МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.02 Информационные системы и технологии,
утвержденной первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Проектирование цифровых платформ и сервисов

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль): Технологии искусственного интеллекта в транспортных системах

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 5665
Подписал: заведующий кафедрой Нутович Вероника Евгеньевна
Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина посвящена проектированию цифровых платформ и сервисов на основе современных способов межсервисного взаимодействия. В ходе изучения рассматриваются контрактное проектирование программных интерфейсов REST, гипермедиа, GraphQL, событийное взаимодействие через RabbitMQ, удаленные вызовы gRPC, уведомления через WebSocket, обработка ошибок, валидация, устойчивость к отказам и согласование контрактов между сервисами. На лабораторных занятиях обучающиеся последовательно создают многосервисную платформу с каталогом предметных объектов, несколькими транспортами взаимодействия, событиями, аналитическим сервисом и уведомлениями пользователя.

Целью освоения дисциплины является формирование способности проектировать архитектуру цифровых платформ и сервисов на основе контрактов взаимодействия, распределения ответственности между сервисами, событийного обмена, удаленных вызовов, потоковых уведомлений, обработки ошибок и требований к сопровождаемости программного продукта.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности – анализировать границы цифровой платформы и состав сервисов, проектировать контракты REST, GraphQL, RabbitMQ, gRPC и WebSocket, реализовывать согласованное взаимодействие сервисов на Java Spring, применять валидацию, обработку ошибок, идемпотентность и защиту от избыточных запросов, проверять корректность взаимодействия компонентов и готовить техническую документацию по архитектуре цифровой платформы.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ПК-1 - Способен проектировать архитектуру программных продуктов и цифровых сервисов на основе современных методологий проектирования.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- принципы архитектурного проектирования цифровых платформ, включая границы сервисов, контракты взаимодействия и ответственность компонентов

- модели синхронного и асинхронного взаимодействия сервисов, включая запрос-ответ, событие, удаленный вызов и постоянное соединение

- назначение контрактного проектирования программных интерфейсов и способы снижения расхождения между клиентом и сервером

- принципы построения REST-интерфейсов, включая ресурсы, методы HTTP, коды ответа, валидацию и семантику операций POST, PUT и PATCH

- назначение HATEOAS и гипермедиа-ссылок при построении навигации по ресурсам цифрового сервиса

- правила подготовки объектов запроса и ответа, включая неизменяемые структуры запроса, представления ответа и разные модели обновления данных

- принципы обработки ошибок программного интерфейса по RFC 7807 и централизованной обработки исключений

- назначение OpenAPI и Swagger при описании контракта REST и согласовании правил безопасности

- основы GraphQL, включая схему SDL, типы, входные структуры, запросы, мутации, переменные, алиасы и фрагменты

- особенности GraphQL по сравнению с REST, включая выбор полей клиентом, вложенные связи, ошибки в теле ответа и интроспекцию

- принципы реализации GraphQL на Java через Netflix DGS, включая выборки данных, вложенные обработчики, пользовательские скаляры и обработку ошибок

- угрозы избыточной глубины и сложности GraphQL-запросов и способы ограничения вычислительной нагрузки

- принципы событийного взаимодействия через RabbitMQ, включая обменники, очереди, ключи маршрутизации и обработчиков сообщений

- назначение контракта событий, оболочки события, идентификатора события, метаданных и идемпотентной обработки сообщений

- принципы обработки ошибочных сообщений, включая очередь недоставленных сообщений и повторные попытки доставки

- основы Protocol Buffers и gRPC, включая описание сообщений, описание служб, генерацию кода, серверную и клиентскую части

- сценарии сочетания события RabbitMQ и синхронного вызова gRPC при обогащении данных без нарушения основного процесса

- принципы WebSocket, включая установление постоянного соединения, отправку уведомлений от сервера и управление пользовательскими сессиями

- способы сквозной проверки распределенной платформы, включая проверку контрактов, событий, удаленных вызовов, уведомлений и обработки отказов

- требования к технической документации архитектуры цифровой платформы, включая контракты, схему взаимодействия, ограничения и результаты проверки

Уметь:

- уметь проектировать границы сервисов при помощи архитектурной схемы и описания ответственности компонентов в условиях многосервисной цифровой платформы

- уметь проектировать контракт REST при помощи Spring MVC аннотаций, DTO и OpenAPI в условиях разделения контракта и реализации на уровне Maven-модулей

- уметь реализовывать валидацию запросов при помощи Jakarta Bean Validation и пользовательских валидаторов в условиях проверки предметных ограничений

- уметь оформлять обработку ошибок при помощи RFC 7807 и ControllerAdvice в условиях единообразного ответа при ошибках клиента и сервера

- уметь реализовывать REST-интерфейс при помощи Spring Boot, HATEOAS и пагинации в условиях навигации по связанным ресурсам

- уметь реализовывать семантику POST, PUT и PATCH при помощи разных моделей запроса в условиях создания, полной замены и частичного изменения ресурса

- уметь проектировать схему GraphQL при помощи SDL в условиях представления типов, связей, входных структур, запросов и мутаций

- уметь реализовывать GraphQL-обработчики при помощи Netflix DGS в условиях получения связанных данных и преобразования моделей

- уметь ограничивать сложность GraphQL-запросов при помощи инструментов контроля глубины и сложности в условиях защиты сервиса от избыточной нагрузки

- уметь проектировать событийный контракт при помощи sealed interface, оболочки события и ключей маршрутизации в условиях асинхронной передачи изменений

- уметь реализовывать публикацию и обработку событий при помощи RabbitMQ и RabbitTemplate в условиях идемпотентной обработки и очереди недоставленных сообщений

- уметь проектировать контракт gRPC при помощи Protocol Buffers в условиях описания сообщений и удаленных методов аналитического сервиса

- уметь реализовывать сервер и клиент gRPC при помощи сгенерированного Java-кода в условиях синхронного обогащения данных по событию

- уметь реализовывать уведомления через WebSocket при помощи Spring WebSocket в условиях отправки событий подключенным пользователям

- уметь проверять сквозное взаимодействие сервисов при помощи журналов, тестовых запросов и сценариев отказа в условиях нескольких транспортов взаимодействия

- уметь готовить техническую документацию при помощи архитектурной схемы, описания контрактов, таблицы событий и результатов проверки в условиях передачи решения на сопровождение

Владеть:

- навыком проектирования архитектуры многосервисной цифровой платформы

- навыком разработки контрактов REST, GraphQL, RabbitMQ, gRPC и WebSocket

- навыком реализации REST-сервиса с HATEOAS, валидацией и обработкой ошибок

- навыком реализации GraphQL-сервиса на Netflix DGS с ограничением глубины и сложности запросов

- навыком организации событийного обмена через RabbitMQ с идемпотентной обработкой сообщений

- навыком разработки gRPC-сервера и gRPC-клиента на основе Protocol Buffers

- навыком реализации уведомлений через WebSocket и проверки сквозного сценария взаимодействия

- навыком подготовки технической документации по архитектуре цифровой платформы

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 4 з.е. (144 академических часа(ов).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №7
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 80 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Архитектура цифровых платформ и сервисов Рассматриваемые вопросы: - понятие цифровой платформы и ее отличие от отдельного программного сервиса; - границы сервисов, ответственность компонентов и контракты взаимодействия; - требования к сопровождаемости, расширяемости и устойчивости платформенного решения.
2	Способы межсервисного взаимодействия Рассматриваемые вопросы: - синхронное взаимодействие по модели запрос-ответ; - асинхронный обмен событиями и обработка изменений состояния; - постоянное соединение для уведомления пользователя о событиях сервиса.
3	Контрактное проектирование REST-интерфейсов Рассматриваемые вопросы: - контракт как единый источник правил взаимодействия клиента и сервера; - ресурсы, методы HTTP, коды ответа и валидация входных данных; - разделение контракта и реализации в модульной структуре Java-приложения.
4	Модели данных REST-интерфейса и правила обновления ресурсов Рассматриваемые вопросы: - объекты запроса и ответа при создании, полной замене и частичном изменении ресурса;

№ п/п	Тематика лекционных занятий / краткое содержание
	- неизменяемые структуры данных и представления ответа с гипермедиа-ссылками; - пользовательская валидация предметных ограничений.
5	Обработка ошибок и описание REST-контракта Рассматриваемые вопросы: - единый формат ошибок по RFC 7807; - централизованная обработка исключений в Spring MVC; - описание контракта и правил безопасности средствами OpenAPI.
6	Гипермедиа и навигация по ресурсам REST-сервиса Рассматриваемые вопросы: - назначение HATEOAS в развитии REST-интерфейса; - представление сущности, набора сущностей и страниц с гипермедиа-ссылками; - формирование ссылок на связанные операции и навигацию по страницам.
7	Служебный слой и согласованность операций REST-сервиса Рассматриваемые вопросы: - разделение контроллеров, служебного слоя и хранилища данных; - обработка полной замены, частичного изменения и каскадного удаления; - предотвращение циклических зависимостей между компонентами.
8	GraphQL как язык запросов к связанным данным Рассматриваемые вопросы: - причины применения GraphQL при сложных связях между данными; - схема SDL, объектные типы, входные типы, запросы и мутации; - переменные, алиасы, фрагменты и вложенные поля запроса.
9	Реализация GraphQL-сервиса на Java Рассматриваемые вопросы: - компоненты Netflix DGS для обработки запросов и мутаций; - получение родительского объекта во вложенном обработчике; - преобразование GraphQL-моделей в модели REST-слоя и обратно.
10	Безопасность и устойчивость GraphQL-интерфейса Рассматриваемые вопросы: - обработка ошибок GraphQL и классификация причин отказа; - пользовательские скалярные типы даты и времени; - ограничение глубины и вычислительной сложности запроса.
11	Событийная архитектура на основе RabbitMQ Рассматриваемые вопросы: - обменники, очереди, ключи маршрутизации и подписчики событий; - контракт события, метаданные, полезная нагрузка и идентификатор события; - публикация события после успешного изменения состояния данных.
12	Надежность обработки событий Рассматриваемые вопросы: - идемпотентность обработчика и защита от повторной обработки события; - очередь недоставленных сообщений и анализ ошибочных событий; - повторные попытки доставки с увеличением интервала ожидания.
13	Удаленные вызовы через gRPC и Protocol Buffers Рассматриваемые вопросы: - описание сообщений и служб в Protocol Buffers; - генерация Java-кода для серверной и клиентской частей gRPC; - синхронный удаленный вызов как способ получения аналитических данных.
14	Сочетание событийного обмена и gRPC-вызова Рассматриваемые вопросы: - запуск удаленного вызова по факту получения события;

№ п/п	Тематика лекционных занятий / краткое содержание
	- обогащение события результатами аналитического сервиса; - сохранение основного процесса при отказе вспомогательного сервиса.
15	Уведомления через WebSocket Рассматриваемые вопросы: - постоянное соединение WebSocket и отличие от обычного HTTP-запроса; - регистрация пользовательских соединений и потокобезопасная отправка сообщений; - передача событий цифровой платформы подключенным пользователям.
16	Сквозная проверка архитектуры цифровой платформы Рассматриваемые вопросы: - проверка согласованности REST, GraphQL, RabbitMQ, gRPC и WebSocket; - анализ журналов, ошибок, повторной обработки событий и отказов зависимостей; - структура технической документации по архитектуре и контрактам взаимодействия.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Разработка модульного контракта REST-интерфейса Студент создает Maven-модуль контракта для предметных ресурсов цифрового сервиса. В контракт добавляются интерфейсы Spring MVC с маршрутами, методами HTTP, правилами валидации и моделями запроса. Для разных операций обновления создаются отдельные модели создания, полной замены и частичного изменения.
2	Описание ответов и ошибок REST-интерфейса Студент оформляет модели ответа с поддержкой гипермедиа-ссылок. Добавляется пользовательская валидация предметного идентификатора и единый формат ошибки по RFC 7807. В модуле контракта настраивается описание OpenAPI и схема авторизации.
3	Реализация REST-интерфейса по заданному контракту Студент создает модуль реализации и подключает контракт как зависимость. Хранение данных выполняется в потокобезопасной структуре с начальным заполнением тестовыми данными. Контроллеры реализуют операции создания, чтения, изменения, удаления и поиска ресурсов.
4	Формирование гипермедиа-ссылок и постраничной навигации Студент добавляет сборщик представлений для единообразного формирования ссылок. Ответы отдельных ресурсов и страниц оформляются с использованием средств Spring HATEOAS. Для списка ресурсов добавляется навигация по первой, предыдущей, следующей и последней странице.
5	Реализация правил изменения связанных ресурсов Студент реализует полную замену, частичное изменение и каскадное удаление связанных объектов. При необходимости циклическая зависимость служебных компонентов разрывается средствами Spring. Поведение операций проверяется через последовательность HTTP-запросов.
6	Разработка схемы GraphQL Студент создает схему SDL с типами, входными структурами, запросами, мутациями и связями между объектами. Для даты и времени задаются пользовательские скалярные типы. Схема проверяется запросами с аргументами, вложенными полями, переменными, алиасами и фрагментами.
7	Реализация обработчиков GraphQL Студент реализует компоненты Netflix DGS для запросов, мутаций и вложенных связей. Родительский объект извлекается из среды выполнения запроса. Входные структуры GraphQL преобразуются в модели существующего служебного слоя.

№ п/п	Наименование лабораторных работ / краткое содержание
8	Обработка ошибок и ограничение сложности GraphQL-запросов Студент добавляет обработчик исключений GraphQL с классификацией ошибок. Настраиваются ограничения глубины и вычислительной сложности запроса. Проверяются корректные запросы, избыточно вложенные запросы и ошибочные обращения к данным.
9	Проектирование и публикация событий RabbitMQ Студент создает модуль контракта событий с общей структурой события, метаданными и ключами маршрутизации. После успешного изменения данных публикуется событие в тематический обменник RabbitMQ. Событие содержит идентификатор, тип действия, время и полезную нагрузку.
10	Обработка событий и очередь недоставленных сообщений Студент реализует отдельный сервис обработки событий с подпиской на очередь RabbitMQ. Обработчик выполняет проверку идемпотентности по идентификатору события. Для ошибочных сообщений настраиваются повторные попытки и очередь недоставленных сообщений.
11	Разработка контракта gRPC на основе Protocol Buffers Студент описывает сообщения и удаленные методы аналитического сервиса в файле proto. По описанию генерируется Java-код серверной и клиентской частей. В проект добавляется сервер gRPC с обработкой входного запроса через StreamObserver.
12	Реализация gRPC-вызова по событию RabbitMQ Студент создает клиент gRPC, вызываемый после получения события RabbitMQ. Результат удаленного вызова используется для формирования обогащенного события. Отказ аналитического сервиса обрабатывается без отмены основного изменения данных.
13	Реализация уведомлений через WebSocket Студент настраивает Spring WebSocket и обработчик текстовых сообщений. Подключенные пользовательские соединения хранятся в потокобезопасном реестре. Сервис отправляет уведомления всем активным соединениям при поступлении события.
14	Интеграция событийной очереди и WebSocket-уведомлений Студент связывает обработчик RabbitMQ с отправкой уведомлений через WebSocket. Для входящих событий выполняется дедупликация по идентификатору. Клиентская страница отображает состояние соединения и журнал полученных уведомлений.
15	Сквозная проверка взаимодействия сервисов Студент запускает сервисы REST, GraphQL, RabbitMQ, gRPC и WebSocket в единой последовательности. Проверяется путь от изменения ресурса до события, аналитического обогащения и уведомления пользователя. Результаты фиксируются по журналам сервисов и контрольным запросам.
16	Анализ устойчивости и подготовка технической документации Студент моделирует ошибочные события, недоступность вспомогательного сервиса и избыточный GraphQL-запрос. Для каждого случая фиксируются реакция системы, сообщения об ошибках и сохранность основного процесса. Техническая документация включает схему взаимодействия, контракты, события, ограничения и результаты проверки.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным работам.
3	Выполнение курсового проекта.
4	Подготовка к промежуточной аттестации.

5	Подготовка к текущему контролю.

4.4. Примерный перечень тем курсовых проектов

Проектирование цифровой платформы управления каталогом учебной литературы.

Проектирование цифровой платформы учета транспортных заявок.

Проектирование цифровой платформы мониторинга сервисных обращений пассажиров.

Проектирование цифровой платформы управления расписанием и уведомлениями.

Проектирование цифровой платформы бронирования ресурсов учебной лаборатории.

Проектирование цифровой платформы учета заявок на ремонт оборудования.

Проектирование цифровой платформы управления складом запасных частей.

Проектирование цифровой платформы обмена сообщениями между подразделениями.

Проектирование цифровой платформы учета мероприятий транспортного университета.

Проектирование цифровой платформы управления заявками на практику обучающихся.

Проектирование цифровой платформы ведения справочника маршрутов.

Проектирование цифровой платформы аналитики обращений пользователей.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Баланов, А. Н. Построение микросервисной архитектуры и разработка высоконагруженных приложений : учебное пособие для вузов / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. — 244 с. — ISBN 978-5-507-48747-9. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/394538 (дата обращения: 19.06.2026)
2	Уоллс, К. Spring в действии : руководство / К. Уоллс ; перевод с английского А. Н. Киселева. —	Лань : электронно-библиотечная система. — URL:

	6-е изд. — Москва : ДМК Пресс, 2022. — 544 с. — ISBN 978-5-93700-112-2. — Текст : электронный	https://e.lanbook.com/book/314828 (дата обращения: 19.06.2026)
3	Васюткина, И. А. Разработка серверной части web-приложений на Java : учебное пособие / И. А. Васюткина. — Новосибирск : НГТУ, 2021. — 83 с. — ISBN 978-5-7782-4394-1. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/216155 (дата обращения: 19.06.2026)
4	Наир, В. Предметно-ориентированное проектирование в Enterprise Java : руководство / В. Наир ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2020. — 306 с. — ISBN 978-5-97060-872-2. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/179503 (дата обращения: 19.06.2026)
5	Аймен, Э. А. Облачные микросервисы в Kubernetes : руководство / Э. А. Аймен ; пер. с англ. В. С. Яценкова. — Москва : ДМК Пресс, 2025. — 276 с. — ISBN 978-5-93700-324-9. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/514873 (дата обращения: 19.06.2026)
6	Хабаров, С. П. Построение распределенных систем на базе WebSocket : учебное пособие / С. П. Хабаров, М. Л. Шилкина. — Санкт-Петербург : Лань, 2020. — 216 с. — ISBN 978-5-8114-4739-8. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/142358 (дата обращения: 19.06.2026)
7	Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А. Н. Свистунов. — 2-е изд. — Москва : ИНТУИТ, 2016. — 317 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100371 (дата обращения: 19.06.2026)
8	Замотайлова, Д. А. Архитектура предприятий и информационных систем : учебное пособие / Д. А. Замотайлова, Е. В. Попова. — Краснодар : КубГАУ, 2021. — 172 с. — ISBN 978-5-907474-07-9. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/254270 (дата обращения: 19.06.2026)

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

ЭБС Лань – https://e.lanbook.com/.

Образовательная платформа Юрайт – https://urait.ru/.

Единый реестр российских программ для ЭВМ и баз данных – https://reestr.digital.gov.ru/reestr/.

Профессиональные стандарты и квалификации, справочная информация КонсультантПлюс – https://www.consultant.ru/document/cons_doc_LAW_157436/.

Транспортная стратегия Российской Федерации до 2030 года с прогнозом на период до 2035 года, КонсультантПлюс – https://www.consultant.ru/document/cons_doc_LAW_402052/.

Документация Java SE – https://docs.oracle.com/en/java/javase/21/docs/api/index.html.

Документация Spring Framework – https://docs.spring.io/spring-framework/reference/.

Документация Spring HATEOAS – https://docs.spring.io/spring-hateoas/docs/current/reference/html/.

Спецификация GraphQL – https://spec.graphql.org/.

Документация Netflix DGS – https://netflix.github.io/dgs/.

Документация RabbitMQ – https://www.rabbitmq.com/docs.

Документация gRPC – https://grpc.io/docs/.

Документация Protocol Buffers – https://protobuf.dev/.

RFC 6455 WebSocket – https://www.rfc-editor.org/rfc/rfc6455.

RFC 7807 Problem Details – https://www.rfc-editor.org/rfc/rfc7807.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Операционные системы – Astra Linux, ALT Linux, РЕД ОС, Debian GNU/Linux.

Платформа разработки – OpenJDK 21+, Maven, IntelliJ IDEA Community Edition.

Серверные технологии – Spring Boot, Spring HATEOAS, Netflix DGS, Spring WebSocket.

Обмен данными – RabbitMQ, gRPC, Protocol Buffers.

Проверка и сопровождение – Git, curl, Swagger UI.

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для лабораторных занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Курсовой проект в 7 семестре.

Экзамен в 7 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»
Института железнодорожного
транспорта Е.А. Заманов

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Председатель учебно-методической
комиссии Н.А. Андриянова