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 первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Проектирование и архитектура программных систем

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Цифровая инженерия транспортных процессов

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: руководитель образовательной программы Проневич Ольга Борисовна
Дата: 08.09.2026

1. Общие сведения о дисциплине (модуле).

Целью освоения учебной дисциплины является формирование у обучающихся экспертных знаний и практических навыков по проектированию архитектуры сложных программных систем, включая выбор архитектурных стилей и паттернов, обеспечение качества системы (производительность, масштабируемость, надежность, безопасность, сопровождаемость), документирование архитектурных решений и управление техническим долгом, что позволяет создавать устойчивые и эволюционирующие data-платформы транспортного предприятия.

Дисциплина предназначена для формирования и закрепления у обучающихся компетенций, необходимых для решения следующих профессиональных задач:

Архитектурное проектирование сложных программных систем для транспортной отрасли.

Применение принципов чистого кода, SOLID, DRY, KISS при разработке.

Использование предметно-ориентированного проектирования (DDD) для моделирования предметной области.

Выбор и применение паттернов проектирования и избегание антипаттернов.

Обеспечение нефункциональных требований: производительность, масштабируемость, надежность, безопасность.

Управление техническим долгом и документирование архитектурных решений.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ОПК-6 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения;

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

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

Знать:

Принципы чистого кода, SOLID, DRY, KISS и др.: основные принципы объектно-ориентированного проектирования, критерии качества кода, рефакторинг, методики написания читаемого и поддерживаемого кода.

Принципы предметно-ориентированного проектирования (DDD): стратегическое проектирование (bounded contexts, ubiquitous language), тактическое проектирование (entities, value objects, aggregates, repositories, domain services, domain events), применение DDD для моделирования сложных предметных областей.

Уметь:

Проектировать архитектуру программных систем с использованием паттернов проектирования и принципов DDD, выполнять декомпозицию предметной области, выделять агрегаты и bounded contexts, проектировать доменные модели.

Обеспечивать качество программной системы: применять принципы чистого кода, проводить рефакторинг, оценивать и управлять техническим долгом, проектировать систему с учетом нефункциональных требований.

Владеть:

Навыками архитектурного проектирования сложных программных систем, включая выбор архитектурных стилей и паттернов, документирование архитектурных решений с использованием ADR (Architecture Decision Records) и C4 Model.

Навыками разработки масштабируемого и поддерживаемого кода с применением принципов SOLID, DDD, паттернов проектирования, а также управления техническим долгом в долгосрочной перспективе.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 3 з.е. (108 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):	80	80
В том числе:		

Занятия лекционного типа	16	16
Занятия семинарского типа	64	64

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 28 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	Тема 1. Введение в проектирование программных систем. Рассматриваемые вопросы: - Роль архитектуры в разработке ПО. - Уровни проектирования: архитектурный, системный, модульный, кодовый. - Качество программной системы: нефункциональные атрибуты. - Жизненный цикл архитектурного решения. - Роль архитектора в команде.
2	Тема 2. Принципы чистого кода и SOLID. Рассматриваемые вопросы: - Принципы чистого кода: читаемость, простота, тестируемость. - SOLID: Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion. - DRY, KISS, YAGNI. - Code Smells и рефакторинг. - Примеры применения на Python.
3	Тема 3. Паттерны проектирования (GoF). Рассматриваемые вопросы: - Порождающие паттерны: Singleton, Factory, Abstract Factory, Builder, Prototype. - Структурные паттерны: Adapter, Bridge, Composite, Decorator, Facade, Proxy. - Поведенческие паттерны: Strategy, Observer, Command, State, Template Method, Visitor. - Выбор паттерна для конкретной задачи. - Реализация паттернов в Python.
4	Тема 4. Архитектурные паттерны и стили. Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- Layered (N-tier) Architecture. - Hexagonal Architecture (Ports & Adapters). - Onion Architecture. - Clean Architecture. - CQRS и Event Sourcing. - Выбор архитектурного стиля.
5	Тема 5. Предметно-ориентированное проектирование (DDD). Рассматриваемые вопросы: - Ubiquitous Language. - Стратегическое проектирование: Bounded Context, Context Mapping. - Тактическое проектирование: Entity, Value Object, Aggregate, Repository, Domain Service, Domain Event. - Применение DDD в data-проектах. - Event Storming для моделирования.
6	Тема 6. Антипаттерны и управление техническим долгом. Рассматриваемые вопросы: - Типовые антипаттерны в проектировании и коде. - Причины возникновения технического долга. - Методы управления техническим долгом. - Оценка и приоритизация технического долга. - Стратегии погашения долга.
7	Тема 7. Нефункциональные требования и архитектура качества. Рассматриваемые вопросы: - Производительность и масштабируемость. - Надежность и отказоустойчивость. - Безопасность. - Сопровождаемость и эволюционируемость. - Архитектурные тактики для обеспечения качества.
8	Тема 8. Документирование архитектуры и управление архитектурными решениями. Рассматриваемые вопросы: - Архитектурная документация: виды и форматы. - Architecture Decision Records (ADR). - C4 Model для документирования. - Современные инструменты для документирования. - Коммуникация архитектурных решений стейкхолдерам.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Анализ архитектуры существующей системы. Рассматриваемые вопросы: - Выбор реальной системы для анализа. - Идентификация архитектурных решений. - Оценка качества архитектуры. - Выявление технического долга.

№ п/п	Тематика практических занятий/краткое содержание
2	Тема 2. Практика применения SOLID. Рассматриваемые вопросы: - Анализ кода на соответствие SOLID. - Рефакторинг кода с нарушением принципов. - Обоснование изменений. - Сравнение до и после.
3	Тема 3. Паттерны: Singleton и Factory. Рассматриваемые вопросы: - Реализация Singleton в Python. - Реализация Factory Method. - Реализация Abstract Factory. - Применение в реальных сценариях.
4	Тема 4. Паттерны: Adapter, Decorator, Facade. Рассматриваемые вопросы: - Реализация Adapter для интеграции. - Реализация Decorator для расширения функциональности. - Реализация Facade для упрощения интерфейса. - Сравнение и выбор.
5	Тема 5. Паттерны: Observer и Strategy. Рассматриваемые вопросы: - Реализация Observer (событийная модель). - Реализация Strategy для алгоритмов. - Применение в транспортном кейсе. - Сравнение подходов.
6	Тема 6. Архитектурные паттерны: Layers. Рассматриваемые вопросы: - Проектирование слоев приложения. - Реализация слоев на Python. - Преимущества и ограничения. - Рефакторинг кода под слоистую архитектуру.
7	Тема 7. Архитектурные паттерны: Hexagonal Architecture. Рассматриваемые вопросы: - Проектирование Ports & Adapters. - Реализация доменной модели. - Интеграция с внешними системами через адаптеры. - Тестирование архитектуры.
8	Тема 8. Архитектурные паттерны: Clean Architecture. Рассматриваемые вопросы: - Структура Clean Architecture. - Реализация на Python. - Применение Dependency Inversion. - Тестирование и поддерживаемость.
9	Тема 9. DDD: Ubiquitous Language и Bounded Contexts. Рассматриваемые вопросы: - Формирование единого языка. - Идентификация Bounded Contexts. - Проектирование контекстов. - Документирование.
10	Тема 10. DDD: Entity и Value Objects. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- Проектирование Entity. - Проектирование Value Objects. - Сравнение и выбор. - Реализация в Python.
11	Тема 11. DDD: Aggregates и Repositories. Рассматриваемые вопросы: - Проектирование Aggregate. - Реализация Repository. - Транзакционная согласованность. - Паттерны работы с агрегатами.
12	Тема 12. DDD: Domain Services и Domain Events. Рассматриваемые вопросы: - Реализация Domain Service. - Реализация Domain Events. - Интеграция с Event Sourcing. - Применение в транспортном кейсе.
13	Тема 13. CQRS и Event Sourcing. Рассматриваемые вопросы: - Проектирование CQRS. - Реализация Command and Query. - Реализация Event Sourcing. - Хранение событий и восстановление состояния.
14	Тема 14. Антипаттерны: идентификация и устранение. Рассматриваемые вопросы: - Анализ кода на наличие антипаттернов. - God Object, Spaghetti Code, Golden Hammer. - Рефакторинг. - Документирование изменений.
15	Тема 15. Управление техническим долгом. Рассматриваемые вопросы: - Оценка технического долга. - Приоритизация задач по погашению долга. - Интеграция в процесс разработки. - Стратегии управления.
16	Тема 16. Проектирование API First. Рассматриваемые вопросы: - Принцип API First. - Проектирование API с OpenAPI. - Генерация кода из спецификации. - Документирование API.
17	Тема 17. Архитектура распределенных транзакций. Рассматриваемые вопросы: - Проблемы распределенных транзакций. - Saga Pattern. - 2PC (Two-Phase Commit). - Реализация Saga на Python.
18	Тема 18. Архитектура событийно-ориентированных систем. Рассматриваемые вопросы: - Проектирование событий. - Брокеры сообщений (Kafka).

№ п/п	Тематика практических занятий/краткое содержание
	- Гарантии доставки. - Масштабирование.
19	Тема 19. Архитектура микросервисов. Рассматриваемые вопросы: - Декомпозиция на микросервисы. - Межсервисное взаимодействие. - Service Discovery и API Gateway. - Управление данными.
20	Тема 20. Архитектура безопасности. Рассматриваемые вопросы: - Аутентификация и авторизация. - OAuth2, OpenID Connect, JWT. - Защита API. - Security by Design.
21	Тема 21. Architecture Decision Records (ADR). Рассматриваемые вопросы: - Структура и формат ADR. - Написание ADR для реального решения. - Управление ADR. - Интеграция в процесс.
22	Тема 22. Документирование с C4 Model. Рассматриваемые вопросы: - Создание C4 диаграмм. - Инструменты: Structurizr. - Документирование архитектуры. - Презентация архитектуры.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение фундаментальных работ по архитектуре ПО (Мартин Фаулер, Роберт Мартин, Эрик Эванс).
2	Самостоятельная реализация паттернов проектирования на Python.
3	Написание ADR для архитектурных решений.
4	Подготовка к защите итогового проекта.
5	Подготовка к промежуточной аттестации.
6	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа

1	Алпатов, А. Н. Архитектура, проектирование и разработка программных средств : учебное пособие / А. Н. Алпатов, И. Е. Рогов. — Москва : РТУ МИРЭА, 2023. — 120 с. — ISBN 978-5-7339-1972-0.	https://e.lanbook.com/book/386189
2	Гусев, К. В. Введение в профессиональную деятельность : учебное пособие / К. В. Гусев. — Москва : РТУ МИРЭА, 2022. — 88 с.	https://e.lanbook.com/book/239945

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

https://habr.com/ru - база знаний в виде статей, обзоров

http://library.miit.ru/ - электронно-библиотечная система научно-технической библиотеки МИИТ

https://superset.apache.org/user-docs/intro/#documentation – документация Apache Superset

https://www.postgresql.org/ - официальная страница PostgreSQL

https://yandex.cloud/ru/docs/datalens/ - официальная документация Yandex Datalens

https://docs.docker.com/ - официальная документация Docker

https://www.metabase.com/docs/latest/ - официальная документация Metabase

https://tableau.pro/ - официальный сайт Tableau с лучшими практиками проектирования дашбордов

https://revealthedata.com/blog/ - Блог о визуализации данных и BI-системах

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

MS Office (Excel, Word, PowerPoint)
Apache Superset
Yandex Datalens
Metabase
PostgreSQL
Docker
Apache Airflow
Браузер Chrome / Яндекс браузер

Draw.io

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

Компьютер преподавателя с доступом к сети Интернет

Компьютеры студентов с установленным Python-окружением и табличным процессором

Доступ к учебной базе событийных данных и экспериментальным выборкам

Доступ к PostgreSQL и BI-системе для визуализации результатов

Доступ к облачной виртуальной машине или Docker-контейнерам для воспроизводимого анализа

Проектор и экран для проектора

Маркерная доска

9. Форма промежуточной аттестации:

Экзамен в 7 семестре.

10. Оценочные материалы.

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

Авторы:

руководитель образовательной программы О.Б. Проневич

Согласовано:

Директор Д.В. Паринов

Руководитель образовательной программы О.Б. Проневич

Председатель учебно-методической комиссии Д.В. Паринов