

Документ подписан Министром науки и высшего образования Российской Федерации

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 17.06.2026 13:24:14

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Ростовский государственный экономический университет (РИНХ)»
Финансово-экономический колледж



УТВЕРЖДАЮ

Директор

Р. А. Сычев

2026г.

Рабочая программа МДК
Автоматизация процессов тестирования программного обеспечения

Специальность
09.02.12 ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ
ИНФОРМАЦИОННЫХ СИСТЕМ

Форма обучения очная

Часов по учебному плану 144

в том числе:

аудиторные занятия 80

самостоятельная работа 58

Ростов-на-Дону
2026 г.

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	6 (3.2)		Итого	
	10			
Неделя	10			
Вид занятий	УП	РП	УП	РП
Лекции	40	40	40	40
Практические	40	40	40	40
Итого ауд.	80	80	80	80
Контактная работа	80	80	80	80
Сам. работа	58	58	58	58
Часы на контроль	6	6	6	6
Итого	144	144	144	144

ОСНОВАНИЕ

Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.12 ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ (Приказ Министерства просвещения Российской Федерации от 10 марта 2025г № 184)

Рабочая программа составлена по образовательной программе 09.02.12 ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ для набора 2026 года.
программа среднего профессионального образования

Учебный план утвержден учёным советом вуза от 03.03.2026 протокол № 9

Программу составил(и): Преподаватель, Журавлёв Д.Г.

Председатель ЦМК: Ламин В.А.

Рассмотрено на заседании ЦМК от 06.03.2026 протокол № 7

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Формирование системных знаний и практических навыков по применению современных инструментов и фреймворков для автоматизации тестирования, включая модульное, интеграционное и UI-тестирование, а также анализ результатов и интеграцию автоматизированных тестов в процессы CI/CD.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ООП:	МДК.02
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Основы информационной безопасности
2.1.2	Производственная практика
2.1.3	Тестирование и эксплуатация информационных систем
2.1.4	Разработка информационных систем

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
3.1 Знать	
ПК 2.3 Тестировать эксплуатационную и техническую документацию на программное обеспечение	
<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; 	
ПК 2.4 Проводить регрессионные виды тестирования по разработанным тестовым случаям в соответствии с документацией на программное обеспечение и анализ результатов тестирования	
<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; - Стандарты оформления кода для используемых языков программирования; - Основные термины и сокращения, используемые в технической документации и принятые в организации; - Основы алгоритмизации и программирования; - Жизненный цикл программного продукта; 	
ПК 2.5 Выполнять восстановление тестов после сбоев, повлекших за собой нарушение работы системы, в том числе автоматизированных тестов	
<ul style="list-style-type: none"> - Архитектуру тестируемой системы; - Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного 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; - 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 Выполнять проверку исправленных дефектов и оформление результатов тестирования

- Жизненный цикл ПО, жизненный цикл дефекта ПО;
- 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;
- Основные инструментальные средства организации работы в команде;

3.2 Уметь

ПК 2.3 Тестировать эксплуатационную и техническую документацию на программное обеспечение

- Читать техническую документацию на ПО в объеме, необходимом для выполнения задания;
- Оформлять техническую документацию на ПО в рамках своей компетенции;
- Составлять отчет о тестировании эксплуатационной и технической документации на ПО;

ПК 2.4 Проводить регрессионные виды тестирования по разработанным тестовым случаям в соответствии с документацией на программное обеспечение и анализ результатов тестирования

- Составлять сценарии поведения пользователей ПО;
- Выполнять интеграционное и модульное тестирование ПО;
- Выполнять статическое тестирование ПО;
- Использовать специальное ПО для автоматизированного тестирования ПО при необходимости;
- Составлять отчет о проведении тестирования ПО по разработанным тестовым случаям;
- Взаимодействовать с членами команды разработчиков ПО;
- Использовать системы автоматизированного тестирования ПО;

ПК 2.5 Выполнять восстановление тестов после сбоев, повлекших за собой нарушение работы системы, в том числе автоматизированных тестов

- Находить и использовать информацию, необходимую для восстановления тестов после сбоя;
- Взаимодействовать с командой разработчиков при восстановлении системы после сбоя;
- Применять языки программирования для написания программного кода;
- Использовать системы автоматизированного тестирования ПО;
- Составлять отчет о восстановлении работоспособности ПО;

ПК 2.6 Выполнять проверку исправленных дефектов и оформление результатов тестирования

- Взаимодействовать с членами команды разработчиков ПО;
- Использовать инструменты командной работы над проектом ПО;
- Вносить изменения в скрипты автоматизированных тестов при необходимости;
- Использовать шаблоны тестов;
- Применять тесты;

3.3 Владеть

ПК 2.3 Тестировать эксплуатационную и техническую документацию на программное обеспечение

- Навыками проверки полноты эксплуатационной и технической документации на ПО;
- Навыками выявления недостатков эксплуатационной и технической документации на ПО и ее несоответствия внутренним стандартам качества организации;
- Навыками проверки эксплуатационной и технической документации на ПО на соответствие требованиям заказчика;
- Навыками выполнения действий по указаниям в эксплуатационной и технической документации на ПО;
- Навыками проверки соответствия действительных и указанных в эксплуатационной и

технической документации на ПО результатов;

- Навыками выявления несовпадений действительных и указанных в эксплуатационной и технической документации результатов регистрация найденных дефектов ПО в системе контроля дефектов;

ПК 2.4 Проводить регрессионные виды тестирования по разработанным тестовым случаям в соответствии с документацией на программное обеспечение и анализ результатов тестирования

- Навыками выполнения начальных настроек для проведения тестирования ПО;

- Навыками выполнения необходимых видов тестирования ПО в соответствии с планом тестирования;

- Навыками проведения автоматизированного тестирования ПО при необходимости;

- Навыками составления статистики выполнения тестов;

- Навыками проведения анализа полученных результатов тестирования ПО по разработанным тестовым случаям на соответствие ожидаемым результатам;

- Навыками оптимизации тестовых наборов;

- Навыками составления новых тестовых случаев и повторение тестирования при необходимости;

- Навыками формирования и представления отчетности о проведенном тестировании ПО в соответствии с установленными регламентами;

ПК 2.5 Выполнять восстановление тестов после сбоев, повлекших за собой нарушение работы системы, в том числе автоматизированных тестов

- Навыками определения причины сбоя системы совместно с разработчиками;

- Навыками устранения причины сбоя системы, если она находится в компетенции специалиста, либо подготовка отчета руководителю и группе разработчиков;

- Навыками выполнения настройки для повторного тестирования после сбоя;

- Навыками восстановления/изменения автоматизированных тестов после сбоя при необходимости в соответствии с планом/регламентом восстановления;

- Навыками проведения повторного тестирования ПО;

- Навыками формирования и представления отчетности о восстановлении работоспособности ПО в соответствии с установленными регламентами;

ПК 2.6 Выполнять проверку исправленных дефектов и оформление результатов тестирования

- Навыками получения обновленной версии ПО;

- Навыками определения масштабов изменений для выявления необходимости проведения регрессионных тестов;

- Навыками определения оптимального перечня тестов для повторного тестирования ПО;

- Навыками выполнения тестовых сценариев, выявивших дефекты ПО, для подтверждения успешности их выполнения после исправления ПО;

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Примечание
	Раздел 1. Модульное тестирование.					
1.1	Основные средства модульного тестирования в актуальных операционных системах. Синтаксис языков программирования для проектирования модульных тестов. Область применения модульных автотестов.	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

	Средства создания и сопровождения библиотеки тестов. Содержание отчета о тестировании. Типовые формы отчетов. /Лек/					
1.2	Основные средства модульного тестирования в актуальных операционных системах. Синтаксис языков программирования для проектирования модульных тестов. Область применения модульных автотестов. Средства создания и сопровождения библиотеки тестов. Содержание отчета о тестировании. Типовые формы отчетов. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.3	Основные средства модульного тестирования в актуальных операционных системах. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.4	Синтаксис языков программирования для проектирования модульных тестов. Область применения модульных автотестов. Средства создания и сопровождения библиотеки тестов. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.5	Содержание отчета о тестировании. Типовые формы отчетов. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.6	Системы контроля дефектов. Логирование. Сбор статистики. Средства развертывания и интеграции автотестов. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.7	Системы контроля дефектов. Логирование. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.8	Логирование. Сбор статистики. Средства	6	2	ПК 2.3. ПК 2.4. ПК 2.5.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

	развертывания и интеграции автотестов. /Ср/			ПК 2.6.		
1.9	Системы контроля дефектов. Логирование. Сбор статистики. Средства развертывания и интеграции автотестов. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.10	Разработка модульных автотестов для настольных, приложений. Запуск автотестов и сбор статистик. Оформление отчета по результатам анализа статистики. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.11	Разработка модульных автотестов для настольных, приложений. Запуск автотестов и сбор статистик. Оформление отчета по результатам анализа статистики. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.12	Разработка модульных автотестов для настольных, приложений.. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.13	Запуск автотестов и сбор статистик. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.14	Оформление отчета по результатам анализа статистики. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.15	Сборка и запуск тестов из консоли. Запуск автотестов и сбор статистик. Оформление отчета по результатам анализа статистики. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
1.16	Оформление и размещение отчетов о тестировании в соответствии с заданием (создание библиотеки тестов). /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
	Раздел 2. Тестирование документации.					

2.1	Виды технической документации. Актуальные стандарты оформления технической документации. Инструменты и методы тестирования документации на ПО. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.2	Программы и методики приемочного тестирования. Альфа- и бета- тестирование, сбор результатов. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.3	Выполнение приемочного тестирования и оформление отчета о его результатах. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.4	Выполнение приемочного тестирования и оформление отчета о его результатах. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.5	Выполнение приемочного тестирования и оформление отчета о его результатах. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.6	Выполнение тестирования пользовательской документации и оформление отчета о его результатах. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.7	Выполнение тестирования пользовательской документации и оформление отчета о его результатах. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
2.8	Выполнение тестирования пользовательской документации и оформление отчета о его результатах. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
	Раздел 3. Основы интеграционного и системного тестирования.					
3.1	Основные инструменты интеграционного тестирования. Особенности	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

	организации системного тестирования. Средства автоматизации тестирования пользовательского интерфейса (тестирование UI). /Лек/					
3.2	Инструменты тестирования интерфейсов (API-тестирования). Встроенные инструменты разработчика для тестирования в браузерах. REST и SOAP: структура запроса, методы, коды ответов. Типовые SQL-запросы для тестирования подключения баз данных. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.3	Инструменты тестирования интерфейсов (API-тестирования). Встроенные инструменты разработчика для тестирования в браузерах. REST и SOAP: структура запроса, методы, коды ответов. Типовые SQL-запросы для тестирования подключения баз данных. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.4	Инструменты тестирования интерфейсов (API-тестирования). /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.5	Встроенные инструменты разработчика для тестирования в браузерах. REST и SOAP: структура запроса, методы, коды ответов. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.6	Типовые SQL-запросы для тестирования подключения баз	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

	данных. /Ср/				
3.7	Настройка автоматического запуска тестов интерфейсов (API-тестирования). Запуск тестов. Формирование отчета о тестировании. Разработка и запуск тестов для тестирования подключения баз данных. Оформление отчета о тестировании. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
3.8	Настройка автоматического запуска тестов интерфейсов (API-тестирования). Запуск тестов. Формирование отчета о тестировании. Разработка и запуск тестов для тестирования подключения баз данных. Оформление отчета о тестировании. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
3.9	Понятие заглушки. Имитаторы (Mock). Настройка тестового окружения. Актуальные фреймворки для тестирования кода. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
3.10	Понятие заглушки. Имитаторы (Mock). Настройка тестового окружения. Актуальные фреймворки для тестирования кода. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
3.11	Разработка автотеста с заглушками и имитаторами (моками). /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
3.12	Разработка автотеста с заглушками и имитаторами (моками). /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
3.13	Разработка и запуск тестов пользовательского интерфейса. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2

3.14	Разработка и запуск тестов пользовательского интерфейса. Оформление отчета о тестировании. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.15	Оформление отчета о тестировании. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.16	Разработка и запуск тестов пользовательского интерфейса. Оформление отчета о тестировании. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.17	Настройка автоматического сбора и хранения логов. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.18	Регрессионное тестирование в жизненном цикле ПО. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.19	Настройка выбранной системы логирования с учетом ротации файлов. Анализ логов и подготовка отчета о результатах мониторинга. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.20	Настройка выбранной системы логирования с учетом ротации файлов. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.21	Анализ логов и подготовка отчета о результатах мониторинга. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.22	Настройка выбранной системы логирования с учетом ротации файлов. Анализ логов и подготовка отчета о результатах мониторинга. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
3.23	Анализ логов и подготовка отчета о результатах мониторинга. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
	Раздел 4. Надежность, и безопасность ПО в процессе эксплуатации.					
4.1	Методы тестирования безопасности. Инструменты выявления уязвимостей	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

	в исходном коде: статический анализ и имитаторы вредоносных атак. Уровни безопасности и защиты данных. Принципы и методы обеспечения безопасности тестирования в процессе эксплуатации. /Лек/					
4.2	Методы тестирования безопасности. Инструменты выявления уязвимостей в исходном коде: статический анализ и имитаторы вредоносных атак. Уровни безопасности и защиты данных. Принципы и методы обеспечения безопасности тестирования в процессе эксплуатации. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.3	Методы тестирования безопасности. Инструменты выявления уязвимостей в исходном коде: статический анализ и имитаторы вредоносных атак. Уровни безопасности и защиты данных. Принципы и методы обеспечения безопасности тестирования в процессе эксплуатации. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.4	Методы тестирования безопасности. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.5	Инструменты выявления уязвимостей в исходном коде: статический анализ и имитаторы вредоносных атак. Уровни безопасности и защиты данных. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

4.6	Методы тестирования безопасности. Инструменты выявления уязвимостей в исходном коде: статический анализ и имитаторы вредоносных атак. Уровни безопасности и защиты данных. Принципы и методы обеспечения безопасности тестирования в процессе эксплуатации. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.7	Выявление типовых уязвимостей кода веб-страниц в ручном режиме. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.8	Тестирование авторизации, в том числе двухфакторной авторизации. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.9	Планирование процесса системного тестирования, оценка рисков и затрат. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.10	Типовые причины сбоя системы в процессе тестирования. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.11	Методы тестирования производительности. Методы нагрузочного тестирования. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.12	Стохастическое, стрессовое и другие виды тестирования, их область применения, достоинства и недостатки. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.13	Стохастическое, стрессовое и другие виды тестирования, их область применения, достоинства и недостатки. /Лек/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.14	Тестирование производительности приложения с помощью заданного инструментария. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	
4.15	Тестирование производительности приложения с помощью заданного	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2	

	инструментария. /Ср/				
4.16	Разработка сценария, плана тестирования, тестовых пакетов и задания для тестирования безопасности приложения. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.17	Комплексное тестирование безопасности приложения и подготовка отчета о результатах тестирования. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.18	Комплексное тестирование безопасности приложения и подготовка отчета о результатах тестирования. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.19	Комплексное тестирование безопасности приложения и подготовка отчета о результатах тестирования. /Пр/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.20	Комплексное тестирование безопасности приложения и подготовка отчета о результатах тестирования. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.21	Комплексное тестирование безопасности приложения и подготовка отчета о результатах тестирования. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.22	Комплексное тестирование безопасности приложения и подготовка отчета о результатах тестирования. /Ср/	6	2	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2
4.23	Экзамен. /Экзамен/	6	6	ПК 2.3. ПК 2.4. ПК 2.5. ПК 2.6.	Л1.1 Л1.2 Л2.1 Л2.2 Э1 Э2

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Фонд оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация проходит в форме экзамена. Перечень вопросов к экзамену:

1. Дайте определение модульного тестирования. Какова его основная цель
2. Какие средства модульного тестирования существуют в современных операционных системах
3. Перечислите основные синтаксические конструкции языков программирования, используемые для проектирования модульных тестов.
4. Какова область применения модульных автотестов
5. Какие инструменты используются для создания и сопровождения библиотек тестов
6. Что такое библиотека тестов и зачем она нужна
7. Какое содержание должен иметь отчёт о тестировании
8. Назовите типовые формы отчётов о тестировании.
9. Что такое система контроля дефектов Приведите примеры.
10. Какие данные фиксируются в системе контроля дефектов
11. Что такое логирование в контексте тестирования Какие уровни логирования существуют
12. Как собирается статистика выполнения тестов
13. Какие средства развертывания и интеграции автотестов вы знаете
14. Как организовать запуск автотестов из консоли
15. Как оформить и разместить отчёты о тестировании в соответствии с заданием
16. Как производится анализ статистики выполнения тестов
17. Что такое покрытие кода тестами Как его измерить
18. Какие метрики качества модульных тестов существуют
19. В чём разница между модульным тестированием и интеграционным
20. Как поддерживать актуальность библиотеки тестов при изменении кода
21. Какие виды технической документации на ПО подлежат тестированию
22. Назовите актуальные стандарты оформления технической документации (ГОСТ, ISO).
23. Какие инструменты и методы используются для тестирования документации
24. Что такое программа и методика приемочного тестирования
25. Опишите альфа- и бета-тестирование. В чём их различия
26. Как собираются результаты приёмного тестирования
27. Как проверяется полнота эксплуатационной и технической документации
28. Как выявить недостатки документации и её несоответствие внутренним стандартам качества
29. Как проверяется соответствие документации требованиям заказчика
30. Как регистрируются найденные дефекты в документации
31. Какие основные инструменты интеграционного тестирования вы знаете
32. В чём особенности организации системного тестирования
33. Какие средства автоматизации тестирования пользовательского интерфейса (UI) существуют
34. Что такое API-тестирование Какие инструменты для него применяются
35. Какие встроенные инструменты разработчика есть в браузерах для тестирования
36. Опишите структуру REST-запроса: методы, заголовки, тело, параметры.
37. Какие коды ответов HTTP вы знаете Что они означают
38. В чём отличие REST от SOAP
39. Какие типовые SQL-запросы используются для тестирования подключения баз данных
40. Что такое заглушка (stub) в тестировании
41. Что такое имитатор (mock) В каких случаях он применяется
42. Как настроить тестовое окружение для интеграционного тестирования
43. Назовите актуальные фреймворки для тестирования кода
44. Как настроить автоматический сбор и хранение логов
45. Что такое регрессионное тестирование Какова его роль в жизненном цикле ПО
46. Как отбирать тесты для регрессионного тестирования
47. Как определить масштабы изменений для выявления необходимости проведения регрессионных тестов
48. Как выполнить настройку для повторного тестирования после сбоя
49. Как восстановить или изменить автоматизированные тесты после сбоя
50. Какие навыки необходимы для выполнения тестовых сценариев, выявивших дефекты, после их исправления
51. Как разработать автотест с использованием заглушек и имитаторов

52. Как выполнить запуск тестов пользовательского интерфейса и оформить отчёт
53. Как настроить автоматический запуск API-тестов
54. Как выполнить тестирование подключения баз данных и оформить отчёт
55. Как анализировать логи и готовить отчёт о результатах мониторинга
56. Какие методы тестирования безопасности ПО вы знаете
57. Какие инструменты используются для выявления уязвимостей в исходном коде
58. Что такое статический анализ безопасности кода (SAST)
59. Что такое динамический анализ безопасности (DAST)
60. Какие уровни безопасности и защиты данных существуют
61. Каковы принципы и методы обеспечения безопасности тестирования в процессе эксплуатации
62. Как выявлять типовые уязвимости кода веб-страниц в ручном режиме
63. Как тестировать авторизацию, включая двухфакторную
64. Какие методы тестирования производительности вы знаете
65. Какие методы нагрузочного тестирования существуют
66. Что такое стресс-тестирование Чем оно отличается от нагрузочного
67. Что такое стохастическое тестирование Когда оно применяется
68. Каковы достоинства и недостатки различных видов тестирования (нагрузочное, стрессовое, стохастическое)
69. Как планировать процесс системного тестирования
70. Как оценивать риски и затраты при планировании тестирования
71. Какие типовые причины сбоя системы могут возникать в процессе тестирования
72. Как определить причину сбоя системы совместно с разработчиками
73. Как устранить причину сбоя, если она находится в компетенции специалиста по тестированию
74. Как подготовить отчёт о восстановлении работоспособности ПО
75. Как проводить повторное тестирование после сбоя
76. Как тестировать производительность приложения с помощью заданного инструментария
77. Как разработать сценарий, план тестирования, тестовые пакеты для тестирования безопасности
78. Как выполнить комплексное тестирование безопасности приложения и подготовить отчёт
79. Какие инструменты нагрузочного тестирования вы знаете (JMeter, Gatling, LoadRunner)
80. Как интерпретировать результаты нагрузочного тестирования
81. Каков жизненный цикл дефекта ПО Перечислите основные статусы.
82. Какие принципы регрессионного тестирования необходимо учитывать
83. Как взаимодействовать с членами команды разработчиков при тестировании
84. Какие инструменты командной работы над проектом ПО вы знаете (Jira, Trello, YouTrack)
85. Как вносить изменения в скрипты автоматизированных тестов при необходимости

Критерии оценивания:

5 баллов выставляется студентам за полный и правильный ответ на все вопросы билета с логическим обоснованием аргументов, в ответе нет ошибок.

4 балла выставляется студентам, если вопросы билета раскрыты полностью, но обоснования доказательства недостаточны, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

3 балла ставится студентам за правильный ответ на вопросы билета, при этом допущено более одной ошибки по изложению фактов или более двух-трёх недочетов в ответе.

2 балла ставится студентам, если допущены существенные ошибки, показавшие, что обучающийся не обладает обязательными умениями по данной теме в полной мере.

5.2. Фонд оценочных средств для проведения текущего контроля

Представлен в Приложении 1 к рабочей программе МДК

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы,	Заглавие	Издательство, год	Колич-во
Л1.1	Щербак, А. В.	Поддержка и тестирование	Москва: Юрайт, 2026	https://urait.ru/bcode/590253 неограниченный доступ

		программных модулей: учебник для СПО		зарегистрированным пользователям
Л1.2	Казарин, О. В.	Основы информационной безопасности: надежность и безопасность программного обеспечения : учебник для СПО	Москва: Юрайт, 2026	https://urait.ru/bcode/587457 неограниченный доступ зарегистрированным пользователям

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л2.1	Богатырев, В. А.	Надежность информационных систем : учебник для СПО	Москва: Юрайт, 2026	https://urait.ru/bcode/589073 неограниченный доступ зарегистрированным пользователям
Л2.2	Щербак, А. В.	Информационная безопасность : учебник для СПО	Москва: Юрайт, 2026	https://urait.ru/bcode/588374 неограниченный доступ зарегистрированным пользователям

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Компьютерные книги
Э2	Технология разработки программного обеспечения

6.3. Перечень программного обеспечения

6.3.1	Офисный пакет - LibreOffice
6.3.2	Веб-браузер - Chromium

6.4 Перечень информационных справочных систем

6.4.1	ИСС «КонсультантПлюс»
6.4.2	ИИС «Гарант»

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1	Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения.
-----	--

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе МДК.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

МДК 02.02 Автоматизация процессов тестирования программного обеспечения

1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

1.1 Показатели и критерии оценивания компетенций:

УУД, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ПК 2.3 Тестировать эксплуатационную и техническую документацию на программное обеспечение			
<p>Знать:</p> <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; 	<p>Получение систематических знаний</p> <p>о нормативно-технических материалах по вопросам испытания и тестирования ПО; основных понятиях о качестве ПО; видах технической документации; требованиях по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирусной защиты; основах работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО.</p>	<p>Уровень знаний:</p> <p>Знает на уровне понимания и воспроизведения: перечисляет нормативные документы, классифицирует виды документации, объясняет требования безопасности и основы работы в целевой ОС.</p>	<p>Т (1-100) ПЗ (1-24)</p>
<p>Уметь:</p> <ul style="list-style-type: none"> - Читать техническую документацию на ПО в объеме, необходимом для выполнения задания; - Оформлять техническую документацию на ПО в рамках своей компетенции; - Составлять отчет о тестировании эксплуатационной и технической документации на ПО; 	<p>Сформировать систематическое умение</p> <p>читать техническую документацию на ПО в объеме, необходимом для выполнения задания; оформлять техническую документацию на ПО в рамках своей компетенции; составлять отчет о тестировании эксплуатационной и технической документации на ПО.</p>	<p>Уровень умения:</p> <p>Умеет самостоятельно анализировать техническую документацию, выявлять несоответствия, оформлять отчеты в соответствии с регламентами.</p>	<p>Т (1-100) ПЗ (1-24)</p>

<p>Владеть:</p> <ul style="list-style-type: none"> - Навыками проверки полноты эксплуатационной и технической документации на ПО; - Навыками выявления недостатков эксплуатационной и технической документации на ПО и ее несоответствия внутренним стандартам качества организации; - Навыками проверки эксплуатационной и технической документации на ПО на соответствие требованиям заказчика; - Навыками выполнения действий по указаниям в эксплуатационной и технической документации на ПО; - Навыками проверки соответствия действительных и указанных в эксплуатационной и технической документации на ПО результатов; - Навыками выявления несовпадений действительных и указанных в эксплуатационной и технической документации результатов регистрации найденных дефектов ПО в системе контроля дефектов; 	<p>Сформировать систематическое владение навыками проверки полноты эксплуатационной и технической документации на ПО; выявления недостатков документации и её несоответствия внутренним стандартам качества организации; проверки документации на соответствие требованиям заказчика; выполнения действий по указаниям в документации; проверки соответствия действительных и указанных в документации результатов; выявления несовпадений и регистрации найденных дефектов в системе контроля дефектов.</p>	<p>Уровень владения: Владеет устойчивыми навыками проверки, анализа и документирования дефектов технической документации, а также использования систем контроля дефектов.</p>	<p>Т (1-100) ПЗ (1-24)</p>
--	---	--	---------------------------------------

ПК 2.4 Проводить регрессионные виды тестирования по разработанным тестовым случаям в соответствии с документацией на программное обеспечение и анализ результатов тестирования

<p>Знать:</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 	<p>Получение систематических знаний о техниках тестирования ПО, базирующихся на интуиции и опыте инженера, на спецификации, ориентированных на код, на условиях использования, на</p>	<p>Уровень знаний: Знает на уровне понимания и воспроизведения: перечисляет техники тестирования, объясняет их применение, знает стандарты кодирования и</p>	<p>Т (1-100) ПЗ (1-24)</p>
---	--	---	---------------------------------------

<p>код;</p> <ul style="list-style-type: none"> - 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; - Жизненный цикл программного продукта; 	<p>природе приложения; тестировании ПО, ориентированном на дефекты и базирующемся на надежности инженерного процесса; стандартах оформления кода для используемых языков программирования; основных терминах и сокращениях, используемых в технической документации; основах алгоритмизации и программирования; жизненном цикле программного продукта.</p>	<p>жизненный цикл ПО.</p>	
<p>Уметь:</p> <ul style="list-style-type: none"> - Составлять сценарии поведения пользователей ПО; - Выполнять интеграционное и модульное тестирование ПО; - Выполнять статическое тестирование ПО; - Использовать специальное ПО для автоматизированного тестирования ПО при необходимости; - Составлять отчет о проведении тестирования ПО по разработанным тестовым случаям; - Взаимодействовать с членами команды разработчиков ПО; - Использовать системы автоматизированного тестирования ПО; 	<p>Сформировать систематическое умение составлять сценарии поведения пользователей ПО; выполнять интеграционное и модульное тестирование ПО; выполнять статическое тестирование ПО; использовать специальное ПО для автоматизированного тестирования ПО при необходимости; составлять отчет о проведении тестирования ПО по разработанным тестовым случаям; взаимодействовать с членами команды разработчиков; использовать системы автоматизированного тестирования ПО.</p>	<p>Уровень умения: Умеет самостоятельно разрабатывать тестовые сценарии, выполнять различные виды тестирования (модульное, интеграционное, статическое), использовать инструменты автоматизации и оформлять отчёты.</p>	<p>Т (1-100) ПЗ (1-24)</p>
<p>Владеть:</p> <ul style="list-style-type: none"> - Навыками выполнения начальных настроек для 	<p>Сформировать систематическое владение</p>	<p>Уровень владения: Владеет устойчивыми</p>	<p>Т (1-100) ПЗ (1-24)</p>

<p>проведения тестирования ПО;</p> <ul style="list-style-type: none"> - Навыками выполнения необходимых видов тестирования ПО в соответствии с планом тестирования; - Навыками проведения автоматизированного тестирования ПО при необходимости; - Навыками составления статистики выполнения тестов; - Навыками проведения анализа полученных результатов тестирования ПО по разработанным тестовым случаям на соответствие ожидаемым результатам; - Навыками оптимизации тестовых наборов; - Навыками составления новых тестовых случаев и повторение тестирования при необходимости; - Навыками формирования и представления отчетности о проведенном тестировании ПО в соответствии с установленными регламентами; 	<p>навыками выполнения начальных настроек для проведения тестирования ПО; выполнения необходимых видов тестирования в соответствии с планом тестирования; проведения автоматизированного тестирования ПО при необходимости;</p> <p>составления статистики выполнения тестов; анализа полученных результатов тестирования на соответствие ожидаемым результатам; оптимизации тестовых наборов; составления новых тестовых случаев и повторения тестирования; формирования и представления отчетности о проведенном тестировании в соответствии с регламентами.</p>	<p>навыками планирования, выполнения и анализа автоматизированного тестирования, а также оптимизации тестовых наборов и подготовки отчетности.</p>	
---	---	--	--

ПК 2.5 Выполнять восстановление тестов после сбоя, повлекших за собой нарушение работы системы, в том числе автоматизированных тестов

<p>Знать:</p> <ul style="list-style-type: none"> - Архитектуру тестируемой системы; - Основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО; - Техники тестирования ПО, базирующиеся на интуиции и опыте инженера; - Техники тестирования ПО, базирующиеся на спецификации; - Техники тестирования ПО, ориентированные на код; 	<p>Получение систематических знаний об архитектуре тестируемой системы; основах работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО; техниках тестирования ПО (на интуиции, спецификации, коде, условиях использования, природе приложения); тестировании, ориентированном на дефекты и на надежности инженерного процесса;</p>	<p>Уровень знаний: Знает на уровне понимания: архитектуру системы, принципы регрессионного тестирования, техники тестирования, алгоритмы решения задач.</p>	<p>Т (1-100) ПЗ (1-24)</p>
---	--	--	---------------------------------------

<ul style="list-style-type: none"> - 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; 	<p>принципах регрессионного тестирования ПО; алгоритмах решения типовых задач; основных терминах и сокращениях.</p>		
<p>Уметь:</p> <ul style="list-style-type: none"> - Находить и использовать информацию, необходимую для восстановления тестов после сбоя; - Взаимодействовать с командой разработчиков при восстановлении системы после сбоя; - Применять языки программирования для написания программного кода; - Использовать системы автоматизированного тестирования ПО; - Составлять отчет о восстановлении работоспособности ПО; 	<p>Сформировать систематическое умение находить и использовать информацию, необходимую для восстановления тестов после сбоя; взаимодействовать с командой разработчиков при восстановлении системы после сбоя; применять языки программирования для написания программного кода; использовать системы автоматизированного тестирования ПО; составлять отчет о восстановлении работоспособности ПО.</p>	<p>Уровень умения: Умеет самостоятельно восстанавливать тестовую среду, взаимодействовать с разработчиками, писать код для восстановления и оформлять отчёты.</p>	<p>Т (1-100) ПЗ (1-24)</p>
<p>Владеть:</p> <ul style="list-style-type: none"> - Навыками определения причины сбоя системы совместно с разработчиками; - Навыками устранения причины сбоя системы, если она находится в компетенции специалиста, 	<p>Сформировать систематическое владение навыками определения причины сбоя системы совместно с разработчиками; устранения причины сбоя (в рамках компетенции) либо подготовки отчета</p>	<p>Уровень владения: Владеет устойчивыми навыками диагностики сбоев, восстановления тестов, повторного тестирования и документирования</p>	<p>Т (1-100) ПЗ (1-24)</p>

<p>либо подготовка отчета руководителю и группе разработчиков;</p> <ul style="list-style-type: none"> - Навыками выполнения настройки для повторного тестирования после сбоя; - Навыками восстановления/изменения автоматизированных тестов после сбоя при необходимости в соответствии с планом/регламентом восстановления; - Навыками проведения повторного тестирования ПО; - Навыками формирования и представления отчетности о восстановлении работоспособности ПО в соответствии с установленными регламентами; 	<p>руководителю; выполнения настройки для повторного тестирования после сбоя; восстановления/изменения автоматизированных тестов после сбоя в соответствии с регламентом; проведения повторного тестирования ПО; формирования и представления отчетности о восстановлении работоспособности ПО.</p>	<p>результатов восстановления.</p>	
---	---	------------------------------------	--

ПК 2.6 Выполнять проверку исправленных дефектов и оформление результатов тестирования

<p>Знать:</p> <ul style="list-style-type: none"> - Жизненный цикл ПО, дефекта ПО; - 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>Получение систематических знаний о жизненном цикле ПО и дефекта ПО; принципах регрессионного тестирования ПО; техниках тестирования ПО (на интуиции, спецификации, коде, условиях использования, природе приложения); тестировании, ориентированном на дефекты и на надежности инженерного процесса; основных инструментальных средствах организации работы в команде.</p>	<p>Уровень знаний: Знает на уровне понимания: жизненный цикл дефекта, принципы регрессии, техники тестирования и инструменты командной работы.</p>	<p>Т (1-100) ПЗ (1-24)</p>
---	--	---	---------------------------------------

- Основные инструментальные средства организации работы в команде;			
Уметь: - Взаимодействовать с членами команды разработчиков ПО; - Использовать инструменты командной работы над проектом ПО; - Вносить изменения в скрипты автоматизированных тестов при необходимости; - Использовать шаблоны тестов; - Применять тесты;	Сформировать систематическое умение взаимодействовать с членами команды разработчиков ПО; использовать инструменты командной работы над проектом ПО; вносить изменения в скрипты автоматизированных тестов при необходимости; использовать шаблоны тестов; применять тесты.	Уровень умения: Умеет работать в команде, модифицировать тестовые скрипты, использовать шаблоны и применять тесты для проверки исправленных дефектов.	Т (1-100) ПЗ (1-24)
Владеть: - Навыками получения обновленной версии ПО; - Навыками определения масштабов изменений для выявления необходимости проведения регрессионных тестов; - Навыками определения оптимального перечня тестов для повторного тестирования ПО; - Навыками выполнения тестовых сценариев, выявивших дефекты ПО, для подтверждения успешности их выполнения после исправления ПО;	Сформировать систематическое владение навыками получения обновленной версии ПО; определения масштабов изменений для выявления необходимости проведения регрессионных тестов; определения оптимального перечня тестов для повторного тестирования ПО; выполнения тестовых сценариев, выявивших дефекты, для подтверждения успешности их выполнения после исправления ПО.	Уровень владения: Владеет устойчивыми навыками управления версиями ПО, планирования регрессионного тестирования, выполнения повторных тестов и подтверждения исправления дефектов.	Т (1-100) ПЗ (1-24)

Т – тестовые задания, ПЗ – практические задания.

2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Тестовые задания:

Раздел 1. Модульное тестирование

1. Что такое модульное тестирование?
 - А) Тестирование отдельных компонентов программы в изоляции
 - В) Тестирование системы в целом
 - С) Тестирование пользовательского интерфейса
 - D) Тестирование документации
2. Какой инструмент чаще всего используется для модульного тестирования в Python?

- A) JUnit
 - B) PyTest
 - C) Selenium
 - D) JMeter
3. Какая аннотация в JUnit обозначает тестовый метод?
- A) @Before
 - B) @After
 - C) @Test
 - D) @Ignore
4. Что такое библиотека тестов?
- A) Набор переиспользуемых тестовых функций и утилит
 - B) Журнал выполнения тестов
 - C) Система контроля версий
 - D) Отчёт о тестировании
5. Какая информация должна содержаться в отчёте о тестировании?
- A) Дата, версия ПО, количество пройденных/упавших тестов, список ошибок
 - B) Только количество пройденных тестов
 - C) Исходный код приложения
 - D) Личные данные разработчиков
6. Что такое покрытие кода тестами (code coverage)?
- A) Процент строк или ветвей кода, выполненных во время тестов
 - B) Количество тестов в библиотеке
 - C) Время выполнения тестов
 - D) Количество найденных дефектов
7. Какой инструмент используется для сбора статистики покрытия кода в Python?
- A) pytest-cov
 - B) JUnit
 - C) Selenium
 - D) Postman
8. Что такое заглушка (stub) в модульном тестировании?
- A) Простая реализация зависимого компонента, возвращающая фиксированные данные
 - B) Реальный компонент системы
 - C) Отчёт о тестировании
 - D) Система логирования
9. Что такое имитатор (mock) в тестировании?
- A) Объект, имитирующий поведение реального компонента с возможностью проверки вызовов
 - B) Заглушка
 - C) Тестовый набор
 - D) Отладчик
10. Какой фреймворк для создания моков используется в Python?
- A) unittest.mock
 - B) Mockito
 - C) EasyMock
 - D) JMock
11. Что такое тест-сьют (test suite)?
- A) Группа тестов, объединённых по какому-либо признаку
 - B) Один тестовый метод
 - C) Отчёт о тестировании
 - D) Система контроля дефектов
12. Какой уровень покрытия считается наиболее строгим?
- A) Покрытие строк
 - B) Покрытие ветвей
 - C) Покрытие функций
 - D) Покрытие классов
13. Какая команда запускает все тесты в PyTest?
- A) python test.py

- B) pytest
 - C) pytest run
 - D) test run
14. Что такое фикстура (fixture) в PyTest?
- A) Функция, подготавливающая окружение для тестов
 - B) Тестовый метод
 - C) Ассерт
 - D) Исключение
15. Как в JUnit отметить, что тест должен выбросить исключение?
- A) @Test(expected=Exception.class)
 - B) @Test(expected = Exception.class)
 - C) @ExceptionTest
 - D) @Throws
16. Что такое параметризованный тест?
- A) Тест, выполняемый с несколькими наборами входных данных
 - B) Тест с параметрами командной строки
 - C) Тест, зависящий от времени
 - D) Тест без утверждений
17. Какая метрика оценивает количество тестов, успешно прошедших выполнение?
- A) Code coverage
 - B) Test pass rate
 - C) Defect density
 - D) Mean time to failure
18. Что такое TDD (Test-Driven Development)?
- A) Разработка через тестирование: сначала пишется тест, затем код
 - B) Разработка без тестов
 - C) Тестирование после разработки
 - D) Автоматическое тестирование
19. Какой инструмент статического анализа кода используется в Python?
- A) pylint
 - B) pytest
 - C) selenium
 - D) jmeter
20. Что такое интеграционное тестирование?
- A) Тестирование взаимодействия между модулями
 - B) Тестирование одного модуля
 - C) Тестирование производительности
 - D) Тестирование безопасности
21. Какой паттерн проектирования тестов рекомендуется для UI-автоматизации?
- A) Page Object Model (POM)
 - B) Singleton
 - C) Factory
 - D) Observer
22. Что такое CI/CD?
- A) Continuous Integration / Continuous Delivery
 - B) Code Integration / Code Deployment
 - C) Computer Interface / Computer Design
 - D) Common Internet / Common Domain
23. Какой инструмент CI/CD наиболее популярен?
- A) Jenkins
 - B) Selenium
 - C) JUnit
 - D) Postman
24. Что такое тестовый набор (test set)?
- A) Коллекция тестовых случаев для конкретной функции

- B) Ожидаемый результат
- C) Исходные данные
- D) Отчёт

25. Какой метод в unittest (Python) проверяет равенство?

- A) assertEquals()
- B) assertTrue()
- C) assertFalse()
- D) assertIs()

Раздел 2. Тестирование документации

26. Какие виды технической документации подлежат тестированию?

- A) Руководство пользователя, руководство администратора, спецификация API
- B) Только исходный код
- C) Только маркетинговые материалы
- D) Только финансовые отчёты

27. Какой стандарт оформления технической документации наиболее распространён в России?

- A) ГОСТ 19
- B) ISO 9001
- C) IEEE 802.11
- D) OWASP

28. Что такое приёмочное тестирование?

- A) Тестирование, проводимое заказчиком для подтверждения соответствия требованиям
- B) Модульное тестирование
- C) Нагрузочное тестирование
- D) Тестирование безопасности

29. Что такое альфа-тестирование?

- A) Внутреннее тестирование силами разработчиков и тестировщиков
- B) Тестирование реальными пользователями на площадке заказчика
- C) Автоматизированное тестирование
- D) Регрессионное тестирование

30. Что такое бета-тестирование?

- A) Тестирование реальными пользователями в реальных условиях
- B) Внутреннее тестирование
- C) Модульное тестирование
- D) Статическое тестирование

31. Как проверяется полнота эксплуатационной документации?

- A) Сравнением с чек-листом требований и функциональностью системы
- B) Только визуально
- C) Автоматически
- D) По отзывам пользователей

32. Что такое программа и методика испытаний?

- A) Документ, описывающий порядок тестирования, тестовые случаи и критерии приемки
- B) Исходный код тестов
- C) Отчёт о тестировании
- D) План проекта

33. Какая информация должна быть в отчёте о тестировании документации?

- A) Выявленные несоответствия, рекомендации, оценка полноты
- B) Только количество страниц
- C) Список авторов
- D) Исходный код

34. Как регистрируются дефекты в документации?

- A) В системе контроля дефектов (Jira, Bugzilla)
- B) В текстовом файле
- C) В устной форме
- D) В электронной почте

35. Что такое статическое тестирование документации?

- A) Анализ документации без выполнения кода
- B) Выполнение инструкций из документации
- C) Автоматическая проверка орфографии
- D) Рецензирование коллегами

Раздел 3. Интеграционное и системное тестирование (36–60)

36. Какие инструменты используются для API-тестирования?

- A) Postman, SoapUI
- B) JUnit, PyTest
- C) Selenium, Cypress
- D) JMeter, Gatling

37. Что такое REST API?

- A) Архитектурный стиль взаимодействия компонентов распределённого приложения
- B) Протокол передачи файлов
- C) База данных
- D) Язык программирования

38. Какой HTTP-метод обычно используется для получения ресурса?

- A) GET
- B) POST
- C) PUT
- D) DELETE

39. Какой HTTP-статус означает «Успешно создано»?

- A) 200 OK
- B) 201 Created
- C) 204 No Content
- D) 404 Not Found

40. Какой HTTP-статус означает «Не найдено»?

- A) 200
- B) 201
- C) 404
- D) 500

41. Чем SOAP отличается от REST?

- A) SOAP использует XML и строгую спецификацию, REST более гибкий
- B) REST использует только JSON
- C) SOAP не использует HTTP
- D) REST не использует HTTP

42. Какой инструмент используется для автоматизации UI-тестирования веб-приложений?

- A) Selenium
- B) JUnit
- C) Postman
- D) JMeter

43. Что такое end-to-end (e2e) тестирование?

- A) Тестирование полного пользовательского сценария от начала до конца
- B) Тестирование одного модуля
- C) Тестирование производительности
- D) Тестирование безопасности

44. Какой фреймворк используется для BDD (Behavior-Driven Development)?

- A) Cucumber
- B) JUnit
- C) PyTest
- D) Selenium

45. Что такое заглушка (stub) в интеграционном тестировании?

- A) Компонент, имитирующий отсутствующую часть системы
- B) Реальная система
- C) Отчёт
- D) Тестовые данные

46. Как настроить тестовое окружение для интеграционного тестирования?
- A) Развернуть изолированную среду с тестовыми БД и моками внешних сервисов
 - B) Использовать продуктивную среду
 - C) Не настраивать
 - D) Использовать только локальные файлы
47. Что такое регрессионное тестирование?
- A) Проверка, что изменения не нарушили существующую функциональность
 - B) Тестирование новых функций
 - C) Тестирование производительности
 - D) Тестирование безопасности
48. Как отбирать тесты для регрессионного тестирования?
- A) Выбирать тесты, покрывающие критическую функциональность и изменённые модули
 - B) Все тесты подряд
 - C) Только новые тесты
 - D) Только тесты, которые падали
49. Какой инструмент используется для нагрузочного тестирования?
- A) JMeter
 - B) Selenium
 - C) Postman
 - D) JUnit
50. Что такое стресс-тестирование?
- A) Тестирование системы за пределами нормальной нагрузки для выявления точки отказа
 - B) Тестирование в обычных условиях
 - C) Функциональное тестирование
 - D) Тестирование документации
51. Что такое пирамида тестирования?
- A) Модель распределения тестов: много модульных, меньше интеграционных, ещё меньше e2e
 - B) Иерархия дефектов
 - C) Структура отчёта
 - D) План тестирования
52. Какой инструмент используется для тестирования производительности веб-приложений?
- A) Gatling
 - B) Selenium
 - C) Postman
 - D) JUnit
53. Что такое дымовое тестирование (smoke test)?
- A) Минимальный набор тестов для проверки, что система запускается и не падает
 - B) Полное регрессионное тестирование
 - C) Нагрузочное тестирование
 - D) Тестирование безопасности
54. Какой тип тестирования проверяет время отклика системы?
- A) Тестирование производительности
 - B) Функциональное тестирование
 - C) Модульное тестирование
 - D) Тестирование документации
55. Что такое логирование в контексте тестирования?
- A) Запись событий выполнения тестов для последующего анализа
 - B) Удаление временных файлов
 - C) Компиляция кода
 - D) Отправка уведомлений
56. Какой уровень логирования самый подробный?
- A) DEBUG
 - B) INFO
 - C) WARNING
 - D) ERROR

57. Как настроить автоматический сбор логов в Jenkins?
- A) Использовать плагин для архивации артефактов
 - B) Отправить по email
 - C) Удалить логи
 - D) Не собирать
58. Что такое CI-пайплайн?
- A) Автоматическая последовательность сборки, тестирования и анализа кода
 - B) Ручной процесс
 - C) Только сборка
 - D) Только тестирование
59. Какой инструмент используется для оркестрации контейнеров при тестировании?
- A) Docker Compose / Kubernetes
 - B) Jenkins
 - C) Git
 - D) Postman
60. Что такое тестовый двойник (test double)?
- A) Общий термин для заглушек, моков, фейков и шпионов
 - B) Настоящий объект
 - C) Отчёт
 - D) Тестовые данные
- Раздел 4. Надежность, безопасность и производительность
61. Какие методы тестирования безопасности существуют?
- A) SAST (статический анализ), DAST (динамический анализ), пентест
 - B) Только модульное тестирование
 - C) Только нагрузочное тестирование
 - D) Только регрессионное
62. Что такое SAST (Static Application Security Testing)?
- A) Анализ исходного кода без его выполнения
 - B) Анализ работающего приложения
 - C) Нагрузочное тестирование
 - D) Функциональное тестирование
63. Что такое DAST (Dynamic Application Security Testing)?
- A) Анализ работающего приложения через отправку запросов
 - B) Статический анализ кода
 - C) Модульное тестирование
 - D) Регрессионное тестирование
64. Какой инструмент используется для статического анализа безопасности кода на Python?
- A) Bandit
 - B) Selenium
 - C) JMeter
 - D) Postman
65. Какой тип уязвимости позволяет внедрять вредоносный код в веб-страницу?
- A) XSS (Cross-Site Scripting)
 - B) SQL-инъекция
 - C) CSRF
 - D) Path traversal
66. Какой тип уязвимости позволяет выполнять произвольные SQL-запросы?
- A) SQL-инъекция
 - B) XSS
 - C) CSRF
 - D) Path traversal
67. Как защититься от SQL-инъекций в коде?
- A) Использовать параметризованные запросы
 - B) Экранировать кавычки
 - C) Не использовать базы данных
 - D) Использовать ORM без параметризации
68. Что такое двухфакторная аутентификация?

- A) Использование двух разных факторов (пароль + код из SMS)
 - B) Два пароля
 - C) Два логина
 - D) Два сервера
69. Как тестировать авторизацию?
- A) Проверять доступ к ресурсам с разными ролями
 - B) Только проверять пароль
 - C) Не тестировать
 - D) Только проверять email
70. Что такое OWASP Top Ten?
- A) Список десяти наиболее критичных уязвимостей веб-приложений
 - B) Инструмент тестирования
 - C) Стандарт документации
 - D) Язык программирования
71. Какие методы нагрузочного тестирования существуют?
- A) Дымовое, нагрузочное, стрессовое, тестирование выносливости
 - B) Только функциональное
 - C) Только модульное
 - D) Только интеграционное
72. Что такое тестирование выносливости (soak test)?
- A) Длительная работа системы под нагрузкой для выявления утечек памяти
 - B) Кратковременная пиковая нагрузка
 - C) Тестирование без нагрузки
 - D) Функциональное тестирование
73. Какой инструмент используется для нагрузочного тестирования сценариев пользователей?
- A) JMeter
 - B) Selenium
 - C) Postman
 - D) JUnit
74. Что такое стохастическое тестирование?
- A) Тестирование со случайными входными данными
 - B) Тестирование по фиксированному плану
 - C) Тестирование производительности
 - D) Тестирование безопасности
75. Какие типовые причины сбоя системы в процессе тестирования?
- A) Ошибки в коде, проблемы с памятью, сетевые сбои, некорректные данные
 - B) Только ошибки в коде
 - C) Только проблемы с сетью
 - D) Только ошибки пользователя
76. Как определить причину сбоя системы?
- A) Анализировать логи, стек вызовов, воспроизводить шаги
 - B) Перезагрузить систему
 - C) Игнорировать
 - D) Удалить тесты
77. Что нужно сделать после исправления дефекта?
- A) Провести повторное тестирование и регрессионное
 - B) Закрыть задачу без проверки
 - C) Удалить тест
 - D) Изменить требования
78. Как составить отчёт о восстановлении работоспособности ПО?
- A) Описать причину сбоя, предпринятые действия, результаты проверки
 - B) Только указать время восстановления
 - C) Не составлять
 - D) Отправить устное сообщение
79. Что такое план тестирования?
- A) Документ, описывающий стратегию, ресурсы, расписание и объём тестирования

- B) Исходный код тестов
 - C) Отчёт о дефектах
 - D) Требования
80. Как оценить риски при планировании тестирования?
- A) Идентифицировать вероятные сбои и их влияние, назначить приоритеты
 - B) Не оценивать
 - C) Только по времени
 - D) Только по бюджету
81. Как тестировать производительность приложения?
- A) Использовать нагрузочные инструменты, измерять время ответа, пропускную способность
 - B) Только визуально
 - C) Только модульными тестами
 - D) Только ручным тестированием
82. Как интерпретировать результаты нагрузочного тестирования?
- A) Сравнить с SLA, выявить узкие места, оценить масштабируемость
 - B) Только посмотреть графики
 - C) Не анализировать
 - D) Только количество ошибок
83. Что такое SLA в контексте производительности?
- A) Соглашение об уровне обслуживания (например, время ответа < 200 мс)
 - B) Система логирования
 - C) Инструмент нагрузочного тестирования
 - D) Отчёт о дефектах
84. Какой инструмент используется для мониторинга производительности приложения в реальном времени?
- A) Prometheus + Grafana
 - B) JUnit
 - C) Selenium
 - D) Postman
85. Что такое профилирование (profiling) приложения?
- A) Измерение потребления ресурсов (CPU, память) отдельными частями кода
 - B) Функциональное тестирование
 - C) Тестирование безопасности
 - D) Документирование
- Раздел 5. Управление тестированием и работа в команде (86–100)
86. Какие статусы дефекта существуют в жизненном цикле?
- A) Новый, В работе, Исправлен, Проверен, Закрыт
 - B) Только Открыт/Закрыт
 - C) Только Важный/Неважный
 - D) Только Найден/Не найден
87. Какой инструмент управления дефектами наиболее популярен?
- A) Jira
 - B) Postman
 - C) Selenium
 - D) JMeter
88. Как взаимодействовать с разработчиками при тестировании?
- A) Оформлять баг-репорты с чёткими шагами воспроизведения, приоритетом, скриншотами
 - B) Устно сообщать
 - C) Игнорировать
 - D) Не сообщать
89. Что такое баг-репорт?
- A) Документ, содержащий описание дефекта, шаги воспроизведения, ожидаемый и фактический результат
 - B) Отчёт о тестировании
 - C) План тестирования

- D) Требование
90. Какие инструменты командной работы над проектом ПО вы знаете?
- A) Jira, Trello, YouTrack, Slack
 - B) Selenium, Postman
 - C) JUnit, PyTest
 - D) Git, SVN
91. Как вносить изменения в скрипты автоматизированных тестов?
- A) Использовать Git для контроля версий, создавать ветки, делать pull request
 - B) Изменять напрямую на сервере
 - C) Не вносить
 - D) Удалять старые
92. Что такое шаблон теста?
- A) Готовая структура тестового метода, которую можно адаптировать
 - B) Исходные данные
 - C) Ожидаемый результат
 - D) Отчёт
93. Как получить обновлённую версию ПО для тестирования?
- A) Из системы контроля версий (git pull) или из CI/CD артефакта
 - B) Попросить разработчика скинуть архив
 - C) Не обновлять
 - D) Копировать с продакшена
94. Как определить, нужно ли проводить регрессионное тестирование?
- A) Оценить масштаб изменений, влияние на другие модули, критичность изменяемого функционала
 - B) Всегда проводить
 - C) Никогда не проводить
 - D) Только по требованию заказчика
95. Как выбрать оптимальный перечень тестов для регрессии?
- A) Включить тесты на изменённые модули и тесты на критическую функциональность
 - B) Запускать все тесты
 - C) Запускать только новые тесты
 - D) Не запускать ничего
96. Как выполнить тесты, выявившие дефект, после его исправления?
- A) Запустить их снова, проверить, что дефект устранён и не появились новые
 - B) Игнорировать
 - C) Удалить
 - D) Переписать
97. Что такое регрессионный тестовый набор (regression suite)?
- A) Набор тестов, предназначенный для проверки того, что изменения не нарушили старую функциональность
 - B) Набор новых тестов
 - C) Набор тестов производительности
 - D) Набор тестов безопасности
98. Как часто следует запускать регрессионные тесты?
- A) После каждого значительного изменения кода или перед релизом
 - B) Один раз в год
 - C) Никогда
 - D) Только при сбое
99. Какая информация должна быть в отчёте о регрессионном тестировании?
- A) Количество пройденных/упавших тестов, список дефектов, оценка рисков
 - B) Только количество тестов
 - C) Только имена разработчиков
 - D) Исходный код
100. Что такое тест-план?
- A) Документ, определяющий цели, стратегию, ресурсы и расписание тестирования
 - B) Исходный код тестов
 - C) Отчёт о дефектах

- D) Инструмент автоматизации

Критерии оценивания:

- 5 баллов выставляется, если правильные ответы даны на 85-100% тестовых заданий
- 4 балла выставляется студенту, если правильные ответы даны на 65-84% тестовых заданий
- 3 балла выставляется студенту, если правильные ответы даны на 50-64% тестовых заданий
- 2 балла выставляется студенту, если правильные ответы даны на менее 50% тестовых заданий

Практические задания:

Задание 1

Тема: Разработка модульных тестов для функции расчёта факториала

Условие: Дана функция `factorial(n)`, которая вычисляет факториал целого неотрицательного числа.

Задание:

1. Написать модульные тесты с использованием PyTest (или JUnit для Java).
2. Покрыть граничные случаи: $n=0$, $n=1$, $n=5$, $n=10$.
3. Добавить тест на выброс исключения при отрицательном n .
4. Измерить покрытие кода тестами с помощью `pytest-cov`.
5. Создать отчёт о тестировании в формате HTML.

Задание 2

Тема: Использование моков (`mock`) для изоляции внешних зависимостей

Условие: Есть класс `UserService`, который вызывает внешний API для получения данных пользователя.

Задание:

1. Написать модульные тесты для `UserService`, используя `unittest.mock` (Python) или `Mockito` (Java).
2. Замокать HTTP-запросы так, чтобы тесты не обращались к реальному API.
3. Проверить, что метод `get_user` возвращает корректные данные при успешном ответе.
4. Проверить обработку ошибок (таймаут, ошибка сервера).
5. Убедиться, что метод был вызван с правильными параметрами.

Задание 3

Тема: Параметризованные тесты

Условие: Функция `is_prime(n)` проверяет, является ли число простым.

Задание:

1. Написать параметризованный тест в PyTest (декоратор `@pytest.mark.parametrize`).
2. Передать несколько наборов данных: простые числа (2, 3, 17), составные (4, 9, 15), граничные (1, 0, отрицательные).
3. Проверить, что функция возвращает правильный результат для всех случаев.
4. Добавить тест на выброс исключения для нецелых типов.

Задание 4

Тема: Создание библиотеки модульных тестов для класса «Калькулятор»

Условие: Разработан класс `Calculator` с методами `add`, `subtract`, `multiply`, `divide`.

Задание:

1. Создать модуль с тестами для всех методов.
2. Использовать фикстуры (`fixtures`) для инициализации экземпляра класса перед каждым тестом.
3. Обработать деление на ноль (ожидать исключение).
4. Организовать тесты в тест-сьют (`test suite`).
5. Запустить тесты из консоли и сгенерировать отчёт в формате JUnit XML.

Задание 5

Тема: Интеграция модульных тестов в CI/CD (GitLab CI)

Условие: Проект хранится в репозитории GitLab.

Задание:

1. Написать файл `.gitlab-ci.yml`, который автоматически запускает модульные тесты при каждом `push`.
2. Настроить этап сборки (`install dependencies`) и этап тестирования.
3. Добавить этап отчёта о покрытии кода (`coverage report`).
4. Настроить публикацию отчётов в артефакты.
5. Проверить, что пайплайн падает при падении хотя бы одного теста.

Раздел 2. Тестирование документации (6–7)

Задание 6

Тема: Тестирование пользовательской документации на соответствие функциональности

Условие: Дано руководство пользователя для веб-приложения «Библиотека».

Задание:

1. Проверить полноту документации: все ли функции, описанные в ТЗ, отражены в руководстве.
2. Выполнить инструкции из документации на тестовом стенде, выявить расхождения.
3. Составить отчёт о тестировании документации с указанием найденных несоответствий.
4. Зарегистрировать дефекты в системе трекинга (имитация в Jira или таблице).
5. Предложить исправления для документации.

Задание 7

Тема: Разработка программы и методики приемочного тестирования

Условие: Разрабатывается система «Электронная запись к врачу».

Задание:

1. На основе требований составить программу и методику испытаний (шаблон по ГОСТ 34).
2. Описать состав тестовых случаев для приемочного тестирования.
3. Указать критерии прохождения (успешность выполнения 100% критических тестов).
4. Провести альфа-тестирование с участием двух «внутренних пользователей» (одного группников).
5. Собрать результаты и оформить протокол приемочного тестирования.

Раздел 3. Интеграционное и системное тестирование (8–12)

Задание 8

Тема: API-тестирование REST сервиса с помощью Postman

Условие: Доступен публичный тестовый API (например, JSONPlaceholder или ReqRes).

Задание:

1. Создать коллекцию Postman для ресурса `/users`: GET, POST, PUT, PATCH, DELETE.
2. Настроить переменные окружения (`baseUrl`, `token`).
3. Написать тесты на статус-коды и структуру ответа (используя `pm.test`).
4. Добавить тест на создание пользователя с проверкой возвращённого ID.
5. Экспортировать коллекцию и окружение, приложить к отчёту.

Задание 9

Тема: Автоматизация API-тестов на Python с использованием `pytest` и `requests`

Условие: Используется тот же тестовый API.

Задание:

1. Написать скрипты для выполнения GET, POST, PUT, DELETE запросов.
2. Реализовать проверки статус-кодов и содержания ответа.
3. Использовать параметризацию для проверки разных ID.
4. Организовать тесты в виде функций с ассертами.
5. Запустить тесты из командной строки, получить отчёт в формате HTML (`pytest-html`).

Задание 10

Тема: Разработка автотеста с заглушками (`stub`) и имитаторами (`mock`)

Условие: Приложение заказывает данные о погоде с внешнего сервиса.

Задание:

1. Написать класс `WeatherService`, который вызывает API погоды.
2. В тестах замочать ответ внешнего сервиса так, чтобы он возвращал фиктивные данные.
3. Проверить, что метод `get_temperature` правильно извлекает температуру из мок-ответа.
4. Проверить обработку ошибок (например, сервер вернул 500).
5. Убедиться, что при недоступности сервиса выбрасывается кастомное исключение.

Задание 11

Тема: UI-тестирование веб-приложения с Selenium WebDriver

Условие: Разработано простое веб-приложение «Список задач» (или используйте тестовый сайт).

Задание:

1. Написать тесты для сценариев: добавление задачи, отметка о выполнении, удаление.
2. Использовать паттерн Page Object Model (POM) для организации кода.
3. Реализовать ожидания (WebDriverWait) для синхронизации.
4. Запустить тесты в Chrome и Firefox.
5. Сделать скриншоты при падении теста.

Задание 12

Тема: Настройка регрессионного тестирования в CI/CD

Условие: Проект с API-тестами и UI-тестами.

Задание:

1. Настроить GitLab CI (или Jenkins) для запуска всех тестов при создании Merge Request.
2. Разделить запуск на параллельные этапы: модульные тесты, API-тесты, UI-тесты.
3. Настроить публикацию отчётов Allure (или любого другого фреймворка).
4. Добавить правило: если падают критические тесты, MR нельзя мёржить.
5. Продемонстрировать успешный и неуспешный запуски.

Задание 13

Тема: Статический анализ безопасности кода (SAST)

Условие: Дан фрагмент кода на Python с потенциальными уязвимостями (SQL-инъекция, XSS, использование eval).

Задание:

1. Установить инструмент Bandit.
2. Запустить анализ кода, выявить уязвимости.
3. Классифицировать их по степени критичности.
4. Исправить код в соответствии с рекомендациями.
5. Сравнить отчёты до и после исправления.

Задание 14

Тема: Тестирование авторизации и двухфакторной аутентификации (имитация)

Условие: Веб-приложение с ролями (пользователь, администратор) и двухфакторной аутентификацией (TOTP).

Задание:

1. Написать тесты для проверки доступа к защищённым ресурсам в зависимости от роли.
2. Реализовать проверку, что пользователь без 2FA не может войти.
3. Имитировать генерацию TOTP-кода и проверить успешный вход.
4. Проверить блокировку после нескольких неудачных попыток.
5. Оформить отчёт о тестировании безопасности.

Задание 15

Тема: Нагрузочное тестирование с Apache JMeter

Условие: Тестовое веб-приложение (или имитация через простой сервер).

Задание:

1. Создать тест-план JMeter с запросом GET к главной странице.
2. Добавить Thread Group с 50 пользователями и 10 циклами.
3. Добавить слушатели: график результатов, отчёт агрегирования.
4. Измерить среднее время ответа, пропускную способность, процент ошибок.
5. Сравнить с заданным SLA (например, время ответа < 500 мс). Выводы оформить.

Задание 16

Тема: Стресс-тестирование и поиск точки отказа

Условие: То же приложение.

Задание:

1. Постепенно увеличивать количество пользователей (100, 200, 500).
2. Найти нагрузку, при которой система начинает давать ошибки.
3. Построить график зависимости времени ответа от нагрузки.
4. Проанализировать узкие места (логи, дампы).
5. Дать рекомендации по масштабированию.

Задание 17

Тема: Тестирование производительности с использованием Locust (Python)

Условие: Тестовое веб-API.

Задание:

1. Написать сценарий Locust, имитирующий поведение пользователя.
2. Запустить Locust в веб-режиме, провести тест с 100 пользователями.
3. Собрать метрики: время ответа, RPS, количество ошибок.
4. Экспортировать статистику в CSV.
5. Сравнить результаты с аналогичными, полученными в JMeter.

Задание 18

Тема: Выявление типовых уязвимостей кода веб-страниц в ручном режиме

Условие: Учебное веб-приложение с уязвимостями (например, Damn Vulnerable Web Application).

Задание:

1. Обнаружить XSS-уязвимость: ввести скрипт в поле ввода.
2. Найти SQL-инъекцию: ввести ' OR '1'='1 в поле логина.
3. Проверить возможность обхода авторизации через изменение параметров URL.
4. Составить отчёт о найденных уязвимостях с шагами воспроизведения.
5. Предложить способы защиты.

Задание 19

Тема: Комплексное тестирование безопасности веб-приложения

Условие: То же приложение.

Задание:

1. Использовать инструмент OWASP ZAP для автоматизированного сканирования.
2. Запустить активное сканирование, проанализировать отчёт.
3. Выявить критические уязвимости (например, отсутствие CSRF-токенов).
4. Вручную подтвердить найденные уязвимости.
5. Подготовить отчёт с рекомендациями по устранению.

Задание 20

Тема: Работа с системой контроля дефектов (Jira)

Условие: Доступна учебная Jira (или имитация).

Задание:

1. Создать проект, настроить типы задач: Bug, Task, Improvement.
2. Завести 3 баг-репорта с различной критичностью (Blocker, Critical, Minor).
3. Указать шаги воспроизведения, ожидаемый и фактический результат.
4. Прикрепить скриншоты.
5. Провести жизненный цикл дефекта: Open → In Progress → Fixed → Closed.

Задание 21

Тема: Настройка автоматизированного логирования и мониторинга тестов

Условие: Существующий набор тестов (любой).

Задание:

1. Настроить логирование в тестах с использованием библиотеки logging (Python) или log4j (Java).
2. Сохранять логи в файл с ротацией.
3. В CI/CD добавить этап сбора логов и архивации.
4. Настроить отправку уведомлений в Telegram/Slack при падении критических тестов.
5. Проанализировать логи после прогона и найти причины падения (имитация).

Задание 22

Тема: Регрессионное тестирование: отбор тестов и автоматизация

Условие: В приложении изменили модуль «Корзина».

Задание:

1. Определить, какие тесты должны быть включены в регрессионный набор.
2. Написать скрипт, который запускает только регрессионные тесты (по метке).
3. Использовать маркировку в PyTest (@pytest.mark.regression).
4. Настроить запуск регрессии в CI после каждого коммита в основную ветку.
5. Оформить отчёт о результатах регрессионного тестирования.

Задание 23

Тема: Интеграция тестов в пайплайн GitLab CI с динамическими тестовыми окружениями

Условие: Проект микросервисной архитектуры.

Задание:

1. Написать `.gitlab-ci.yml` с этапами: `build`, `test`, `deploy`.
2. На этапе `test` поднимать тестовое окружение с помощью `Docker Compose`.
3. Запускать API-тесты и UI-тесты.
4. После тестов автоматически разрушать окружение.
5. Настроить отчёт о покрытии кода (используя `pytest-cov` и `coverage.py`).

Задание 24

Тема: Восстановление автоматизированных тестов после сбоя

Условие: Тест упал из-за временной недоступности внешнего сервиса.

Задание:

1. Добавить в тесты механизм повторных попыток (`retry`) с экспоненциальной задержкой.
2. Настроить логирование факта повтора.
3. Проверить, что после восстановления сервиса тест проходит.
4. Смоделировать сбой (отключить интернет на время).
5. Написать отчёт о восстановлении.

Критерии оценивания:

- 5 баллов выставляется, если правильные ответы даны на 85-100% практических заданий

- 4 балла выставляется студенту, если правильные ответы даны на 65-84% практических заданий

- 3 балла выставляется студенту, если правильные ответы даны на 50-64% практических заданий

- 2 балла выставляется студенту, если правильные ответы даны на менее 50% практических заданий

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций состоит из текущего контроля.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации и учитываются при оценивании знаний, умений, навыков и (или) опыта деятельности.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

МДК 02.02 Автоматизация процессов тестирования программного обеспечения

Методические указания для студентов по освоению дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения являются частью рабочей программы дисциплины (РПД) (приложением к рабочей программе).

РПД – рабочая программа, утвержденная директором колледжа для изучения дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения. Она определяет цели и задачи дисциплины, формируемые в ходе ее изучения компетенции и их компоненты, содержание изучаемого материала, виды занятий и объем выделяемого учебного времени, а также порядок изучения и преподавания дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения.

Для самостоятельной учебной работы студента важное значение имеют разделы «Структура и содержание дисциплины (модуля)» и «Учебно-методическое и информационное обеспечение дисциплины (модуля)». В первом указываются разделы и темы изучаемой дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения, а также виды занятий и планируемый объем (в академических часах), во втором – рекомендуемая литература и перечень ресурсов информационно-телекоммуникационной сети «Интернет».

Для подготовки к текущему контролю студенты могут воспользоваться оценочными средствами, представленными в Приложении 1 к рабочей программе дисциплины.

1. Описание последовательности действий студента

Приступая к изучению дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения необходимо в первую очередь ознакомиться содержанием РПД, где в разделе «Структура и содержание дисциплины (модуля)» приведено общее распределение часов аудиторных занятий и самостоятельной работы по темам дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения.

Залогом успешного освоения дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения является регулярное посещение занятий и выполнение предусмотренных программой заданий. Пропуск одного, а тем более нескольких занятий может осложнить освоение разделов курса.

Лекции имеют целью дать систематизированные основы научных знаний по содержанию дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения. При изучении и проработке теоретического материала необходимо:

– повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;

– при самостоятельном изучении теоретической темы подготовить конспект, используя рекомендованные в РПД литературные источники и электронные образовательные ресурсы.

Практические занятия проводятся с целью углубления и закрепления знаний, полученных на лекциях и в процессе самостоятельной работы с учебной литературой.

В процессе практического занятия, как вида учебных занятий, обучающиеся выполняют одно или несколько практических заданий под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

Выполнение обучающимся практических работ проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений;
- углубления теоретических знаний в соответствии с заданной темой;
- формирования умений применять теоретические знания при решении поставленных задач;
- развития профессиональных компетенций у обучающихся;
- развития творческой инициативы, самостоятельности, ответственности и организованности.

Выполнение обучающимися практических заданий направлено на:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплины;
- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;
- выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

При подготовке к практическому занятию необходимо изучить или повторить лекционный материал по соответствующей теме.

2. Самостоятельная работа студента

Самостоятельная работа студента – самостоятельная учебная деятельность студента, организуемая колледжем и осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- воспитание самостоятельности, как личностного качества будущего специалиста.

Самостоятельная работа студента по дисциплине выполняется:

- самостоятельно вне расписания учебных занятий;
- с использованием современных образовательных технологий;
- работа со специальной литературой для подготовки к тестовым, практическим заданиям.

3. Рекомендации по работе с литературой и источниками

Работу с литературой следует начинать с анализа РПД, содержащей список основной и дополнительной литературы, а также знакомства с учебно-методическими разработками.

В случае возникновения затруднений в понимании учебного материала следует обратиться к другим источникам, где изложение может оказаться более доступным.

Работа с литературой не только полезна как средство более глубокого изучения дисциплины МДК 02.02 Автоматизация процессов тестирования программного обеспечения, но и является неотъемлемой частью профессиональной деятельности будущего выпускника.