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
09.03.01 Информатика и вычислительная техника,
утвержденной директором ФГАОУ ВО РУТ
(МИИТ) Париновым Д.В.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Предметно-ориентированное проектирование

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки программного обеспечения

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: 937226
Подписал: директор центра Проневич Ольга Борисовна
Дата: 12.11.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина формирует у будущих инженеров-архитекторов критически важный навык перевода хаотичных бизнес-требований в строгую математическую и программную модель. В условиях острого кадрового дефицита на рынке системных аналитиков и Middle+/Senior архитекторов, курс выступает ядром межпредметной интеграции. Студент выступает в роли «ведущего архитектора домена», чьи артефакты (карты контекстов, гексагональная архитектура, API-контракты и схемы доменных событий) становятся прямым техническим заданием для смежных команд, параллельно изучающих серверную, клиентскую и мобильную разработку. На практике обучающиеся осваивают методологию Domain-Driven Design (DDD), учатся проводить сессии Event Storming, выделять агрегаты и защищать ядро бизнес-логики от инфраструктурных утечек. Особый акцент делается на событийно-ориентированной архитектуре, генерации OpenAPI-спецификаций для фронтенда и настройке маршрутизации WebSocket/Push-уведомлений для мобильных клиентов с использованием импортозамещенного стека и открытого программного обеспечения.

Целью освоения дисциплины является формирование системного архитектурного мышления и способности проектировать слабовязанные, эволюционирующие программные продукты на основе методологии предметно-ориентированного проектирования в условиях гетерогенной и импортозамещенной ИТ-инфраструктуры.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности: выявлять скрытые бизнес-инварианты и формировать единый язык коммуникации с заказчиками, декомпозировать сложные предметные области на изолированные ограниченные контексты, инкапсулировать бизнес-логику в тактические паттерны DDD, изолировать доменное ядро от инфраструктуры посредством гексагональной архитектуры, а также проектировать строгие контракты и механизмы асинхронной доставки событий для интеграции с клиентскими и мобильными приложениями.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ДПК-БП-4 - Способен проектировать и реализовывать архитектуру данных нагруженного бизнес-приложения;

ИП-ДРБ-1 - Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению информационных систем (ИС), автоматизирующих задачи организационного управления и бизнес-процессы;

ППК-Р5 - Способен разрабатывать требования и проектировать программное обеспечение;

ППК-Р6 - Способен участвовать в промышленной разработке программного обеспечения;

ППК-У2 - Способен выстраивать и анализировать взаимосвязь технических решений и задач разработки с бизнес-целями и показателями компании;

УК-1 - Способен осмысленно подходить к решению задач, выявлять проблемы, ставить цели, вырабатывать стратегию действий.

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

Знать:

- философия и эвристики предметно-ориентированного проектирования как метода борьбы со случайной и существенной сложностью бизнес-логики;
- концепция Единого Языка и методы его формализации для устранения семантических разрывов между разработчиками и бизнес-экспертами;
- методология событийного моделирования и техники выявления скрытых бизнес-процессов, команд и инвариантов;
- принципы стратегического проектирования и критерии декомпозиции предметной области на ограниченные контексты;
- паттерны интеграции ограниченных контекстов и правила построения Карты Контекстов;
- концепция агрегатов, корневых сущностей и правила обеспечения строгой транзакционной консистентности их границ;
- тактическое моделирование сущностей и управление их жизненным циклом посредством диаграмм конечных автоматов;
- природа, правила проектирования и сценарии применения неизменяемых объектов-значений для борьбы с антипаттерном примитивной одержимости;
- роль и сценарии инкапсуляции сложной, распределенной бизнес-логики в доменных сервисах и фабриках;
- принципы построения гексагональной архитектуры для жесткой изоляции доменного ядра от инфраструктуры;

- архитектурные слои приложения и строгое разделение ответственности между доменом, слоем применения и инфраструктурным слоем;

- концепция доменных событий и их роль в обеспечении слабой связности и согласованности в конечном счете;

- паттерн Transactional Outbox и механизмы гарантированной асинхронной доставки событий через брокеры сообщений;

- антикоррозионные слои и паттерны трансляции данных для защиты собственной доменной модели при интеграции с внешними системами;

- абстракция репозиториев, их интерфейс и реализация в виде адаптеров для реляционных и NoSQL СУБД;

- эвристики рефакторинга доменной модели, выявления запахов предметной области и управления архитектурным техническим долгом;

- специфика применения практик DDD в условиях системного импортозамещения и генерации API-контрактов для смежных команд разработки.

Уметь:

- уметь проводить сессии событийного моделирования при помощи инструментов визуального моделирования в условиях первичного выявления бизнес-процессов и скрытых инвариантов предметной области;

- уметь формировать глоссарий Единого Языка при помощи матриц соответствия бизнес-терминов и программных сущностей в условиях устранения семантической двусмысленности между разработчиками и бизнес-заказчиками;

- уметь проектировать карту ограниченных контекстов при помощи нотаций стратегического DDD в условиях декомпозиции сложной корпоративной системы на слабосвязанные микросервисы;

- уметь проектировать границы Агрегатов и корневых Сущностей при помощи UML-диаграмм классов и анализа транзакционных инвариантов в условиях обеспечения строгой консистентности данных внутри доменной модели;

- уметь выделять и реализовывать Объекты-значения при помощи паттернов неизменяемых объектов в выбранном языке программирования в условиях борьбы с антипаттерном Primitive Obsession и защиты бизнес-правил;

- уметь моделировать жизненные циклы Сущностей при помощи диаграмм конечных автоматов в условиях управления сложными статусными переходами бизнес-объектов;

- уметь изолировать доменную модель от инфраструктуры при помощи паттернов Гексагональной архитектуры в условиях обеспечения независимости бизнес-логики от конкретных реализаций СУБД и внешних API;

- уметь писать код Доменных Сервисов и Сервисов Приложения при помощи принципов SOLID и инверсии зависимостей в условиях оркестрации бизнес-логики без нарушения границ Агрегатов;

- уметь проектировать и генерировать Доменные события и OpenAPI-контракты при помощи паттерна Transactional Outbox и стандартов описания API в условиях обеспечения гарантированной асинхронной доставки сообщений и интеграции с клиентскими приложениями;

- уметь проектировать антикоррозионные слои и BFF-адаптеры при помощи паттернов адаптации данных и фасадов в условиях интеграции с внешними системами и агрегации данных под специфичные нужды мобильных и веб-клиентов.

Владеть:

- навыками фасилитации сессий совместного моделирования для выявления скрытых бизнес-инвариантов и формирования архитектурного видения продукта;

- приемами объектно-ориентированного программирования для инкапсуляции бизнес-логики в агрегаты, объекты-значения и доменные сервисы;

- методами настройки инфраструктурных адаптеров и маршрутизации доменных событий для обновления состояний на клиентских и мобильных устройствах;

- способами генерации, версионирования и валидации строгих API-контрактов для кросс-функциональных команд серверной и клиентской разработки;

- инструментарием контейнеризации и интеграционного тестирования для проверки работоспособности гексагональной архитектуры в изолированной среде.

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 №5
Контактная работа при проведении учебных занятий (всего):	80	80
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	48	48

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 64 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	Философия и эвристики предметно-ориентированного проектирования Рассматриваемые вопросы: - концепция существенной и случайной сложности в разработке программного обеспечения; - исторические предпосылки возникновения методологии DDD и ее эволюция; - роль архитектора в преодолении хаоса бизнес-требований и формировании контрактов для смежных команд.
2	Единый язык и событийное моделирование Рассматриваемые вопросы: - принципы формирования глоссария единого языка для устранения семантических разрывов; - методология event storming и техники выявления скрытых бизнес-процессов; - визуализация команд, событий и политик на интерактивных досках.
3	Стратегическое проектирование и ограниченные контексты Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- критерии декомпозиции предметной области на изолированные домены; - понятие ограниченного контекста и его роль в микросервисной архитектуре; - выявление ядра предметной области, поддерживающих и общих поддоменов.
4	Интеграция контекстов и построение карты Рассматриваемые вопросы: - паттерны взаимодействия ограниченных контекстов и их экономическое обоснование; - модели отношений customer-supplier, partnership и conformist; - правила визуализации и документирования карты контекстов.
5	Агрегаты и обеспечение транзакционной консистентности Рассматриваемые вопросы: - концепция агрегатов и выбор корневой сущности для управления границами; - правила обеспечения строгой консистентности внутри агрегата; - эвристики определения оптимального размера агрегата и ссылок по идентификаторам.
6	Тактическое моделирование сущностей и управление состоянием Рассматриваемые вопросы: - отличия сущностей от других строительных блоков доменной модели; - моделирование жизненных циклов объектов при помощи диаграмм конечных автоматов; - обработка инвариантов и бизнес-правил при изменении состояний.
7	Объекты-значения и борьба с примитивной одержимостью Рассматриваемые вопросы: - природа неизменяемых объектов-значений и сценарии их применения; - антипаттерн primitive obsession и методы его устранения в коде; - реализация логики валидации и сравнения внутри объектов-значений.
8	Доменные сервисы и фабрики Рассматриваемые вопросы: - инкапсуляция распределенной бизнес-логики в доменных сервисах; - сценарии выделения фабрик для создания сложных агрегатов и сущностей; - разграничение ответственности между доменными сервисами и сервисами приложения.
9	Гексагональная архитектура, изоляция и паттерн BFF Рассматриваемые вопросы: - принципы построения архитектуры портов и адаптеров; - строгое разделение слоев приложения и инверсия зависимостей; - использование адаптеров в роли BFF для агрегации данных под специфичные нужды мобильных и веб-клиентов
10	Доменные события, согласованность и схемы контрактов Рассматриваемые вопросы: - роль событий в обеспечении слабой связности и триггеризации UI-обновлений; - концепция eventual consistency и компромиссы распределенных систем; - проектирование схем событий для команд мобильной и клиентской разработки.
11	Гарантированная доставка событий и интеграция с брокерами Рассматриваемые вопросы: - паттерн transactional outbox для атомарной фиксации состояния и события; - механизмы публикации сообщений в очередях и стриминговых платформах; - обработка идемпотентности и повторных доставок в подписчиках.
12	Антикоррозионные слои и публикации API-контрактов Рассматриваемые вопросы: - защита собственной доменной модели от влияния внешних систем; - генерация спецификаций OpenAPI и gRPC на основе портов слоя приложения; - создание открытых сервисов и публикации языков для сторонних потребителей.

№ п/п	Тематика лекционных занятий / краткое содержание
13	Абстракция репозиториев и инфраструктурные адаптеры Рассматриваемые вопросы: - интерфейс репозитория и его место в доменной модели; - реализация адаптеров для реляционных и документоориентированных субд; - оптимизация запросов и паттерн unit of work в инфраструктурном слое.
14	Рефакторинг доменной модели и управление техническим долгом Рассматриваемые вопросы: - выявление запахов предметной области и эвристики улучшения архитектуры; - стратегии эволюционного рефакторинга агрегатов и контекстов; - метрики качества кода и их связь с бизнес-ценностью продукта.
15	Специфика DDD в условиях импортозамещения и гетерогенной инфраструктуры Рассматриваемые вопросы: - адаптация практик проектирования под отечественные субд и операционные системы; - обеспечение информационной безопасности на уровне доменной модели; - интеграция с государственными сервисами и корпоративными шинами данных.
16	Итоговая архитектура и защита проектных решений Рассматриваемые вопросы: - синтез стратегических и тактических артефактов в единый архитектурный паспорт; - методы презентации и обоснования проектных решений перед стейкхолдерами; - анализ типовых ошибок и антипаттернов при внедрении ddd в реальных проектах.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Проведение сессии событийного моделирования (Event Storming) Студент разворачивает локальный или облачный аналог интерактивной доски и переносит на нее стикеры с доменными событиями предметной области грузоперевозок. Затем инженер добавляет команды, инициирующие эти события, и выявляет проблемные зоны бизнес-процессов. В завершение работы формируется первичная хронологическая лента жизненного цикла груза.
2	Определение границ ограниченных контекстов и построение Context Map Студент анализирует полученную ленту событий и группирует их в изолированные кластеры, формируя границы будущих микросервисов. Инженер отрисовывает карту контекстов в специализированном редакторе диаграмм, связывая блоки паттернами взаимодействия. Результатом является схема стратегической архитектуры платформы с указанием направлений обмена данными.
3	Формализация глоссария Единого Языка (Ubiquitous Language) Студент открывает текстовый редактор из отечественного офисного пакета и создает таблицу соответствия бизнес-терминов и программных сущностей. Инженер описывает строгие определения для каждого концепта, исключая двусмысленности между разными контекстами. Итоговый документ экспортируется в формат, пригодный для версионирования в системе контроля исходного кода.
4	Проектирование Агрегатов и корневых Сущностей Студент инициализирует пакет доменной модели в среде разработки и создает базовые классы корневых сущностей. Инженер реализует методы изменения состояния, внутри которых зашивает проверки бизнес-инвариантов и правил согласованности. Код покрывается модульными тестами, подтверждающими невозможность перевода Агрегата в невалидное состояние.
5	Реализация Объектов-значений (Value Objects) тудент выявляет в предметной области концепции, не имеющие идентификатора, и описывает их в

№ п/п	Наименование лабораторных работ / краткое содержание
	виде неизменяемых классов. Инженер переопределяет методы сравнения и хеширования, обеспечивая корректную работу структур данных. Базовые типы данных в сигнатурах методов Агрегатов заменяются на созданные строго типизированные объекты-значения.
6	Моделирование жизненного цикла Сущности (State Machine) Студент описывает возможные статусы заказа и правила переходов между ними в виде матрицы состояний. Инженер внедряет в код Агрегата паттерн конечного автомата, блокирующий некорректные переходы и генерирующий исключения при нарушении бизнес-логики. Поведение сущности проверяется серией сценарных тестов, имитирующих различные пути выполнения заказа.
7	Инкапсуляция логики в Доменных Сервисах и Фабриках Студент выделяет из Агрегатов распределенную бизнес-логику и переносит ее в отдельные классы доменных сервисов. Инженер проектирует фабрики, инкапсулирующие сложные правила создания кластеров взаимосвязанных сущностей. Написанный код интегрируется в общий каркас проекта без нарушения принципа единственной ответственности.
8	Генерация и обработка Доменных Событий Студент добавляет в корневые сущности механизм регистрации фактов изменения их состояния в виде объектов доменных событий. Инженер реализует паттерн сбора событий, позволяющий внешнему коду получать список произошедших изменений после выполнения бизнес-операции. Работоспособность механизма подтверждается тестами, проверяющими корректность генерации полезной нагрузки событий.
9	Создание Портов (Интерфейсов) в Гексагональной архитектуре Студент описывает абстрактные интерфейсы репозиториев и внешних сервисов непосредственно в доменном слое приложения. Инженер строго регламентирует сигнатуры методов, используя только типы из доменной модели и избегая любых инфраструктурных зависимостей. Созданные порты становятся контрактами для последующей реализации адаптеров.
10	Реализация слоя Приложения (Use Cases) и генерация API-контрактов Студент пишет классы сервисов приложения, которые оркестрируют выполнение бизнес-сценариев и вызывают методы Агрегатов. Инженер настраивает автоматическую генерацию спецификации OpenAPI на основе портов входящих команд для команд клиентской и мобильной разработки. Слой приложения покрывается интеграционными тестами с использованием заглушек вместо реальных инфраструктурных компонентов.
11	Настройка инверсии зависимостей (DI) и связывание слоев Студент конфигурирует контейнер внедрения зависимостей для автоматического связывания абстрактных портов с конкретными реализациями. Инженер настраивает маршрутизацию входящих запросов от пользовательского интерфейса к соответствующим сервисам приложения. В результате сборки получается запускаемый каркас системы с полностью изолированным доменным ядром.
12	Реализация Адаптера для реляционной СУБД Студент проектирует схему реляционной базы данных и пишет SQL-скрипты для создания таблиц, отражающих структуру Агрегатов. Инженер реализует классы инфраструктурных репозиториев, обеспечивающие маппинг строк из базы данных в объекты доменной модели и обратно. Работоспособность адаптера проверяется путем сохранения и последующего чтения сложного Агрегата.
13	Реализация паттерна Transactional Outbox Студент добавляет в схему базы данных специальную таблицу для исходящих сообщений и модифицирует код репозитория. Инженер обеспечивает атомарную запись изменений Агрегата и соответствующих доменных событий в рамках одной транзакции с СУБД. Корректность реализации подтверждается тестами, имитирующими сбои и гарантирующими сохранность событий.
14	Интеграция с брокером и маршрутизация событий на клиенты Студент разворачивает локальный экземпляр брокера сообщений и настраивает очереди для приема доменных событий. Инженер пишет фоновый процесс, который вычитывает записи из Outbox и маршрутизирует их в адаптеры, формирующие JSON-пейлоады для Push-уведомлений на

№ п/п	Наименование лабораторных работ / краткое содержание
	мобильные устройства и WebSocket-кадров для веб-клиентов. Успешность интеграции демонстрируется путем эмуляции получения события конечным устройством.
15	Проектирование Антикоррозионного слоя (BFF-адаптера) Студент генерирует мокальный внешний API, возвращающий данные в чужеродном для системы формате. Инженер пишет адаптер, который перехватывает эти данные, транслирует их в собственные Объекты-значения и агрегирует специально под ограничения UI мобильного приложения. Реализованный BFF-слой защищает ядро системы от проникновения чужой семантики и структур данных.
16	Итоговая сборка прототипа и оформление Архитектурного Паспорта Студент запускает сквозной интеграционный сценарий, демонстрирующий полный жизненный цикл заказа от создания до обновления статуса на веб-клиенте через WebSocket. Инженер фиксирует результаты работы в виде структурированного отчета, прикладывая диаграммы, фрагменты кода и сгенерированные API-контракты. Сформированный архитектурный паспорт и работающий прототип подготавливаются к финальной защите.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Анализ предметной области и выявление бизнес-инвариантов Студент разбирает кейс предметной области грузоперевозок и проводит ролевое интервью с «бизнес-экспертом», фиксируя требования, ограничения и точки неопределенности. Инженер отделяет существенную сложность бизнес-правил от случайной сложности технических решений и составляет реестр скрытых инвариантов с указанием источника правила и цены его нарушения. Результатом работы является структурированное описание домена и перечень бизнес-правил, подлежащих защите внутри границ Агрегатов.
2	Классификация поддоменов и обоснование границ ограниченных контекстов Студент классифицирует выделенные кластеры событий на ядро, поддерживающие и общие поддомены, оценивая их вклад в конкурентное преимущество платформы. Инженер уточняет глоссарий по границам контекстов, устраняя омонимы и семантические коллизии, и обосновывает выбор паттернов интеграции в записях об архитектурных решениях (ADR). Итогом является уточненная карта контекстов с экономическим обоснованием стоимости интеграционных связей.
3	Проектирование границ Агрегатов и матрицы состояний Студент анализирует транзакционные инварианты и определяет, какие объекты обязаны изменяться в одной транзакции, а какие допускают ссылки по идентификатору и согласованность в конечном счете. Инженер строит UML-диаграмму классов Агрегатов и матрицу допустимых статусных переходов заказа с указанием запрещенных переходов и условий их блокировки. Результатом является проектная спецификация Агрегатов, служащая техническим заданием для последующей реализации и модульного тестирования.
4	Ревью доменной модели и план рефакторинга тактических паттернов Студент проводит разбор учебного фрагмента кода с анемичной моделью и выявляет запахи предметной области: примитивную одержимость, раздутые сервисы и утечки инфраструктурных зависимостей в домен. Инженер составляет каталог нарушений и пошаговый план рефакторинга с выделением Объектов-значений, Доменных Сервисов и Фабрик. Итогом является отчет об архитектурном техническом долге с приоритетами устранения и оценкой влияния на бизнес-ценность продукта.
5	Проектирование портов и адаптеров, экспертиза направления зависимостей Студент разрабатывает схему портов и адаптеров прототипа и распределяет компоненты между доменом, слоем приложения и инфраструктурой. Инженер проверяет направление зависимостей на диаграммах компонентов и C4, выявляя недопустимые обращения ядра к СУБД, брокеру и внешним API, и проектирует схемы доменных событий с правилами их версионирования. Результатом

№ п/п	Тематика практических занятий/краткое содержание
	является архитектурная схема и каталог событий, согласованный со смежными командами клиентской и мобильной разработки.
6	Contract-first: проектирование и экспертиза API-контрактов Студент проектирует синхронные и асинхронные контракты в нотациях OpenAPI и AsyncAPI, описывая модели запросов и ответов, коды ошибок и правила пагинации. Инженер проводит экспертизу обратной совместимости версий, выявляя ломающие изменения, и настраивает валидацию спецификаций по принятым командным правилам. Итогом является версионированный набор контрактов, пригодный для генерации клиентов и передачи командам серверной, клиентской и мобильной разработки.
7	Обоснование стратегий хранения и гарантированной доставки событий Студент сопоставляет реляционную и документоориентированную модели хранения Агрегатов применительно к отечественной СУБД и оценивает влияние выбора на транзакционные границы. Инженер анализирует сценарии сбоев, обосновывает применение Transactional Outbox, идемпотентности подписчиков и допустимых окон рассогласования данных. Результатом является расчетно-аналитическая записка с выбранными стратегиями хранения, гарантиями доставки и планом интеграционного тестирования отказоустойчивости.
8	Архитектурный паспорт в условиях импортозамещения и защита проектных решений Студент формирует архитектурный паспорт продукта, сводя воедино карту контекстов, схемы Агрегатов, контракты и ADR средствами отечественного офисного пакета по требованиям ГОСТ. Инженер обосновывает миграцию с зарубежного стека на отечественные операционные системы, СУБД и брокеры, оценивая риски, требования информационной безопасности и особенности интеграции с государственными сервисами. Итогом является публичная защита проектных решений перед стейкхолдерами с протоколом замечаний и планом их устранения.

4.3. Самостоятельная работа обучающихся.

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

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

№ п/п	Библиографическое описание	Место доступа
1	Миловидова, А. А. Предметно-ориентированное проектирование : учебное пособие / А. А. Миловидова, А. Д. Лагунова. — Москва : РТУ МИРЭА, 2024. — 71 с. — ISBN 978-5-7339-2222-5. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/421112 (дата обращения: 22.06.2026)

2	Верхова, Г. В. Предметно-ориентированное проектирование автоматизированных систем : учебно-методическое пособие / Г. В. Верхова, С. В. Акимов, А. П. Шабанов. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2023. — 49 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/426014 (дата обращения: 22.06.2026)
3	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: 22.06.2026)

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

Электронно-библиотечная система «Лань» (e.lanbook.com) – для доступа к фундаментальным учебным пособиям по архитектуре ПО.

Электронно-библиотечная система «Юрайт» (urait.ru) – для доступа к современным переводам зарубежных авторов и отечественным монографиям.

Официальная документация Postgres Professional (postgrespro.ru/docs) – сертифицированная документация отечественной СУБД.

Спецификация OpenAPI (spec.openapis.org) – стандарт описания RESTful API для интеграции с клиентскими приложениями.

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

Операционные системы – отечественные дистрибутивы Linux для развертывания инфраструктуры и тестирования переносимости кода.

Офисные пакеты – российские аналоги для оформления архитектурных паспортов и глоссариев по ГОСТ.

Среды разработки и системы контроля версий – популярные open-source IDE и корпоративные Git-платформы для версионирования кода и схем.

Инструменты моделирования – платформы для совместной работы с канбан-досками и генераторы UML/C4 диаграмм из текстового кода.

Технологический стек реализации – строго типизированные языки программирования и фреймворки для построения гексагональной архитектуры.

Брокеры сообщений – open-source очереди и стриминговые платформы для асинхронной интеграции и маршрутизации событий.

Системы управления базами данных – сертифицированные отечественные реляционные СУБД и in-memory хранилища.

Контракты и контейнеризация – стандарты описания REST/AsyncAPI и инструменты изолированного запуска сервисов для интеграционного тестирования.

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:

Зачет в 5 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 И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Директор ЦС IT - программ ВИШ О.Б. Проневич

Председатель учебно-методической
комиссии Н.А. Андриянова