МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО Университет биотехнологий
Институт цифровых технологий
Кафедра информационных технологий и моделирования

ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ

Методические указания для проведения лабораторно-практических занятий и самостоятельной работы студентов

Новосибирск 2026

УДК 004.415.53(075.32)
ББК 32.973.26-018.2я723
Ч 65

Кафедра информационных технологий и моделирования

Составители: Чирков С.В., Власенко А.А.

Рецензент: Краснов С.Д., преподаватель кафедры информационных технологий и моделирования ФГБОУ ВО Университет биотехнологий

Тестирование информационных ресурсов: методические указания для проведения лабораторно-практических занятий и самостоятельной работы студентов / сост. С. В. Чирков, А. А. Власенко. – Новосибирск: Университет биотехнологий, 2026.

Методические указания предназначены для студентов, обучающихся по специальности 09.02.09 Веб-разработка (дисциплина МДК.01.03 «Тестирование информационных ресурсов»). Содержат описание практических занятий, задания для самостоятельной работы, требования к оформлению отчётов и критерии оценки.

Методические указания разработаны в соответствии с рабочей программой дисциплины и фондом оценочных средств.

Содержание

Введение 4
1. Общие методические указания 6
1.1. Организация практических занятий 6
1.2. Правила работы на занятиях 6
1.3. Требования к отчёту 6
1.4. Организация самостоятельной работы 7
1.5. Критерии оценки 7
2. Методические указания к практическим занятиям 8
Тема 1.1. Качество программного обеспечения. Виды и уровни тестирования 8
Тема 1.2. Техники тестирования информационных ресурсов 9
Тема 1.3. Тест-планы и тестовая документация 11
Тема 2.1. Разработка тестовых сценариев 13
Тема 2.2. Инструменты подготовки тестовых данных 15
Тема 2.3. Функциональное тестирование веб-ресурса 17
Тема 3.1. Регистрация дефектов. Отчёт по результатам тестирования 19
Тема 3.2. Автоматизация тестирования: обзор подходов 21
3. Методические указания к самостоятельной работе 24
4. Рекомендуемая литература и информационные ресурсы 25
Приложение А. Образец титульного листа отчёта 26

Введение

На изучение дисциплины МДК.01.03 «Тестирование информационных ресурсов» запланировано 98 ч, в том числе 88 ч аудиторных (32 ч – занятия лекционного типа, 50 ч – занятия семинарского типа (практические занятия), 6 ч – консультации) и 10 ч (10,2 %) отведено для самостоятельной работы. Форма промежуточной аттестации – итоговая оценка по результатам текущего контроля (2 семестр). Практические занятия реализуются в форме практической подготовки в объёме 48 ч.

Основная цель междисциплинарного курса – формирование умений планировать и проводить тестирование информационных ресурсов, разрабатывать тестовую документацию, регистрировать дефекты и составлять отчёты по результатам тестирования.

В процессе изучения дисциплины решаются следующие задачи:

- изучение понятий качества программного обеспечения, видов и уровней тестирования;
- освоение техник тест-дизайна: эквивалентного разбиения, граничных значений, таблиц решений, попарного тестирования;
- освоение разработки тест-планов, тест-кейсов, чек-листов и матрицы трассировки требований;
- освоение подготовки тестовых данных и функционального тестирования веб-ресурса;
- освоение регистрации дефектов и подготовки отчёта по результатам тестирования;
- формирование представлений о подходах к автоматизации тестирования.

В соответствии с требованиями ФГОС СПО дисциплина направлена на формирование следующих компетенций по специальности 09.02.09 Веб-разработка:

- выбирать способы решения задач профессиональной деятельности применительно к различным контекстам (ОК 01);

– использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности (ОК 02);
– осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста (ОК 05);
– пользоваться профессиональной документацией на государственном и иностранном языках (ОК 09);
– проектировать информационные ресурсы (ПК 1.1);
– разрабатывать интерфейсы пользователя (ПК 1.2);
– интегрировать программный код в соответствующую инфраструктуру (ПК 1.3);
– использовать систему контроля версий в процессе коллективной (параллельной) разработки (ПК 1.4);
– выполнять процедуры тестирования программного кода (ПК 1.5).

Методические указания составлены в соответствии с рабочей программой дисциплины и фондом оценочных средств. Нумерация и содержание практических занятий соответствуют темам рабочей программы; контрольные вопросы и практические задания согласованы с фондом оценочных средств.

Распределение практических занятий и самостоятельной работы по темам дисциплины приведено в таблице 1.

Таблица 1 – Распределение часов по темам

Тема	**Практические занятия**	**ПЗ, ч**	**СР, ч**
1.1. Качество программного обеспечения. Виды и уровни тестирования	№ 1–2	4	1
1.2. Техники тестирования информационных ресурсов	№ 3–5	7	1
1.3. Тест-планы и тестовая документация	№ 6–8	6	1
2.1. Разработка тестовых сценариев	№ 9–12	9	2

Тема	Практические занятия	ПЗ, ч	СР, ч
2.2. Инструменты подготовки тестовых данных	№ 13–15	6	1
2.3. Функциональное тестирование веб-ресурса	№ 16–18	6	1
3.1. Регистрация дефектов. Отчёт по результатам тестирования	№ 19–21	6	2
3.2. Автоматизация тестирования: обзор подходов	№ 22–24	6	1
Итого	24 занятий	50	10

1. Общие методические указания

1.1. Организация практических занятий

Практические занятия проводятся в компьютерном классе. Используются операционная система ALT Linux, браузер Mozilla Firefox с инструментами разработчика, Selenium IDE, система учёта дефектов Redmine, среда Node.js, редактор кода VSCodium, LibreOffice (Writer, Calc), система контроля версий Git. Объектом тестирования служит учебный веб-ресурс (разработанный в МДК.01.01–01.02 или предоставленный преподавателем).

Перед занятием необходимо изучить краткие теоретические сведения и повторить лекционный материал. На занятии студент разрабатывает тестовую документацию, выполняет проверки, регистрирует дефекты, оформляет отчёт и отвечает на контрольные вопросы.

1.2. Правила работы на занятиях

- соблюдать правила электробезопасности и санитарные нормы работы за персональным компьютером, делать перерывы для гимнастики для глаз;
- не устанавливать и не удалять программное обеспечение без разрешения преподавателя;
- сохранять результаты работы в личном каталоге и делать резервные копии;
- тестирование выполнять только на учебных стендах и на ресурсах, указанных преподавателем; проверки безопасности чужих систем запрещены;
- для тестирования использовать только вымышленные или обезличенные данные; реальные персональные данные в тестах не применять;
- каждый найденный дефект регистрировать в системе учёта с шагами воспроизведения; результаты тестов фиксировать сразу, а не по памяти;

– самостоятельно выполнять задания; использование чужих решений и готовых ответов без указания источника не допускается.

1.3. Требования к отчёту

Отчёт по каждому занятию оформляется в электронном виде (LibreOffice Writer, Calc, PDF) и содержит: тему и цель, описание объекта тестирования, тестовую документацию (план, чек-листы, тест-кейсы), результаты выполнения (таблицы, снимки экрана), перечень дефектов, выводы. Титульный лист – по образцу приложения А.

1.4. Организация самостоятельной работы

Самостоятельная работа (10 ч) включает изучение теоретического материала и выполнение заданий по темам 1.1, 1.2, 1.3, 2.1, 2.2, 2.3, 3.1, 3.2 (раздел 3 настоящих указаний). Результаты проверяются преподавателем на очередном практическом занятии в форме собеседования.

1.5. Критерии оценки

Результаты выполнения практических занятий и самостоятельной работы учитываются при выставлении оценки по результатам текущего контроля (2 семестр). Для оценки используется традиционная четырёхбалльная система:

Оценка	**Критерии**
Отлично	Все задания выполнены полностью и верно, отчёт оформлен по требованиям. Ответ полный, логичный, обоснованный; студент свободно отвечает на дополнительные вопросы, приводит примеры.
Хорошо	Задания выполнены, есть незначительные неточности в решении или оформлении. Ответ по существу верный; допускаются несущественные ошибки при ответе на дополнительные вопросы.
Удовлетворительно	Выполнена основная часть заданий. Ответ неполный: знание основных положений, затруднения с пояснениями и примерами.

Оценка	Критерии
Неудовлетворительно	Задания не выполнены или выполнены неверно. Основной материал не усвоен, на дополнительные вопросы студент ответить не может.

2. Методические указания к практическим занятиям

Ниже приведены практические занятия по темам рабочей программы. Для каждого занятия указаны цель, краткие теоретические сведения, задания и контрольные вопросы.

Раздел 1. Основы тестирования

Тема 1.1. Качество программного обеспечения. Виды и уровни тестирования

Характеристики качества ПО, понятие дефекта, уровни тестирования (модульное, интеграционное, системное, приёмочное), виды: функциональное, регрессионное, нагрузочное, безопасности и удобства. Объём практических занятий – 4 ч.

Практическое занятие № 1. Качество ПО и дефекты (2 ч)

Цель работы: изучить характеристики качества программного обеспечения и понятие дефекта.

Краткие теоретические сведения.

Качество ПО – соответствие требованиям и ожиданиям пользователей. Модель качества ISO/IEC 25010 включает функциональную пригодность, производительность, совместимость, удобство использования, надёжность, безопасность, сопровождаемость, переносимость. Дефект – несоответствие фактического результата ожидаемому; причина дефекта – ошибка человека (error), проявление – отказ (failure).

Задания.

1. Для учебного веб-ресурса сопоставьте характеристики качества ISO 25010 с конкретными проверками (таблица).
2. Определите объект тестирования, опишите его назначение и состав.
3. Приведите по примеру дефектов для пяти характеристик качества.

Контрольные вопросы.

1. Что такое качество программного обеспечения и по каким характеристикам его оценивают?
2. Что такое дефект и каков его жизненный цикл?
3. Чем ошибка отличается от отказа?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 2. Уровни и виды тестирования (2 ч)

Цель работы: изучить уровни и виды тестирования и их применение.

Краткие теоретические сведения.

Уровни: модульное (отдельные функции), интеграционное (взаимодействие компонентов), системное (всё приложение), приёмочное (заказчик). Виды: функциональное, регрессионное, нагрузочное, тестирование безопасности и удобства использования, совместимости. Пирамида тестирования рекомендует больше быстрых модульных тестов и меньше медленных сквозных.

Задания.

1. Составьте таблицу «уровень – объект – кто выполняет – инструменты».
2. Для функций учебного ресурса (регистрация, поиск, оформление заказа) определите уровни и виды проверок.
3. Нарисуйте пирамиду тестирования для проекта и обоснуйте пропорции.

Контрольные вопросы.

1. Перечислите виды тестирования по уровням: модульное, интеграционное, системное, приёмочное.
2. Что такое регрессионное тестирование?
3. Чем нагрузочное тестирование отличается от функционального?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Тема 1.2. Техники тестирования информационных ресурсов

Подходы «чёрного», «белого» и «серого ящика», эквивалентное разбиение, анализ граничных значений, таблицы решений, попарное тестирование, выбор техники. Объём практических занятий – 7 ч.

Практическое занятие № 3. Подходы «чёрного», «белого» и «серого ящика» (2 ч)

Цель работы: освоить подходы к тестированию и выбор подхода.

Краткие теоретические сведения.

«Чёрный ящик» – проверка по входам и выходам без знания кода; «белый ящик» – проверка с учётом внутренней структуры (покрытие операторов, ветвей); «серый ящик» – частичное знание внутреннего устройства (например, структуры БД и API).

Задания.

1. Для функции расчёта скидки опишите проверки «чёрным» и «белым» ящиком.
2. Составьте таблицу сравнения трёх подходов (знания, виды ошибок, кто применяет).
3. Определите подход для каждой из пяти задач тестирования учебного ресурса.

Контрольные вопросы.

1. В чём разница между тестированием «чёрным» и «белым» ящиком?
2. Что такое «серый ящик»?
3. Что такое покрытие кода?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 4. Эквивалентное разбиение и анализ граничных значений (2 ч)

Цель работы: освоить построение классов эквивалентности и граничных значений.

Краткие теоретические сведения.

Эквивалентное разбиение делит входные данные на классы, в пределах которых система ведёт себя одинаково; из каждого класса берут один представитель. Анализ граничных значений проверяет значения на границах классов (минимум, максимум и соседние значения), где ошибки встречаются чаще.

Задания.

1. Для поля «возраст» (допустимый диапазон 18–65) постройте классы эквивалентности и граничные значения, составьте набор тестов.
2. То же для поля «количество товаров» (от 1 до 100).
3. Проверьте тесты на реальной форме учебного ресурса и зафиксируйте результаты.

Контрольные вопросы.

1. Что такое эквивалентное разбиение и как оно применяется?
2. Что такое анализ граничных значений и почему ошибки чаще возникают на границах?
3. Сколько тестов нужно для диапазона 1–100 при двух границах?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 5. Таблицы решений и попарное тестирование (3 ч)

Цель работы: освоить таблицы решений и попарное тестирование.

Краткие теоретические сведения.

Таблица решений описывает комбинации условий и соответствующие действия; позволяет полностью проверить бизнес-правила. Попарное

тестирование (pairwise) сокращает число комбинаций параметров, гарантируя проверку каждой пары значений.

Задания.

1. Составьте таблицу решений для правила скидки (клиент постоянный, сумма заказа, промокод) и тесты по ней.
2. Сократите набор комбинаций параметров (браузер × ОС × язык) попарным методом и сравните число тестов.
3. Выберите технику тест-дизайна для пяти разных функций ресурса и обоснуйте выбор.

Контрольные вопросы.

1. Что такое таблица решений?
2. В чём идея попарного тестирования?
3. Как выбрать технику в зависимости от характера проверяемой функции?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Тема 1.3. Тест-планы и тестовая документация

Структура тест-плана, цели, объём и ресурсы, тест-кейс, чек-лист, матрица трассировки, критерии начала и завершения тестирования, оценка трудоёмкости. Объём практических занятий – 6 ч.

Практическое занятие № 6. Тест-план (2 ч)

Цель работы: научиться составлять тест-план.

Краткие теоретические сведения.

Тест-план определяет объект, цели, объём (что тестируется и что нет), подходы, ресурсы и график, критерии входа и выхода, риски и ответственных. Критерии входа: готовность сборки, окружения и данных; критерии выхода: выполнение плана, отсутствие критических дефектов.

Задания.

1. Составьте тест-план для формы регистрации учебного веб-ресурса: объект, цели, уровни, критерии входа и выхода.
2. Определите риски тестирования и меры снижения.
3. Оцените трудоёмкость тестирования (экспертно в часах) по видам работ.

Контрольные вопросы.

1. Что такое тест-план и из каких разделов он состоит?
2. Что такое критерии входа и выхода?
3. Как оценить трудоёмкость тестирования?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 7. Тест-кейсы и чек-листы (2 ч)

Цель работы: освоить оформление тест-кейсов и чек-листов.

Краткие теоретические сведения.

Тест-кейс содержит идентификатор, название, предусловия, шаги, тестовые данные, ожидаемый результат и приоритет. Чек-лист – перечень проверок без подробных шагов; применяется для быстрых проверок. Требования к тест-кейсу: атомарность, однозначность, независимость.

Задания.

1. Составьте чек-лист для страницы входа (не менее 15 пунктов).
2. Оформите на его основе пять подробных тест-кейсов по шаблону.
3. Проверьте качество тест-кейсов по критериям (атомарность, однозначность) и исправьте недостатки.

Контрольные вопросы.

1. Чем тест-кейс отличается от чек-листа?
2. Из каких полей состоит тест-кейс?
3. Как оценить качество тест-кейса?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 8. Матрица трассировки и критерии завершения тестирования (2 ч)

Цель работы: научиться связывать требования и тесты и определять критерии завершения.

Краткие теоретические сведения.

Матрица трассировки показывает соответствие требований тестам и позволяет оценить покрытие. Критерии завершения: выполнение заданного процента тестов, отсутствие открытых критических дефектов, достижение целевого покрытия.

Задания.

1. Составьте матрицу трассировки «требование – тест-кейсы» для учебного ресурса (не менее 10 требований).
2. Рассчитайте покрытие требований тестами и найдите непокрытые требования.
3. Сформулируйте критерии завершения тестирования для проекта.

Контрольные вопросы.

1. Что такое матрица трассировки?
2. Как вычислить покрытие требований?
3. Когда тестирование можно считать завершённым?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Раздел 2. Проведение тестирования

Тема 2.1. Разработка тестовых сценариев

Сценарии на основе требований и пользовательских историй, позитивные и негативные проверки, предусловия, шаги и ожидаемый результат, контроль покрытия требований. Объём практических занятий – 9 ч.

Практическое занятие № 9. Сценарии по требованиям и пользовательским историям (2 ч)

Цель работы: научиться проектировать тестовые сценарии по требованиям.

Краткие теоретические сведения.

Сценарии выводят из требований и пользовательских историй с критериями приёмки: каждый критерий превращается в одну или несколько проверок. Сценарий включает предусловия, шаги, тестовые данные и ожидаемый результат.

Задания.

1. Для пяти пользовательских историй учебного ресурса выпишите критерии приёмки.
2. Разработайте по ним тестовые сценарии (минимум восемь) и оформите таблицей.
3. Определите предусловия и тестовые данные.

Контрольные вопросы.

1. Как разрабатывается тестовый сценарий и на основании чего?
2. Как связаны критерии приёмки и тесты?
3. Что такое предусловие?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 10. Позитивные и негативные проверки (2 ч)

Цель работы: освоить позитивные и негативные тестовые сценарии.

Краткие теоретические сведения.

Позитивный сценарий проверяет корректную работу при допустимых данных, негативный – реакцию на недопустимые данные и действия (пустые поля, неверный формат, превышение длины, спецсимволы, повторная отправка). Хороший набор сочетает оба вида.

Задания.

1. Разработайте не менее шести тестовых сценариев для формы входа (позитивные и негативные), оформите их в таблицу с ожидаемым результатом.
2. Составьте список негативных проверок для формы регистрации (не менее десяти).
3. Выполните сценарии на учебной форме и зафиксируйте фактические результаты.

Контрольные вопросы.

1. Что такое позитивные и негативные тестовые сценарии?
2. Какие негативные проверки обязательны для формы?
3. Как оформить фактический результат?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 11. Наборы сценариев и покрытие требований (2 ч)

Цель работы: научиться формировать набор сценариев и контролировать покрытие.

Краткие теоретические сведения.

Набор сценариев группируют по функциям и приоритетам; критичные сценарии (smoke) выполняются при каждой сборке. Покрытие контролируют по матрице трассировки; избыточные сценарии удаляют.

Задания.

1. Сформируйте набор из 15–20 сценариев для учебного ресурса, распределите по приоритетам.
2. Выделите смоук-набор (5–7 сценариев) и обоснуйте выбор.
3. Проверьте покрытие требований и дополните набор.

Контрольные вопросы.

1. Что такое смоук-тестирование?
2. Как расставляют приоритеты сценариев?
3. Как избежать избыточности набора?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 12. Тестовые сценарии для интерфейса, интеграции и безопасности (3 ч)

Цель работы: разработать сценарии для пользовательского интерфейса, интеграций и базовых проверок безопасности.

Краткие теоретические сведения.

Для интерфейса проверяют соответствие макету, состояния элементов, адаптивность и доступность. Для интеграций – корректность обмена с внешними сервисами (API) и обработку сбоев. Базовые проверки безопасности: права доступа, защита от подстановки данных, экранирование ввода, защищённое соединение.

Задания.

1. Разработайте сценарии проверки соответствия интерфейса прототипу (пять сценариев).
2. Разработайте сценарии проверки работы с внешним API (успешный ответ, ошибка, таймаут).
3. Разработайте базовые проверки безопасности формы (права, ввод HTML-символов, HTTPS) и выполните их на учебном стенде.

4. Используйте нейросеть для получения черновика тест-кейсов; проверьте и доработайте (ПК 1.3).

Контрольные вопросы.

1. Какие проверки выполняют для интерфейса?
2. Как тестировать интеграцию с внешним сервисом?
3. Какие базовые проверки безопасности применимы к формам?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Тема 2.2. Инструменты подготовки тестовых данных

Требования к тестовым данным, генерация, маскирование реальных сведений, наборы для граничных случаев, инструменты подготовки и загрузки. Объём практических занятий – 6 ч.

Практическое занятие № 13. Требования к тестовым данным (2 ч)

Цель работы: изучить требования к тестовым данным и способы их получения.

Краткие теоретические сведения.

Тестовые данные должны быть достаточными, разнообразными (типичные, граничные, недопустимые), воспроизводимыми и безопасными. Источники: ручное создание, генерация, копирование производственных данных с обезличиванием.

Задания.

1. Составьте перечень сущностей и полей, для которых требуются тестовые данные в учебном ресурсе.
2. Определите для каждого поля типичные, граничные и недопустимые значения.
3. Опишите риски использования реальных данных.

Контрольные вопросы.

1. Какие требования предъявляются к тестовым данным?

2. Какие инструменты используются для подготовки тестовых данных?
3. Почему нельзя использовать реальные данные без обезличивания?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 14. Генерация и обезличивание данных (2 ч)

Цель работы: освоить генерацию тестовых данных и маскирование.

Краткие теоретические сведения.

Данные генерируют скриптами (случайные значения, библиотеки генерации), в электронных таблицах или специальными генераторами. Маскирование заменяет реальные сведения вымышленными с сохранением формата.

Задания.

1. Подготовьте тестовые данные (не менее 20 записей) для таблицы пользователей: ФИО, адрес электронной почты, телефон, дата рождения.
2. Выполните обезличивание предложенной выгрузки (замена ФИО, телефонов, почты с сохранением формата).
3. Проверьте уникальность и корректность форматов.

Контрольные вопросы.

1. Как генерируются и обезличиваются тестовые наборы данных?
2. Что такое маскирование данных?
3. Как обеспечить воспроизводимость данных?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 15. Загрузка тестовых данных и проверка страницы поиска (2 ч)

Цель работы: загрузить тестовые данные в стенд и выполнить проверки.

Краткие теоретические сведения.

Данные загружают в СУБД скриптами INSERT или командой загрузки из CSV (COPY, LOAD DATA). После загрузки проверяют количество записей и целостность. Страницу поиска проверяют на типичных, граничных, пустых и специальных запросах.

Задания.

1. Загрузите подготовленные данные в базу учебного стенда из CSV и проверьте количество записей.
2. Проверьте страницу поиска на подготовленных данных: точное совпадение, часть слова, регистр, пустой запрос, спецсимволы.
3. Зафиксируйте результаты и найденные дефекты.

Контрольные вопросы.

1. Как организовать загрузку тестовых данных?
2. Как проверить страницу поиска?
3. Как очистить стенд после тестирования?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Тема 2.3. Функциональное тестирование веб-ресурса

Проверка функций, форм, навигации и бизнес-сценариев, валидация данных, обработка ошибок, сессии и права доступа, кроссбраузерное тестирование и разрешения экрана. Объём практических занятий – 6 ч.

Практическое занятие № 16. Тестирование функций, форм и навигации (2 ч)

Цель работы: освоить функциональное тестирование форм, навигации и ссылок.

Краткие теоретические сведения.

Функциональное тестирование проверяет работу функций по требованиям. Для форм проверяют обязательность, форматы, ограничения

длины, сообщения об ошибках; для навигации – работу меню, ссылок, кнопки «назад», хлебных крошек, отсутствие битых ссылок.

Задания.

1. Проверьте формы, навигацию и ссылки учебного ресурса по чек-листу и оформите результаты в таблицу.
2. Проверьте сайт на битые ссылки и ошибки консоли браузера.
3. Проверьте бизнес-сценарий (регистрация – вход – оформление заказа).

Контрольные вопросы.

1. Что включает функциональное тестирование веб-ресурса?
2. Как проверяются формы, навигация и ссылки на сайте?
3. Как найти битые ссылки?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 17. Валидация, ошибки, сессии и права доступа (2 ч)

Цель работы: проверить валидацию данных, обработку ошибок, сессии и права доступа.

Краткие теоретические сведения.

Проверяют валидацию на клиенте и на сервере (отключение JavaScript), реакцию на недопустимые данные, обработку ошибок (страницы 404, 500), сроки жизни сессии, выход из системы, доступ к страницам без прав.

Задания.

1. Проверьте валидацию формы на клиенте и на сервере; опишите различия.
2. Проверьте работу сессии: выход, истечение, повторный вход, доступ к закрытым страницам по прямой ссылке.

3. Проверьте права ролей (администратор, пользователь) по матрице прав.

Контрольные вопросы.

1. Какие проверки выполняются при тестировании авторизации?
2. Почему валидацию необходимо проверять на сервере?
3. Как проверить разграничение прав?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 18. Кроссбраузерное тестирование и разрешения экрана (2 ч)

Цель работы: выполнить кроссбраузерную проверку и проверку на разных экранах.

Краткие теоретические сведения.

Проверку выполняют в браузерах разных движков и на разных разрешениях (режим адаптивного просмотра инструментов разработчика, реальные устройства). Фиксируют различия в отображении и поведении.

Задания.

1. Проверьте ключевые страницы в двух браузерах и на трёх разрешениях экрана; оформите матрицу совместимости.
2. Зафиксируйте отличия и опишите дефекты.
3. Составьте рекомендации по поддерживаемым браузерам.

Контрольные вопросы.

1. Что такое кроссбраузерное тестирование?
2. Как проверить адаптивность страницы?
3. Как выбрать набор браузеров и устройств?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Раздел 3. Результаты тестирования

Тема 3.1. Регистрация дефектов. Отчёт по результатам тестирования

Структура отчёта о дефекте, шаги воспроизведения, вложения и журналы, приоритет и серьёзность, жизненный цикл дефекта, итоговый отчёт с метриками и выводами. Объём практических занятий – 6 ч.

Практическое занятие № 19. Регистрация дефектов в Redmine (2 ч)

Цель работы: освоить оформление и регистрацию дефектов.

Краткие теоретические сведения.

Отчёт о дефекте содержит краткое название, описание, шаги воспроизведения, ожидаемый и фактический результат, окружение (браузер, ОС, версия), приоритет и серьёзность, вложения (снимки экрана, журналы). Серьёзность – влияние на систему; приоритет – очерёдность исправления.

Задания.

1. Найдите и оформите в Redmine не менее пяти дефектов в учебном веб-ресурсе с полным описанием, шагами воспроизведения и приоритетом.
2. Прикрепите снимки экрана и фрагменты журналов.
3. Проверьте качество оформления дефектов друг у друга (взаимная проверка) и исправьте замечания.

Контрольные вопросы.

1. Как регистрируется дефект: какие поля обязательны в описании?
2. Как определяется приоритет и серьёзность дефекта?
3. Что такое воспроизводимость дефекта?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 20. Жизненный цикл дефекта и повторное тестирование (2 ч)

Цель работы: изучить жизненный цикл дефекта и выполнить повторное тестирование.

Краткие теоретические сведения.

Состояния дефекта: новый, назначен, в работе, исправлен, проверен, закрыт (или отклонён, открыт повторно). После исправления выполняют повторную проверку дефекта и регрессионное тестирование связанных функций.

Задания.

1. Проведите в Redmine дефекты по жизненному циклу (смена статусов, назначение, комментарии) в паре «тестировщик – разработчик».
2. Выполните регрессионное тестирование после исправления и зафиксируйте результат.
3. Определите, какие дефекты следует отклонить и почему.

Контрольные вопросы.

1. Какой жизненный цикл дефекта?
2. Как организовать регрессионное тестирование после исправления дефекта?
3. Когда дефект отклоняют?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 21. Метрики и отчёт по результатам тестирования (2 ч)

Цель работы: подготовить итоговый отчёт о тестировании с метриками.

Краткие теоретические сведения.

Метрики: число выполненных и пройденных тестов, процент прохождения, покрытие требований, число дефектов по серьёзности и

статусам, плотность дефектов. Отчёт включает объём, результаты, открытые дефекты, риски, выводы о готовности и рекомендации.

Задания.

1. Составьте отчёт о тестировании: объём, пройденные и непройденные тесты, открытые дефекты, выводы и рекомендации.
2. Рассчитайте метрики (процент прохождения, покрытие, дефекты по серьёзности) в LibreOffice Calc и постройте диаграммы.
3. Сделайте вывод о готовности ресурса к приёмке.

Контрольные вопросы.

1. Что включает отчёт по результатам тестирования?
2. Какие метрики покрытия и прохождения тестов используются?
3. Как документируются результаты тестирования для заказчика?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Тема 3.2. Автоматизация тестирования: обзор подходов

Цели автоматизации, критерии отбора сценариев, инструменты автоматизации веб-тестирования, структура автотеста, достоинства, ограничения и стоимость сопровождения. Объём практических занятий – 6 ч.

Практическое занятие № 22. Цели автоматизации и отбор сценариев (2 ч)

Цель работы: определить цели автоматизации и критерии отбора сценариев.

Краткие теоретические сведения.

Автоматизируют повторяемые, стабильные, критичные и трудоёмкие при ручной проверке сценарии (регрессия, смоук). Не стоит автоматизировать нестабильные функции, разовые проверки, проверки удобства. Окупаемость зависит от числа запусков и стоимости сопровождения.

Задания.

1. Из набора сценариев, разработанного ранее, отберите кандидатов на автоматизацию и обоснуйте выбор (таблица критериев).
2. Оцените ориентировочную окупаемость автоматизации (часы ручных проверок против часов разработки и сопровождения).
3. Составьте список рисков автоматизации.

Контрольные вопросы.

1. Что такое автоматизация тестирования и какие задачи она решает?
2. Когда автоматизация целесообразна, а когда нет?
3. Из чего складывается стоимость сопровождения автотестов?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 23. Запись и воспроизведение: Selenium IDE (2 ч)

Цель работы: освоить запись и воспроизведение сценария в Selenium IDE.

Краткие теоретические сведения.

Selenium IDE – расширение браузера для записи действий пользователя и их воспроизведения. Сценарий состоит из команд (открыть страницу, нажать, ввести, проверить). Проверки (assert, verify) подтверждают результат.

Задания.

1. Запишите в Selenium IDE сценарий входа в учебный ресурс с проверкой результата.
2. Добавьте проверки (assert) и параметризацию значений.
3. Выполните сценарий несколько раз и сохраните результаты.

Контрольные вопросы.

1. Какие подходы к автоматизации тестирования существуют?
2. Как устроен автотест: действия и проверки?

3. Что такое устойчивость локаторов?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

Практическое занятие № 24. Структура автотеста и обзор инструментов (2 ч)

Цель работы: изучить структуру автотеста и обзор инструментов.

Краткие теоретические сведения.

Автотест состоит из подготовки (предусловия), действий, проверок и очистки. Инструменты: Selenium WebDriver, Playwright, Cypress; для API – Postman, pytest с requests. Сценарии хранят в репозитории Git и запускают при каждой сборке (CI).

```
import requests

def test_home_page():
    r = requests.get('http://localhost/', timeout=5)
    assert r.status_code == 200
    assert 'Главная' in r.text
```

Задания.

1. Напишите простой автотест на JavaScript (Node.js) или Python для проверки заголовка страницы или статуса ответа HTTP.
2. Сравните три инструмента автоматизации (возможности, язык, лицензия) в таблице.
3. Опишите, как автотесты встраиваются в процесс разработки (CI) и хранятся в Git.

Контрольные вопросы.

1. Как устроен автотест и какова его структура?
2. Какие инструменты автоматизации веб-тестирования вы знаете?
3. Чем автотест интерфейса отличается от автотеста API?

Отчёт: описание объекта тестирования, тестовая документация, результаты выполнения, перечень дефектов, ответы на контрольные вопросы.

3. Методические указания к самостоятельной работе

Объём самостоятельной работы – 10 ч. Задания выполняются в свободное от занятий время; результаты сдаются в электронном виде и проверяются на очередном практическом занятии.

Таблица 2 – Задания для самостоятельной работы

№	Тема	Задание	Время, ч
1	1.1. Качество ПО	Подготовьте сообщение о моделях качества ПО (ISO/IEC 25010) с таблицей характеристик и примерами проверок.	1
2	1.2. Тест-дизайн	Постройте классы эквивалентности и граничные значения для пяти полей формы заказа; оформите тесты в таблице.	1
3	1.3. Тестовая документация	Подготовьте шаблоны тест-плана, тест-кейса и чек-листа для учебного проекта.	1
4	2.1. Тестовые сценарии	Разработайте 15 тестовых сценариев для модуля учебного проекта, свяжите их с требованиями в матрице трассировки.	2
5	2.2. Тестовые данные	Подготовьте набор тестовых данных для трёх таблиц учебной базы (не менее 30 записей) и скрипт загрузки.	1
6	2.3. Функциональное тестирование	Проверьте один реальный сайт по чек-листу функционального тестирования и оформите отчёт о найденных замечаниях.	1
7	3.1. Отчёт о тестировании	Подготовьте итоговый отчёт по тестированию модуля проекта: метрики, диаграммы, дефекты, выводы.	2
8	3.2. Автоматизация	Автоматизируйте три сценария учебного проекта (любым инструментом) и приложите результаты запуска.	1

4. Рекомендуемая литература и информационные ресурсы

Основная литература

1. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для СПО / Т. М. Зубкова. – 4-е изд., стер. – Санкт-Петербург: Лань, 2025. – 252 с. – ISBN 978-5-507-50993-5. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/505511
2. Старолетов, С. М. Основы тестирования программного обеспечения: учебное пособие для СПО / С. М. Старолетов. – 5-е изд., стер. – Санкт-Петербург: Лань, 2025. – 192 с. – ISBN 978-5-507-53820-1. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/499391

Дополнительная литература

1. Старолетов, С. М. Основы тестирования программного обеспечения: учебное пособие для СПО / С. М. Старолетов. – 4-е изд., стер. – Санкт-Петербург: Лань, 2024. – 192 с. – ISBN 978-5-507-47492-9. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/382343
2. Игнатьев, А. В. Тестирование программного обеспечения: учебное пособие для СПО / А. В. Игнатьев. – 4-е изд., стер. – Санкт-Петербург: Лань, 2026. – 56 с. – ISBN 978-5-507-54395-3. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/508067

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Единый реестр российского ПО. – URL: https://reestr.digital.gov.ru/
2. КонсультантПлюс. – URL: https://www.consultant.ru/

Приложение А. Образец титульного листа отчёта

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО Университет биотехнологий
Институт цифровых технологий
Кафедра информационных технологий и моделирования

Отчёт по практическому занятию № …

по дисциплине «Тестирование информационных ресурсов»

Тема: …

Выполнил(а): студент(ка) … курса, группы …
ФИО

Проверил: должность,
ФИО

Новосибирск 202…