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)

Тестирование

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).

Цель дисципилны «Тестирование» заключается в формировании у студентов знаний и умений в области тестирования и обеспечения качества программного обеспечения.

Задачи данной дисциплины:

- формирование знаний в области видов тестирования;
- формирование знаний в области жизненного цикла тестирования;
- формирование умения по работе с различными инструментами тестирования;
- формирование умений в области проведения различных видов тестирования.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ПК-2 - Способен проводить тестирование разработанного программного продукта с использованием инструментов функционального, модульного и интеграционного тестирования.;

ПК-3 - Способен разрабатывать и разворачивать программные продукты используя инструменты CI/CD в рамках методологии DevOps.;

ПК-4 - Способен разрабатывать программные продукты используя инструменты поддержки процесса разработки в соответствии с гибкими методологиями разработки.

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

Знать:

- жизненный цикл тестирования программного обеспечения;
- виды тестирвоания;
- методы тестирования;
- техники тестирования.

Уметь:

- тестировать программное обеспечение с использованием различных техник.
- тестировать программное обеспечение с использованием различных методов.

Владеть:

- навыком проведения различных видов тестирования программного обеспечения с помощью различных инструментов тестирования с применением методов и техник тестирования.

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	Введение в тестирование ПО. Рассматриваемые вопросы: - что такое тестирование, цели и задачи тестировани; - почему тестирование важно, последствия низкого качества ПО; - роль и качества тестировщика; - 7 принципов тестирования; - тестирование vs отладка; - тестирование в разных моделях разработки (Waterfall, V-model, Agile - кратко).
2	Жизненный цикл тестирования (STLC) и его связь с жизненным циклом разработки ПО (SDLC). Рассматриваемые вопросы: - этапы STLC: анализ, планирование, подготовка, выполнение, завершение; - критерии начала и завершения этапов; - V-модель как пример интеграции тестирования; - концепция "Shift-Left": тестирование на ранних этапах.
3	Виды тестирования. Рассматриваемые вопросы: - функциональное тестирование; - нефункциональное тестирование; - структурное тестирование (белый ящик); - изменения: дымовое (smoke), санитарное (sanity), регрессионное (regression); - подтверждающее тестирование (re-testing).
4	Уровни функционального тестирования Рассматриваемые вопросы: - unit тестирование; - integration тестирование; - acceptance тестирование; - system тестирование; - end-to-end тестирование.
5	Методы проектирования тестов. Рассматриваемые вопросы: - метод тестирование черного ящика; - метод тестирвоание серого ящика; - метод тестирование белого ящика; - применимость методов на разных уровнях тестирования.
6	Техники тест-дизайна. Рассматриваемые вопросы: - цель и важность техник тест-дизайн; - техника классов эквивалентности; - техника анализа граничных значений; - практические примеры применения для полей ввода и форм.
7	Техники тест-дизайна. Продолжение. Рассматриваемые вопросы: - техника таблиц принятия решений; - техника диаграмм состояний и переходов; - тестирование на основе сценариев использования; - применение техник для сложной логики и бизнес-процессов.
8	Нефункциональное тестирование. Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- тестирование производительности (нагрузочное, стрессовое, объемное), метрики, основные подходы; - тестирование юзабилити (UI/UX), принципы и эвристики; - тестирование защищенности; - инсталляционное и конфигурационное тестирование; - тестирование надежности; - тестирование локализации и интернационализации; - тестирование совместимости; - тестирование доступности.
9	Управление дефектами. Рассматриваемые вопросы: - что такое дефект, жизненный цикл дефекта; - как написать качественный баг-репорт: структура, обязательные поля, характеристики; - различие между приоритетом и важностью; - системы отслеживания ошибок (Bug Tracking Systems - BTS): назначение, примеры.
10	Управление тестированием. Рассматриваемые вопросы: - тест-кейс: структура и атрибуты; - чек-листы и их отличие от тест-кейсов; - тестовые наборы (Test Suites); - матрица покрытия требований; - тестовая документация: тест-план (основные разделы), отчеты о тестировании; - основные метрики тестирования; - системы управления тестированием (Test Management Systems - TMS): назначение, примеры.
11	Управление тестированием. Продолжение. Рассматриваемые вопросы: - тестовая документация: тест-план (основные разделы), отчеты о тестировании; - основные метрики тестирования; - системы управления тестированием (Test Management Systems - TMS): назначение, примеры.
12	Тестирование в гибких методологиях (Agile Testing) Рассматриваемые вопросы: - роль и место тестировщика в Scrum и Kanban; - планирование тестирования в спринте; - работа с User Stories и критериями приемки; - кросс-функциональность команды и "Все отвечают за качество"; - коммуникация в Agile команде; - концепция тестовой пирамиды.
13	Введение в API Testing. Рассматриваемые вопросы: - что такое API, зачем его тестировать; - основные понятия REST и SOAP; - протокол HTTP: методы (GET, POST и др.), статусы ответов; - структура запроса и ответа (headers, body, parameters); - инструменты для тестирования API; - отличия и преимущества тестирования на уровне API по сравнению с UI.
14	Инструменты тестирования. Рассматриваемые вопросы: - классификация инструментов по задачам; - краткий обзор популярных инструментов в каждой категории; - как выбрать подходящий инструмент.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Написание тест-кейсов. В результате выполнения практических работ студент получит навыки разработки тест-кейсов для тестирования программного обеспечения.
2	Применение техник тест-дизайна: классы эквивалентности и граничные значения. В результате выполнения практических работ студент получит навыки применения техник EQP и BVA для эффективного проектирования тест-кейсов для полей ввода и форм.
3	Применение техник тест-дизайна: таблицы решений и диаграммы состояний. В результате выполнения практических работ студент получит навыки применения Decision Tables для тестирования сложной бизнес-логики и State Transition Diagrams для тестирования логики с переходами состояний.
4	Написание качественного баг-репорта. В результате выполнения практических работ студент научится находить дефекты в тестовом приложении и оформлять их в виде четких, воспроизводимых баг-репортов, определять приоритет и важность.
5	Работа с системой отслеживания ошибок (Bug Tracking System). В результате выполнения практических работ студент получит навыки работы с одной из BTS: создание, редактирование, комментирование, смена статусов баг-репортов.
6	Работа с системой управления тестированием (Test Management System). В результате выполнения практических работ студент получит навыки работы с одной из TMS: создание и организация тест-кейсов, создание тест-сьютов, выполнение тестов, привязка тестов к требованиям.
7	Тестирование API с использованием Postman. В результате студент получит представление о том, как работают инструменты UI автоматизации (например, Cypress или Selenium), увидит пример выполнения автоматизированных тестов, возможно, выполнит очень простую запись/запуск готового скрипта.
8	Обзор и базовое взаимодействие с инструментом UI автоматизации. В результате студент получит представление о том, как работают инструменты UI автоматизации, увидит пример выполнения автоматизированных тестов, возможно, выполнит очень простую запись/запуск готового скрипта.
9	Базовое тестирование производительности с Apache JMeter. В результате студент получит представление о том, как выглядят инструменты для нагрузочного тестирования, сможет запустить готовый тестовый план для проверки доступности или базовой нагрузки.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Работа с лекционным материалом
2	Работа с литературой
3	Текущаяя подготовка к занятиям
4	Подготовка к промежуточной аттестации.

5	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	Игнатьев, А. В. Тестирование программного обеспечения / А. В. Игнатьев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 56 с. — ISBN 978-5-507-45425-9.	https://e.lanbook.com/book/269873 (дата обращения: 01.11.2022)
2	Барнум, К. М. Основы юзабилити-тестирования / К. М. Барнум ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2022. — 408 с. — ISBN 978-5-97060-960-6.	https://e.lanbook.com/book/241163 (дата обращения: 01.11.2022)
3	Аграновский, А. В. Тестирование веб-приложений : учебное пособие / А. В. Аграновский. — Санкт-Петербург : ГУАП, 2020. — 155 с. — ISBN 978-5-8088-1515-5.	https://e.lanbook.com/book/216533 (дата обращения: 01.11.2022)
4	Миронов, А. И. Тестирование и верификация программного обеспечения: Практикум : учебное пособие / А. И. Миронов, С. М. Трушин, А. А. Петренко. — Москва : РТУ МИРЭА, 2022. — 65 с.	https://e.lanbook.com/book/240095 (дата обращения: 01.11.2022)
5	Попова, Ю. Б. Тестирование и отладка программного обеспечения : учебное пособие / Ю. Б. Попова. — Минск : БНТУ, 2020. — 66 с. — ISBN 978-985-583-056-7.	https://e.lanbook.com/book/248642 (дата обращения: 01.11.2022)

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/)

Документация по Selenium (https://www.selenium.dev/documentation/)

Документация по Cypress (https://docs.cypress.io/)

Документация по JMeter (https://jmeter.apache.org/usermanual/get-started.html)

Документация по Postman (https://learning.postman.com/docs/getting-started/introduction/)

Документация по Appium (https://appium.io/docs/en/about-appium/intro/)
Документация по Cypress (https://docs.cypress.io/)

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

JetBrains IntelliJ IDEA CE
Java 17
Selenium
Cypress
JMeter
Postman
Appium
Пакет офисных приложений
Браузер с доступом в интернет

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:

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» И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Председатель учебно-методической
комиссии Н.А. Андриянова