

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))



Рабочая программа дисциплины (модуля),
как компонент образовательной программы
специализированного высшего образования
по направлению подготовки
27.04.04 Управление в технических системах,
утвержденной первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Программная инженерия

Направление подготовки: 27.04.04 Управление в технических системах

Направленность (профиль): Интеллектуальное управление 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 РУТ (МИИТ)
ID подписи: 2053
Подписал: заведующий кафедрой Баранов Леонид Аврамович
Дата: 01.06.2026

1. Общие сведения о дисциплине (модуле).

Целью дисциплины является формирование у студентов способности разрабатывать комплект конструкторской документации автоматизированной системы управления технологическими процессами.

Задача: провести исследование существующих архитектурных подходов к повышению отказоустойчивости ПО для управления техническими системами.

В результате изучения данной дисциплины студент должен:

- иметь ясные представления в области различных методологий организации процессов производства программного обеспечения, методы сбора и анализа требований, ресурсного планирования, модели жизненного цикла программного продукта, планирование хода работ над проектом;

- знать методы управления проектами, создавать набор необходимых артефактов в соответствии со стандартами, провести ресурсное планирование, построить календарный план проекта;

- уметь разрабатывать необходимую документацию, сопровождающую все этапы проекта, планировать и осуществлять контроль необходимых контрольных точек проекта, разрабатывать календарный план проекта в соответствии с выбранной моделью жизненного цикла.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ПК-4 - Способен применять современные теоретические и экспериментальные методы разработки математических моделей исследуемых объектов и процессов, относящихся к профессиональной деятельности по направлению подготовки;

ПК-16 - Способен разрабатывать комплект конструкторской документации автоматизированной системы управления технологическими процессами.

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

Знать:

- Знать методы управления проектами.

Уметь:

- Уметь разрабатывать необходимую документацию, сопровождающую все этапы проекта.

- Уметь планировать и осуществлять контроль необходимых контрольных точек проекта.

Владеть:

- навыками разработки документации по разработке и сопровождению проектов.

- Навыками разработки календарного плана проекта в соответствии с выбранной моделью жизненного цикла.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 6 з.е. (216 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 №2
Контактная работа при проведении учебных занятий (всего):	48	48
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	16	16

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 168 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	<p>Программная инженерия как научно-практическая дисциплина: истоки, место и структура в системе компьютерных наук.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- Предпосылки возникновения программной инженерии (далее ПИ).- Место ПИ среди компьютерных наук.- Составляющие ПИ.- Функции программной инженерии.- Дисциплины программной инженерии.- Составляющие ПИ как науки.- Определение целевых объектов ПИ.- Программная инженерия как инженерная дисциплина.- Инфраструктура программной инженерии.- Стандарты программной инженерии.- Применение компонентов повторного использования.- Программная инженерия как дисциплина управления.- Стандарт РМВОК.- Экономическая дисциплина программной инженерии
2	<p>Ядро знаний SWEBOOK: углубленный анализ ключевых областей — от требований к сопровождению.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- Ядро знаний SWEBOOK.- Основные области знаний SWEBOOK.- Требования к ПО.- Инженерия требований.- Составляющие инженерии требований.- Проектирование программного обеспечения.- Архитектура ПО.- Шаблоны проектирования.- Инструменты проектирования.- Конструирование ПО.- Стандарты при конструировании.- Управление конструированием.- Тестирование ПО.- Основные понятия тестирования.- Методы тестирования.- Типы тестирования.- Управление тестированием.- Сопровождение ПО.- Категории сопровождения.- Методики сопровождения ПО.
3	<p>Управление инженерными практиками: конфигурация, измерения, процессы и инструментарий создания ПО.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- Методики сопровождения ПО.- Управление конфигурацией ПО.

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - Составляющие управления конфигурацией. - Управление процессом конфигурации. - Идентификация конфигурации ПО. - Контроль конфигурации ПО. - Учет статуса и аудит конфигурации. - Управление выпусками и доставкой. - Управление инженерией ПО. - Организационное управление. - Управление программным проектом. - Инженерия измерения ПО. - Процесс программной инженерии. - Определение процессов ЖЦ. - Оценка процессов ЖЦ. - Измерение процессов и продукта. - Методы и инструменты ПИ.
4	<p>Стандартизация и адаптация жизненного цикла: модели процессов и философия гибкой разработки.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Стандарт и модели жизненного цикла. - Жизненный цикл. - Структура процессов ЖЦ. - Классификация процессов. - Классификация моделей ЖЦ. - Методы гибкой разработки.
5	<p>Системная инженерия требований: классификация, спецификация и управление изменениями в сложных программных комплексах.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Методы гибкой разработки. - Требования к ПО. - Классификация требований. - Функциональные и нефункциональные требования. - Связь между требованиями. - Спецификация требований. - Процесс инженерии требований. - Определение и анализ требований. - Процесс изменения требований.
6	<p>Визуальное моделирование программных систем: язык UML, структурный и поведенческий анализ.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Моделирование программных систем. - Моделирование в разработке ПО. - Цели моделирования. - Представления системы. - Язык моделирования UML. - Модели взаимодействия. - Модель процесса. - Диаграммы последовательностей. - Структурные модели. - Модели поведения.

№ п/п	Тематика лекционных занятий / краткое содержание
7	<p>Архитектурное проектирование: стили, шаблоны и представления архитектуры программных систем.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Архитектура программных систем. - Архитектурные решения. - Требования и архитектура. - Представления архитектуры. - Создание представлений. - Инструменты создания представлений. - Архитектурные шаблоны. - Многослойная архитектура. - Конвейерная архитектура.
8	<p>Объектно-ориентированная парадигма и каталоги шаблонов проектирования (GoF) для создания гибких архитектур.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Объектно-ориентированное проектирование. - Шаблоны проектирования. - Концепции ООП. - Процесс ООП. - Шаблоны проектирования. - Архитектурный шаблон: ActiveRecord. - Порождающий шаблон: Singleton. - Порождающий шаблон: Builder. - Структурный шаблон: Bridge. - Структурный шаблон: Decorator. - Поведенческий шаблон: Iterator. - Поведенческий шаблон: Observer.
9	<p>Эволюция парадигм программирования: сравнительный анализ императивного, декларативного и объектно-ориентированного подходов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Парадигмы программирования. - Место парадигмы в разработке. - Классификация парадигм программирования. - Декларативное программирование. - Императивное программирование. - Функциональное программирование. - Логическое программирование. - Структурное программирование. - Объектно-ориентированное программирование.
10	<p>Современные архитектурные стили и парадигмы: компонентный подход, АОП и сервис-ориентированная архитектура (SOA).</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Причины появления компонентов. - Компонентно-ориентированное программирование. - Особенности парадигмы. - Характеристики компонентов. - Отличия компонентов от объектов. - Интерфейсы компонентов. - Компонентная модель. - Аспектно-ориентированное программирование.

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - Сквозная функциональность. - Сквозная функциональность. - Внедрение аспектов. - Сервис-ориентированная архитектура. - Веб-сервисы.
11	<p>Формальные и семантические основы языков программирования: система типов, полиморфизм и метапрограммирование.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Языки программирования. - Метапрограммирование. - Классификация языков программирования. - Различия между синтаксисом и семантикой. - Система типов. - Статическая и динамическая типизация. - Утиная типизация. - Полиморфизм. - Метапрограммирование. - Рефлексия. - Варианты использования рефлексии. DSL.
12	<p>Стратегии и процессы тестирования ПО: интеграция в жизненный цикл и разработка через тестирование (TDD).</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Тестирование программного обеспечения. - Основные понятия тестирования. - Тестирование ПО. - Виды тестирования. - Место тестирования в разработке ПО. - Процесс тестирования. - Фазы тестирования. - Тестирование при разработке. - Разработка через тестирование.
13	<p>Обеспечение качества и корректности ПО: верификация, валидация, инспекции и формальные методы проверки моделей.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Верификация и валидация. - Методы верификации. - Необходимость верификации. - Инспекции. - Преимущества инспекций. - Области проверки кода. - Парное программирование. - Автоматизация инспекций. - Формальные методы. - Логика Хоара. - Применение языков спецификации. - Математический аппарат проверки моделей.
14	<p>Управление эволюцией и сопровождением ПО: рефакторинг, реинженерия и борьба с техническим долгом.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Эволюция ПО.

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - Процесс разработки и эволюции. - Режимы внесения изменений. - Эволюция ПО в гибкой методологии. - Динамика эволюции ПО. - Сопровождение ПО. - Проблемы добавления функциональности. - Оценка процесса сопровождения. - Процессы реинженерии. - Рефакторинг. - Признаки «плохого» кода.
15	<p>Управление конфигурацией и версионирование: организация изменений и функциональность современных СУВ.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Признаки «плохого» кода. - Составляющие управления конфигурацией. - Идентификация конфигурации ПО. - Управление изменениями. - Запросы на изменение. - Оценка запросов на изменение. - Функциональность систем управления версиями. - Понятия управления версиями. - Организация управления версиями. - Характеристики СУВ.
16	<p>Документирование в программной инженерии: системная, пользовательская документация и концепция грамотного программирования.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - Документирование ПО. - Типы документации. - Документация в гибкой методологии. - Структура документации. - Форматы документации. - Документирование процессов разработки. - Объем документации на процесс разработки. - Пользовательская документация. - Виды пользовательской документации. - Системная документация. - Генерация документации. - Грамотное программирование.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	<p>UML. Диаграмма классов</p> <p>В результате выполнения студент умеет анализировать предметную область, выявлять классы-кандидаты, определять их атрибуты и операции, а также устанавливать статические связи (ассоциации, наследование, агрегацию, композицию) между ними для построения концептуальной и логической модели системы на основе диаграммы классов.</p>

№ п/п	Наименование лабораторных работ / краткое содержание
2	UML. Диаграммы объектов. В результате выполнения студент умеет создавать снимки состояния системы в конкретный момент времени, используя диаграммы объектов для верификации разработанной диаграммы классов и демонстрации примеров взаимодействия экземпляров классов с конкретными значениями атрибутов.
3	UML. Диаграмма вариантов использования. В результате выполнения студент умеет проводить текстуальный анализ требований, идентифицировать действующих лиц (актеров) и прецеденты (варианты использования), а также моделировать функциональные требования к системе с помощью диаграммы вариантов использования, определяя границы системы и отношения между ее элементами.
4	UML. Диаграмма последовательности. В результате выполнения студент умеет моделировать временную последовательность обмена сообщениями между объектами в рамках конкретного сценария варианта использования, детализируя процессы взаимодействия с помощью диаграмм последовательности.
5	UML. Диаграмма сотрудничества. В результате выполнения студент умеет моделировать структурные и поведенческие аспекты взаимодействия объектов, акцентируя внимание на связях между ними и порядке передачи сообщений для реализации определенного варианта использования.
6	UML. Диаграмма схем состояний. В результате выполнения студент умеет анализировать жизненный цикл отдельных объектов системы, определяя их возможные состояния, события, переводящие объект из одного состояния в другое, и действия, выполняемые в этих состояниях, с последующим построением диаграммы состояний.
7	UML. Диаграмма деятельности. UML. Компонентная диаграмма. В результате выполнения студент умеет моделировать логику и последовательность шагов бизнес-процессов или алгоритмов выполнения операций, используя диаграммы деятельности для описания потоков управления и обработки данных; разрабатывать физическое представление системы путем моделирования ее компонентов (файлов, библиотек, модулей) и их зависимостей, показывая организацию кода на этапе реализации.
8	UML. Диаграмма размещения. В результате выполнения студент умеет проектировать физическую архитектуру системы, определяя вычислительные узлы (серверы, рабочие станции, устройства) и размещение на них компонентов, с учетом требований к распределенным и встроенным системам управления.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение дополнительной 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	Программная инженерия. Интеграционный подход к разработке Турнецкая Е. Л., Аграновский А. В. Учебник Издательство "Лань", - 2-е изд., стер. - 216 с. - ISBN 978-5-507-50848-8 , 2025	https://reader.lanbook.com/book/480161
2	Методы научных исследований в программной инженерии Пантелеев Е. Р. Учебное пособие 2-е изд., стер.- Санкт-Петербург: Лань. - 136 с. -ISBN 978-5-8114-6781-5 , 2021	https://reader.lanbook.com/book/152439#2

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

Официальный сайт РУТ (МИИТ) (<https://www.miit.ru/>).

Научно-техническая библиотека РУТ (МИИТ) (<http://library.miit.ru>).

Образовательная платформа «Юрайт» (<https://urait.ru/>).

Общие информационные, справочные и поисковые системы «Консультант Плюс», «Гарант».

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

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

<http://htbs-miit.ru:9999/> - Сайт дистанционного обучения Московского государственного университета путей сообщения (МИИТ);

<http://edu.emiit.ru/> - портал ДОТ ИЭФ МИИТ.

Intuit.ru – интернет университет информационных технологий;

<http://www.knigafund.ru>,

Mirknig.com – электронные книги.

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

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

Microsoft Internet Explorer (или другой браузер).

Операционная система Microsoft Windows.

Microsoft Office.

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.

9. Форма промежуточной аттестации:

Зачет во 2 семестре.

10. Оценочные материалы.

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

Авторы:

доцент, доцент, к.н. кафедры
"Интеллектуальное управление и
информационная безопасность в
высокоавтоматизированных
транспортных системах" Института
железнодорожного транспорта

Л.Н. Логинова

Согласовано:

Заведующий кафедрой УиЗИ

Л.А. Баранов

Председатель учебно-методической
комиссии

С.В. Володин