

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В.Н. Татищева»
(Астраханский государственный университет им. В.Н. Татищева)

Филиал АГУ им. В.Н. Татищева в г. Знаменске Астраханской области

СОГЛАСОВАНО
Руководитель ОПОП
Бориско С.Н.
«13» ноября 2025 г.

УТВЕРЖДАЮ
Председатель ЦК (МО)
Фисенко Т.Ю.
протокол заседания ЦК (МО) №3
от «13» ноября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по профессиональному модулю

ПМ.02 СОПРОВОЖДЕНИЕ ПРОЦЕССОВ ТЕСТИРОВАНИЯ В ПРОЦЕССЕ
ЭКСПЛУАТАЦИИ

ПМ.02.МДК.01 ОБЕСПЕЧЕНИЕ КАЧЕСТВА ПРОГРАММНОГО
ОБОЕСПЕЧЕНИЯ

ПМ.02.МДК.02 АВТОМАТИЗАЦИЯ ПРОЦЕССОВ ТЕСТИРОВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ПМ.02 УЧЕБНАЯ ПРАКТИКА

ПМ.02 ПРОИЗВОДСТВЕННАЯ ПРАКТИКА

ПМ.03 ПРОИЗВОДСТВЕННАЯ (ПРЕДДИПЛОМНАЯ) ПРАКТИКА

| | |
|------------------------------|---|
| Составители: | Бориско С.Н., к.т.н., доцент, завкафедрой ЗнМИ; Мустафаев Н.Г., к.т.н., доцент кафедры ЗнМИ; Тимошкин А.А., к.т.н., доцент кафедры ЗнМИ; Устинов А.С., к.т.н., доцент кафедры ЗнМИ; Каштанов Д.Ю., ассистент кафедры ЗнМИ |
| Согласовано с работодателями | Литвинов С.П., к.т.н., заместитель командира войсковой части 15644 по научно-исследовательской и испытательной работе; Кириянов М.Н., ведущий инженер ПАО «Ростелеком» |
| Наименование специальности | 09.02.12 Техническая эксплуатация и сопровождение информационных систем |
| Квалификация выпускника | Специалист по технической эксплуатации и сопровождению информационных систем |
| Форма обучения | очная |
| Год приема (курс) | 2026 (3 курс) |

Знаменск, 2025 г.

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ

**2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ,
ПОДЛЕЖАЩИЕ ПРОВЕРКЕ**

**3. ФОРМЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ ЭЛЕМЕНТОВ
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

**4. КОНТРОЛЬНЫЕ ЗАДАНИЯ ДЛЯ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ
ПРАКТИКИ**

**5. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ**

1. Общие положения

Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки результатов освоения обучающимися междисциплинарного курса ПМ.02 Сопровождение процессов тестирования в процессе эксплуатации.

Результатом освоения профессионального модуля является готовность обучающимися к выполнения следующих видов профессиональной деятельности (в соответствии с ФГОС СПО): сопровождение процессов тестирования в процессе эксплуатации.

2. Результаты освоения профессионального модуля, подлежащие проверке

| Код компетенции | Планируемые результаты освоения учебного предмета (междисциплинарного курса) | | |
|-----------------|--|---|---|
| | Практический опыт | Умения | Знания |
| ПК 2.1 | <ul style="list-style-type: none"> – Изучения необходимых для проведения тестирования ПО действий, перечисленных в задании на тестирование – Подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости) – Оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения – Настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции Формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами | <ul style="list-style-type: none"> - Устанавливать корректную последовательность операций при выполнении тестирования ПО - Выявлять недостающую информацию для выполнения тестирования ПО в заданном объеме - Устанавливать операционные системы - Выполнять базовую настройку операционных систем - Подготавливать необходимые средства и ресурсы для выполнения задания по тестированию ПО - Составлять отчет о результатах подготовки к выполнению тестирования ПО | <ul style="list-style-type: none"> - Основную терминологию по тестированию ПО - Язык, на котором написана техническая документация тестируемого ПО на уровне, достаточном для чтения технической документации - Основные термины и сокращения, используемые в технической документации и принятые в организации - Процедуры обеспечения безопасности при выполнении тестирования ПО - Область применения инструментальных средств для выполнения тестирования ПО - Особенности основных операционных систем |

| | | | |
|--------|--|--|---|
| | | | <ul style="list-style-type: none"> - Требования по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирус |
| ПК 2.2 | <ul style="list-style-type: none"> – Проверки компонентов инструментария и тестируемого ПО на корректное начальное состояние для начала тестирования – Выполнения тестовых процедур на тестовых данных – Сравнения фактического и ожидаемого результатов выполнения тестовых процедур Формирования и представления отчетности о выполнении процесса тестирования ПО в соответствии с установленными регламентами | <ul style="list-style-type: none"> – Нормативно-технические материалы по вопросам испытания и тестирования ПО – Основные термины и сокращения, используемые в технической документации и принятые в организации – Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования ПО соответствующего типа – Основы теории алгоритмов и дискретной математики в объеме полученного профессионального образования Синтаксис языка программирования тестируемого ПО, особенности программирования на этом языке, стандартные | <ul style="list-style-type: none"> – Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования – Использовать системы контроля дефектов ПО – Составлять отчет о выполнении тестирования ПО Работать в команде со специалистами по тестированию ПО и разработчиками |

| | | | |
|--------|---|--|---|
| | | библиотеки языка программирования | |
| ПК 2.3 | <ul style="list-style-type: none"> – Проверки полноты эксплуатационной и технической документации на ПО – Выявления недостатков эксплуатационной и технической документации на ПО и ее несоответствия внутренним стандартам качества организации – Проверки эксплуатационной и технической документации на ПО на соответствие требованиям заказчика – Выполнения действий по указаниям в эксплуатационной и технической документации на ПО – Проверки соответствия действительных и указанных в эксплуатационной и технической документации на ПО результатов Выявления несовпадений действительных и указанных в эксплуатационной и технической документации результатов регистрации найденных дефектов ПО в системе контроля дефектов | <ul style="list-style-type: none"> – Нормативно-технические материалы по вопросам испытания и тестирования ПО – Основные понятия о качестве ПО – Виды технической документации – Требования по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирусной защиты Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО | <ul style="list-style-type: none"> – Основы современных СУБД – Теорию баз данных – Основы программирования – Современные объектно-ориентированные языки программирования – Современные структурные языки программирования – Языки современных бизнес-приложений – Современные методики тестирования разрабатываемых ИС: инструменты и методы модульного тестирования – Методы верификации программного обеспечения – Источники информации, необходимой для профессиональной деятельности в рамках технической поддержки процессов создания (модификации) и сопровождения ИС Лучшие практики создания (модификации) и сопровождения ИС в экономике |
| ПК 2.4 | <ul style="list-style-type: none"> – Выполнения начальных настроек для проведения тестирования ПО – Выполнения необходимых видов | <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера | <ul style="list-style-type: none"> – Кодировать на языках программирования ИС – Тестировать |

| | | | |
|--------|---|--|--|
| | <p>тестирования ПО в соответствии с планом тестирования</p> <ul style="list-style-type: none"> – Проведения автоматизированного тестирования ПО при необходимости – Составления статистики выполнения тестов – Проведения анализа полученных результатов тестирования ПО по разработанным тестовым случаям на соответствие ожидаемым результатам – Оптимизации тестовых наборов – Составления новых тестовых случаев и повторение тестирования при необходимости <p>Формирования и представления отчетности о проведенном тестировании ПО в соответствии с установленными регламентами</p> | <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения – Стандарты оформления кода для используемых языков программирования – Основные термины и сокращения, используемые в технической документации и принятые в организации – Основы алгоритмизации и программирования <p>Жизненный цикл программного продукта</p> | <p>результаты разработки ИС</p> <p>Работать с записями по качеству (в том числе с корректирующими действиями, предупреждающими действиями, запросами на исправление несоответствий) при выполнении технической поддержки процессов создания (модификации) и сопровождения ИС</p> |
| ПК 2.5 | <ul style="list-style-type: none"> – Определения причины сбоя системы совместно с разработчиками – Устранения причины сбоя системы, если она находится в компетенции специалиста, либо подготовка отчета | <ul style="list-style-type: none"> – Находить и использовать информацию, необходимую для восстановления тестов после сбоя – Взаимодействовать | <ul style="list-style-type: none"> – Архитектуру тестируемой системы – Основы работы в операционной системе, в которой производится тестирование, на |

| | | | |
|---------|---|--|---|
| | <p>руководителю и группе разработчиков</p> <ul style="list-style-type: none"> – Выполнения настройки для повторного тестирования после сбоя – Восстановления/изменения автоматизированных тестов после сбоя при необходимости в соответствии с планом/регламентом восстановления – Проведения повторного тестирования ПО <p>Формирования и представления отчетности о восстановлении работоспособности ПО в соответствии с установленными регламентами</p> | <p>ь с командой разработчиков при восстановлении системы после сбоя</p> <ul style="list-style-type: none"> – Применять языки программирования для написания программного кода – Использовать системы автоматизированного тестирования ПО <p>Составлять отчет о восстановлении работоспособности ПО</p> | <p>уровне, необходимом для тестирования разработанного ПО</p> <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения – Принципы регрессионного тестирования ПО – Алгоритмы решения типовых задач, области и способы их применения <p>Основные термины и сокращения, используемые в технической документации и принятые в организации</p> |
| ПК 2.6. | <ul style="list-style-type: none"> – Получения обновленной версии ПО – Определения масштабов | <ul style="list-style-type: none"> – Жизненный цикл ПО, жизненный цикл дефекта ПО | <ul style="list-style-type: none"> – Основы системного администрирования |

| | | | |
|--|---|---|---|
| | <p>изменений для выявления необходимости проведения регрессионных тестов</p> <ul style="list-style-type: none"> – Определения оптимального перечня тестов для повторного тестирования ПО <p>Выполнения тестовых сценариев, выявивших дефекты ПО, для подтверждения успешности их выполнения после исправления ПО</p> | <ul style="list-style-type: none"> – Принципы регрессионного тестирования ПО – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения <p>Основные инструментальные средства организации работы в команде</p> | <ul style="list-style-type: none"> – Основы администрирования баз данных – Коммуникационное оборудование – Сетевые протоколы – Основы современных операционных систем – Основы современных СУБД – Устройство и функционирование современных ИС – Основы архитектуры мультиарендного программного обеспечения – Основы ИБ организации – Источники информации, необходимой для профессиональной деятельности в рамках технической поддержки процессов создания (модификации) и сопровождения ИС <p>Лучшие практики создания (модификации) и сопровождения ИС в экономике</p> |
|--|---|---|---|

3. Формы контроля и оценивания элементов профессионального модуля

| Элемент модуля | Наименование оценочного средства текущего контроля и промежуточной аттестации | |
|---|---|--|
| | Текущий контроль | Промежуточная аттестация |
| <p>МДК.01 обеспечение качества программного обеспечения</p> <p>МДК.02 автоматизация процессов тестирования программного обеспечения</p> | <p>Тестовые вопросы, вопросы для устного/письменного опроса</p> <p>Практические задания: решения методических задач</p> | <p>Вопросы к экзамену</p> <p>Курсовые проекты (работы)</p> <p>Прохождения практики</p> |

4. Контрольные задания для оценивания результатов освоения практики

4.1. Виды контрольных заданий

1. Тестовые задания
2. Практические задания
3. Контрольная работа

4.2. Требования к дифференциальному зачету по производственной практике

Дифференциальный зачет по производственной практике выставляется на основании характеристики профессиональной деятельности обучающегося на практике с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и требованиями организации, в которой проходила практика.

5. Контрольно-оценочные материалы задания для промежуточной аттестации

Вопросы к экзамену

1. Дайте определение модульному тестированию. Что считается «модулем» (юнитом) в контексте такого тестирования?
2. Сформулируйте основные цели и преимущества модульного тестирования для процесса разработки и качества кода.
3. Что такое тестовое покрытие (code coverage)? Назовите основные его типы (покрытие строк, ветвей, условий) и объясните, какой из них является наиболее значимым.
4. Опишите цикл «Красный — Зеленый — Рефакторинг» в методологии разработки через тестирование (TDD). Какова роль модульных тестов в этом цикле?
5. Что такое изолированный unit-тест? Почему изоляция является ключевым принципом модульного тестирования?
6. Объясните разницу между заглушками (Stubs), моками (Mocks) и фейками (Fakes). В каких сценариях модульного тестирования применяется каждый из них?
7. Что такое параметризованные тесты (Parameterized Tests)? Как они помогают улучшить тестовое покрытие и избежать дублирования кода?
8. Назовите и охарактеризуйте три основных компонента структуры модульного теста (Arrange-Act-Assert / Given-When-Then).
9. Какие атрибуты (аннотации) в популярных фреймворках (JUnit, NUnit, pytest) используются для обозначения тестового метода, методов настройки и очистки?
10. В чем заключаются основные трудности и ограничения модульного тестирования? Приведите примеры кода, который сложно покрыть юнит-тестами.
11. Что является объектом тестирования документации? Перечислите основные виды документации, которые требуют проверки.
12. Объясните, почему тестирование документации является важной частью обеспечения качества ПО. Какие риски оно помогает смягчить?
13. Какие аспекты проверяются при тестировании пользовательской документации (руководства пользователя, справка)? Назовите не менее 5 критериев.
14. Что такое тестирование на соответствие требованиям (Requirements Traceability)? Какую роль в этом процессе играет техническая документация?
15. Опишите процесс проверки документации по установке и развертыванию (Deployment/Installation Guide). Что является критически важным в таких документах?
16. Как можно автоматизировать проверку актуальности документации? Приведите примеры инструментов или практик (например, встроенная документация в код — docstrings, Swagger/OpenAPI).

17. Что такое «живая документация» (Living Documentation) в контексте BDD? Как фреймворки вроде Cucumber связывают тесты и документацию?
18. Какие типовые дефекты чаще всего встречаются в документации (фактические, структурные, стилистические)? Приведите примеры каждого.
19. Кто обычно отвечает за тестирование документации в проекте? Обоснуйте свою точку зрения.
20. Как тестирование API-документации (например, в формате OpenAPI/Swagger) связано с интеграционным тестированием?
21. Дайте определение интеграционному тестированию. Какова его основная цель в отличие от модульного тестирования?
22. Опишите основные стратегии (подходы) проведения интеграционного тестирования: «снизу вверх» (Bottom-Up), «сверху вниз» (Top-Down) и «большой взрыв» (Big Bang). Укажите их преимущества и недостатки.
23. Что такое драйверы (Drivers) и заглушки (Stubs) в интеграционном тестировании? В каких стратегиях они применяются?
24. Дайте определение системному тестированию. Какие аспекты системы проверяются на этом уровне?
25. Чем функциональное системное тестирование отличается от нефункционального (например, тестирования производительности, надежности, удобства использования)?
26. Что такое сквозное (End-to-End, E2E) тестирование? Как оно соотносится с системным тестированием?
27. Опишите типичный сценарий для интеграционного тестирования взаимодействия веб-клиента с бэкенд-API и базой данных.
28. Какие основные сложности возникают при автоматизации интеграционных и системных тестов по сравнению с модульными?
29. Что такое тестовое окружение для интеграционного и системного тестирования? Каковы ключевые требования к его состоянию и конфигурации?
30. Как регрессионное тестирование связано с интеграционным и системным уровнями? Почему на этих уровнях особенно важна автоматизация регрессии?
31. Дайте определение надежности программного обеспечения в контексте эксплуатации. Какие метрики используются для ее оценки (MTBF, MTTR, Availability)?
32. Как автоматизированное тестирование (в частности, интеграционное и системное) способствует повышению надежности ПО перед выпуском в эксплуатацию?
33. Что такое тестирование восстановления после сбоев (Recovery Testing) и тестирование отказоустойчивости (Failover Testing)? Какие аспекты надежности они проверяют?
34. Как нагрузочное (Load) и стрессовое (Stress) тестирование связаны с проверкой надежности системы в условиях, близких к эксплуатационным?
35. Объясните, как практика мониторинга в production-среде (логи, метрики, трассировка) позволяет оценивать и поддерживать надежность работающей системы.
36. Дайте определение безопасности ПО. Назовите три основных принципа информационной безопасности (CIA Triad).
37. Что такое статический анализ безопасности кода (SAST) и как его можно интегрировать в процесс автоматизированного тестирования на ранних этапах?
38. Что такое динамический анализ безопасности приложений (DAST)? Приведите пример типовой уязвимости, которую он помогает обнаружить.
39. Как автоматизированные интеграционные тесты могут проверять корректность работы механизмов аутентификации и авторизации в системе?
40. Опишите концепцию DevSecOps. Как она меняет подход к тестированию безопасности, интегрируя его в цикл разработки и эксплуатации?

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В.Н. Татищева»
(Астраханский государственный университет им. В.Н. Татищева)

Филиал АГУ им. В.Н. Татищева в г. Знаменске Астраханской области

СОГЛАСОВАНО
Руководитель ОПОП
Бориско С.Н.
«13» ноября 2025 г.

УТВЕРЖДАЮ
Председатель ЦК (МО)
Фисенко Т.Ю.
протокол заседания ЦК (МО) №3
от «13» ноября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по междисциплинарному курсу

ПМ.02.МДК.01 ОБЕСПЕЧЕНИЕ КАЧЕСТВА ПРОГРАММНОГО
ОБОЕСПЕЧЕНИЯ

| | |
|------------------------------|---|
| Составитель | Бориско С.Н., к.т.н., доцент, завкафедрой ЗнМИ; Мустафаев Н.Г., к.т.н., доцент кафедры ЗнМИ; Тимошкин А.А., к.т.н., доцент кафедры ЗнМИ; Устинов А.С., к.т.н., доцент кафедры ЗнМИ; Каштанов Д.Ю., ассистент кафедры ЗнМИ |
| Согласовано с работодателями | Литвинов С.П., к.т.н., заместитель командира войсковой части 15644 по научно-исследовательской и испытательной работе; Кириянов М.Н., ведущий инженер ПАО «Ростелеком» |
| Наименование специальности | 09.02.12 Техническая эксплуатация и сопровождение информационных систем |
| Квалификация выпускника | Специалист по технической эксплуатации и сопровождению информационных систем |
| Форма обучения | очная |
| Год приема (курс) | 2026 (3 курс) |

Знаменск, 2025 г.

СОДЕРЖАНИЕ

- 1. ОБЩИЕ ПОЛОЖЕНИЯ**
- 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА,
ПОДЛЕЖАЩИЕ ПРОВЕРКЕ**
- 3. ФОРМЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ ЭЛЕМЕНТОВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**
- 4. КОНТРОЛЬНЫЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ
ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА**

1. Общие положения

Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки результатов освоения обучающимися междисциплинарного курса ПМ.01.МДК.02 «Разработка информационных систем».

ФОС включает контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, разработанные в соответствии с требованиями ФГОС СПО и содержанием рабочей программы междисциплинарного курса.

2. Результаты освоения междисциплинарного курса, подлежащие проверке

| Код компетенции | Планируемые результаты освоения учебного предмета (междисциплинарного курса) | | |
|-----------------|--|---|--|
| | Практический опыт | Умения | Знания |
| ПК 2.1 | <ul style="list-style-type: none"> – Изучения необходимых для проведения тестирования ПО действий, перечисленных в задании на тестирование – Подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости) – Оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения – Настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции Формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами | <ul style="list-style-type: none"> - Устанавливать корректную последовательность операций при выполнении тестирования ПО - Выявлять недостающую информацию для выполнения тестирования ПО в заданном объеме - Устанавливать операционные системы - Выполнять базовую настройку операционных систем - Подготавливать необходимые средства и ресурсы для выполнения задания по тестированию ПО - Составлять отчет о результатах подготовки к выполнению тестирования ПО | <ul style="list-style-type: none"> - Основную терминологию по тестированию ПО - Язык, на котором написана техническая документация тестируемого ПО на уровне, достаточном для чтения технической документации - Основные термины и сокращения, используемые в технической документации и принятые в организации - Процедуры обеспечения безопасности при выполнении тестирования ПО - Область применения инструментальных средств для выполнения тестирования ПО - Особенности основных операционных систем - Требования по |

| | | | |
|--------|--|---|---|
| | | | <p>обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирус</p> |
| ПК 2.2 | <ul style="list-style-type: none"> – Проверки компонентов инструментария и тестируемого ПО на корректное начальное состояние для начала тестирования – Выполнения тестовых процедур на тестовых данных – Сравнения фактического и ожидаемого результатов выполнения тестовых процедур <p>Формирования и представления отчетности о выполнении процесса тестирования ПО в соответствии с установленными регламентами</p> | <ul style="list-style-type: none"> – Нормативно-технические материалы по вопросам испытания и тестирования ПО – Основные термины и сокращения, используемые в технической документации и принятые в организации – Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования ПО соответствующего типа – Основы теории алгоритмов и дискретной математики в объеме полученного профессионального образования Синтаксис языка программирования тестируемого ПО, особенности программирования на этом языке, стандартные библиотеки языка | <ul style="list-style-type: none"> – Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования – Использовать системы контроля дефектов ПО – Составлять отчет о выполнении тестирования ПО <p>Работать в команде со специалистами по тестированию ПО и разработчиками</p> |

| | | | |
|--------|---|--|---|
| | | программирования | |
| ПК 2.3 | <ul style="list-style-type: none"> – Проверки полноты эксплуатационной и технической документации на ПО – Выявления недостатков эксплуатационной и технической документации на ПО и ее несоответствия внутренним стандартам качества организации – Проверки эксплуатационной и технической документации на ПО на соответствие требованиям заказчика – Выполнения действий по указаниям в эксплуатационной и технической документации на ПО – Проверки соответствия действительных и указанных в эксплуатационной и технической документации на ПО результатов Выявления несовпадений действительных и указанных в эксплуатационной и технической документации результатов регистрация найденных дефектов ПО в системе контроля дефектов | <ul style="list-style-type: none"> – Нормативно-технические материалы по вопросам испытания и тестирования ПО – Основные понятия о качестве ПО – Виды технической документации – Требования по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирусной защиты Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО | <ul style="list-style-type: none"> – Основы современных СУБД – Теорию баз данных – Основы программирования – Современные объектно-ориентированные языки программирования – Современные структурные языки программирования – Языки современных бизнес-приложений – Современные методики тестирования разрабатываемых ИС: инструменты и методы модульного тестирования – Методы верификации программного обеспечения – Источники информации, необходимой для профессиональной деятельности в рамках технической поддержки процессов создания (модификации) и сопровождения ИС Лучшие практики создания (модификации) и сопровождения ИС в экономике |
| ПК 2.4 | <ul style="list-style-type: none"> – Выполнения начальных настроек для проведения тестирования ПО – Выполнения необходимых видов тестирования ПО в | <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники | <ul style="list-style-type: none"> – Кодировать на языках программирования ИС – Тестировать результаты |

| | | | |
|--------|---|--|---|
| | <p>соответствии с планом тестирования</p> <ul style="list-style-type: none"> – Проведения автоматизированного тестирования ПО при необходимости – Составления статистики выполнения тестов – Проведения анализа полученных результатов тестирования ПО по разработанным тестовым случаям на соответствие ожидаемым результатам – Оптимизации тестовых наборов – Составления новых тестовых случаев и повторение тестирования при необходимости <p>Формирования и представления отчетности о проведенном тестировании ПО в соответствии с установленными регламентами</p> | <p>тестирования ПО, базирующиеся на спецификации</p> <ul style="list-style-type: none"> – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения – Стандарты оформления кода для используемых языков программирования – Основные термины и сокращения, используемые в технической документации и принятые в организации – Основы алгоритмизации и программирования <p>Жизненный цикл программного продукта</p> | <p>разработки ИС</p> <p>Работать с записями по качеству (в том числе с корректирующими действиями, предупреждающими действиями, запросами на исправление несоответствий) при выполнении технической поддержки процессов создания (модификации) и сопровождения ИС</p> |
| ПК 2.5 | <ul style="list-style-type: none"> – Определения причины сбоя системы совместно с разработчиками – Устранения причины сбоя системы, если она находится в компетенции специалиста, либо подготовка отчета руководителю и группе | <ul style="list-style-type: none"> – Находить и использовать информацию, необходимую для восстановления тестов после сбоя – Взаимодействовать с командой | <ul style="list-style-type: none"> – Архитектуру тестируемой системы – Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом |

| | | | |
|---------|---|---|---|
| | <p>разработчиков</p> <ul style="list-style-type: none"> – Выполнения настройки для повторного тестирования после сбоя – Восстановления/изменения автоматизированных тестов после сбоя при необходимости в соответствии с планом/регламентом восстановления – Проведения повторного тестирования ПО Формирования и представления отчетности о восстановлении работоспособности ПО в соответствии с установленными регламентами | <p>разработчиков при восстановлении системы после сбоя</p> <ul style="list-style-type: none"> – Применять языки программирования для написания программного кода – Использовать системы автоматизированного тестирования ПО Составлять отчет о восстановлении работоспособности ПО | <p>для тестирования разработанного ПО</p> <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения – Принципы регрессионного тестирования ПО – Алгоритмы решения типовых задач, области и способы их применения Основные термины и сокращения, используемые в технической документации и принятые в организации |
| ПК 2.6. | <ul style="list-style-type: none"> – Получения обновленной версии ПО – Определения масштабов изменений для выявления | <ul style="list-style-type: none"> – Жизненный цикл ПО, жизненный цикл дефекта ПО – Принципы | <ul style="list-style-type: none"> – Основы системного администрирования – Основы |

| | | | |
|--|---|--|--|
| | <p>необходимости проведения регрессионных тестов</p> <ul style="list-style-type: none"> – Определения оптимального перечня тестов для повторного тестирования ПО Выполнения тестовых сценариев, выявивших дефекты ПО, для подтверждения успешности их выполнения после исправления ПО | <p>регрессионного тестирования ПО</p> <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения <p>Основные инструментальные средства организации работы в команде</p> | <p>администрирования баз данных</p> <ul style="list-style-type: none"> – Коммуникационное оборудование – Сетевые протоколы – Основы современных операционных систем – Основы современных СУБД – Устройство и функционирование современных ИС – Основы архитектуры мультиарендного программного обеспечения – Основы ИБ организации – Источники информации, необходимой для профессиональной деятельности в рамках технической поддержки процессов создания (модификации) и сопровождения ИС <p>Лучшие практики создания (модификации) и сопровождения ИС в экономике</p> |
|--|---|--|--|

2. Распределение оценивания результатов обучения по видам контроля

| Наименование элемента практического опыта, умений или знаний | Наименование оценочного средства текущего контроля и промежуточной аттестации | |
|--|---|--------------------------|
| | Текущий контроль | Промежуточная аттестация |
| ПО.1 Изучения необходимых для проведения тестирования ПО действий, перечисленных в задании на тестирование | Компьютерное тестирование на знание терминологии по теме; | Вопросы к экзамену |
| У1. Устанавливать корректную последовательность операций при | | |

| | | |
|--|---|--|
| выполнении тестирования ПО; | Практических и лабораторных задания по теме | |
| У2. Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; | | |
| У3. Читать техническую документацию на ПО в объеме, необходимом для выполнения задания; | | |
| У4. Составлять сценарии поведения пользователей ПО; | | |
| У5. Устанавливать программное обеспечение, необходимое для функционирования ИС; | | |
| У6. Находить и использовать информацию, необходимую для восстановления тестов после сбоя; | | |
| У7. Использовать системы автоматизированного тестирования ПО; | | |
| У8. Взаимодействовать с членами команды разработчиков ПО; | | |
| З1. Основную терминологию по тестированию ПО; | | |
| З2. Язык, на котором написана техническая документация тестируемого ПО на уровне, достаточном для чтения технической документации; | | |
| З3. Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; | | |
| З4. Читать техническую документацию на ПО в объеме, необходимом для выполнения задания; | | |
| З5. Составлять сценарии поведения пользователей ПО | | |

4. Контрольные задания для оценки результатов освоения учебной дисциплины

4.1. Контрольные задания для текущего контроля

1. Тестовые задания

1. (Один ответ) Какой из процессов в цикле разработки ПО направлен на ПРЕДУПРЕЖДЕНИЕ дефектов, а не на их поиск?

- а) Тестирование
- б) Статический анализ кода (Code Review, линтеры)

- в) Регрессионное тестирование
- г) Нагрузочное тестирование

2. (Несколько ответов) Какие из следующих утверждений о качестве ПО являются верными согласно стандарту ISO 25010? (Выберите 2-3 варианта)

- а) Качество ПО — это только отсутствие багов.
- б) Качество ПО включает функциональную пригодность, производительность, безопасность и удобство использования.
- в) Качество использования зависит от характеристик продукта, пользователей и контекста.
- г) Главный критерий качества — скорость написания кода.

3. (Один ответ) Что означает аббревиатура «SQA» (Software Quality Assurance)?

- а) Система быстрых ответов (System Quick Answers)
- б) Обеспечение качества программного обеспечения
- в) Стандартные вопросы и ответы
- г) Специальная оценка качества

4. (Свободный ответ) В чем ключевая разница между Quality Assurance (QA) и Quality Control (QC)? Дайте краткое определение каждому.

5. (Один ответ) Практика, при которой тестировщик участвует в обсуждении требований на ранних этапах для выявления противоречий и неоднозначностей, называется:

- а) Регрессионное тестирование
- б) Приемочное тестирование
- в) Смещение тестирования влево (Shift-Left Testing)
- г) Альфа-тестирование

6. (Несколько ответов) Какие из перечисленных факторов влияют на стоимость исправления дефекта на разных этапах жизненного цикла? (Выберите 2-3 варианта)

- а) Стоимость исправления дефекта, найденного на этапе анализа требований, обычно ниже, чем дефекта, найденного в продакшене.
- б) Исправление дефекта после релиза продукта клиентам является самым дорогим.
- в) Стоимость исправления дефекта всегда постоянна.
- г) На этапе кодирования найти и исправить баг дешевле, чем на этапе тестирования.

7. (Один ответ) Какой техникой тест-дизайна вы воспользуетесь для проверки поля, принимающего значения от 1 до 100?

- а) Таблица принятия решений
- б) Тестирование состояний и переходов
- в) Анализ граничных значений (Boundary Value Analysis)
- г) Парное тестирование (Pairwise Testing)

8. (Несколько ответов) Для чего используется техника «Эквивалентное разделение» (Equivalence Partitioning)? (Выберите 2-3 варианта)

- а) Для сокращения количества тестовых случаев за счет группировки входных данных, которые, как ожидается, обрабатываются программой одинаково.
- б) Для проверки всех возможных комбинаций входных параметров.
- в) Для определения представительных значений из каждого класса эквивалентности для тестирования.
- г) Для тестирования исключительно пользовательского интерфейса.

9. (Один ответ) Метод тест-дизайна, который позволяет эффективно покрыть множество комбинаций входных параметров, проверяя все возможные пары значений, — это:

- а) Исчерпывающее тестирование
- б) Парное тестирование (Pairwise)
- в) Тестирование на основе сценариев использования
- г) Тестирование методом «черного ящика»

10. (Свободный ответ) Опишите, в какой ситуации вы бы применили «Таблицу принятия решений» (Decision Table) для создания тест-кейсов. Приведите краткий пример.

11. (Один ответ) Тестирование, при котором тестировщик знает внутреннюю структуру и код приложения, называется:

- а) Тестирование «черного ящика»
- б) Тестирование «белого ящика»
- в) Тестирование «серого ящика»
- г) Интуитивное тестирование

12. (Один ответ) Какой критерий покрытия «белого ящика» требует, чтобы каждое условие в решении принимало значения true и false хотя бы один раз?

- а) Покрытие операторов
- б) Покрытие решений (ветвей)
- в) Покрытие условий
- г) Покрытие комбинаций условий

13. (Один ответ) Как называется документ, который содержит пошаговые инструкции для проверки определенной функциональности, ожидаемый и фактический результат?

- а) План тестирования (Test Plan)
- б) Тест-кейс (Test Case)
- в) Чек-лист (Checklist)
- г) Баг-репорт (Bug Report)

14. (Несколько ответов) Какая информация является ОБЯЗАТЕЛЬНОЙ для корректного баг-репорта? (Выберите 2-3 варианта)

- а) Заголовок (Title), кратко описывающий проблему.
- б) Детальные шаги для воспроизведения (Steps to Reproduce).
- в) Ожидаемый и фактический результат.
- г) Предложения по дизайну нового функционала.
- д) Информация об окружении (ОС, браузер, версия приложения).

15. (Один ответ) Уровень серьезности дефекта (Severity) указывает на:

- а) Насколько срочно нужно исправить дефект (приоритет).
- б) Степень воздействия дефекта на работу системы (критичность).
- в) Сложность исправления дефекта для разработчика.
- г) Имя тестировщика, нашедшего дефект.

16. (Свободный ответ) Что такое «ложноположительный» (False Positive) результат в тестировании? Приведите пример из области тестирования безопасности или производительности.

17. (Один ответ) Полный набор тестов, необходимый для проверки того, что исправление одного дефекта не привело к появлению новых в ранее работоспособном функционале, — это:

- а) Набор для дымового тестирования
- б) Набор для регрессионного тестирования
- в) Набор для интеграционного тестирования
- г) Набор для исследовательского тестирования

18. (Несколько ответов) Какие из перечисленных артефактов относятся к тестовой документации? (Выберите 2-3 варианта)

- а) Стратегия тестирования (Test Strategy)
- б) Пользовательские истории (User Stories)
- в) Матрица трассируемости требований (Traceability Matrix)
- г) Техническое задание (SRS)
- д) Отчет о выполнении тестирования (Test Summary Report)

19. (Один ответ) Краткий цикл тестирования, выполняемый для проверки основных функций системы после сборки, чтобы убедиться, что она «не дымит», называется:

- а) Приемочное тестирование
- б) Дымовое тестирование (Smoke Testing)
- в) Нагрузочное тестирование
- г) Санитарное тестирование (Sanity Testing)

20. (Один ответ) Приоритет дефекта (Priority) определяет:

- а) Влияние дефекта на конечного пользователя.
- б) Очередность исправления дефекта.
- в) Модуль системы, в котором обнаружен дефект.
- г) Версию ПО, в которой дефект будет исправлен.

Ответы и критерии оценки

1. **б) Статический анализ кода** (*Предупредительная мера, в отличие от тестирования, которое является проверочной*).
2. **б, в** (*ISO 25010 определяет модель качества как набор характеристик, включая указанные; качество использования контекстуально. Отсутствие багов — часть надежности, а не всё качество. Скорость кодирования не является характеристикой качества*).
3. **б) Обеспечение качества программного обеспечения.**
4. **Пример ответа (2 балла): Quality Assurance (QA, Обеспечение качества)** — это **процессо-ориентированная** деятельность, направленная на предотвращение дефектов путем улучшения процессов разработки. **Quality Control (QC, Контроль качества)** — это **продукто-ориентированная** деятельность, направленная на выявление дефектов в готовом продукте (тестирование, проверки). QA — как *построить правильно*, QC — *правильно ли построили*.
5. **в) Смещение тестирования влево (Shift-Left Testing).**
6. **а, б** (*Стоимость исправления растет экспоненциально к поздним этапам. Утверждение "г" неверно, так как кодирование предшествует тестированию, и исправление там дешевле*).
7. **в) Анализ граничных значений** (*Классическая задача для BVA: значения 0,1,2,99,100,101*).

8. **а, в** (*EP* используется для сокращения тест-кейсов и выбора представительных значений. Не для всех комбинаций и не только для UI).
9. **б)** **Попарное тестирование (Pairwise)** (*Оптимизированный метод для проверки взаимодействий параметров*).
10. **Пример ответа (2 балла):** **Таблицу принятия решений** применяют, когда поведение системы зависит от **комбинации нескольких логических условий**. Пример: решение о выдаче кредита зависит от условий: «Возраст > 21», «Стаж > 1 года», «Кредитная история положительная». Таблица переберет все комбинации этих условий (True/False) и определит ожидаемый результат (Выдать/Отказать) для каждого набора.
11. **б)** **Тестирование «белого ящика»** (*Знание внутренней структуры*).
12. **в)** **Покрывание условий** (*Condition Coverage* требует, чтобы каждое элементарное условие принимало оба значения).
13. **б)** **Тест-кейс (Test Case)**.
14. **а, б, в, д** (*Краткий заголовок, шаги воспроизведения, сравнение ожидаемого/фактического и окружение — обязательны. Предложения по новому функционалу не являются частью баг-репорта*).
15. **б)** **Степень воздействия дефекта на работу системы (критичность)**. (*Severity = Влияние на систему. Priority = Очередность исправления*).
16. **Пример ответа (2 балла):** **Ложноположительный результат** — это когда инструмент тестирования **ошибочно сигнализирует о наличии проблемы, которой на самом деле нет**. Пример 1 (Безопасность): Сканер уязвимостей сообщает об XSS в поле ввода, но на самом деле данные корректно экранируются. Пример 2 (Производительность): Тест нагрузки показывает таймаут, вызванный проблемами сети, а не реальной перегрузкой приложения.
17. **б)** **Набор для регрессионного тестирования**.
18. **а, в, д** (*Стратегия, матрица трассируемости и отчет — часть тестовой документации. User Stories и SRS — документы требований*).
19. **б)** **Дымовое тестирование (Smoke Testing)** (*Базовая проверка стабильности сборки*).
20. **б)** **Очередность исправления дефекта**. (*Priority = когда исправлять. Severity = насколько это серьезно*).

Критерии оценки (рекомендация):

Вопросы 1, 3, 5, 7, 9, 11, 12, 13, 15, 17, 19, 20 (12 вопросов): По 1 баллу за правильный ответ.

Вопросы 2, 6, 8, 14, 18 (5 вопросов): По 2 балла за полностью правильный выбор. Частично правильный — 1 балл.

Вопросы 4, 10, 16 (3 вопроса): По 3 балла за полный, точный ответ с примером. 2 балла — правильный, но неполный ответ. 1 балл — поверхностное понимание.

Максимальный балл: $12 \cdot 1 + 5 \cdot 2 + 3 \cdot 3 = 12 + 10 + 9 = 31$ балл.

Шкала перевода в 5-балльную систему:

28-31 балл (90-100%) = 5 (Отлично)

23-27 баллов (74-89%) = 4 (Хорошо)

17-22 балла (55-73%) = 3 (Удовлетворительно)

0-16 баллов (0-54%) = 2 (Неудовлетворительно)

2. Практические и лабораторные задания

Блок 1: Анализ и планирование

Задание 1.1: Анализ требований технического задания на непротиворечивость

Цель: Научиться выявлять дефекты в требованиях до начала разработки.

Задача: Проанализируйте предоставленный фрагмент ТЗ для модуля «Корзина»:

«Пользователь может добавить до 10 единиц одного товара в корзину».

«Пользователь может добавить не более 50 различных товаров в корзину».

«Максимальное общее количество всех товаров в корзине — 100».

«При добавлении товара, если его уже 10 штук в корзине, система должна предложить купить оптом со скидкой».

1. Выявите **логические противоречия, неоднозначности и непроверяемые требования**.
2. Составьте список из **5 уточняющих вопросов** к аналитику/продукт-менеджеру.
3. Предложите исправленную, однозначную формулировку для одного из проблемных требований.

Задание 1.2: Определение целей тестирования для заданных уровней

Цель: Четко формулировать фокус тестирования на каждом уровне.

Задача: Для модуля «Корзина» сформулируйте **конкретные цели тестирования** для:

1. **Модульного (Unit) тестирования:** (Пример: «Проверить корректность расчета итоговой суммы с учетом скидок и налогов в изоляции от БД и UI»).
2. **Интеграционного тестирования:** (Пример: «Проверить корректность взаимодействия модуля корзины с сервисом товаров (Product Service) и сервисом скидок (Discount Service)»).
3. **Системного (E2E) тестирования:** (Пример: «Проверить полный сценарий покупки от выбора товара до оформления заказа через веб-интерфейс»).

Блок 2: Проектирование и подготовка тестовых артефактов

Задание 2.1: Подготовка тестового пакета и задания на тестирование модуля

Цель: Создать структурированный набор документов для начала тестирования.

Задача:

1. Создайте в Git-репозитории (например, GitHub) папку qa/shopping-cart/.
2. Внутри папки создайте **README.md**, где опишите:
 - Краткий обзор тестируемого модуля.
 - Ссылки на ТЗ и документацию.
 - Инструкции по запуску тестового стенда.

- Цели тестирования (из Задания 1.2).

3. Создайте файл `test-assignment.md` с четким **техническим заданием на тестирование** (Test Assignment), включающим: объем тестирования, критерии начала/окончания, необходимые ресурсы, риски.

Задание 2.2: Проектирование тест-кейсов для интеграционного тестирования

Цель: Разработать тесты для проверки взаимодействия компонентов.

Задача: Спроектируйте тест-кейсы для проверки интеграции «Модуль Корзины — Сервис Цен».

1. Используйте технику «**Таблица принятия решений**». Условия: наличие товара на складе, актуальность цены, наличие персональной скидки. Определите тестовые сценарии.
2. Оформите **3 тест-кейса** в формате Gherkin (Given-When-Then) или в табличном виде с полями: ID, Название, Предусловия, Шаги, Ожидаемый результат.
3. Сохраните тест-кейсы в папке репозитория как `integration-test-cases.md` или в формате `.feature` (Cucumber).

Задание 2.3: Подготовка E2E тестового сценария для веб-приложения

Цель: Создать детальный сценарий проверки полного пользовательского пути.

Задача:

1. Разработайте **E2E сценарий**: «Авторизованный пользователь добавляет два разных товара в корзину, применяет промокод, переходит к оформлению».
2. **Вручную** выполните этот сценарий на тестовом стенде, фиксируя каждый шаг.
3. Создайте **автоматизированный скрипт** на базе Selenium WebDriver (Python/Java) или Cypress (JavaScript), который выполняет этот сценарий. Скрипт должен проверять ключевые состояния (товар добавлен, сумма изменилась, промокод применен).
4. Разместите код скрипта в папке `e2e/` вашего репозитория.

Блок 3: Разработка и оптимизация тестовых наборов

Задание 3.1: Разработка тестового набора методом «белого ящика»

Цель: Написать юнит-тесты с высокой степенью покрытия кода.

Задача: Вам предоставлен код функции `calculateTotal(cartItems, discountCode)` на выбранном языке (Python/Java/JS).

1. Проанализируйте код, постройте **граф потока управления** (Control Flow Graph).
2. Используя критерий «**Покрытие условий**» (**Condition Coverage**), определите минимальный набор тестовых данных.

3. Напишите **юнит-тесты** (используя JUnit, pytest, Jest), покрывающие все ветвления логики. Используйте моки (mock) для зависимостей (например, для вызова сервиса проверки промокода).

4. Запустите инструмент анализа покрытия кода (jacoco, pytest-cov, istanbul) и получите отчет. Цель — **не менее 80% покрытия ветвей** для данной функции.

Задание 3.2: Разработка тестового набора с имитацией действий пользователя

Цель: Создать автоматизированные UI-тесты, отражающие реальное поведение.

Задача:

1. Разработайте **набор из 5 UI-тестов** для модуля корзины с помощью инструмента записи/воспроизведения (например, Selenium IDE, Katalon Recorder) или через написание кода.

2. Тесты должны покрывать: добавление/удаление товара, изменение количества, очистку корзины, применение неверного промокода.

3. Настройте **параметризацию** одного из тестов (например, для проверки добавления разных товаров из CSV-файла).

4. Интегрируйте запуск этих тестов в **локальный пайплайн** (например, с помощью Makefile или npm scripts).

Задание 3.3: Генерация тестовых данных для интеграции с БД

Цель: Создать реалистичные и разнообразные данные для тестирования.

Задача:

1. Создайте **SQL-скрипт** или **файл фикстур** (например, для pytest или Rails), который заполняет тестовую БД данными:

- Товары разных категорий, с разными ценами и остатками.
- Промокоды (действующие, истекшие, одноразовые).
- Тестовых пользователей.

2. Используйте инструмент **Faker** (или аналог) для генерации реалистичных имен, email, адресов.

3. Реализуйте **шаблон «Object Mother»** или **«Test Data Builder»** в коде ваших интеграционных тестов для удобного создания сложных объектов корзины с товарами.

Задание 3.4: Оптимизация тестовых наборов и оценка тестов

Цель: Повысить эффективность тестового набора.

Задача:

1. **Анализ дублирования:** Проведите ревизию всех созданных тест-кейсов (блоки 2 и 3). Выявите и устранили **функциональные дубликаты**. Создайте единый реестр тестов.

2. **Приоритизация:** Присвойте каждому тест-кейсу **приоритет (P0-P2)**, где P0 — критический функционал (например, «Расчет итоговой суммы»), P2 — второстепенный (например, «Визуальное отображение иконки корзины»).

3. **Оценка эффективности:** Предложите **3 метрики** для оценки качества вашего тестового набора (например, «% покрытия критического функционала», «Среднее время выполнения smoke-набора», «Соотношение автоматизированных/ручных тестов»). Рассчитайте их для вашего проекта.

4. **Создайте «Золотой набор» (Sanity Suite):** Отберите **10 самых важных тестов** (ручных или авто), которые должны проходить всегда для подтверждения базовой работоспособности модуля после любой сборки.

Методические рекомендации:

Стек технологий:

Управление тестами: **Git, Markdown, TestRail** (или аналоги).

Автоматизация: **Selenium WebDriver, Cypress, Playwright** (для UI); **JUnit, pytest, Jest** (для юнитов); **RestAssured, Postman** (для API, если потребуется).

Данные: **Faker, SQL, fixtures**.

Формат сдачи: Git-репозиторий с четкой структурой папок:

text

```
/qa-shopping-cart-project
├── README.md
├── docs/
│   ├── test-assignment.md
│   └── requirements-analysis.md
├── test-cases/
│   ├── integration/
│   ├── e2e/
│   └── unit/
├── automation/
│   ├── ui-tests/ (Selenium/Cypress)
│   ├── api-tests/
│   └── unit-tests/
├── test-data/ (SQL, fixtures, generators)
└── reports/ (покрытие, результаты прогонов)
```

Критерии оценки:

Глубина анализа (25%): Качество анализа ТЗ, постановка вопросов, определение целей.

Качество артефактов (30%): Полнота, четкость, однозначность тест-кейсов и документации.

Техническая реализация (30%): Работоспособность автоматизированных тестов, корректность использования инструментов, покрытие кода.

Оптимизация и структура (15%): Логическая организация репозитория, устранение дублей, приоритизация, предложенные метрики.

Вопросы к экзамену

1. Дайте определение качества программного обеспечения (ПО) согласно стандарту ISO 25010. Назовите основные характеристики качества из данного стандарта.
2. В чем разница между Quality Assurance (QA, обеспечение качества) и Quality Control (QC, контроль качества)? Опишите цели и методы каждого подхода.
3. Что такое Verification (верификация) и Validation (валидация)? Проиллюстрируйте разницу на примере разработки программного модуля.
4. Опишите модель зрелости процессов разработки ПО CMMI (Capability Maturity Model Integration). Какие уровни она включает и что они характеризуют?
5. Назовите основные роли в процессе обеспечения качества (QA-инженер, тест-аналитик, тест-менеджер). Кратко охарактеризуйте их обязанности.
6. Что такое тест-план? Какие ключевые разделы он должен содержать (согласно стандарту IEEE 829)?
7. Объясните, как принципы гибкой методологии разработки (Agile/Scrum) влияют на процессы обеспечения качества. Что такое Definition of Done (DoD)?
8. Опишите концепцию "Сдвиг влево" (Shift Left) в тестировании. Какие практики и инструменты его поддерживают?
9. Что такое тестовая стратегия и чем она отличается от тест-плана?
10. Какие существуют уровни тестирования? Дайте краткую характеристику каждому: модульное (Unit), интеграционное (Integration), системное (System), приемочное (Acceptance).
11. Дайте определение тест-дизайну. Какова его основная цель в процессе тестирования?
12. Объясните разницу между техниками тест-дизайна, основанными на спецификации (черный ящик), и основанными на структуре кода (белый ящик).
13. Что такое классы эквивалентности (Equivalence Partitioning)? Приведите практический пример разбиения входных данных на классы.
14. Опишите технику анализа граничных значений (Boundary Value Analysis). Почему ошибки часто возникают на границах?
15. Что такое таблица принятия решений (Decision Table)? Опишите этапы ее создания и применения для тест-дизайна.
16. Для чего используется техника тестирования переходов между состояниями (State Transition Testing)? Приведите пример системы, поведение которой удобно моделировать с помощью диаграммы состояний.
17. Что такое Use Case Testing (тестирование на основе сценариев использования)? Как use case-диаграммы и сценарии помогают в создании тестов?
18. Объясните технику тест-дизайна "Попарное тестирование" (Pairwise Testing). Какую проблему производительности тестов она решает?
19. Что такое тестовые сценарии (Test Scenarios) и тестовые случаи (Test Cases)? Как они соотносятся друг с другом?
20. Что такое положительное (positive) и отрицательное (negative) тестирование? Приведите примеры тестовых данных для каждого типа.
21. Дайте определение дефекту (багу) в программном обеспечении. Каков жизненный цикл дефекта (Defect/Bug Life Cycle)?
22. Как правильно оформить отчет об ошибке (Bug Report)? Перечислите обязательные и желательные поля качественного баг-репорта.
23. Какие атрибуты дефекта являются ключевыми для его классификации и приоритизации? (Например: Severity (критичность) vs Priority (приоритет)).

24. Опишите общепринятые уровни критичности (Severity) дефекта: Critical, Major, Minor, Trivial. Приведите примеры для каждого.
25. Что такое "не воспроизводимый" дефект? Какие действия должен предпринять тестировщик при его обнаружении?
26. Дайте определение набора тестов (Test Suite). По каким критериям тестовые случаи объединяются в наборы?
27. Что такое регрессионное тестирование (Regression Testing)? Когда оно проводится и как формируется регрессионный тестовый набор?
28. Что такое дымовое тестирование (Smoke Testing) и для чего оно используется? Как формируется набор для smoke-тестов?
29. Объясните разницу между Sanity Testing и Smoke Testing. В каких ситуациях применяется каждый из этих видов?
30. Что такое тестовые данные? Какие подходы к подготовке тестовых данных вы знаете (ручное создание, автоматическая генерация, использование "грязных" данных из прод)?
31. Как осуществляется управление тестовыми данными, особенно при работе с конфиденциальной информацией (например, персональными данными)?
32. Что такое "ложноположительный" (False Positive) и "ложноотрицательный" (False Negative) результат тестирования? Чем они опасны?
33. Что такое дефект-сессия (Bug Bash, Defect Triage Meeting)? Какова цель ее проведения и кто в ней участвует?
34. Как метрики, связанные с дефектами (например, плотность дефектов, время жизни дефекта), используются для оценки качества продукта и процесса?
35. Что такое корневая причина (Root Cause) дефекта? Для чего проводится анализ корневых причин (RCA)?
36. Какие существуют типы дефектов по месту их возникновения? (Дефекты требований, архитектурные, кодовые, дефекты сборки/развертывания).
37. Что такое тестовое покрытие (Test Coverage)? Какие его основные типы существуют (покрытие требований, покрытие кода, покрытие функций)?
38. Как связаны между собой тестовые артефакты: требования -> тест-кейсы -> дефекты? Почему важна прослеживаемость (traceability)?
39. Что такое "золотой" тестовый набор (Golden Test Suite) и как он используется при регрессионном тестировании?
40. Опишите, как процесс управления тестовыми наборами и дефектами поддерживается специализированными инструментами (например, Jira, TestRail, qTest).

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В.Н. Татищева»
(Астраханский государственный университет им. В.Н. Татищева)

Филиал АГУ им. В.Н. Татищева в г. Знаменске Астраханской области

СОГЛАСОВАНО
Руководитель ОПОП
Бориско С.Н.
«13» ноября 2025 г.

УТВЕРЖДАЮ
Председатель ЦК (МО)
Фисенко Т.Ю.
протокол заседания ЦК (МО) №3
от «13» ноября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по междисциплинарному курсу

ПМ.02.МДК.02 АВТОМАТИЗАЦИЯ ПРОЦЕССОВ ТЕСТИРОВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

| | |
|------------------------------|---|
| Составитель | Бориско С.Н., к.т.н., доцент, завкафедрой ЗнМИ; Мустафаев Н.Г., к.т.н., доцент кафедры ЗнМИ; Тимошкин А.А., к.т.н., доцент кафедры ЗнМИ; Устинов А.С., к.т.н., доцент кафедры ЗнМИ; Каштанов Д.Ю., ассистент кафедры ЗнМИ |
| Согласовано с работодателями | Литвинов С.П., к.т.н., заместитель командира войсковой части 15644 по научно-исследовательской и испытательной работе; Кирьянов М.Н., ведущий инженер ПАО «Ростелеком» |
| Наименование специальности | 09.02.12 Техническая эксплуатация и сопровождение информационных систем |
| Квалификация выпускника | Специалист по технической эксплуатации и сопровождению информационных систем |
| Форма обучения | очная |
| Год приема (курс) | 2026 (3 курс) |

Знаменск, 2025 г.

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ

**2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА,
ПОДЛЕЖАЩИЕ ПРОВЕРКЕ**

**3. ФОРМЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ ЭЛЕМЕНТОВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

**4. КОНТРОЛЬНЫЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

1. Общие положения

Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки результатов освоения обучающимися междисциплинарного курса ПМ.02.МДК.02 Автоматизация процессов тестирования программного обеспечения.

ФОС включают контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, разработанные в соответствии с требованиями ФГОС СПО и содержанием рабочей программы междисциплинарного курса.

2. Результаты освоения профессионального модуля, подлежащие проверке

| Код компетенции | Планируемые результаты освоения учебного предмета (междисциплинарного курса) | | |
|-----------------|--|--|--|
| | Практический опыт | Умения | Знания |
| ПК 2.1 | <ul style="list-style-type: none">– Изучения необходимых для проведения тестирования ПО действий, перечисленных в задании на тестирование– Подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости)– Оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения– Настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенцииФормирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами | <ul style="list-style-type: none">- Устанавливать корректную последовательность операций при выполнении тестирования ПО- Выявлять недостающую информацию для выполнения тестирования ПО в заданном объеме- Устанавливать операционные системы- Выполнять базовую настройку операционных систем- Подготавливать необходимые средства и ресурсы для выполнения задания по тестированию ПО- Составлять отчет о результатах подготовки к выполнению тестирования ПО | <ul style="list-style-type: none">- Основную терминологию по тестированию ПО- Язык, на котором написана техническая документация тестируемого ПО на уровне, достаточном для чтения технической документации- Основные термины и сокращения, используемые в технической документации и принятые в организации- Процедуры обеспечения безопасности при выполнении тестирования ПО- Область применения инструментальных средств для выполнения тестирования ПО- Особенности основных операционных систем- Требования по |

| | | | |
|--------|--|---|---|
| | | | <p>обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирус</p> |
| ПК 2.2 | <ul style="list-style-type: none"> – Проверки компонентов инструментария и тестируемого ПО на корректное начальное состояние для начала тестирования – Выполнения тестовых процедур на тестовых данных – Сравнения фактического и ожидаемого результатов выполнения тестовых процедур <p>Формирования и представления отчетности о выполнении процесса тестирования ПО в соответствии с установленными регламентами</p> | <ul style="list-style-type: none"> – Нормативно-технические материалы по вопросам испытания и тестирования ПО – Основные термины и сокращения, используемые в технической документации и принятые в организации – Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования ПО соответствующего типа – Основы теории алгоритмов и дискретной математики в объеме полученного профессионального образования Синтаксис языка программирования тестируемого ПО, особенности программирования на этом языке, стандартные библиотеки языка | <ul style="list-style-type: none"> – Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования – Использовать системы контроля дефектов ПО – Составлять отчет о выполнении тестирования ПО <p>Работать в команде со специалистами по тестированию ПО и разработчиками</p> |

| | | | |
|--------|---|--|---|
| | | программирования | |
| ПК 2.3 | <ul style="list-style-type: none"> – Проверки полноты эксплуатационной и технической документации на ПО – Выявления недостатков эксплуатационной и технической документации на ПО и ее несоответствия внутренним стандартам качества организации – Проверки эксплуатационной и технической документации на ПО на соответствие требованиям заказчика – Выполнения действий по указаниям в эксплуатационной и технической документации на ПО – Проверки соответствия действительных и указанных в эксплуатационной и технической документации на ПО результатов Выявления несовпадений действительных и указанных в эксплуатационной и технической документации результатов регистрация найденных дефектов ПО в системе контроля дефектов | <ul style="list-style-type: none"> – Нормативно-технические материалы по вопросам испытания и тестирования ПО – Основные понятия о качестве ПО – Виды технической документации – Требования по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирусной защиты Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО | <ul style="list-style-type: none"> – Основы современных СУБД – Теорию баз данных – Основы программирования – Современные объектно-ориентированные языки программирования – Современные структурные языки программирования – Языки современных бизнес-приложений – Современные методики тестирования разрабатываемых ИС: инструменты и методы модульного тестирования – Методы верификации программного обеспечения – Источники информации, необходимой для профессиональной деятельности в рамках технической поддержки процессов создания (модификации) и сопровождения ИС Лучшие практики создания (модификации) и сопровождения ИС в экономике |
| ПК 2.4 | <ul style="list-style-type: none"> – Выполнения начальных настроек для проведения тестирования ПО – Выполнения необходимых видов тестирования ПО в | <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники | <ul style="list-style-type: none"> – Кодировать на языках программирования ИС – Тестировать результаты |

| | | | |
|--------|---|--|---|
| | <p>соответствии с планом тестирования</p> <ul style="list-style-type: none"> – Проведения автоматизированного тестирования ПО при необходимости – Составления статистики выполнения тестов – Проведения анализа полученных результатов тестирования ПО по разработанным тестовым случаям на соответствие ожидаемым результатам – Оптимизации тестовых наборов – Составления новых тестовых случаев и повторение тестирования при необходимости <p>Формирования и представления отчетности о проведенном тестировании ПО в соответствии с установленными регламентами</p> | <p>тестирования ПО, базирующиеся на спецификации</p> <ul style="list-style-type: none"> – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения – Стандарты оформления кода для используемых языков программирования – Основные термины и сокращения, используемые в технической документации и принятые в организации – Основы алгоритмизации и программирования <p>Жизненный цикл программного продукта</p> | <p>разработки ИС</p> <p>Работать с записями по качеству (в том числе с корректирующими действиями, предупреждающими действиями, запросами на исправление несоответствий) при выполнении технической поддержки процессов создания (модификации) и сопровождения ИС</p> |
| ПК 2.5 | <ul style="list-style-type: none"> – Определения причины сбоя системы совместно с разработчиками – Устранения причины сбоя системы, если она находится в компетенции специалиста, либо подготовка отчета руководителю и группе | <ul style="list-style-type: none"> – Находить и использовать информацию, необходимую для восстановления тестов после сбоя – Взаимодействовать с командой | <ul style="list-style-type: none"> – Архитектуру тестируемой системы – Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом |

| | | | |
|---------|---|---|---|
| | <p>разработчиков</p> <ul style="list-style-type: none"> – Выполнения настройки для повторного тестирования после сбоя – Восстановления/изменения автоматизированных тестов после сбоя при необходимости в соответствии с планом/регламентом восстановления – Проведения повторного тестирования ПО Формирования и представления отчетности о восстановлении работоспособности ПО в соответствии с установленными регламентами | <p>разработчиков при восстановлении системы после сбоя</p> <ul style="list-style-type: none"> – Применять языки программирования для написания программного кода – Использовать системы автоматизированного тестирования ПО Составлять отчет о восстановлении работоспособности ПО | <p>для тестирования разработанного ПО</p> <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения – Принципы регрессионного тестирования ПО – Алгоритмы решения типовых задач, области и способы их применения Основные термины и сокращения, используемые в технической документации и принятые в организации |
| ПК 2.6. | <ul style="list-style-type: none"> – Получения обновленной версии ПО – Определения масштабов изменений для выявления | <ul style="list-style-type: none"> – Жизненный цикл ПО, жизненный цикл дефекта ПО – Принципы | <ul style="list-style-type: none"> – Основы системного администрирования – Основы |

| | | | |
|--|---|--|--|
| | <p>необходимости проведения регрессионных тестов</p> <ul style="list-style-type: none"> – Определения оптимального перечня тестов для повторного тестирования ПО Выполнения тестовых сценариев, выявивших дефекты ПО, для подтверждения успешности их выполнения после исправления ПО | <p>регрессионного тестирования ПО</p> <ul style="list-style-type: none"> – Техники тестирования ПО, базирующиеся на интуиции и опыте инженера – Техники тестирования ПО, базирующиеся на спецификации – Техники тестирования ПО, ориентированные на код – Тестирование ПО, ориентированное на дефекты – Техники тестирования ПО, базирующиеся на условиях использования – Тестирование ПО, базирующееся на надежности инженерного процесса – Техники тестирования ПО, базирующиеся на природе приложения <p>Основные инструментальные средства организации работы в команде</p> | <p>администрирования баз данных</p> <ul style="list-style-type: none"> – Коммуникационное оборудование – Сетевые протоколы – Основы современных операционных систем – Основы современных СУБД – Устройство и функционирование современных ИС – Основы архитектуры мультиарендного программного обеспечения – Основы ИБ организации – Источники информации, необходимой для профессиональной деятельности в рамках технической поддержки процессов создания (модификации) и сопровождения ИС <p>Лучшие практики создания (модификации) и сопровождения ИС в экономике</p> |
|--|---|--|--|

3. Распределение оценивания результатов обучения по видам контроля

| Наименование элемента практического опыта, умений или знаний | Наименование оценочного средства текущего контроля и промежуточной аттестации | |
|--|---|--------------------------|
| | Текущий контроль | Промежуточная аттестация |
| ПО.1 Изучения необходимых для проведения тестирования ПО действий, перечисленных в задании на тестирование | Компьютерное тестирование на знание терминологии по теме; | Вопросы к экзамену |
| У1. Устанавливать корректную последовательность операций при | | |

| | | |
|--|---|--|
| выполнении тестирования ПО; | Практических и лабораторных задания по теме | |
| У2. Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; | | |
| У3. Читать техническую документацию на ПО в объеме, необходимом для выполнения задания; | | |
| У4. Составлять сценарии поведения пользователей ПО; | | |
| У5. Устанавливать программное обеспечение, необходимое для функционирования ИС; | | |
| У6. Находить и использовать информацию, необходимую для восстановления тестов после сбоя; | | |
| У7. Использовать системы автоматизированного тестирования ПО; | | |
| У8. Взаимодействовать с членами команды разработчиков ПО; | | |
| З1. Основную терминологию по тестированию ПО; | | |
| З2. Язык, на котором написана техническая документация тестируемого ПО на уровне, достаточном для чтения технической документации; | | |
| З3. Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; | | |
| З4. Читать техническую документацию на ПО в объеме, необходимом для выполнения задания; | | |
| З5. Составлять сценарии поведения пользователей ПО | | |

4. Контрольные задания для оценки результатов освоения учебной дисциплины

4.1. Контрольные задания для текущего контроля

1. Тестовые задания

1. (Один ответ) Какой из перечисленных объектов используется в модульном тестировании для замены реальной зависимости (например, базы данных или внешнего API), чтобы изолировать тестируемый модуль и контролировать его поведение?

- а) Заглушка (Stub)
- б) Системный тест
- в) Ручной тестировщик
- г) Продакшен-сервер

2. (Несколько ответов) Какие из следующих утверждений о принципах хороших модульных тестов (FIRST) верны? (Выберите 2-3 варианта)

- а) Fast (Быстрые): тесты должны выполняться за миллисекунды.
- б) Interesting (Интересные): тесты должны иметь сложную, запутанную логику.
- в) Repeatable (Повторяемые): результат теста не должен зависеть от внешних факторов.
- г) System-oriented (Системно-ориентированные): тесты должны проверять работу всей системы.
- д) Timely (Своевременные): тесты пишутся незадолго до написания кода (TDD).

3. (Один ответ) Как называется практика разработки, при которой сначала пишется падающий тест, затем код, который делает тест зеленым, и после этого проводится рефакторинг?

- а) Behavior-Driven Development (BDD)
- б) Test-Driven Development (TDD)
- в) Acceptance Test-Driven Development (ATDD)
- г) Экстремальное программирование (XP)

4. (Свободный ответ) Объясните, в чем разница между Mock-объектом и Stub-объектом в контексте модульного тестирования. Приведите простой пример использования каждого.

5. (Один ответ) Тестирование, направленное на проверку соответствия фактического поведения системы ее пользовательской документации (руководству, справке), называется:

- а) Тестирование удобства использования
- б) Тестирование документации
- в) Тестирование производительности
- г) Регрессионное тестирование

6. (Несколько ответов) Какие из перечисленных артефактов могут и должны быть протестированы? (Выберите 2-3 варианта)

- а) Техническое задание (SRS) на непротиворечивость и полноту
- б) Руководство пользователя (User Manual)
- в) Внутренние комментарии в коде
- г) README-файл в репозитории проекта
- д) Личные заметки разработчика

7. (Свободный ответ) Какие типичные дефекты можно обнаружить при тестировании руководства пользователя? Назовите не менее трех.

8. (Один ответ) Какой вид тестирования проверяет взаимодействие между двумя или более модулями/сервисами после их объединения?

- а) Модульное тестирование
- б) Интеграционное тестирование
- в) Приемочное тестирование
- г) Нагрузочное тестирование

9. (Один ответ) Какой подход к интеграционному тестированию предполагает последовательное добавление и тестирование модулей сверху вниз, начиная с главного модуля, используя заглушки для нижних уровней?

- а) Нисходящее тестирование (Top-Down)
- б) Восходящее тестирование (Bottom-Up)
- в) Большой взрыв (Big Bang)
- г) Сэндвич (Sandwich)

10. (Несколько ответов) Какие из следующих задач решаются с помощью системного тестирования (System Testing)? (Выберите 2-3 варианта)

- а) Проверка корректности алгоритма внутри одной функции.
- б) Проверка соответствия системы всем требованиям из технического задания.
- в) Проверка взаимодействия системы с внешним окружением (ОС, БД, сеть).
- г) Проверка удобства интерфейса для конечного пользователя.
- д) Проверка покрытия кода операторами.

11. (Свободный ответ) Что такое «сквозное тестирование» (End-to-End, E2E) и чем оно отличается от классического системного тестирования? Приведите пример E2E-сценария для интернет-магазина.

12. (Один ответ) Какой тип нефункционального тестирования оценивает способность системы выдерживать пиковые нагрузки без потери функциональности?

- а) Тестирование безопасности
- б) Тестирование надежности (Reliability Testing)
- в) Нагрузочное тестирование (Load Testing)
- г) Тестирование удобства использования

13. (Один ответ) Метрика, показывающая среднее время работы системы между сбоями, называется:

- а) MTTR (Mean Time To Repair)
- б) RPO (Recovery Point Objective)
- в) MTBF (Mean Time Between Failures)
- г) Uptime (Время доступности)

14. (Несколько ответов) Какие из перечисленных практик DevOps способствуют повышению надежности системы в эксплуатации? (Выберите 2-3 варианта)

- а) Развертывание в продакшен раз в полгода (Big Bang релиз).
- б) Внедрение мониторинга и алертинга (Prometheus, Grafana).
- в) Использование Feature Toggles (флагов функций) для безопасного включения нового кода.
- г) Полное отсутствие автоматических тестов.
- д) Автоматическое масштабирование (Auto-scaling) в облачной инфраструктуре.

15. (Один ответ) Тестирование, при котором специалист моделирует действия злоумышленника с целью поиска уязвимостей (например, инъекции, XSS, несанкционированный доступ), — это:

- а) Юзабилити-тестирование
- б) Тестирование производительности
- в) Пентест (Penetration Testing)
- г) Модульное тестирование

16. (Один ответ) Практика регулярного применения обновлений безопасности для операционных систем и программных компонентов — это:

- а) Резервное копирование
- б) Патч-менеджмент
- в) Шифрование данных
- г) Аудит доступа

17. (Свободный ответ) Что такое «непрерывное тестирование» (Continuous Testing) в контексте DevOps и как оно связано с надежностью системы?

18. (Один ответ) Какой из перечисленных инструментов является фреймворком для автоматизации веб-интерфейса (UI-тестирования)?

- а) Jenkins
- б) Selenium WebDriver
- в) JUnit
- г) Postman

19. (Несколько ответов) Какие из перечисленных элементов являются типичными компонентами пайплайна непрерывной интеграции (CI)? (Выберите 2-3 варианта)

- а) Сборка проекта (Build)
- б) Запуск модульных и интеграционных тестов
- в) Ручное ревью всех строк кода
- г) Статический анализ кода (SAST)
- д) Развертывание непосредственно в продакшен без проверок

20. (Один ответ) Формат сценариев Given-When-Then, понятный как техническим, так и нетехническим специалистам, используется в методологии:

- а) Test-Driven Development (TDD)
- б) Behavior-Driven Development (BDD)
- в) Водопадная модель (Waterfall)
- г) Экстремальное программирование (XP)

Ответы и критерии оценки

1. **а) Заглушка (Stub)** (*Или Mock, но заглушка — более общий термин для замены зависимости*).

2. **а, в, д** (*Быстрые, Повторяемые, Своевременные (в TDD). Интересные — не принцип FIRST, системно-ориентированные — противоречит идее модульности*).

3. **б) Test-Driven Development (TDD)**

4. **Пример ответа (2 балла): Stub** — это объект-заглушка, который возвращает заранее заданные, жестко запрограммированные ответы на вызовы во время теста (например, всегда возвращает user_id=5). **Mock** — это объект, который проверяет, как тестируемый код взаимодействует с ним (например, проверяет, что метод save() был вызван ровно один раз с определенными аргументами). Stub используется для обеспечения данных, Mock — для проверки поведения.

5. **б) Тестирование документации**

6. **а, б, г** (*SRS, руководство пользователя и README — критически важные документы, требующие проверки. Комментарии в коде и личные заметки обычно не тестируются формально*).

7. **Пример ответа (2 балла за 3 пункта):** 1) **Фактические ошибки:** несоответствие шагов в документации реальному поведению программы. 2) **Неполнота:** отсутствие описания важных функций или сценариев ошибок. 3) **Неясность/двусмысленность:** формулировки, которые можно понять по-разному.

4) Устаревшая информация: описание функционала, который был изменен или удален.

5) Орфографические и грамматические ошибки.

8. б) Интеграционное тестирование

9. а) Нисходящее тестирование (Top-Down)

10. б, в, г (Системное тестирование проверяет соответствие системы ТЗ в целом, работу в окружении и нефункциональные требования, включая юзабилити. Проверка алгоритма — модульное, покрытие кода — метрика).

11. Пример ответа (2 балла): Сквозное тестирование (E2E) — это вид тестирования, который проверяет полный рабочий поток приложения от начала до конца в среде, максимально приближенной к реальной, с вовлечением всех интегрированных систем (UI, бэкэнд, БД, внешние сервисы). Отличие от системного: E2E фокусируется на пользовательских сценариях и бизнес-процессах, а системное — на проверке системы как единого целого против спецификаций. Пример E2E-сценария: «Пользователь ищет товар, добавляет его в корзину, проходит процесс оформления заказа с оплатой и получает confirmation email».

12. в) Нагрузочное тестирование (Load Testing) (Проверка под нагрузкой. Надежность — более широкое понятие, включающее тестирование на отказоустойчивость).

13. в) MTBF (Mean Time Between Failures) (Среднее время между отказами. MTTR — среднее время восстановления).

14. б, в, д (Мониторинг, Feature Toggles и автоскейлинг — ключевые практики надежности. Редкие релизы и отсутствие тестов ведут к снижению надежности).

15. в) Пентест (Penetration Testing)

16. б) Патч-менеджмент

17. Пример ответа (2 балла): Непрерывное тестирование — это практика выполнения автоматизированных тестов на протяжении всего конвейера доставки ПО (CI/CD) для получения немедленной обратной связи о рисках, связанных с выпуском. Оно связано с надежностью, потому что позволяет быстро обнаруживать регрессии и дефекты, не давая им попасть в продакшен, тем самым повышая стабильность работающей системы и снижая количество инцидентов.

18. б) Selenium WebDriver

19. а, б, г (Сборка, запуск автоматических тестов и статический анализ — ключевые автоматизированные этапы CI. Ручное ревью — не этап CI пайплайна, развертывание в прод без проверок — антипаттерн).

20. б) Behavior-Driven Development (BDD) (Gherkin-синтаксис — основа BDD).

Критерии оценки (рекомендация):

Вопросы 1, 3, 5, 8, 9, 12, 13, 15, 16, 18, 20 (11 вопросов): По 1 баллу за правильный ответ.

Вопросы 2, 6, 10, 14, 19 (5 вопросов): По 2 балла за полностью правильный выбор.

Частично правильный — 1 балл.

Вопросы 4, 7, 11, 17 (4 вопроса): По 3 балла за полный, точный ответ с примерами. 2 балла — правильный, но неполный ответ. 1 балл — поверхностное понимание.

Максимальный балл: $11 \cdot 1 + 5 \cdot 2 + 4 \cdot 3 = 11 + 10 + 12 = 33$ балла.

Шкала перевода в 5-балльную систему:

30-33 балла (91-100%) = 5 (Отлично)

25-29 баллов (76-90%) = 4 (Хорошо)

18-24 балла (55-75%) = 3 (Удовлетворительно)

0-17 баллов (0-54%) = 2 (Неудовлетворительно)

2. Практических и лабораторных задания

Этап 1: Модульное тестирование и инфраструктура

Задание 1.1: Разработка модульных автотестов для бэкенд-логики

Цель: Создать основу библиотеки модульных тестов.

Задача:

1. Выберите язык (Python/pytest или Java/JUnit) и фреймворк для модульного тестирования.

2. Напишите **5 модульных тестов** для ключевых функций бэкенда:

- `calculate_balance(transactions)` — расчет текущего баланса.
- `categorize_transaction(amount, description)` — автоматическая категоризация трат.
- `validate_password(password)` — проверка сложности пароля.

3. Используйте **параметризацию тестов** для проверки граничных значений.

4. Реализуйте **фикстуры** (fixtures) для переиспользуемых данных.

Задание 1.2: Оформление и размещение отчетов о тестировании

Цель: Создать структурированную библиотеку тестов и настроить отчетность.

Задача:

1. Создайте Git-репозиторий `fintrack-autotests`.

2. Организуйте структуру папок:

`text`

`/tests`

├── `unit/` (задание 1.1)

├── `integration/`

├── `api/`

├── `ui/`

├── `security/`

└── `performance/`

`/reports`

`/test_data`

`/utils`

3. Настройте генерацию **детализированных HTML-отчетов** о прохождении тестов (используйте `pytest-html`, `Allure` или `ExtentReports`).

4. Создайте `README.md` с описанием проекта, инструкцией по запуску и примерами отчетов.

Задание 1.3: Запуск автотестов и сбор статистик. Анализ отчетов

Цель: Настроить автоматизированный сбор метрик тестирования.

Задача:

1. Напишите скрипт/конфигурацию для запуска **всех тестов из консоли** одной командой.

2. Настройте сбор **ключевых метрик**: общее время выполнения, количество пройденных/упавших тестов, процент покрытия (если настроено).

3. Проведите 5 запусков тестового набора, соберите статистику.

4. **Оформите отчет по анализу статистики**: постройте график времени выполнения, выявите самые медленные/нестабильные тесты, предложите план оптимизации.

Задание 1.4: Сборка и запуск тестов в CI/CD

Цель: Интегрировать тесты в пайплайн.

Задача:

1. Настройте **GitHub Actions** или **GitLab CI/CD** конфигурацию.

2. Создайте джобу, которая при пуше в ветку `develop`:

- Устанавливает зависимости.

- Запускает модульные тесты из консоли.
 - Публикует сгенерированный отчет как артефакт сборки.
3. Настройте **уведомление о падении тестов** в Telegram/Slack.

Этап 2: Приемочное и документационное тестирование

Задание 2.1: Выполнение приемочного тестирования (UAT)

Цель: Проверить соответствие системы бизнес-требованиям.

Задача:

1. На основе пользовательских историй (например, «Как пользователь, я хочу видеть отчет по расходам за месяц, чтобы контролировать бюджет») разработайте **чек-лист из 10 пунктов** для приемочного тестирования.

2. Выполните проверки **вручную** на тестовом стенде.

3. **Оформите отчет о результатах UAT:** укажите процент пройденных проверок, найденные расхождения с требованиями, рекомендации к выпуску.

Задание 2.2: Тестирование пользовательской документации

Цель: Проверить точность и полноту документации.

Задача:

1. Получите черновик **Руководства пользователя** для модуля «Добавление транзакции».

2. Проведите тестирование методом **прослеживания (Traceability):** выполните каждый шаг из документации в системе и проверьте соответствие.

3. **Оформите отчет:** создайте таблицу с дефектами документации (несоответствие, неполнота, опечатки) и приоритетом исправления.

Этап 3: Интеграционное, UI и API тестирование

Задание 3.1: Разработка и запуск UI-тестов

Цель: Автоматизировать проверку пользовательского интерфейса.

Задача:

1. Используя **Selenium WebDriver** или **Playwright**, напишите 3 UI-теста:

- Успешная авторизация.
- Добавление новой траты.
- Просмотр отчета за текущий месяц.

2. Реализуйте **Page Object Pattern** для работы с элементами страниц.

3. Настройте **перехват скриншотов** при падении тестов.

4. **Оформите отчет о тестировании** с приложенными скриншотами и логами.

Задание 3.2: Автоматизация API-тестирования

Цель: Настроить автоматический запуск тестов REST API.

Задача:

1. Используя **Postman** или **RestAssured/requests (Python)**, создайте коллекцию тестов для CRUD операций с транзакциями.

2. Настройте **автоматический запуск коллекции** из командной строки с помощью Newman или в рамках CI-пайплайна.

3. Настройте **валидацию ответов:** статус-коды, схему JSON (JSON Schema), значения полей.

4. **Сформируйте отчет** о прохождении API-тестов.

Задание 3.3: Тестирование подключения к БД

Цель: Проверить корректность работы с базой данных.

Задача:

1. Напишите **интеграционные тесты**, которые:

- Подключаются к тестовой БД (PostgreSQL/MySQL).
- Вставляют тестовую запись о транзакции.
- Читают и проверяют эту запись.

▪ Очищают тестовые данные после выполнения (используйте фикстуры setUp/tearDown или аннотации @Before/@After).

2. Используйте **Docker** для поднятия изолированной тестовой БД.
3. **Оформите отчет**, включив в него логи SQL-запросов и время отклика БД.

Задание 3.4: Разработка автотеста с моками и стабами

Цель: Научиться изолировать тестируемый код от внешних зависимостей.

Задача:

1. Протестируйте функцию `generate_monthly_report(user_id)`, которая зависит от:
 - Внешнего `CurrencyService` (курсы валют).
 - Внутреннего `EmailService` (отправка отчета).
2. Создайте **Mock** для `CurrencyService`, который возвращает фиксированный курс.
3. Создайте **Stub/Spy** для `EmailService`, чтобы проверить, что метод отправки был вызван с правильными аргументами.
4. Напишите тест, проверяющий корректность формирования отчета с использованием этих заглушек.

Задание 3.5: Настройка системы логирования и ротации

Цель: Организовать сбор логов для анализа.

Задача:

1. Настройте логирование в ваших автотестах (например, с помощью `log4j`, `loguru` или `winston`).
2. Настройте **ротацию лог-файлов** по размеру (например, не более 10 МБ) или по времени (ежедневно). Используйте `RotatingFileHandler` или `logrotate`.
3. Уровни логирования: `INFO` для начала/окончания теста, `DEBUG` для деталей запросов, `ERROR` для сбоев.

Задание 3.6: Анализ логов и подготовка отчета мониторинга

Цель: Научиться извлекать полезную информацию из логов.

Задача:

1. Проведите 10 запусков UI-тестов.
2. С помощью скриптов (`grep/awk`) или лог-агрегатора (можно использовать `ELK/Seq`) проанализируйте логи:
 - Найдите самый частый тип ошибок.
 - Рассчитайте среднее время выполнения тестов.
 - Выявите тесты с наибольшим числом повторных попыток (flaky tests).
3. **Подготовьте отчет мониторинга** с выводами и рекомендациями по стабилизации тестов.

Этап 4: Тестирование безопасности и производительности

Задание 4.1: Ручное выявление уязвимостей веб-страниц

Цель: Провести базовый ручной security-аудит.

Задача:

1. Используя браузерные инструменты разработчика (DevTools), проверьте:
 - **XSS:** Попробуйте ввести `<script>alert(1)</script>` в поля ввода.
 - **Информационная утечка:** Проверьте ответы сервера на наличие служебной информации в заголовках или телах ошибок.
 - **Небезопасные куки:** Проверьте флаги `HttpOnly` и `Secure` у сессионных кук.
2. **Оформите отчет** с обнаруженными потенциальными уязвимостями и рекомендациями.

Задание 4.2: Тестирование производительности

Цель: Оценить поведение системы под нагрузкой.

Задача:

1. С помощью **JMeter** или **k6** создайте сценарий нагрузки:
 - 100 виртуальных пользователей в течение 5 минут.
 - Сценарий: авторизация -> добавление транзакции -> просмотр отчета.
2. Запустите тест, соберите метрики: **Response Time, Throughput, Error Rate**.

3. **Проанализируйте отчет JMeter/Grafana.** Найдите «узкое место». Предложите гипотезу по оптимизации (кэширование, индексы БД).

Задание 4.3: Тестирование авторизации (включая 2FA)

Цель: Проверить механизмы контроля доступа.

Задача:

1. Напишите **негативные тесты** для проверки авторизации:
 - Неверный логин/пароль.
 - Просмотр защищенных страниц без токена.
 - Попытка использовать чужой user_id для доступа к данным.
2. Если реализована 2FA, протестируйте сценарии: корректный код, неверный код, повторное использование кода.
3. **Документируйте** результаты проверки.

Задание 4.4: Разработка плана и пакетов для тестирования безопасности

Цель: Спланировать комплексную security-проверку.

Задача:

1. Разработайте **План тестирования безопасности (Security Test Plan)** для "FinTrack", включив: цели, области тестирования (OWASP Top-10), подходы (ручной/авто), инструменты, критерии успеха.
2. Создайте **тестовый пакет (Security Test Suite)** из 10 тестов, включающий проверки на SQL-инъекцию, XSS, IDOR, безопасную передачу данных.

Задание 4.5: Комплексное тестирование безопасности и финальный отчет

Цель: Провести итоговый security-аудит.

Задача:

1. **Выполните** ключевые тесты из плана (используйте также автоматические сканеры, например, **OWASP ZAP** в пассивном режиме).
2. **Оцените риски** для каждого найденного дефекта (Severity + Probability).
3. **Подготовьте итоговый отчет о тестировании безопасности**, включающий:
 - Резюме и общую оценку.
 - Детали уязвимостей с шагами воспроизведения и CVSS-оценкой (если применимо).
 - Приоритизированный план исправлений.
 - Рекомендации для разработчиков (Security Guidelines).

Итоговая сдача проекта:

Требуется предоставить:

1. Git-репозиторий со всей структурой кода и конфигураций.
2. Исполняемые автотесты всех типов.
3. Полный набор сгенерированных отчетов (Allure/HTML, результаты JMeter, security-отчет).
4. Итоговую **презентацию/отчет**, демонстрирующую архитектуру тестовой системы, ключевые метрики и выводы по каждому этапу.

Критерии оценки:

Полнота реализации (35%) – выполнены все задания.

Качество кода и архитектуры (30%) – читаемость, использование паттернов, модульность.

Качество отчетов и документации (25%) – четкость, информативность, профессиональное оформление.

Аналитика и выводы (10%) – глубина анализа статистики, логов, предложения по оптимизации.

4.2 Контрольные задания для промежуточной аттестации

Вопросы к экзамену

1. Дайте определение модульному тестированию. Что считается «модулем» (юнитом) в контексте такого тестирования?
2. Сформулируйте основные цели и преимущества модульного тестирования для процесса разработки и качества кода.
3. Что такое тестовое покрытие (code coverage)? Назовите основные его типы (покрытие строк, ветвей, условий) и объясните, какой из них является наиболее значимым.
4. Опишите цикл «Красный — Зеленый — Рефакторинг» в методологии разработки через тестирование (TDD). Какова роль модульных тестов в этом цикле?
5. Что такое изолированный unit-тест? Почему изоляция является ключевым принципом модульного тестирования?
6. Объясните разницу между заглушками (Stubs), моками (Mocks) и фейками (Fakes). В каких сценариях модульного тестирования применяется каждый из них?
7. Что такое параметризованные тесты (Parameterized Tests)? Как они помогают улучшить тестовое покрытие и избежать дублирования кода?
8. Назовите и охарактеризуйте три основных компонента структуры модульного теста (Arrange-Act-Assert / Given-When-Then).
9. Какие атрибуты (аннотации) в популярных фреймворках (JUnit, NUnit, pytest) используются для обозначения тестового метода, методов настройки и очистки?
10. В чем заключаются основные трудности и ограничения модульного тестирования? Приведите примеры кода, который сложно покрыть юнит-тестами.
21. Что является объектом тестирования документации? Перечислите основные виды документации, которые требуют проверки.
22. Объясните, почему тестирование документации является важной частью обеспечения качества ПО. Какие риски оно помогает смягчить?
23. Какие аспекты проверяются при тестировании пользовательской документации (руководства пользователя, справка)? Назовите не менее 5 критериев.
24. Что такое тестирование на соответствие требованиям (Requirements Traceability)? Какую роль в этом процессе играет техническая документация?
25. Опишите процесс проверки документации по установке и развертыванию (Deployment/Installation Guide). Что является критически важным в таких документах?
26. Как можно автоматизировать проверку актуальности документации? Приведите примеры инструментов или практик (например, встроенная документация в код — docstrings, Swagger/OpenAPI).
27. Что такое «живая документация» (Living Documentation) в контексте BDD? Как фреймворки вроде Cucumber связывают тесты и документацию?
28. Какие типовые дефекты чаще всего встречаются в документации (фактические, структурные, стилистические)? Приведите примеры каждого.
29. Кто обычно отвечает за тестирование документации в проекте? Обоснуйте свою точку зрения.
30. Как тестирование API-документации (например, в формате OpenAPI/Swagger) связано с интеграционным тестированием?
31. Дайте определение интеграционному тестированию. Какова его основная цель в отличие от модульного тестирования?
32. Опишите основные стратегии (подходы) проведения интеграционного тестирования: «снизу вверх» (Bottom-Up), «сверху вниз» (Top-Down) и «большой взрыв» (Big Bang). Укажите их преимущества и недостатки.
33. Что такое драйверы (Drivers) и заглушки (Stubs) в интеграционном тестировании? В каких стратегиях они применяются?

34. Дайте определение системному тестированию. Какие аспекты системы проверяются на этом уровне?
35. Чем функциональное системное тестирование отличается от нефункционального (например, тестирования производительности, надежности, удобства использования)?
36. Что такое сквозное (End-to-End, E2E) тестирование? Как оно соотносится с системным тестированием?
37. Опишите типичный сценарий для интеграционного тестирования взаимодействия веб-клиента с бэкенд-API и базой данных.
38. Какие основные сложности возникают при автоматизации интеграционных и системных тестов по сравнению с модульными?
39. Что такое тестовое окружение для интеграционного и системного тестирования? Каковы ключевые требования к его состоянию и конфигурации?
40. Как регрессионное тестирование связано с интеграционным и системным уровнями? Почему на этих уровнях особенно важна автоматизация регрессии?
41. Дайте определение надежности программного обеспечения в контексте эксплуатации. Какие метрики используются для ее оценки (MTBF, MTTR, Availability)?
42. Как автоматизированное тестирование (в частности, интеграционное и системное) способствует повышению надежности ПО перед выпуском в эксплуатацию?
43. Что такое тестирование восстановления после сбоев (Recovery Testing) и тестирование отказоустойчивости (Failover Testing)? Какие аспекты надежности они проверяют?
44. Как нагрузочное (Load) и стрессовое (Stress) тестирование связаны с проверкой надежности системы в условиях, близких к эксплуатационным?
45. Объясните, как практика мониторинга в production-среде (логи, метрики, трассировка) позволяет оценивать и поддерживать надежность работающей системы.
46. Дайте определение безопасности ПО. Назовите три основных принципа информационной безопасности (CIA Triad).
47. Что такое статический анализ безопасности кода (SAST) и как его можно интегрировать в процесс автоматизированного тестирования на ранних этапах?
48. Что такое динамический анализ безопасности приложений (DAST)? Приведите пример типовой уязвимости, которую он помогает обнаружить.
49. Как автоматизированные интеграционные тесты могут проверять корректность работы механизмов аутентификации и авторизации в системе?
50. Опишите концепцию DevSecOps. Как она меняет подход к тестированию безопасности, интегрируя его в цикл разработки и эксплуатации?

Критерии оценки

Оценка «5» - (отлично)

При ответе материал изложен грамотным языком в определенной логической последовательности, точно использована терминология, полно раскрыто содержание материала в объеме, предусмотренном программой, продемонстрировано усвоение ранее изученных сопутствующих вопросов. Возможны одна - две неточности при освещении второстепенных вопросов.

Оценка «4» - (хорошо)

Ответ удовлетворяет в основном требованиям на оценку «5», но при этом имеет один из недостатков: в изложении допущены небольшие пробелы; допущены один – два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя; допущены ошибка или более двух недочетов при освещении второстепенных вопросов, легко исправленные по замечанию преподавателя.

Оценка «3» - (удовлетворительно)

При ответе неполно или непоследовательно раскрыто содержание материала, но показано общее понимание, имелись затруднения или допущены ошибки в определении понятий.

Оценка «2» - (неудовлетворительно)

При ответе не раскрыто основное содержание учебного материала; обнаружено незнание или непонимание обучающимся большей или наиболее важной части учебного материала; допущены ошибки в определении понятий, допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными умениями по данной теме в полной мере.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В.Н. Татищева»
(Астраханский государственный университет им. В.Н. Татищева)

Филиал АГУ им. В.Н. Татищева в г. Знаменске Астраханской области

СОГЛАСОВАНО
Руководитель ОПОП
Бориско С.Н.

«13» ноября 2025 г.

УТВЕРЖДАЮ
Председатель методического
объединения преподавателей
технологических дисциплин
Фисенко Т.Ю.
протокол заседания ЦК и МО № 3
от «13» ноября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной практике

по профессиональному модулю
ПМ.02 СОПРОВОЖДЕНИЕ ПРОЦЕССОВ ТЕСТИРОВАНИЯ В ПРОЦЕССЕ
ЭКСПЛУАТАЦИИ

| | |
|---------------------------------|---|
| Составители | Бориско С.Н., к.т.н., доцент, завкафедрой ЗнМИ; Мустафаев Н.Г., к.т.н., доцент кафедры ЗнМИ; Тимошкин А.А., к.т.н., доцент кафедры ЗнМИ; Устинов А.С., к.т.н., доцент кафедры ЗнМИ; Каштанов Д.Ю., ассистент кафедры ЗнМИ |
| Согласовано с работодателями | Литвинов С.П., к.т.н., заместитель командира войсковой части 15644 по научно-исследовательской и испытательной работе; |
| Наименование специальности | Кириянов М.Н., ведущий инженер ПАО «Ростелеком» 09.02.12 Техническая эксплуатация и сопровождение информационных систем |
| Квалификация выпускника | Специалист по технической эксплуатации и сопровождению информационных систем |
| Форма обучения | очная |
| Год приема | 2026 (2 курс) |

Знаменск, 2025 г.

СОДЕРЖАНИЕ

- 1. Общие положения**
- 2. Результаты освоения учебной практики, подлежащие проверке**
- 3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания**
- 4. Контрольные задания для оценки результатов освоения учебной практики**

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки результатов прохождения учебной практики обучающимися в рамках профессионального модуля ПМ.02 «Сопровождение процессов тестирования в процессе эксплуатации».

ФОС включают контрольные материалы для оценки результатов освоения общих и профессиональных компетенций, полученных в период прохождения практики, разработанные в соответствии с требованиями ФГОС СПО и содержанием рабочей программы практики.

2. Результаты освоения учебной практики, подлежащие проверке

Соответствие изучаемых разделов, результатов обучения и оценочных средств

| № п/п | Контролируемые разделы (этапы) практики | Код контролируемой компетенции (компетенций) | Наименование оценочных средств |
|-------|---|--|---|
| 1 | Раздел 1 Подготовительный этап | ОК 1, ПК 2.1 | Дневник, Характеристика профессиональной деятельности студента на практике, Аттестационный лист, Отчет |
| 2 | Раздел 2 Практический этап | ОК 1, ПК 2.2 | |
| 3 | Раздел 3 Отчетный этап | ОК 1, ПК 2.1, ПК 2.2 | |

3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

- оценка «отлично» выставляется обучающемуся, если пороговый или высокий уровень сформированности профессиональных компетенций в период практики, высокий уровень знаний, высокая степень выполнения практических навыков, активный подход к решению профессиональных задач разной степени сложности, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

- оценка «хорошо» – пороговый или высокий уровень сформированности компетенций, в период практики, адекватный уровень знаний, адекватная степень выполнения практических навыков, адекватная способность к решению стандартных профессиональных задач, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

- оценка «удовлетворительно» – пороговый уровень сформированности компетенций в период практики, уровня знаний и степени выполнения практических навыков достаточно для решения типовых профессиональных задач, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

- оценка «неудовлетворительно» – пороговый или ниже уровень сформированности компетенций в период практики, уровня знаний и степени выполнения практических навыков недостаточно для решения типовых профессиональных задач, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

4. Контрольные задания для проверки результатов освоения учебной практики

Темы для учебной практики

Виды работ:

1. Анализ требований и выявление некорректных пользовательских требований и сценариев
2. Составление чек-листов на основе граничных значений
3. Разработка и генерация тест-кейсов для авторизации
4. Проверка логики бизнес-правил через данные в PostgreSQL
5. Генерация тестовых данных с использованием инструментария
6. Разработка тестовых сценариев и тестовых пакетов на основе таблицы решений
7. Разработка сценариев для e2e тестирования различных типов приложений
8. Сценарий тестирования отказа в базе (drop connection)
9. Проектирование тестов на основе пользовательских историй
10. Работа с файловой системой (проверка загрузки, чтения)
11. Разработка задания на дымовое тестирование
12. Автоматическая генерация тестовых данных с использованием интерфейса (API)
13. Планирование подготовки тестового окружения для нагрузочного тестирования
14. Планирование подготовки тестового окружения для тестирования безопасности
15. Проведение дымового тестирования веб-приложения
16. Оформление отчета по дефекту пользовательского интерфейса
17. Выполнение API-запросов
18. Анализ и тестирование ответов REST API с JSON
19. SQL-запросов различных видов для проверки записей в БД
20. Разработка и запуск модульных тестов с применением языков (не менее двух) программирования
21. Формирование библиотеки тестов.
22. Проверка логики бизнес-правил через данные в СУБД
23. Работа с ошибками в консоли разработчика в браузере
24. Реализация автотеста с использованием параметров
25. Настройка репозитория в системе контроля версий
26. Анализ логов приложения
27. Оформление отчетов о тестировании по каждой из выполненных работ
28. Запуск тестовых наборов из библиотеки тестов на выполнение в автоматическом режиме
29. Тестирование загрузки файлов и валидации форматов.
30. Тестирование обновлений в структуре БД
31. Настройка и проверка взаимодействия тестов с системой логирования.
32. Тестирование обновлений файлов конфигурации

Виды контрольных заданий по учебной практике

1. Разработка требований к информационной системе:
2. Разработка тестовых заданий.
3. Проведение тестов
4. Формирование библиотеки тестов.

Критерии оценки тестов:

Студенту выставляются следующие оценки:

- менее 50% правильных ответов – неудовлетворительно;
- 50 – 70% правильных ответов – удовлетворительно;
- 70 – 90% правильных ответов – хорошо;
- 90 – 100% правильных ответов – отлично.

Критерии оценки ситуационных задач

«Отлично» ставится, если дан полный, развернутый ответ на поставленный вопрос; показана совокупность осознанных знаний об объекте изучения, доказательно раскрыты основные положения (свободно оперирует понятиями, терминами, категориями и др.); в ответе отслеживается четкая структура, выстроенная в логической последовательности; ответ изложен научным грамотным языком; на возникшие вопросы преподавателя студент давал четкие, конкретные ответы, показывая умение выделять существенные и несущественные моменты материала.

«Хорошо» ставится, если дан полный, развернутый ответ на поставленный вопрос, показано умение выделять существенные и несущественные моменты материала; ответ четко структурирован, выстроен в логической последовательности, изложен литературным грамотным языком; однако были допущены неточности в определении понятий, персоналий, терминов, дат и др.

«Удовлетворительно» ставится, если дан неполный ответ на поставленный вопрос, логика и последовательность изложения имеют некоторые нарушения, допущены несущественные ошибки в изложении теоретического материала и употреблении терминов, персоналий; в ответе не присутствуют доказательные выводы; сформированность умений показана слабо, речь неграмотная.

«Неудовлетворительно» ставится, если дан неполный ответ на поставленный вопрос, логика и последовательность изложения имеют существенные нарушения, допущены существенные ошибки в теоретическом материале (фактах, понятиях, категориях); в ответе отсутствуют выводы, сформированность умений не показана, речь неграмотная.

Критерии оценки практических заданий:

«Отлично» - рабочее место оснащается с соблюдением всех требований к подготовке для выполнения манипуляций; практические действия выполняются последовательно, в соответствии с алгоритмом выполнения манипуляций; соблюдаются все требования к безопасности пациента и медперсонала; выдерживается соответствие с алгоритмом действий; рабочее место убирается, в соответствии с требованиями санэпидрежима; все действия обосновываются.

«Хорошо» - рабочее место не полностью самостоятельно оснащается для выполнения практических манипуляций; практические действия выполняются последовательно, но неуверенно; соблюдаются все требования к безопасности пациента и медперсонала; нарушается регламент времени, в соответствии с алгоритмом действий; рабочее место убирается, в соответствии с требованиями санэпидрежима; все действия обосновываются с уточняющими вопросами педагога.

«Удовлетворительно» - рабочее место не полностью оснащается для выполнения практических манипуляций; нарушена последовательность их выполнения; действия неуверенные, для обоснования действий необходимы наводящие и дополнительные вопросы и комментарии педагога; соблюдаются все требования к безопасности пациента и медперсонала; рабочее место убирается, в соответствии с требованиями санэпидрежима;

«Неудовлетворительно» - затруднения с подготовкой рабочего места, невозможность самостоятельно выполнить практические манипуляции; совершаются действия, нарушающие безопасность пациента и медперсонала, нарушаются требования санэпидрежима, техники безопасности при работе с аппаратурой, используемыми материалами.

4.2. Требования к дифференциальному зачету по учебной практике

Дифференциальный зачет по учебной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности обучающегося на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и требованиями организации, в которой проходила практика.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В.Н. Татищева»
(Астраханский государственный университет им. В.Н. Татищева)

Филиал АГУ им. В.Н. Татищева в г. Знаменске Астраханской области

СОГЛАСОВАНО
Руководитель ОПОП
Бориско С.Н.

«13» ноября 2025 г.

УТВЕРЖДАЮ
Председатель методического
объединения преподавателей
технологических дисциплин
Фисенко Т.Ю.
протокол заседания ЦК и МО № 3
от «13» ноября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по производственной практике
по профессиональному модулю
ПМ.02 СОПРОВОЖДЕНИЕ ПРОЦЕССОВ ТЕСТИРОВАНИЯ В ПРОЦЕССЕ
ЭКСПЛУАТАЦИИ.

| | |
|---------------------------------|---|
| Составители | Бориско С.Н., к.т.н., доцент, завкафедрой ЗнМИ; Мустафаев Н.Г., к.т.н., доцент кафедры ЗнМИ; Тимошкин А.А., к.т.н., доцент кафедры ЗнМИ; Устинов А.С., к.т.н., доцент кафедры ЗнМИ; Каштанов Д.Ю., ассистент кафедры ЗнМИ |
| Согласовано с работодателями | Литвинов С.П., к.т.н., заместитель командира войсковой части 15644 по научно-исследовательской и испытательной работе; |
| Наименование специальности | Кириянов М.Н., ведущий инженер ПАО «Ростелеком» 09.02.12 Техническая эксплуатация и сопровождение информационных систем |
| Квалификация выпускника | Специалист по технической эксплуатации и сопровождению информационных систем |
| Форма обучения | очная |
| Год приема | 2026 (3 курс) |

Знаменск, 2025 г.

СОДЕРЖАНИЕ

- 1. Общие положения**
- 2. Результаты освоения производственной практики, подлежащие проверке**
- 3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания**
- 4. Контрольные задания для оценки результатов освоения производственной практики**

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки результатов прохождения производственной практики обучающимися в рамках профессионального модуля ПМ.02 «Сопровождение процессов тестирования в процессе эксплуатации».

ФОС включают контрольные материалы для оценки результатов освоения общих и профессиональных компетенций, полученных в период прохождения практики, разработанные в соответствии с требованиями ФГОС СПО и содержанием рабочей программы практики.

2. Результаты освоения производственной практики, подлежащие проверке

Соответствие изучаемых разделов, результатов обучения и оценочных средств

| № п/п | Контролируемые разделы (этапы) практики | Код контролируемой компетенции (компетенций) | Наименование оценочных средств |
|-------|---|--|---|
| 1 | Раздел 1 Подготовительный этап | ОК 1 | Дневник, Характеристика профессиональной деятельности студента на практике, Аттестационный лист, Отчет |
| 2 | Раздел 2 Практический этап | ОК 1, ПК 2.3, ПК 2.4, ПК 2.5, ПК 2.6 | |
| 3 | Раздел 3 Отчетный этап | ОК 1, ПК 2.6 | |

3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

- оценка «отлично» выставляется обучающемуся, если пороговый или высокий уровень сформированности профессиональных компетенций в период практики, высокий уровень знаний, высокая степень выполнения практических навыков, активный подход к решению профессиональных задач разной степени сложности, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

- оценка «хорошо» – пороговый или высокий уровень сформированности компетенций, в период практики, адекватный уровень знаний, адекватная степень выполнения практических навыков, адекватная способность к решению стандартных профессиональных задач, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

- оценка «удовлетворительно» – пороговый уровень сформированности компетенций в период практики, уровня знаний и степени выполнения практических навыков достаточно для решения типовых профессиональных задач, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

- оценка «неудовлетворительно» – пороговый или ниже уровень сформированности компетенций в период практики, уровня знаний и степени выполнения практических навыков недостаточно для решения типовых профессиональных задач, предусмотренных программой практики в соответствии с ФГОС ВО и профессиональным стандартом и/или квалификационными характеристиками.

4. Контрольные задания для проверки результатов освоения производственной практики

Виды контрольных заданий по производственной практике

1. Подготовка тестирования
2. Разработка тестов
3. Проведение тестирования
4. Анализ дефектов
5. Документирование

Критерии оценки тестов:

Студенту выставляются следующие оценки:

- менее 50% правильных ответов – неудовлетворительно;
- 50 – 70% правильных ответов – удовлетворительно;
- 70 – 90% правильных ответов – хорошо;
- 90 – 100% правильных ответов – отлично.

Критерии оценки ситуационных задач

«Отлично» ставится, если дан полный, развернутый ответ на поставленный вопрос; показана совокупность осознанных знаний об объекте изучения, доказательно раскрыты основные положения (свободно оперирует понятиями, терминами, категориями и др.); в ответе отслеживается четкая структура, выстроенная в логической последовательности; ответ изложен научным грамотным языком; на возникшие вопросы преподавателя студент давал четкие, конкретные ответы, показывая умение выделять существенные и несущественные моменты материала.

«Хорошо» ставится, если дан полный, развернутый ответ на поставленный вопрос, показано умение выделять существенные и несущественные моменты материала; ответ четко структурирован, выстроен в логической последовательности, изложен литературным грамотным языком; однако были допущены неточности в определении понятий, персоналий, терминов, дат и др.

«Удовлетворительно» ставится, если дан неполный ответ на поставленный вопрос, логика и последовательность изложения имеют некоторые нарушения, допущены несущественные ошибки в изложении теоретического материала и употреблении терминов, персоналий; в ответе не присутствуют доказательные выводы; сформированность умений показана слабо, речь неграмотная.

«Неудовлетворительно» ставится, если дан неполный ответ на поставленный вопрос, логика и последовательность изложения имеют существенные нарушения, допущены существенные ошибки в теоретическом материале (фактах, понятиях, категориях); в ответе отсутствуют выводы, сформированность умений не показана, речь неграмотная.

Критерии оценки практических заданий:

«Отлично» - рабочее место оснащается с соблюдением всех требований к подготовке для выполнения манипуляций; практические действия выполняются последовательно, в соответствии с алгоритмом выполнения манипуляций; соблюдаются все требования к безопасности пациента и медперсонала; выдерживается соответствие с алгоритмом действий; рабочее место убирается, в соответствии с требованиями санэпидрежима; все действия обосновываются.

«Хорошо» - рабочее место не полностью самостоятельно оснащается для выполнения практических манипуляций; практические действия выполняются последовательно, но неуверенно; соблюдаются все требования к безопасности пациента и медперсонала; нарушается регламент времени, в соответствии с алгоритмом действий;

рабочее место убирается, в соответствии с требованиями санэпидрежима; все действия обосновываются с уточняющими вопросами педагога.

«Удовлетворительно» - рабочее место не полностью оснащается для выполнения практических манипуляций; нарушена последовательность их выполнения; действия неуверенные, для обоснования действий необходимы наводящие и дополнительные вопросы и комментарии педагога; соблюдаются все требования к безопасности пациента и медперсонала; рабочее место убирается, в соответствии с требованиями санэпидрежима;

«Неудовлетворительно» - затруднения с подготовкой рабочего места, невозможность самостоятельно выполнить практические манипуляции; совершаются действия, нарушающие безопасность пациента и медперсонала, нарушаются требования санэпидрежима, техники безопасности при работе с аппаратурой, используемыми материалами.

4.2. Требования к дифференциальному зачету по производственной практике

Дифференциальный зачет по производственной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности обучающегося на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и требованиями организации, в которой проходила практика.