

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Вычислительная и прикладная математика»

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Распределенные системы обработки информации»

Направление подготовки – 09.04.04 Программная инженерия

ОПОП академической магистратуры

«Программно-алгоритмическое обеспечение систем искусственного интеллекта»

Квалификация (степень) выпускника – магистр

Форма обучения – очная (2 года)

Рязань 2022 г.

1. СПИСОК ТЕОРЕТИЧЕСКИХ ВОПРОСОВ К ЗАЧЕТУ/ЭКЗАМЕНУ

1. РСОИ и какие задачи они решают? Определение РСОИ и следствия из определения.
2. Требования ко РСОИ.
3. Монолитное приложение, SOA и микросервисы. Для чего используется каждый подход.
4. Микросервисная архитектура.
5. Протоколы сериализации.
6. Протокол HTTP. На каком уровне работает, какие плюсы и минусы. Методы HTTP, статусы и заголовки.
7. Сессия. Что такое Cookie и для чего они используются. Stateful и Stateless.
8. Угрозы в сети. DDoS, XSS, CSRF, Man-In-The-Middle. Что такое CORS?
9. HTTPS.
10. Авторизация и аутентификация. OAuth, OAuth 2, OpenID Connect.
11. Что такое Front-End и какие задачи он выполняет? Масштабирование Front-end'ов.
12. Виды кэширования (code, проху, browser). Как работает кэш в браузере.
13. Что такое RESTful, принципы его использования.
14. Заявки, жизненный цикл заявки. Идентификаторы заявок. Обработка таймаутов.
15. Проблема синхронизации в распределенных системах.
16. Асинхронное взаимодействие сервисов.
17. Некорректное выполнение операций. Повторная попытка, отмена операций, ХА-транзакции.
18. Теорема CAP.
19. Транзакции. ACID. Проблемы транзакций. Оптимистическая и пессимистическая блокировка.
20. Write Ahead Log. Связь с транзакциями.
21. Что такое NoSQL, когда имеет смысл их применять. Типы NoSQL-баз данных.
22. Что такое BASE. Что такое eventual consistency, почему этот подход применим.
23. Map-Reduce.
24. Хранение данных RAID. Репликация.
25. Шардирование.

2. ПЕРЕЧЕНЬ ЛАБОРАТОРНЫХ/ПРАКТИЧЕСКИХ РАБОТ ПО ДИСЦИПЛИНЕ

Лабораторные работы №1.1 - №1.4. Создание простейшего веб-серверного приложения

Цель работ: изучение и реализация простейшего веб-серверного приложения.

В рамках первых 4 лабораторных работ требуется написать простейшее веб-серверное приложение и реализовать сборки, unit-тестирования и деплоя на сервер. Требования для выполнения работы:

1. Исходный проект хранится в удаленном репозитории (Github, Bitbucket, Gitlab).
2. Для сборки можно использовать Travis CI или воспользоваться готовыми решениями, которые предоставляют Bitbucket и Gitlab.
3. В build-плане сделать раздел сборки, тестирования и деплоя.
4. Для деплоя выбрать платформу PaaS типа Heroku, OpenShift или подобных. Если есть интерес, можно выбрать бесплатный VPS-сервер и натсроить деплой на него.

5. (опционально) Реализовать интеграционное тестирование.

Лабораторные работы №1.5 - №1.8. Реализация взаимодействия сервисов (12 час.)

Цель работ: изучение и реализация взаимодействия друг с другом сервисов.

В рамках данных 4 лабораторных работ требуется выбрать предметную область и реализовать несколько взаимодействующих друг с другом сервисов. Сервисы должны решать только свою бизнес-функцию, которая является частью общей системы. Система должна предоставлять REST API для выполнения операций. Получение данных нужно выполняться только через сервис, к которому относятся эти данные.

Агрегацию результатов выполнять на отдельном сервисе.
Требования для выполнения работы:

1. Придумать предметную область и реализовать 3-4 сервиса, выполняющих некоторую свою бизнес-функцию в рамках предметной области.
2. Помимо 3-4 сервисов выделить сервис агрегации (gateway), проксирующий запросы от других систем и агрегирующий данные. Все запросы пользователя проходят через эту систему.
3. Данные каждого сервиса можно хранить как в SQL, так и в NoSQL базе. Для упрощения допускается хранить данные на одной базе, но в разных схемах. При этом каждый сервис должен взаимодействовать только со своей схемой, получение данных, не относящихся к текущему сервису строго запрещено.
4. Должен быть хотя бы один запрос, требующий агрегированную информацию с нескольких сервисов (двух и более).
5. Должно быть минимум два запроса, выполняющие обновление данных на нескольких сервисах в рамках одной операции.
6. При получении списка данных предусмотреть пагинацию.
7. Сделать подробное логгирование выполняемых операций на каждом сервисе.
8. При реализации использовать шаблоны MVC, смешивание представления и бизнес-логики строго запрещено.
9. Все публичные методы требуется покрыть unit-тестами (покрытие функций). Покрытие тестами должно быть не менее 70%, для демонстрации покрытия использовать какую-нибудь утилиту.
10. Подготовить шаблоны запросов или маленький скрипт для демонстрации работы.

Лабораторные работы №2.1 - №2.4 Реализация механизмов, увеличивающих отказоустойчивость системы.

Цель работ: реализовать механизмы, увеличивающие отказоустойчивость системы.

В рамках данных 4 лабораторных работ требуется, используя ПО, разработанное ранее в лабораторных работах 1-8, реализовать механизмы, увеличивающие отказоустойчивость системы.

Требования для выполнения работы:

1. В случае недоступности части систем или внутренних ошибок сервера, возвращать ошибку с пояснением (т.е. http статус 4xx или 5xx + json с пояснением).
2. Реализовать валидацию входных данных. Если данные имеют некорректный формат возвращать 400 Bad Request.
3. Для агрегирующего запроса на чтение (п. 4 требований ЛР 2) в случае недоступности одной из систем, выполнять деградацию функциональности.

4. Для одного из запросов, выполняющих обновление данных на нескольких системах (п. 5 требований ЛР 2), в случае недоступности одной из систем, выполнять полный откат операции.
5. Для другого запроса, в случае недоступности одной из систем, возвращать пользователю успешный результат, а операцию ставить в очередь для повторного выполнения. Когда система, недоступная в процессе операции, будет поднята, операция должна быть выполнена. Для очереди можно использовать in-memory очередь, например, Redis, ZeroMQ, или реализовать очередь на примитивах языка (BlockingQueue для Java).

Лабораторные работы №2.5 - №2.7 Реализация пользовательского интерфейса, выполняющего все операции, реализованные в REST API.

Цель работ: реализовать пользовательский интерфейс, выполняющий все операции, реализованные в REST API.

В рамках данных 3 лабораторных работ требуется, используя ПО, разработанное ранее в лабораторных работах 1-8, реализовать пользовательский интерфейс, выполняющий все операции, реализованные в REST API.

Требования для выполнения работы:

1. Для всех операций, представленных через REST API, написать вызов с пользовательского интерфейса.
2. В рамках лабораторной никакой авторизации не требуется. Если для выполнения операции требуется пользователь, то просто выбирать одного из заранее заданных.
3. Реализовать валидацию данных как на клиенте, так и на сервере.
4. Ошибки валидации требуется отображать на форме и пометить некорректные поля с кратким пояснением.
5. В случае неуспешного ответа от сервера, выводить пользователю уведомление на странице или всплывающим окном.

Лабораторная работа №2.8 Реализация OAuth2-авторизации для API и пользовательского интерфейса.

Цель работы: реализовать OAuth2 авторизацию для API и пользовательского интерфейса.

В рамках данной лабораторной работы требуется, используя ПО, разработанное ранее в лабораторных работах 1-8, реализовать OAuth2 авторизацию для API и пользовательского интерфейса, а также использование токена при взаимодействии между сервисами. Требования для выполнения работы:

1. Все операции между сервером и клиентом разделить на публичные и закрытые. В REST API для выполнения закрытых операций использовать токен, полученный с помощью OAuth 2 (code flow).
2. Для пользовательского интерфейса реализовать login/password авторизацию, в результате на успешную авторизацию выдавать токен, идентифицирующий авторизованного пользователя. Все закрытые операции должны выполняться с помощью этого токена. Если для UI используется SPA и вся логика работы с сервером реализуется с помощью JavaScript, то токен передавать в заголовке Authorization: Bearer <token>. Если используется server rendering и для отправки формы выполняется синхронный submit, то токен можно хранить в cookies и передавать через заголовок Cookie: <token-name>=<token>.
3. Регистрацию нового пользователя делать не обязательно, достаточно создать несколько тестовых пользователей.
4. Пароль в базе хранить в хэшированном виде.
5. Выделить отдельный сервис, отвечающий за авторизацию. Все запросы от UI и API на проверку токенов делегируются ему через Aggregation Service.

6. Выданные токены хранить в некотором хранилище (можно в БД, redis, hazelcast и т.п.). На сервере не использовать встроенную сессию.
7. Для взаимодействия между сервисами использовать токен, каждый сервис имеет <appId>/<appSecret>, по этим данным выдается токен с некоторым временем жизни. По окончании времени жизни, токен требуется перезапросить. Авторизационные данные для доступа к каждому сервису знают сами сервисы, эти запросы не делегируются на Session Service.
8. Все токены должны иметь время жизни.
9. Для токена, полученного с помощью OAuth2, реализовать refresh токен.
10. Токен, используемый в пользовательском интерфейсе, должен инвалидироваться после 30 минут бездействия.
11. Подготовить пример получения/обновления OAuth 2 токена, выполнения запросов к REST API.
12. (опционально) Каждому сервису присвоить доменное имя (через /etc/hosts), между сервисами использовать HTTPS. Использовать nginx в качестве reverse-proxy, настройки HTTPS выполнять на нем.

Лабораторная работа №2.9 Реализация сервиса статистики.

Цель работы: реализовать сервис статистики.

В рамках данной лабораторной работы требуется, используя ПО, разработанное ранее в лаб. работах 1-8, выделить сервис статистики, и отправлять на него информацию обо всех произошедших в системе действиях.

Требования для выполнения работы:

1. Выделить отдельный сервис, на который будут приходить информация с других сервисов через очередь (т.е. событийная модель). Среди этой информации должна быть оповещение об удачной/неудачной авторизации, запросах, изменяющих данные в системе и т.п.
2. Реализовать отказоустойчивую доставку событий (подтверждения успешной обработки). При этом выполнять ожидание подтверждения обработки в другом потоке не блокируя сам запрос. Обрабатывать ситуации отсутствия ответа (таймаут ожидания) и ответ, что запрос не смог обработаться корректно из-за ошибок в данных. В случае если ответ не пришел, реализовать N раз повторную отправку. Если пришел ответ о некорректных данных в сообщении, писать подробную информацию в лог.
3. Реализовать идемпотентную обработку событий на стороне приемщика.
4. Реализовать три сводных отчета по полученным данным. Реализовать графическое представление этих отчетов.

Добавить отдельную роль ADMIN, который будет иметь доступ к графическому представлению отчетов в специальной секции сайта, доступной только пользователям с этой ролью.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Изучение дисциплины проходит в течение одного семестра. Основные темы дисциплины осваиваются в ходе аудиторных занятий, однако важная роль отводится и самостоятельной работе студентов.

Самостоятельная работа как вид учебной работы может использоваться на лабораторных работах, а также иметь самостоятельное значение – внеаудиторная самостоятельная работа обучающихся – при подготовке к лабораторным работам, при подготовке к дифференцированному зачету.

Самостоятельная работа включает в себя следующие этапы:

- изучение теоретического материала (работа над конспектом лекции);
- самостоятельное изучение дополнительных информационных ресурсов (доработка конспекта лекции);
- выполнение заданий текущего контроля успеваемости (подготовка к лабораторным работам);
- итоговая аттестация по дисциплине (подготовка к дифференцированному зачету/экзамену).

4. РЕКОМЕНДАЦИИ ПО ПЛАНИРОВАНИЮ И ОРГАНИЗАЦИИ ВРЕМЕНИ, НЕОБХОДИМОГО ДЛЯ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ. ОПИСАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ДЕЙСТВИЙ СТУДЕНТА («СЦЕНАРИЙ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ»)

Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины.

Для освоения лекционного материала следует: изучить конспект лекции в тот же день, после лекции: 10 – 15 минут, повторно прочитать конспект лекции за день перед следующей лекцией: 10 – 15 минут. Также следует изучить теоретический лекционный материал по рекомендуемому учебнику/учебному пособию: 1 час в неделю.

Следует максимально использовать лекционное время для изучения дисциплины, понимания лекционного материала и написания конспекта лекций. В процессе лекционного занятия студент должен уметь выделять важные моменты и основные положения. При написании *конспекта лекций* следует придерживаться следующих правил и рекомендаций.

1. При ведении конспекта рекомендуется структурировать материал по разделам, главам, темам. Вести нумерацию формул. Выделять по каждой теме постановку задачи, основные положения, выводы. Кратко записывать те пояснения лектора, которые оказались особенно важными. Это позволит при подготовке к сдаче зачёта не запутаться в структуре лекционного материала.

2. Лекционный материал следует записывать в конспект лишь после того, как излагаемый лектором тезис будет вами дослушан до конца и понят.

3. При конспектировании следует отмечать непонятные, на данном этапе, положения, доказательства и пр.

4. Рекомендуется по каждой теме выразить свое мнение, комментарий, вывод.

Доработка конспекта лекции с применением учебника, методической литературы, дополнительной литературы, интернет-ресурсов: этот вид самостоятельной работы студентов особенно важен в том случае, когда одну и ту же задачу можно решать различными способами, а на лекции изложен только один из них. Кроме того, рабочая программа предполагает рассмотрение некоторых относительно несложных тем только во время самостоятельных занятий, без чтения лектором.

Подготовка к лабораторным работам состоит в теоретической подготовке (изучение конспекта лекций, методических указаний к данной лабораторной работе и дополнительной литературы) и выполнении индивидуального задания. Выполнение каждой из запланированных работ заканчивается предоставлением отчета. Требования к форме и содержанию отчета приведены в методических указаниях к лабораторным работам или определяются преподавателем на первом занятии. Допускаясь к лабораторной работе, каждый студент должен представить преподавателю «заготовку» отчета, содержащую: оформленный титульный лист, цель работы, задание, проект решения, полученные результаты, выводы.

Важным этапом является защита лабораторной работы. В процессе защиты студент отвечает на вопросы преподавателя, касающиеся теоретического материала, относящегося к данной работе, и проекта, реализующего его задание, комментирует полученные в ходе работы результаты. При подготовке к защите лабораторной работы рекомендуется ознакомиться со списком вопросов по изучаемой теме и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендуемую литературу.

Подготовка к сдаче дифференцированного зачета.

Зачет – форма промежуточной проверки знаний, умений, навыков, степени освоения дисциплины. Главная задача зачета состоит в том, чтобы у студента по окончании изучения данной дисциплины сформировались определенное представление об общем содержании дисциплины, определенные теоретические знания и практические навыки, определенный кругозор. Готовясь к зачету, студент приводит в систему знания, полученные на лекциях, на практических и лабораторных занятиях, разбирается в том, что осталось непонятным, и тогда изучаемая им дисциплина может быть воспринята в полном объеме с присущей ей строгостью и логичностью, ее практической направленностью.

Зачеты дают возможность преподавателю определить теоретические знания студента и его практические навыки при решении определенных прикладных задач. Оцениваются: понимание и степень усвоения теоретического материала; степень знакомства с основной и дополнительно литературой, а также с современными публикациями; умение применить теорию к практике, решать определенные практические задачи данной предметной области, правильно проводить расчеты и т. д.; знакомство с историей данной науки; логика, структура и стиль ответа, умение защищать выдвигаемые положения.

Значение зачета не ограничивается проверкой знаний, являясь естественным завершением обучения студента по данной дисциплине, они способствуют обобщению и закреплению знаний и умений, приведению их в стройную систему, а также устранению возникших в процессе обучения пробелов.

Подготовка к зачету – это тщательное изучение и систематизация учебного материала, осмысление и запоминание теоретических положений, формулировок, формул, установление и осмысление внутрисубъектных связей между различными темами дисциплины, закрепление теоретических знаний путем решения определенных задач.

Планируйте подготовку к зачету, учитывая сразу несколько факторов: неоднородность в сложности учебного материала и степени его проработки в ходе обучения, свои индивидуальные способности. Рекомендуется делать перерывы в занятиях через каждые 50-60 минут на 10 минут. После 3-4 часов занятий следует сделать часовой перерыв. Чрезмерное утомление приведет к снижению тонуса интеллектуальной деятельности. Целесообразно разделять весь рабочий день на три рабочих периода – с утра до обеда, с обеда до ужина и с ужина до сна. Каждый рабочий период дня должен заканчиваться отдыхом не менее 1 часа. Работая в сессионном режиме, студент имеет возможность увеличить время занятий с 10 (как требовалось в семестре) до 12 часов в сутки.

Подготовку к зачету следует начинать с общего планирования своей деятельности. С определения объема материала, подлежащего проработке, необходимо внимательно сверить конспекты с программой дисциплины, чтобы убедиться, все ли разделы отражены в лекциях, отсутствующие темы изучить по учебнику. Второй этап предусматривает системное изучение материала по данному предмету с обязательной записью всех выкладок, выводов, формул. На третьем этапе – этапе закрепления – полезно чередовать углубленное повторение особенно сложных вопросов с беглым повторением всего материала.

5. РЕКОМЕНДАЦИИ ПО РАБОТЕ С ЛИТЕРАТУРОЙ

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по данному предмету. Литературу по дисциплине рекомендуется читать как в бумажном, так и в электронном виде (если отсутствует бумажный аналог). Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Кроме того, полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?», «зачем мне это нужно по специальности?».

Рекомендуется самостоятельно изучать материал, который еще не прочитан на лек-

ции и не применялся на лабораторном или практическом занятии, тогда занятия будут гораздо понятнее. В течение недели рекомендуется выбрать время (1 час) для работы с литературой.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СОГЛАСОВАНО **ФГБОУ ВО "РГРТУ", РГРТУ**, Овечкин Геннадий
Владимирович, Заведующий кафедрой ВПМ

15.08.24 08:38 (MSK)

Простая подпись