например 1234.</p> <p>1) Выполняется метод контроллера и формируется ответ 2) Запрос поступает в DispatcherServlet 3) Выбирается подходящий контроллер и метод 4) Возвращается HTTP-ответ клиенту</p>	2314	ОПК-3
7.	<p>Какая аннотация в Spring Boot объединяет в себе @Configuration, @EnableAutoConfiguration и @ComponentScan, позволяя быстро настроить и запустить приложение? Напишите название аннотации без пакета и символа '@'.</p>	SpringBootApplication	ПК-6
8.	<p>Какая аннотация в Spring используется для пометки метода, который будет автоматически вызываться при получении сообщения из очереди, работающей по технологии JMS? Напишите название аннотации без пакета и символа '@'.</p>	JmsListener	ПК-6
9.	<p>Какой интерфейс в Spring позволяет выполнять дополнительную обработку бинов до и после их инициализации? Напишите название интерфейса без пакета.</p>	BeanPostProcessor	ПК-6
10.	<p>Какие из следующих утверждений о тестировании Spring Boot-приложений являются верными?</p> <p>A) @SpringBootTest поднимает весь контекст приложения B) @WebMvcTest загружает только контроллеры и связанные компоненты C) @MockBean заменяет бин в контексте мок-версией D) В @SpringBootTest нельзя использовать TestContainers</p>	А, В, С/abc/ABC	ПК-6
11.	<p>Какое ключевое слово в Java используется для того, чтобы ограничить одновременный доступ нескольких потоков к критической секции кода или объекту?</p>	synchronized	ПК-6

12.	Какой аннотацией помечается класс, обрабатывающий глобальные ошибки в Spring MVC? <i>Напишите название аннотации без пакета и символа '@`.</i>	ControllerAdvice	ПК-6
13.	Какой класс в Java используется для создания и управления потоками выполнения в многопоточных программах? <i>Напишите название класса без пакета.</i>	Thread / thread	ОПК-3