

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Лельчицкий Игорь Давыдович

Должность: и.о. проректора по образовательной деятельности

Дата подписания: 16.06.2026 09:51:50

Уникальный программный ключ:

aa5b5ee17d97a2e4d94e98e995320af94f043ce2

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Тверской государственный университет»

Утверждаю:

Руководитель ООП

С.М.Дудаков

2023 г.



Рабочая программа дисциплины (с аннотацией)

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Направление подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль)

Искусственный интеллект и анализ данных

Для студентов 4 курса

очная форма

Составитель: ООО «РЕД СОФТ», ведущий инженер по качеству Логинов Денис

Александрович

Тверь, 2023

I. Аннотация

1. Цель и задачи дисциплины:

Ознакомить обучающихся с основными принципами и методами обеспечения качества программного обеспечения, дать систематические знания в области ручного и автоматизированного тестирования, сформировать практические навыки разработки тест-кейсов, написания автотестов и интеграции тестирования в процессы разработки.

2. Место дисциплины в структуре ООП

Дисциплина входит в раздел «Элективные дисциплины» части, формируемой участниками образовательных отношений, блока 1.

Предварительные знания и навыки. Студент должен знать принципы объектно-ориентированного программирования, уметь программировать на языке Java или Python, знать основы реляционных баз данных и SQL, иметь представление о сетевых протоколах и HTTP.

Дальнейшее использование. Полученные знания применяются при прохождении производственной практики, при выполнении выпускной квалификационной работы, а также в дальнейшей профессиональной деятельности в области разработки и обеспечения качества программного обеспечения.

3. Объем дисциплины: 3 зачетных единицы, 108 академических часа, в том числе:

контактная аудиторная работа: лекции 30 часов, практические занятия 30 часов;

контактная внеаудиторная работа: контроль самостоятельной работы 0 часов, в том числе курсовая работа 0 часов;

самостоятельная работа: 48 часов, в том числе контроль 0 часов.

4. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Планируемые результаты освоения образовательной программы (формируемые компетенции)	Планируемые результаты обучения по дисциплине
ПК-3, Способен осуществлять концептуальное моделирование проблемной области и проводить формализацию представления знаний в	ПК-3.1, Разрабатывает концептуальную модель проблемной области системы искусственного интеллекта

системах искусственного интеллекта	
ПК-5, Способен использовать инструментальные средства для решения задач машинного обучения	ПК-5.1, Осуществляет оценку и выбор инструментальных средств для решения поставленной задачи
ПК-7 Способен осуществлять сбор и подготовку данных для систем искусственного интеллекта	ПК-7.1 Осуществляет поиск данных в открытых источниках, специализированных библиотеках и репозиториях

5. Форма промежуточной аттестации и семестр прохождения: зачёт в 7 семестре.

6. Язык преподавания: русский.

II. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Учебная программа – наименование разделов и тем	Всего (час.)	Контактная работа (час.)					Самостоя- тельная ра- бота, в том числе Кон- троль (час.)
		Лекции		Практиче- ские заня- тия			
		Всего	в т.ч. прак- тическая подготовка	Всего	в т.ч. прак- тическая	Контроль сам. раб., в т.ч. курсовая ра- бота	
1	2	3	4	5	6	7	8
Часть 1. Ручное те- стирование							
Введение в тестирова- ние ПО	7	2	0	2/0	0	0	3
Виды тестирования	7	2	0	2/0	0	0	3
Жизненный цикл раз- работки и тестирова- ния ПО	7	2	0	2/0	0	0	3
Тест-дизайн	7	2	0	2/0	0	0	3

Инструменты и технологии тестировщика	7	2	0	2/0	0	0	3
Тестирование в разных сферах и доменах	7	2	0	2/0	0	0	3
Работа в команде и soft skills	7	2	0	2/0	0	0	3
Часть 2. Автоматизированное тестирование							
Введение в автотестирование	7	2	0	2/0	0	0	3
Теория автотестирования	7	2	0	2/0	0	0	3
Создание проекта автотестов (Gradle, Git, JUnit 5)	7	2	0	2/0	0	0	3
Assertions, логирование, Allure, конфигурация	5	0	0	2/0	0	0	3
Паттерны проектирования в автотестах	7	2	0	2/0	0	0	3
Web-автотесты (Selenium / Selenide)	7	2	0	2/0	0	0	3
CI/CD и Jenkins	7	2	0	2/0	0	0	3
API-автотесты (REST Assured)	7	2	0	2/0	0	0	3
Тестирование производительности (JMeter / Gatling)	5	2	0	0/0	0	0	3
Итого	108	30	0	30/0	0	0	48

Учебная программа дисциплины

Часть 1. Ручное тестирование

1. Введение в тестирование программного обеспечения
 - Понятие, цели и задачи тестирования ПО
 - Семь принципов тестирования (ISTQB)
 - Ключевые термины: ошибка, дефект, баг, отказ
 - QA, QC и тестирование: различия
 - Верификация и валидация
 - Серьёзность (severity) и приоритет (priority)

2. Виды тестирования
 - Функциональное и нефункциональное тестирование
 - Уровни тестирования: модульное, интеграционное, системное, приёмочное
 - Тестирование «белого», «чёрного» и «серого» ящика
 - Регрессионное, дымовое и санитарное тестирование
 - Нагрузочное, стрессовое, тестирование безопасности

3. Жизненный цикл разработки и тестирования ПО
 - Модели SDLC: Водопадная, V-модель, Agile, Scrum, Kanban
 - Жизненный цикл дефекта (Bug Life Cycle)
 - Тест-план: структура и содержание
 - Метрики тестирования
 - Тест-стратегия

4. Тест-дизайн
 - Техники тест-дизайна: классы эквивалентности, граничные значения
 - Таблицы решений, диаграммы перехода состояний
 - Попарное тестирование (Pairwise / PICT)
 - Тест-кейсы и чек-листы
 - Risk-based testing

5. Инструменты и технологии тестировщика
 - Системы баг-трекинга: Jira
 - Системы управления тест-кейсами: TestRail, Qase
 - Тестирование REST API: Postman
 - Работа с базами данных: SQL для QA
 - DevTools браузера

6. Тестирование в разных сферах и доменах
 - Тестирование веб-, мобильных и настольных приложений
 - Специфика финтех-, e-commerce- и медицинских систем
 - Тестирование API

7. Работа в команде и soft skills
 - Роль QA-инженера в Scrum-команде

- Правила написания баг-репорта
- Профессиональный рост: ISTQB, специализации

Часть 2. Автоматизированное тестирование

8. Введение в автотестирование

- Что такое автоматизация тестирования (ISTQB)
- История развития автоматизации: от 1980-х до AI-assisted testing
- Преимущества и недостатки автоматизации
- Пирамида тестирования
- Технологический стек курса

9. Теория автотестирования

- Признаки хорошего автотеста
- Нестабильные тесты (flaky tests): причины и методы борьбы
- Метрики качества автотестов
- Тест-фреймворки: JUnit 5, TestNG, pytest

10. Создание проекта автотестов (Gradle, Git, JUnit 5)

- Структура проекта автотестов
- Gradle: build.gradle, управление зависимостями
- Git: рабочие процессы в контексте QA
- JUnit 5: аннотации жизненного цикла, параметризованные тесты

11. Assertions, логирование, Allure, конфигурация

- AssertJ: цепочки утверждений, soft assertions
- Логирование: SLF4J + Log4j 2
- Allure Report: аннотации, генерация HTML-отчёта
- Конфигурация через Owner (cfg4j)

12. Паттерны проектирования в автотестах

- Page Object Model (POM)
- Паттерн Builder для тестовых данных
- Принципы SOLID применительно к тестовому коду
- Рефакторинг автотестов

13. Web-автотесты (Selenium / Selenide)

- Selenium WebDriver: архитектура
- Selenide: преимущества, основные методы
- Локаторы: CSS-селекторы, XPath
- Явные ожидания, работа с фреймами и модальными окнами

14. CI/CD и Jenkins

- Что такое CI/CD

- Jenkins: Freestyle и Pipeline jobs, Jenkinsfile
- GitHub Actions: workflow YAML
- Allure-отчёты в CI

15. API-автотесты (REST Assured)

- REST-архитектура: ресурсы, методы, статус-коды
- REST Assured: структура Given-When-Then
- JSONPath, JSON Schema Validation
- Авторизация: Bearer Token, OAuth2

16. Тестирование производительности (JMeter / Gatling)

- Виды нагрузочного тестирования: load, stress, soak, spike
- Apache JMeter: тест-план, Thread Group, Listeners
- Gatling: DSL на Java, сценарии, отчёты

III. Образовательные технологии

Учебная программа – наименование разделов и тем	Вид занятия	Образовательные технологии
Введение в тестирование ПО	лекции, практические занятия	изложение теоретического материала, разбор реальных кейсов
Виды тестирования	лекции, практические занятия	изложение теоретического материала, решение задач
Жизненный цикл разработки и тестирования ПО	лекции, практические занятия	изложение теоретического материала, анализ жизненного цикла дефекта
Тест-дизайн	лекции, практические занятия	изложение теоретического материала, практическое проектирование тест-кейсов
Инструменты и технологии тестировщика	лекции, практические занятия	демонстрация инструментов, практическая работа с Jira и Postman
Тестирование в разных сферах и доменах	лекции, практические занятия	разбор кейсов из разных доменов

Работа в команде и soft skills	лекции, практические занятия	групповые упражнения, деловая игра
Введение в автотестирование	лекции, практические занятия	изложение теоретического материала, обзор стека технологий
Теория автотестирования	лекции, практические занятия	изложение теоретического материала, анализ примеров
Создание проекта автотестов	лекции, практические занятия	практическая работа в IDE, настройка проекта
Assertions, логирование, Allure, конфигурация	лекции, практические занятия	практическая работа, разбор примеров
Паттерны проектирования в автотестах	лекции, практические занятия	изложение теоретического материала, рефакторинг кода
Web-автотесты (Selenium / Selenide)	лекции, практические занятия	демонстрация, практическая работа с браузером
CI/CD и Jenkins	лекции, практические занятия	практическая настройка пайплайна
API-автотесты (REST Assured)	лекции, практические занятия	практическая работа с REST API
Тестирование производительности	лекции, практические занятия	Практическая работа в JMeter

IV. Оценочные материалы для проведения текущей и промежуточной аттестации

Типовые контрольные задания и/или критерии для проверки индикатора ПК-3.1

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>Уметь разрабатывать концептуальные модели предметной области для систем с элементами ИИ/ML: выделять сущности, атрибуты, связи, определять точки интеграции тестовых сценариев с моделью данных и поведения системы.</p>	<p>Примеры задач для контрольных работ и самостоятельных работ:</p> <ul style="list-style-type: none"> • Разработайте концептуальную модель тестового окружения для системы рекомендаций: выделите сущности (пользователь, товар, история просмотров, модель ML), опишите их атрибуты и взаимосвязи. Определите, какие данные необходимы для тестирования качества рекомендаций. • Постройте модель данных для тестирования чат-бота с NLP-компонентом: укажите источники входных данных (интенты, сущности, контекст диалога), ожидаемые выходы и метрики качества (accuracy, F1-score). • Спроектируйте концептуальную схему тест-дата-сета для проверки модели классификации: определите классы, балансировку данных, признаки для валидации. 	<p>оценка 3 — знает основные элементы концептуального моделирования (сущности, атрибуты, связи) и понимает специфику тестирования систем с ИИ; оценка 4 — кроме того умеет разрабатывать концептуальную модель предметной области, выделяя ключевые сущности и их взаимосвязи для задач тестирования; оценка 5 — кроме того умеет интегрировать в модель требования к тестовым данным, метрики качества и сценарии валидации, обосновывая выбор элементов модели с учётом особенностей ИИ-систем (недетерминированность, дрейф данных, этические аспекты).</p>

Типовые контрольные задания и/или критерии для проверки индикатора ПК-5.1

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>Уметь проводить сравнительный анализ инструментальных средств тестирования (фреймворки, библиотеки, CI/CD-инструменты) и обоснованно выбирать оптимальный стек под задачу проекта.</p>	<p>Примеры задач для контрольных работ и лабораторных работ:</p> <ul style="list-style-type: none"> • Проведите сравнительный анализ Selenium WebDriver, Selenide и Playwright для автоматизации тестирования SPA-приложения. Составьте таблицу критериев (поддержка языков, скорость, стабильность, документация, сообщество) и обоснуйте выбор инструмента. • Оцените применимость JUnit 5 vs TestNG vs pytest для проекта с параметризованными тестами и параллельным запуском. Предложите конфигурацию и аргументируйте решение. • Выберите инструмент нагрузочного тестирования (JMeter / Gatling / k6) для API микросервиса с требованиями к RPS и отчётности. Обоснуйте выбор с учётом критериев: язык сценариев, интеграция с CI, визуализация результатов. 	<p>оценка 3 — знает основные инструментальные средства тестирования и их назначение;</p> <p>оценка 4 — кроме того умеет проводить сравнительный анализ инструментов по заданным критериям и выбирать подходящий под типовую задачу;</p> <p>оценка 5 — кроме того умеет проводить многокритериальную оценку, учитывать контекст проекта (стек технологий, команда, бюджет, сроки), прогнозировать риски внедрения и формировать обоснованное технико-экономическое обоснование выбора.</p>

Типовые контрольные задания и/или критерии для проверки индикатора ПК-7.1

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>Уметь осуществлять целенаправленный поиск тестовых данных, библиотек, утилит и документации в открытых источниках: GitHub, Maven Central, PyPI, официальных репозиториях, специализированных ресурсах (OWASP, ISTQB, Allure).</p>	<p>Примеры задач для самостоятельных и лабораторных работ:</p> <ul style="list-style-type: none"> • Найдите в открытом доступе набор тестовых данных (JSON/CSV) для тестирования API интернет-магазина. Оцените качество данных (полнота, реалистичность, анонимизация) и предложите способ их интеграции в автотесты. • Найдите и сравните три open-source библиотеки для генерации фейковых данных (Faker, DataFactory, Mimesis). Подготовьте краткий обзор: лицензия, поддержка языков, примеры использования, рекомендации по выбору. • Используя GitHub Search и Maven Repository, найдите актуальную версию REST Assured с поддержкой JSON Schema Validation. Проверьте наличие документации, примеров и активных issue. Сформулируйте выводы о пригодности для учебного проекта. 	<p>оценка 3 — умеет выполнять базовый поиск информации в открытых источниках по заданным ключевым словам; оценка 4 — кроме того умеет фильтровать результаты поиска по релевантности, актуальности и лицензии, отбирать подходящие ресурсы для решения задачи тестирования; оценка 5 — кроме того умеет проводить критическую оценку найденных ресурсов (качество кода, активность поддержки, совместимость), документировать процесс поиска и формировать структурированную подборку инструментов/данных с обоснованием выбора.</p>

V. Учебно-методическое и информационное обеспечение дисциплины

1. Рекомендованная литература

а) Основная литература

- [1] Ананьева, Т. Н. Стандартизация, сертификация и управление качеством программного обеспечения: учебное пособие / Т.Н. Ананьева, Н.Г. Новикова, Г.Н. Исаев. Москва: ИНФРА-М, 2026. — 232 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014887-8. - Текст: электронный. - URL: <https://znanium.ru/catalog/product/2216049> (дата обращения: 13.04.2026). – Режим доступа: по подписке.
- [2] Морозова, Ю. В. Тестирование программного обеспечения: учебное пособие / Ю. В. Морозова. - Томск: Эль-Контент, 2019. - 120 с. - ISBN 978-5-4332-0279-5. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1845910> (дата обращения: 13.04.2026). – Режим доступа: по подписке.
- [3] Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения: учебное пособие / А. В. Проскуряков; Южный федеральный университет. - Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2022. - 197 с. - ISBN 978-5-9275-4044-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/2057599> (дата обращения: 3.04.2026). – Режим доступа: по подписке.

б) Дополнительная литература

- [1] Искусственный интеллект: модели, сети и методы машинного обучения : учебное пособие / В. В. Антонов, А. С. Дьячков, Л. А. Кромина [и др.]. — Москва : Русайнс, 2026. — 165 с. — ISBN 978-5-466-11506-2. — URL: <https://book.ru/book/962906> (дата обращения: 06.05.2026). — Текст : электронный.
- [2] Сидоркина, И. Г. Системы искусственного интеллекта : учебное пособие / И. Г. Сидоркина. — Москва : КноРус, 2026. — 245 с. — ISBN 978-5-406-14910-2. — URL: <https://book.ru/book/959436> (дата обращения: 06.05.2026). — Текст : электронный.

2. Программное обеспечение

Наименование помещений	Программное обеспечение
Ауд. 201а (компьютерная лаборатория ПМиК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Перечень программного обеспечения (со свободными лицензиями): Linux Ubuntu, KDE, IntelliJ IDEA Community Edition, Java Development Kit 17 Eclipse Temurin, Git, Apache JMeter, Google Chrome, Postman, Docker, Python, Vim.

3. Современные профессиональные базы данных и информационные справочные системы

- [1] ЭБС «ZNANIUM.COM» <http://www.znanium.com>
- [2] ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru>
- [3] ЭБС IPRbooks <http://www.iprbookshop.ru>
- [4] ЭБС «Лань» <http://e.lanbook.com>
- [5] ЭБС «Юрайт» <https://urait.ru>
- [6] ЭБС ТвГУ <http://megapro.tversu.ru/megapro/Web>
- [7] Научная электронная библиотека eLIBRARY.RU (подписка на журналы) https://elibrary.ru/projects/subscription/rus_titles_open.asp
- [8] Репозиторий ТвГУ <http://eprints.tversu.ru>

4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- [1] ОЦЕНКА КАЧЕСТВА СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА, <https://files.stroyinf.ru/Data/769/76946.pdf>
- [2] Метрики ИИ: как измерить качество и эффективность искусственного интеллекта, <https://neuro-core.ru/blogs/machine-learning-metrics>
- [3] Оценка результатов машинного обучения, <https://4brain.ru/aibasics/results.php>
- [4] Метрики качества работы моделей машинного обучения: как их понимать и использовать, <https://webiomed.ru/blog/metriki-kachestva-raboty-modelei>

VI. Методические материалы для обучающихся по освоению дисциплины

Важной составляющей данного раздела РПД являются требования к рейтинг-контролю с указанием баллов, распределенных между модулями и видами работы обучающихся.

Максимальная сумма баллов по учебной дисциплине, заканчивающейся зачетом, по итогам семестра составляет 100 баллов (50 баллов - 1-й модуль и 50 баллов - 2-й модуль).

Студенту, набравшему 40 баллов и выше по итогам работы в семестре, в экзаменационной ведомости и зачетной книжке выставляется оценка «зачтено». Студент, набравший до 39 баллов включительно, сдает зачет.

Распределение баллов по модулям устанавливается преподавателем и может корректироваться.

Требования к рейтинг контролю (7 семестр)

Контрольная работа №1

Темы: Ручное тестирование, тест-дизайн, жизненный цикл дефекта, инструменты QA, концептуальное моделирование тестовых сред для ИИ/ML.
Формат: Письменно-практическая работа, 90 минут.

Примеры заданий:

Для формы регистрации с полями Email, Пароль (мин. 8 символов, 1 цифра, 1 спецсимвол), Дата рождения примените техники классов эквивалентности и граничных значений. Составьте 8 тест-кейсов. Найдите в открытых источниках актуальный гайд по тест-дизайну для веб-форм, оцените его применимость и укажите, какие ограничения вы учли при проектировании.

Дана ситуация: «В рекомендательной системе при повторном запросе одного и того же пользователя выводятся товары из разных категорий без явного изменения профиля». Оформите баг-репорт в формате Jira (Summary, Steps, Actual/Expected, Severity, Priority, Environment, Logs). Обоснуйте выбор Severity и Priority с учётом бизнес-рисков и недетерминированности ML-выдачи.

Разработайте концептуальную модель тестового окружения для чат-бота с NLP-компонентом. Выделите сущности (интененты, сущности, контекст диалога, модель, источники данных), опишите их связи. Определите, какие метрики качества (accuracy, F1-score, latency) необходимо валидировать и как организовать тестовые данные с учётом баланса классов.

Используя GitHub и Maven Central, найдите три open-source библиотеки для генерации тестовых данных. Подготовьте сравнительную таблицу (лицензия, поддержка Java/Python, примеры использования, активность поддержки). Обоснуйте выбор одной библиотеки для интеграции в автоматизированный фреймворк.

Контрольная работа №2

Темы: Автоматизация, JUnit 5, Selenide, REST Assured, Page Object, CI/CD, анализ отчётов, выбор инструментов.

Формат: Практическая работа в IDE, 120 минут.

Примеры заданий:

1. Проведите сравнительный анализ Selenium WebDriver, Selenide и Playwright для тестирования SPA-приложения. Составьте таблицу критериев (стабильность локаторов, скорость, встроенные ожидания, интеграция с CI, документация). Обоснуйте выбор инструмента под контекст учебного проекта с учётом требований к поддержке и обучаемости команды.

2. Реализуйте класс LoginPage по паттерну **Page Object** с использованием **Selenide**. Напишите 2 параметризованных теста (@ParameterizedTest) с применением SoftAssertions. Добавьте явные ожидания и обработку модальных окон. Прокомментируйте, как ваш код следует принципам SOLID и FIRST.

3. Напишите 3 API-теста на **REST Assured**: GET с валидацией JSON Schema, POST с авторизацией через Bearer Token, проверка 400 Bad Request. Настройте логирование через SLF4J. Укажите, как вы будете изолировать тестовые данные между запусками.

4. Создайте Jenkinsfile или .github/workflows/test.yml с шагами: Checkout → Build → Test → Allure Report. Опишите стратегию работы с нестабильными (flaky) тестами в пайплайне (retry-механизмы, тегирование, изоляция окружения). Проанализируйте типичный Allure-отчёт: как по нему выявить корневую причину падения и предложить оптимизацию процесса?

Вопросы к зачёту

Блок 1. Фундаментальные основы и ручное тестирование

1. Семь принципов тестирования (ISTQB). Различия между ошибкой, дефектом и отказом. Верификация vs Валидация.

2. Уровни тестирования: модульное, интеграционное, системное, приёмочное. Пирамида тестирования и её экономическое обоснование.

3. Функциональное и нефункциональное тестирование. Классификация: регрессионное, дымовое, санитарное, безопасность.

4. Жизненный цикл дефекта (Bug Life Cycle). Severity vs Priority: правила выставления, типичные конфликты, приоритизация в Agile.

5. Тест-план и тест-стратегия: структура, ключевые метрики качества тестирования, роль QA в определении Definition of Done.

Блок 2. Тест-дизайн и работа с данными

6. Техники тест-дизайна чёрного ящика: классы эквивалентности, граничные значения, таблицы решений, диаграммы состояний.

7. Попарное тестирование (Pairwise / PICT): область применения, преимущества перед полным перебором, ограничения.

8. Risk-based testing: как оценивать риски, приоритизировать тестовое

покрытие и распределять усилия в условиях сжатых сроков.

9. Управление тестовыми данными: генерация (Faker, DataFactory), мок-серверы, изоляция данных, работа с Docker в QA-процессах.

10. Критерии отбора и оценки открытых источников, библиотек и документации для QA. Как проверять достоверность, актуальность и лицензионную чистоту инструментов?

Блок 3. Автоматизация и архитектура фреймворков

11. Критерии целесообразности автоматизации. Признаки хорошего автотеста (FIRST, AIR). Признаки плохого кода в тестах.

12. JUnit 5: аннотации жизненного цикла, параметризованные тесты (@ValueSource, @CsvSource), динамические тесты.

13. fluent-интерфейс, цепочки утверждений, SoftAssertions. Логирование в тестах (SLF4J + Log4j2).

14. Архитектура Selenium WebDriver. Преимущества Selenide. Локаторы (CSS, XPath), явные и неявные ожидания, обработка фреймов.

15. Page Object Model (POM): принципы реализации, применение SOLID, рефакторинг автотестов, борьба с дублированием кода.

Блок 4. API, CI/CD и нагрузочное тестирование

16. REST API: ресурсы, HTTP-методы, статус-коды. REST Assured (Given-When-Then), JSONPath, JSON Schema Validation.

17. Авторизация в API-тестах: Bearer Token, OAuth2, refresh-токены, тестирование защищённых эндпоинтов.

18. CI/CD для QA: Jenkins Pipeline vs GitHub Actions. Параллельный запуск, генерация Allure-отчётов, интеграция quality gates.

19. Нестабильные тесты (flaky): причины возникновения, стратегии обнаружения, методы стабилизации (ожидания, изоляция, retry, анализ логов CI).

20. Нагрузочное тестирование: виды (load, stress, soak, spike). JMeter vs Gatling. Интерпретация метрик (RPS, latency, error rate, throughput), поиск bottleneck.

Блок 5. Специфика тестирования систем с ИИ/ML и анализом данных

21. Концептуальное моделирование тестовых окружений для ИИ-систем: сущности, потоки данных, точки интеграции, валидация поведения модели.

22. Недетерминированность ML-выводов: как тестировать рекомендательные системы, чат-боты, компьютерное зрение? Допустимые погрешности, статистическая валидация, A/B-тестирование.

23. Мониторинг качества моделей в продакшене: data drift, concept drift, интеграция метрик (accuracy, precision, recall, F1) в QA-процесс.

24. Многокритериальный выбор стека автоматизации под проект: как учитывать команду, бюджет, сроки, поддерживаемость и риски внедрения? Формирование технико-экономического обоснования.

25. Поиск и оценка тестовых датасетов, open-source библиотек и руководств в репозиториях. Критический анализ активности поддержки, совместимости и качества кода. Документирование процесса отбора.

VII. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для аудиторной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 308 (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, экран проектор.

Для самостоятельной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 201а (компьютерная лаборатория ПМиК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, доска маркерная, компьютер, сервер (системный блок), концентратор сетевой.

VIII. Сведения об обновлении рабочей программы дисциплины

№ п/п	Обновленный раздел рабочей программы дисциплины	Описание внесённых изменений	Дата и протокол заседания кафедры, утвердившего изменения