МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
высшего образования - программа бакалавриата
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Разработка корпоративных приложений

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: 5665
Подписал: заведующий кафедрой Нутович Вероника Евгеньевна
Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Целями освоения данной дисциплины являются получение базовых, теоретических знаний о разработке корпоративных приложений с использованием паттернов проектирования, практики их применения и реализации на языках высокого уровня.

Основной задачей изучения дисциплины является формирование базовых представлений, знаний и умений о ключевых принципах организации кода, порождающих, структурных паттернах и паттернах поведения, шаблонах организации объектно-ориентированного кода, об антипаттернах, причинах их возникновения, связанных с ними проблемами и путями решения.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ПК-6 - Способен проектировать разрабатываемый программный продукт на основе принятых методологий и практик для корпоративного рынка.;

ПК-7 - Способен разрабатывать программные продукты под разные платформы для корпоративного рынка.

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

Знать:

- ключевые принципы хорошего кода;
- UML диаграммы и основные компоненты;
- различные типы паттернов и практики их применения;
- принципы SOLID.

Уметь:

- применять техники рефакторинга в существующем коде;
- применять традиционные подходы к организации кода;
- применять традиционные принципы хорошего кода.

Владеть:

- навыками рефакторинга с применением различных типов паттернов;
- навыками применения принципов SOLID.

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):	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 44 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	Введение в паттерны проектирования. Рассматриваемые вопросы: - знакомство с паттернами проектирования. Ключевые принципы хорошего кода; - классификация паттернов проектирования. «Банда четырех» (Gang of Four, GoF); - антипаттерны в программировании и проектировании архитектуры; - рефакторинг.

№ п/п	Тематика лекционных занятий / краткое содержание
2	Unified Modeling Language. Рассматриваемые вопросы: - унифицированный язык моделирования (Unified Modeling Language, UML); - UML диаграммы.
3	Традиционные подходы к организации кода. Порождающие паттерны. Рассматриваемые вопросы: - фабричный метод (Factory Method); - абстрактная фабрика (Abstract Factory); - одиночка (Singleton); - прототип (Prototype); - строитель (Builder).
4	Традиционные подходы к организации кода. Структурные паттерны. Рассматриваемые вопросы: - адаптер (Adapter); - декоратор (Decorator); - заместитель (Proxy); - компоновщик (Composite); - мост (Bridge); - приспособленец (Flyweight); - фасад (Facade).
5	Традиционные подходы к организации кода. Паттерны поведения. Рассматриваемые вопросы: - интерпретатор (Interpreter); - шаблонный метод (Template Method); - итератор (Iterator); - команда (Command); - наблюдатель (Observer); - посетитель (Visitor); - посредник (Mediator); - состояние (State); - стратегия (Strategy); - хранитель (Memento); - цепочка обязанностей (Chain of Responsibility).
6	Дополнительные шаблоны организации объектно-ориентированного кода. Рассматриваемые вопросы: - основные шаблоны проектирования: Delegation (Делегирование), Interface (Интерфейс), Abstract Superclass (Абстрактный суперкласс),Interface and Abstract Class (Интерфейс и абстрактный класс), Immutable (Неизменный), Marker Interface (Маркер-интерфейс); - порождающие шаблоны проектирования: Object Pool (Пул объектов); - разделяющие шаблоны проектирования: Filter (Фильтр), Read-Only Interface (Интерфейс, предназначенный только для чтения); - структурные шаблоны проектирования: Dynamic Linkage (Динамическая компоновка), Virtual Proxy (Виртуальный заместитель), Cache Management (Упрамение кэшем); - поведенческие шаблоны проектирования: Command (Команда), Little Language (Малый язык), Snapshot (Моментальный снимок), Null Object (Нулевой объект); - шаблоны проектирования для конкурирующих операций: Single Threaded Execution (Однопоточное выполнение), Lock Object (Объект блокировки), Guarded Suspension (Охраняемая приостановка), Balking (Отмена), Scheduler (Планировшик), ReadjWrite Lock (Блокировка чтения/записи), Producer/Consumer (Производитель-потребитель), Two-Phase Termination (Двухфазное завершение), Double Buffering (Двойная буферизация), Asynchronous Processing (Асинхронная обработка), Future (Будущее).

№ п/п	Тематика лекционных занятий / краткое содержание
7	Ключевые принципы хорошего кода. Рассматриваемые вопросы: - «Не повторяй себя» (DRY – Don’t repeat yourself); - «Делайте вещи проще» (KISS – keep it simple stupid); - «Вам это не понадобится» (YAGNI – You ain’t gonna need it); - «Глобальное проектирование прежде всего» (BDUF – Big Design Up Front); - «Избегайте преждевременной оптимизации» (APO – Avoid Premature Optimization); - принципы SOLID: принцип единственной ответственности (single responsibility principle), принцип открытости/закрытости (open-closed principle), принцип подстановки Барбары Лисков (Liskov substitution principle), принцип разделения интерфейсов (interface segregation principle), принцип инверсии зависимостей (dependency inversion principle).
8	Принципы SOLID. Рассматриваемые вопросы: - принципы SOLID: принцип единственной ответственности (single responsibility principle), принцип открытости/закрытости (open-closed principle), принцип подстановки Барбары Лисков (Liskov substitution principle), принцип разделения интерфейсов (interface segregation principle), принцип инверсии зависимостей (dependency inversion principle).
9	Антипаттерны архитектуры и проектирования. Рассматриваемые вопросы: - ад зависимостей (Dependency hell); - божественный объект (God object); - золотой молоток (Golden hammer); - одиночка (Singleton); - полтергейст (Poltergeist); - проблема йо йо (Yo-yo problem).

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Unified Modeling Language. В результате выполнения лабораторной работы студент получает навык UML моделирования компонентов архитектуры, алгоритмов и элементов программного кода.
2	Рефакторинг. В результате выполнения лабораторной работы студент получает навык рефакторинга программного кода на типовых кейсах.
3	Порождающие паттерны. В результате выполнения лабораторной работы студент получает навык применения порождающих паттернов на типовых кейсах.
4	Структурные паттерны. В результате выполнения лабораторной работы студент получает навык применения структурных паттернов на типовых кейсах.
5	Паттерны поведения. В результате выполнения лабораторной работы студент получает навык применения поведенческих паттернов на типовых кейсах.
6	Организация объектно-ориентированного кода. В результате выполнения лабораторной работы студент знакомится с подходами организации объектно-ориентированного кода на типовых проектах.

№ п/п	Наименование лабораторных работ / краткое содержание
7	Ключевые принципы хорошего кода. В результате выполнения лабораторной работы студент знакомится с принципиальными подходами к программной инженерии профессионального качества.
8	Антипаттерны. В результате выполнения лабораторной работы студент знакомится как с антипаттернами, так и практическими кейсами, в которых паттерны становятся антипаттернами.
9	Организация объектно-ориентированного кода. В результате выполнения лабораторной работы студент знакомится с подходами организации объектно-ориентированного кода на типовых проектах.
10	Принципы SOLID. В результате выполнения лабораторной работы студент получает навык применения принципов SOLID на типовых кейсах.

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	Э. Гамма. Паттерны объектно-ориентированного проектирования. - Санкт-Петербург : Питер, 2021. - 448 с. - ISBN 978-5-4461-1595-2.	URL: https://ibooks.ru/bookshelf/371734/reading (дата обращения: 25.05.2023). - Текст: электронный.
2	Приемы объектно ориентированного проектирования. Паттерны проектирования : справочник / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. — Москва : ДМК Пресс, 2007. — 368 с. — ISBN 5-93700-023-4. — Текст : электронный // Лань : электронно-библиотечная система. Справочное пособие	URL: https://e.lanbook.com/book/1220 (дата обращения: 25.05.2023). — Режим доступа: для авториз. пользователей.
3	Фримен Э. Head First. Паттерны проектирования. Обновленное юбилейное издание. — (Серия «Head First O’Reilly»). / Э. Фримен, Э. Робсон, К. Сьерра, Б. Бейтс. -	URL: https://ibooks.ru/bookshelf/377150/reading (дата обращения: 25.05.2023)

	Санкт-Петербург : Питер, 2018. - 656 с. - ISBN 978-5-496-03210-0.	
4	Тепляков С.В. Паттерны проектирования на платформе .NET / С.В. Тепляков. - Санкт-Петербург : Питер, 2015. - 320 с. - ISBN 978-5-496-01649-0.	URL: https://ibooks.ru/bookshelf/344646/reading (дата обращения: 25.05.2023).
5	Арораа Гаурав. Паттерны проектирования для C# и платформы .NET Core. — (Серия «Для профессионалов»). - Санкт-Петербург : Питер, 2021. - 352 с. - ISBN 978-5-4461-1523-5.	URL: https://ibooks.ru/bookshelf/374458/reading (дата обращения: 25.05.2023).
6	Пикус, Ф. Идиомы и паттерны проектирования в современном C++ : руководство / Ф. Пикус ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2020. — 452 с. — ISBN 978-5-97060-786-2. — Текст : электронный // Лань : электронно-библиотечная система. —	URL: https://e.lanbook.com/book/140598 (дата обращения: 25.05.2023). — Режим доступа: для авториз. пользователей.
7	Бабич А.В. UML: Первое знакомство / А.В. Бабич. - Москва : Национальный Открытый Университет ИНТУИТ, 2016. - 208 с. - ISBN 978-5-94774-878-9.	URL: https://ibooks.ru/bookshelf/362747/reading (дата обращения: 25.05.2023).

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

Электронно-библиотечная система Научно-технической библиотеки РУТ(МИИТ) (http://library.miit.ru/)

Электронно-библиотечная система издательства «Лань» (http://e.lanbook.com/)

Электронно-библиотечная система ibooks.ru (http://ibooks.ru/)

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

Браузер Microsoft Internet Explorer или его аналоги

Пакет офисных программ Microsoft Office или его аналоги

.NET 6

Java 17

Microsoft Visual Studio CE

JetBrains IntelliJ IDEA Community Edition

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 практических занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Зачет 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» Е.А. Заманов

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Председатель учебно-методической
комиссии Н.А.Клычева