

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ**

МДК.05.03 Тестирование информационных систем

программы подготовки специалистов среднего звена /
квалифицированных рабочих и служащих

09.02.07 Информационные системы и программирование

Форма обучения: *очная*

Находка 2026

Рабочая программа учебной дисциплины *МДК.05.03 Тестирование информационных систем* разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности / профессии *09.02.07 Информационные системы и программирование*, утвержденного приказом Минобрнауки России от «09» декабря 2016 г. № 1547, примерной образовательной программой.

Разработчик(и): *Лихацкий М.Б. преподаватель ОСПО филиала ФГБОУ ВО «ВВГУ» в г. Находке*

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 10 от «29» июня 2026 г.

Председатель ЦМК _____ *В.В Плахотина*
подпись

Содержание

- 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ
УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ
УЧЕБНОЙ ДИСЦИПЛИНЫ**

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1 Место дисциплины в структуре основной образовательной программы

Учебная дисциплина *МДК.05.03 Тестирование информационных систем* является частью учебного цикла основной образовательной программы (далее ООП) в соответствии с ФГОС СПО по специальности/профессии *09.02.07 Информационные системы и программирование*.

1.2 Цель и планируемые результаты освоения дисциплины

По итогам освоения дисциплины, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

Код компетенции	Умения	Знания
ОК 1 ОК 2	Обрабатывать текстовую и числовую информацию.	Назначение и виды информационных технологий, технологии сбора, накопления, обработки, передачи и распространения информации.
ОК 4 ОК 5	Применять мультимедийные технологии обработки и представления информации.	Состав, структуру, принципы реализации и функционирования информационных технологий.
ОК 9 ОК 10	Обрабатывать экономическую и статистическую информацию, используя средства пакета прикладных программ.	Базовые и прикладные информационные технологии
ПК 5.4 ПК 5.5	Собирать исходные данные для проведения тестирования. Выполнять тестирование (функциональное, регрессионное, интеграционное, нагрузочное) согласно тест-плану. Регистрировать и классифицировать обнаруженные дефекты, оформлять баг-репорты Разрабатывать тест-кейсы и чек-листы на основе требований заказчика и технического задания. Разрабатывать методики проведения испытаний. Анализировать результаты и составлять итоговые отчеты.	Инструментальные средства тестирования; виды и уровни тестирования; жизненный цикл дефекта; классификация дефектов по серьезности и приоритету. Техники тест-дизайна (классы эквивалентности, граничные значения, таблицы принятия решений); методологии разработки ПО и их влияние на процесс тестирования.

2 Структура и содержание учебной дисциплины

2.1 Объем учебной дисциплины и виды учебной работы

Таблица 2 - Распределение трудоемкости дисциплины по видам учебной работы

Вид учебной работы	Объем часов
Объем образовательной программы учебной дисциплины	66
в том числе:	
– теоретическое обучение	30
– практические занятия <i>(если предусмотрено)</i>	34
– самостоятельная работа	
– консультации	
– промежуточная аттестация – <i>(дифференцируемый зачет)</i>	2

2.2 Тематический план и содержание учебной дисциплины «МДК.05.03 Тестирование информационных систем»

Таблица 3 - Тематический план дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Тема 1.1 Основные понятия организации тестирования.	Содержание учебного материала: Понятие тестирования. Основные определения. Роль тестирования в жизненном цикле информационной системы. Цели и задачи тестирования. Концепция тестирования. Качество программного продукта и тестирование. Организация тестирования. Различные технологии тестирования: Agile – гибкое тестирование, "разработка через тестирование" - XP-программирование. Планирование тестирования. Фазы тестирования. Документирование тестирования. Оценка полноты тестирования.	10	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 5.4 ПК 5.5
	Практическая работа №1: Метод внедрения «агентов» тестирования (Операторы протоколирования выполнения программы). Пошаговое выполнение программы. Контрольные точки. Разбор и анализ примеров. Практическая работа №2: Критерии выбора тестов. Структурные критерии. Модель тестирования «Белого ящика». Примеры применения критериев тестирования веток, путей графа программы. Стохастические критерии тестирования.	11	
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Тема 1.2. Методы, виды и принципы тестирования программных продуктов	Содержание учебного материала: Виды тестирования. Структурное тестирование. Функциональное тестирование (модульное). Управляющий граф программы, ветви, пути. Тестирование модулей. Интеграционное тестирование. Регрессионное тестирование. Системное тестирование. Критерии выбора тестов. Требования к идеальному критерию. Классы критериев. Структурные критерии. Функциональные критерии. Стохастические критерии.	10	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 5.4 ПК 5.5
	Практическая работа №3: «Восстановление данных», «Восстановление работоспособности системы», «Создание резервной копии базы данных»	11	
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
Тема 1.3 Автоматизация тестирования. Инструментарий тестирования	Содержание учебного материала: Методики и инструменты автоматизированного тестирования. Нагрузочное тестирование. Тестирование безопасности. Планирование тестирования. Издержки тестирования В том числе практических занятий и лабораторных работ	10	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 5.4 ПК 5.5
	Практическая работа №4: «Сбор информации об ошибках. Формирование отчетов об ошибках». «Выявление и устранение ошибок программного кода информационных систем». «Выполнение обслуживания информационной системе в соответствии с пользовательской документацией».	12	
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
Промежуточная аттестация - дифференцированный зачет		2	

ИТОГО часов		66	
Теоретические занятия		30	
Практические занятия		34	
Самостоятельная работа		0	
Дифференцированный зачет		2	

3 Условия реализации учебной дисциплины

3.1 Для реализации программы учебной дисциплины «МДК.05.03 Тестирование информационных систем» образовательной организации, предусмотрено наличие следующих специальных помещений:

Для реализации программы учебной дисциплины предусмотрено наличие следующих специальных помещений:

Компьютерный класс информационных технологий: количество посадочных мест - 28 шт., доска меловая - 1 шт., стол для преподавателя - 1 шт., стул для преподавателя - 1 шт.; учебно-методические пособия, дидактические и презентационные материалы

Компьютерный класс имеет следующее оборудование:

Учебно-лабораторное оборудование: Процессор Intel Количество ядер - 2 . Объем кэш-памяти - 8Мб • Базовая частота процессора - 2,0 ГГц. Объем установленной памяти - 4Гб Накопитель твердотельный объемом 512 видео карта - встроенная. Максимальное разрешение - 1920x1080 пикс. Диагональ монитора 23" количество мониторов 1 шт. Разрешение: FHD (1920x1080 пикс.) клавиатура мышь ОС Microsoft Windows 7 Pro Мультимедийный комплект в составе: проектор Sanyo, крепление Kromax, экран Lumien Eco Picture: Технология проецирования изображения - микрозеркала на полупроводниковом чипе. Базовое разрешение проектора: 1280x800 пикс. Входы - VGA, HDMI 2 шт., Порты - RS-232C, USB 2 шт., LAN Размер полотна: ширина 200 см, высота 127 см. ПО: Microsoft Office 2010, Пакет Adobe CC 2019, Pixologic Zbrush Academic Volume License, Adobe Substance Painter, Autodesk 3Ds max, Maya, Blender, Unity, Unreal Engine

3.2 Информационное обеспечение реализации программы

Для реализации программы учебной дисциплины библиотечный фонд ВВГУ укомплектован печатными и электронными изданиями.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья обеспечены печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Основная литература:

1. Бизяев, А.А. Информационные технологии: практикум: [Электронный ресурс]/А.А. Бизяев, К.А. Куратов ; Новосибирский государственный технический университет. - Новосибирск: Новосибирский государственный технический университет, 2022. - 96 с.: ил., табл. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=575330>

2. Информационные технологии [Электронный ресурс] : метод. рекомендации по самостоят. изучению дисциплины / Г.И. Попов, С.И. Бажинов, В.Г. Конюхов, Е.Н. Яшкина, В.А. Пирогов .- М. : РГУФКСМиТ, 2022 .-167 с. : ил. - Режим доступа: <https://lib.rucont.ru/efd/671375>

3. Информационные технологии в профессиональной деятельности [Электронный ресурс] / Т.В. Марзаева .— Улан-Удэ : Бурятский государственный университет, 2025 .- 71 с. -Режим доступа: <https://lib.rucont.ru/efd/348195>

4. Информационные технологии: учеб. пособие [Электронный ресурс]/ Л.Г. Гагарина, Я.О. Теплова, Е.Л. Румянцева, А.М. Баин / под ред. Л.Г. Гагариной. — Москва: ИД «ФОРУМ»: ИНФРА- М, 2021. — 320 с. -Режим доступа: <https://znanium.com/catalog/product/471464>.

5. Филимонова, Е.В. Информационные технологии в профессиональной деятельности : учебник / Филимонова Е.В. — Москва : КноРус, 2022. — 482 с. — (СПО). — ISBN 978-5-406-04887-0. — URL: <https://book.ru/book/922139> (дата обращения:

06.08.2024). — Текст : электронный.

6. Хныкина, А.Г. Информационные технологии [Электронный ресурс] : учеб. пособие / Т.В. Минкина, А.Г. Хныкина .— Ставрополь : изд-во СКФУ, 2023 .— 126 с. : ил. — Режим доступа: <https://lib.rucont.ru/efd/671178>

Дополнительная литература:

1. Богданова, С.В. Информационные технологии [Электронный ресурс]: учебное пособие / С.В. Богданова, А.Н. Ермакова. - Ставрополь: Сервисшкола, 2024. - 211 с. - Режим доступа: <https://znanium.com/catalog/product/514867>

2. Самойленко, А.П. Информационные технологии статистической обработки данных [Электронный ресурс] : учеб. пособие / О.А. Усенко, Южный федеральный ун-т, А.П. Самойленко .— Ростов н/Д. : Изд-во ЮФУ, 2024 .— 128 с. : ил. — ISBN 978-5-9275-2521-8 .— Режим доступа: <https://lib.rucont.ru/efd/692393>

3. Суханова, О.Н. Информационные технологии [Электронный ресурс] : лаб. практикум / О.В.

4. Ментюкова, О.Н. Суханова .— Пенза : РИО ПГСХА, 2022 .- 116 с. : ил. - Режим доступа: <https://lib.rucont.ru/efd/323727>.

Интернет-ресурсы:

1. www.informatics.mccme.ru/moodle/ - дистанционная подготовка по информатике.
2. www.ixbt.com - сайт о высоких технологиях, новости индустрии из мира компьютерного «железа», тестовые испытания и обзоры оборудования. Конференция. Платформа ПК. Цифровой звук. Цифровое фото. Мобильные устройства. Форум.
3. www.lib.ru/INTERMET/termwww.txt - толковый словарь терминов.
4. www.computerra.ru - новости про компьютеры, железо, новые технологии, информационные технологии. Рубрики: В поле зрения; Компьютерралаб; Дюжина полезных программ; Реклама на сайте; Блоги; Наука и жизнь. Архив номеров журнала.
5. www.novtex.ru/IT/arhiv.htm
6. <https://www.planttext.com/>
7. <https://kroki.io/#>
8. <https://www.canva.com/ru ru/>
9. <https://prezi.com/>
10. www.jir.ru/ - электронный научный журнал.

4. Контроль результатов освоения учебной дисциплины

Результаты обучения	Критерии оценки	Методы оценки
- актуальный профессиональный и социальный контекст, в котором приходится работать и жить; - основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; - алгоритмы выполнения работ в профессиональной и смежных областях; - методы работы в профессиональной и смежных сферах; - структуру плана для решения задач; - порядок оценки результатов решения задач профессиональной деятельности - основные методы и средства сбора, обработки, хранения, передачи и накопления информации; - технологию поиска информации в сети Интернет; - номенклатура информационных источников применяемых в профессиональной деятельности; приемы структурирования информации; - формат оформления результатов поиска информации - содержание актуальной нормативно-правовой документации;	Уровень правильных ответов при письменном и устном контроле, четкость изложения материала. Быстрота ориентации в представляемом материале, быстрота реакции на вопросы. Правильность, полнота выполнения заданий, точность формулировок, точность расчетов, соответствие требованиям.	устный опрос, практические работы

- современная научная и профессиональная терминология; - возможные траектории профессионального развития и самообразования - психологические основы деятельности коллектива, психологические особенности личности; - основы проектной деятельности - особенности социального и культурного контекста; - правила оформления документов и построения устных сообщений - назначение, состав, основные характеристики организационной и компьютерной техники;		
- основные компоненты информационных систем, принципы внедрения информационных систем, организацию межсетевого взаимодействия; - назначение и принципы использования системного и прикладного программного обеспечения; - принципы защиты информации от несанкционированного доступа; - правовые аспекты использования информационных технологий и программного обеспечения; - основные понятия	Оптимальность выбора способов действий, методов, последовательностей действий. Соответствие требованиям инструкций, регламентов, рациональность действий. Точность распознавания проблемных ситуаций в различных контекстах; Адекватность анализа сложных ситуаций при решении задач профессиональной деятельности, эффективность поиска. Точность определения источников нужных ресурсов. Разработка детального плана действий, правильность оценки рисков на каждом шагу; Точность оценки плюсов и минусов полученного результата, своего плана и его реализации, предложение критериев оценки и рекомендаций по улучшению плана	Текущий контроль: экспертное наблюдение и оценка в процессе выполнения заданий для практических занятий;

автоматизированной обработки информации;		
---	--	--

Для оценки достижения запланированных результатов обучения по дисциплине разработаны контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации, которые прилагаются к рабочей программе дисциплины.

Для оценки достижения запланированных результатов обучения по дисциплине разработаны контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации, которые прилагаются к рабочей программе дисциплины.

Оценка результатов обучения производится при помощи бально-рейтинговой системы.

Основные критерии оценки:

- конспектирование 0,5 баллов;
- активность на уроке 0,5 баллов;
- посещаемость занятий 0,5 баллов;
- выполнение индивидуальных домашних заданий 1 задание- 1 балл;
- выполнение самостоятельной работы 1 балл;
- лабораторная работа 3 балла;
- контрольная работа 1 задание -1 балл;
- участие в научно-исследовательской работе 15 баллов;
- дифференцированный зачет 20 баллов.

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля и промежуточной аттестации
по учебной дисциплине

МДК.05.03 Тестирование информационных систем

программы подготовки специалистов среднего звена /
квалифицированных рабочих и служащих

09.02.07 Информационные системы и программирование

Форма обучения: очная

Оценочные средства для проведения текущего контроля и промежуточной аттестации по учебной дисциплине *МДК.05.03 «Тестирование информационных систем»* разработаны в соответствии с требованиями ФГОС СПО по специальности / профессии *09.02.07 Информационные системы и программирование*, утвержденного приказом Минобрнауки России от «09» декабря 2016 г. № 1547, примерной образовательной программой.

Разработчик(и): *Лихацкий М.Б. преподаватель ОСПО филиала ФГБОУ ВО «ВВГУ» в г. Находке*

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 10 от «29» июня 2026 г.

Председатель ЦМК _____ *В.В Плахотина*
подпись

1 Общие сведения

Фонд оценочных средств (далее – ФОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины *МДК.05.03 «Тестирование информационных систем»* ФОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта (с использованием оценочного средства - *тестирование*)

2 Планируемые результаты обучения по дисциплине, обеспечивающие результаты освоения образовательной программы

Код ОК, ПК	Код результата обучения	Наименование результата обучения ¹
ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 5.4 ПК 5.5	31	Виды, уровни и методы тестирования ПО, жизненный цикл дефекта
	32	Методики тест-дизайна (классы эквивалентности, граничные значения), принципы построения тест-планов и чек-листов, принципы реализации и функционирования информационных систем
	33	Методы оценки качества программного продукта, метрики тестирования
	34	Инструментальные средства тестирования, а также стандарты оформления тестовой документации.
	У1	Оформлять тестовую документацию, работать с баг-трекером
	У2	Фиксировать результаты тестирования, составлять отчеты о проведенных испытаниях
	У3	Анализировать статистику дефектов, выявлять тренды качества и рассчитывать показатели покрытия тестами

3 Соответствие оценочных средств контролируемым результатам обучения

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель ² овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
Тема 1.1 Основные понятия организации и тестирования.	31	Способность понимать цели, задачи, основные понятия (дефект, ошибка, отказ) и роль тестирования в жизненном цикле разработки ПО.	<i>Устный опрос (п 5.1 вопросы 1-5) Реферат (п 5.2 темы 1-10) Практическая работа №1-2 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	32	Способность знать классификацию видов тестирования (по уровням: модульное, интеграционное, системное; по целям: функциональное, нефункциональное).		
	У1	Способность грамотно классифицировать виды тестирования и выбирать стратегию тестирования в зависимости от проекта и требований.		
Тема 1.2. Методы, виды и принципы тестирования программных продуктов	31	Способность знать техники тест-дизайна (классы эквивалентности, граничные значения, таблицы решений) и правила разработки тестовых сценариев.	<i>Устный опрос (п 5.1 вопросы 11-20) Реферат (п 5.2 темы 16-30) Практическая работа №3 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У2	Способность разрабатывать тест-кейсы, чек-листы и тест-планы, основываясь на требованиях заказчика и техническом задании.		
	У3	Способность анализировать полученные результаты, выявлять и локализовывать ошибки в логике программного продукта.		
Тема 1.3 Автоматизация тестирования. Инструментарий тестирования	У1	Способность применять инструментальные средства автоматизации	<i>Устный опрос (п 5.1 вопросы 11-18) Реферат (п 5.2 темы 21-30) Практическая работа №4 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У2	Способность обрабатывать и систематизировать результаты тестирования, составлять отчеты о найденных дефектах и анализировать статистику ошибок.		

4. Описание процедуры оценивания

Результаты обучения по дисциплине, уровень сформированности компетенций оцениваются по четырёх бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Текущая аттестация по дисциплине проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по дисциплине результатов обучения, обеспечивающих результаты освоения образовательной программы в целом. *Оценка на зачете выставляется с учетом оценок, полученных при прохождении текущей аттестации.*

Критерии оценивания устного ответа

(оценочные средства: *собеседование, устное сообщение*)

5 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания письменной работы

(оценочные средства: *реферат, конспект, контрольная работа, расчетно-графическая работа, доклад (сообщение), в том числе выполненный в форме презентации, творческое задание*).

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы. Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

Критерии оценивания тестового задания

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Количество правильных ответов	91 % и $\geq$	от 81% до 90,9 %	не менее 70%	менее 70%

Критерии выставления оценки студенту на зачете

(оценочные средства: устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных разноуровневых задач и заданий, комплексная расчетно-графическая работа, творческое задание, кейс-задача, портфолио, проект и т.п.)

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного

	материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«не зачтено» / «неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

5. Примеры оценочных средств для проведения текущей аттестации

5.1 Вопросы для устного опроса

1. Что такое тестирование программного обеспечения и каковы его основные цели?
2. Объясните разницу между верификацией и валидацией в контексте тестирования ПО.
3. Что такое ошибка (defect/bug), отказ (failure) и инцидент (incident) в тестировании?
4. Какие основные принципы тестирования вы знаете? Опишите **至少3** принципа.
5. Что такое тестовый случай (test case) и из каких элементов он состоит?
6. Что такое тест-план и какую роль он играет в процессе тестирования?
7. Объясните понятия "покрытие требований" (requirements coverage) и "покрытие кода" (code coverage).
8. Какие основные роли в команде тестирования вы знаете?
9. Что такое жизненный цикл дефекта (bug lifecycle)?
10. Как документация связана с процессом тестирования?
11. Какие основные уровни тестирования вы знаете? Опишите каждый из них.
12. В чем разница между функциональным и нефункциональным тестированием?
13. Что такое регрессионное тестирование и когда оно проводится?
14. Объясните разницу между тестированием "черного ящика" и "белого ящика".
15. Какие методы тестирования "черного ящика" вы знаете? Опишите 2 метода).
16. Что такое тестирование пользовательского интерфейса (UI testing)?
17. Какие виды нефункционального тестирования вы знаете? (например, нагрузочное, безопасность)
18. Что такое исследовательское тестирование (exploratory testing)?

5.2 Темы рефератов

Тема 1.1. Основные понятия организации тестирования

1. Понятие тестирования: цели, задачи и его роль в жизненном цикле информационной системы.
2. Качество программного продукта и его взаимосвязь с тестированием.
3. Современные технологии тестирования: Agile, TDD, XP (экстремальное программирование) и их особенности.
4. Планирование тестирования: этапы, структура и содержание тест-плана.

5. Документирование процесса тестирования: виды отчетов и требования к их оформлению.
6. Метод «внедрения агентов» тестирования: операторы протоколирования выполнения программы.
7. Пошаговое выполнение программы и использование контрольных точек при отладке.
8. Критерии выбора тестов: структурные, функциональные, стохастические и их применение.
9. Модель тестирования «белого ящика»: принципы и примеры применения.
10. Тестирование веток, путей графа программы: построение графа и расчет покрытия.

Тема 1.2. Методы, виды и принципы тестирования программных продуктов

11. Виды тестирования: функциональное, модульное, интеграционное, регрессионное и системное.
12. Структурное тестирование: понятие, методы и инструменты.
13. Управляющий граф программы: вершины, ветви, пути и его роль в тестировании.
14. Тестирование модулей и интеграционное тестирование: отличия и особенности.
15. Классы эквивалентности: метод черного ящика и правила разбиения.
16. Анализ граничных значений: проверка на границах допустимых диапазонов.
17. Тестирование «черного ящика»: стратегии выбора тестовых наборов.
18. Восстановление данных и работоспособности системы после сбоев.
19. Создание резервной копии базы данных и ее роль в надежности ИС.
20. Структурные, функциональные и стохастические критерии выбора тестов.

Тема 1.3. Автоматизация тестирования. Инструментарий тестирования

21. Автоматизация тестирования: цели, преимущества и недостатки.
22. Нагрузочное тестирование: цели, методы и основные инструменты.
23. Тестирование безопасности: виды уязвимостей и методы их обнаружения.
24. Инструментальные средства автоматизации тестирования: обзор и сравнение.
25. Выявление ошибок программного кода: основные методы и приемы.
26. Сбор информации об ошибках и формирование отчетов о них.
27. Обработка исключительных ситуаций и их влияние на качество тестирования.
28. Планирование работ по устранению найденных ошибок.

29. Выполнение обслуживания информационной системы в соответствии с пользовательской документацией.
30. Современные подходы к автоматизации регрессионного тестирования.

5.3 Примеры практических работ

Практическая работа №1

Тема 1.1. Метод внедрения «агентов» тестирования. (Операторы протоколирования выполнения программы). Пошаговое выполнение программы. Контрольные точки. Разбор и анализ примеров.

Цель: Научиться отслеживать выполнение кода, используя отладчик среды разработки и операторы протоколирования (агентов).

Задание: Создайте простую программу (на VBA или Python), которая считает сумму чисел от 1 до введенного N. Внедрите «агент» (логгер), который выводит текущее значение переменной на каждом шаге. Используя контрольные точки и пошаговое выполнение, определите, на каком шаге происходит выход за пределы допустимых значений (например, происходит переполнение переменной).

Пример выполнения (на VBA):

```
vba
Sub PracticalWork1()
    Dim N As Integer, i As Integer, Sum As Long
    N = InputBox("Введите N")
    Sum = 0

    ' Внедрение "агента" - оператора протоколирования
    Debug.Print "Начало цикла. Текущее значение Sum = " & Sum

    For i = 1 To N
        Sum = Sum + i
        ' Пошаговое выполнение и контрольная точка (F9) здесь покажет изменение переменной i
        Debug.Print "Шаг " & i & ": Sum = " & Sum
        ' Поставим контрольную точку, когда i = 10, чтобы проанализировать поведение
    Next i

    MsgBox "Итоговая сумма: " & Sum
End Sub
```

Как выполняется: Студент ставит точку останова (F9), нажимает F8 (пошагово), смотрит в окно «Locals» и видит, как меняются переменные. Задание: проанализировать, что произойдет, если ввести отрицательное N.

Практическая работа №2

Тема 1.1. Критерии выбора тестов. Структурные критерии. Модели тестирования «белого ящика». Примеры применения критериев тестирования веток, путей графа программы. Стохастические критерии тестирования.

Цель: Научиться строить граф потока управления (Control Flow Graph) и рассчитывать критерии покрытия (веток, путей) для тестирования «белого ящика».

Задание: Для следующего фрагмента кода постройте граф программы. Определите минимальное количество тест-кейсов, необходимых для достижения 100% покрытия веток, и 100% покрытия путей.

```
pascal
if (a > 0) then
    b = a * 2;
```

```

else
    b = a + 1;
end if;
for i = 1 to 5 do
    b = b + i;
end for;
print b;

```

Пример выполнения:

1. **Строим граф:** Вершина 1 (начало), 2 (условие $a > 0$), 3 (ветка $b = a * 2$), 4 (ветка $b = a + 1$), 5 (узел слияния), 6 (проверка цикла $i=1$ to 5), 7 (тело цикла $b = b + i$), 8 (конец цикла), 9 (вывод).
2. **Покрывтие операторов:** Нужно пройти по всем узлам. Тест 1 ($a > 0$) и Тест 2 ($a \leq 0$) покрывают все узлы.
3. **Покрывтие веток:** Нужно, чтобы обе ветки условия (Then/Else) выполнились хотя бы один раз. И цикл должен выполниться хотя бы один раз. Это все те же **2 теста** (например: $a=5$ и $a=-5$).
4. **Покрывтие путей:** Для полного покрытия путей нужно учесть, что цикл может не выполниться (0 раз), выполниться 1 раз, 2 раза... 5 раз. Это математически огромное количество. Делаем вывод: **Стохастический критерий** (случайный выбор тестов) не подходит; нужно использовать структурный критерий покрытия веток как оптимальный баланс между трудоемкостью и качеством.

Практическая работа №3

Тема 1.2. «Восстановление данных», «Восстановление работоспособности системы», «Создание резервной копии базы данных».

Цель: Отработать навыки создания резервной копии и восстановления системы после сбоя.

Задание: Создайте тестовую базу данных (Access или таблицу в Excel). Напишите скрипт/процедуру, которая делает резервную копию файла БД (создание копии с новым именем и датой). Затем симулируйте сбой (удалите или испортите файл) и продемонстрируйте процесс восстановления из резервной копии.

Пример выполнения (VBA для копирования и восстановления файла):

```

vba
Sub BackupDatabase()
    Dim sourcePath As String, backupPath As String
    sourcePath = "C:\Tests\MyDB.accdb"
    ' Создаем имя резервной копии с датой
    backupPath = "C:\Tests\Backup_" & Format(Now, "yyyy-mm-dd_hh-mm") & ".accdb"
    ' Копируем файл (создание копии БД)
    FileCopy sourcePath, backupPath
    MsgBox "Резервная копия создана: " & backupPath
End Sub

```

```

Sub RestoreDatabase()
    Dim backupPath As String, sourcePath As String
    backupPath = "C:\Tests\Backup_2023-10-27_10-30.accdb" ' Путь к последней копии
    sourcePath = "C:\Tests\MyDB.accdb"
    ' Удаляем испорченную базу
    Kill sourcePath
    ' Восстанавливаем из копии (Восстановление работоспособности)
    FileCopy backupPath, sourcePath
    MsgBox "База данных успешно восстановлена"
End Sub

```

Задание для студента: Описать действия, которые необходимо выполнить в корпоративной информационной системе для восстановления работоспособности, если резервная копия окажется поврежденной (например, проверить целостность файла, использование журналов транзакций).

Практическая работа №4

Тема 1.3. «Сбор информации об ошибках. Формирование отчетов об ошибках», «Выявление и устранение ошибок программного кода информационных систем», «Выполнение обслуживания информационной системы в соответствии с пользовательской документацией».

Цель: Научиться оформлять дефекты, исправлять ошибки и составлять пользовательскую документацию.

Задание: Вам предоставлен программный код (фрагмент). В нем содержится 2 логические ошибки.

Ваша задача:

1. "Протестировать" код (собрать информацию).
2. Оформить "Баг-Репорт" (Отчет об ошибке) по стандартному шаблону.
3. Исправить ошибки в коде.
4. Написать краткую инструкцию пользователя (памятку по обслуживанию) для этой функции.

Пример кода с ошибками (VBA):

```
vba
Sub CalcSalary()
    Dim Hours As Integer, Rate As Double, Sum As Double
    Hours = InputBox("Введите отработанные часы")
    Rate = InputBox("Введите ставку за час")
    ' Ошибка 1: Деление на ноль или пустую строку
    Sum = Hours * Rate ' Ошибка 2: Если пользователь вводит "abc", будет ошибка типа
    MsgBox "Зарплата: " & Sum
End Sub
```

Пример заполнения Баг-Репорта:

- **Заголовок:** Ошибка при вводе текстовых данных в поля "Часы" или "Ставка".
- **Окружение:** Windows 10, Office 2019.
- **Шаги воспроизведения:** 1) Запустить процедуру CalcSalary. 2) Ввести буквы "abc" в поле часов. 3) Нажать ОК.
- **Ожидаемый результат:** Программа должна выдать сообщение об ошибке ("Введите числовое значение") и не завершать работу.
- **Фактический результат:** Вылетает системная ошибка Type Mismatch. Программа зависает.
- **Приоритет:** Высокий (High).

Пример исправления (устранение ошибки):

```
vba
Sub CalcSalary()
    Dim Hours As Integer, Rate As Double, Sum As Double
    ' Сбор данных и проверка на числовой ввод
    On Error GoTo ErrorHandler
    Hours = Val(InputBox("Введите отработанные часы"))
    Rate = Val(InputBox("Введите ставку за час"))
    Sum = Hours * Rate
    MsgBox "Зарплата: " & Sum
    Exit Sub
ErrorHandler:
    MsgBox "Ошибка! Введены некорректные данные. Пожалуйста, введите числа."
End Sub
```

Задание по документации: Составить памятку для конечного пользователя: «При обслуживании системы введите данные в строгом формате чисел через запятую (например, 8,5) и не используйте посторонние символы».

6. Оценочные средства для проведения промежуточной аттестации

6.1 Вопросы для собеседования при проведении дифференцированного зачета

Тема 1.1. Основные понятия организации и тестирования

1. **Дайте определение термину «Тестирование программного обеспечения».** Чем тестирование отличается от отладки (debugging)?
2. **Что такое «дефект» (bug), «ошибка» (error) и «отказ» (failure)?** Опишите их взаимосвязь и приведите примеры.
3. **Какова цель и основные задачи тестирования на разных этапах жизненного цикла разработки ПО?** Расскажите о роли тестировщика в команде.

Тема 1.2. Методы, виды и принципы тестирования программных продуктов

4. **Перечислите основные виды тестирования по уровню проведения** (модульное, интеграционное, системное). В чем их принципиальное отличие?
5. **Объясните разницу между функциональным и нефункциональным тестированием.** Приведите примеры нефункциональных видов (нагрузочное, стрессовое, тестирование безопасности).
6. **Что такое «тест-дизайн»?** Опишите такие техники, как классы эквивалентности и анализ граничных значений. Приведите пример.
7. **Что такое «регрессионное тестирование»?** Когда и зачем оно проводится?
8. **Что такое позитивное и негативное тестирование?** Как правильно строить тест-кейсы для проверки негативных сценариев?
9. **Что такое тест-план и тест-кейс?** Из каких обязательных полей они состоят? Чем чек-лист отличается от тест-кейса?

Тема 1.3. Автоматизация тестирования. Инструментарий тестирования

10. **Каковы плюсы и минусы автоматизации тестирования?** В каких случаях автоматизация целесообразна, а в каких лучше использовать ручное тестирование?
11. **Что такое баг-трекер?** Назовите основные системы для отслеживания ошибок (Jira, Bugzilla, Redmine). Опишите жизненный цикл дефекта (Bug Lifecycle) от создания до закрытия.
12. **Какие инструменты автоматизации тестирования вы знаете?** Расскажите о назначении

инструментов для тестирования API (например, Postman) и веб-интерфейсов (например, Selenium).

13. Как собирается и обрабатывается статистика по дефектам? Какие метрики качества ПО существуют (плотность дефектов, покрытие тестами)?

Вопросы на профессиональные компетенции (ПК) и документацию:

14. Представьте, что вам выдали техническое задание на разработку информационной системы. Какие исходные данные вам необходимо собрать, чтобы начать составлять тестовую документацию (тест-план, чек-листы)?
15. Опишите порядок действий при обнаружении критического дефекта в модуле безопасности системы. Как вы его зарегистрируете, оформите документацию и передадите разработчикам? (Здесь проверяется понимание ПК 5.1, 5.2, 5.6, 6.3).

6.2 Тестовые материалы

Вариант 1

Часть 1. Выбор одного правильного ответа (вопросы 1–3)

- Какой метод тестирования предполагает проверку внутренней логики кода, ветвей и условий программы?
А) Метод «черного ящика»
Б) Метод «белого ящика»
В) Метод «серого ящика»
Г) Исследовательское тестирование
- Что является главной целью регрессионного тестирования?
А) Проверка удобства использования интерфейса
Б) Поиск новых ошибок в только что написанном коде
В) Убедиться, что новые изменения не сломали существующий функционал
Г) Тестирование системы под максимальной нагрузкой
- В рамках какой методологии разработки (указанной в программе) активно применяется "парное программирование" и "разработка через тестирование" (TDD)?
А) Каскадная модель (Waterfall)
Б) Экстремальное программирование (XP)
В) Спиральная модель
Г) Модель V-образная

Часть 2. Установление соответствия (вопросы 4–6)

- Установите соответствие между уровнем тестирования и его описанием.
Уровень: 1. Модульное, 2. Интеграционное, 3. Системное
Описание: А. Проверка взаимодействия нескольких модулей и API, Б. Проверка отдельных функций и методов кода, В. Полная проверка всей системы целиком.
- Установите соответствие между видом тестирования и его целью.
Вид: 1. Нагрузочное, 2. Функциональное, 3. Тестирование безопасности
Цель: А. Проверить корректность работы логики и бизнес-процессов, Б. Проверить устойчивость системы к взлому и утечкам данных, В. Проверить производительность системы

при большом количестве пользователей.

6. Установите соответствие между методом тестирования и его принципом.

Метод: 1. Классы эквивалентности, 2. Анализ граничных значений, 3. Покрытие кода

Принцип: А. Деление входных данных на группы, внутри которых поведение системы одинаково, Б. Проверка значений на границах допустимых диапазонов, В. Оценка процента исполненных операторов и ветвей программы.)

Часть 3. Установление последовательности (вопросы 7–9)

7. Расставьте в правильной последовательности этапы жизненного цикла дефекта (Bug Lifecycle):

1. Перепроверка тестировщиком (Retest)
2. Взятие дефекта в работу разработчиком (In Progress)
3. Обнаружение и регистрация дефекта (New)
4. Исправление дефекта (Fixed)
5. Закрытие дефекта (Closed)

8. Расставьте в правильном порядке этапы процесса отладки программы:

1. Устранение причины ошибки в коде
2. Воспроизведение ошибки
3. Написание отчета об ошибке (если найдена тестировщиком)
4. Анализ причин возникновения ошибки

9. Расставьте в правильной последовательности процесс тестирования:

1. Разработка тест-кейсов
2. Анализ требований и тест-планирование
3. Выполнение тестов и сравнение результатов
4. Составление отчетов и оформление баг-репортов

Часть 4. Открытые вопросы (вопросы 10–12)

10. Дайте определение тестированию и перечислите не менее трех техник тест-дизайна («черного ящика»).
11. В чем разница между «Верификацией» (Verification) и «Валидацией» (Validation)? Приведите примеры.
12. Что такое исследовательское тестирование (Exploratory testing)? Опишите, когда его целесообразно применять.

Вариант 2

Часть 1. Выбор одного правильного ответа (вопросы 1–3)

1. Какая техника тест-дизайна основана на проверке значений на стыках допустимых диапазонов (например, 0, 1, 9, 10)?
А) Классы эквивалентности
Б) Анализ граничных значений
В) Таблица решений
Г) Стохастические критерии
2. К какому виду нефункционального тестирования относится проверка восстановления работоспособности системы после сбоя (например, восстановление резервной копии)?
А) Тестирование безопасности
Б) Нагрузочное тестирование
В) Тестирование восстановления
Г) Функциональное тестирование

3. Что из перечисленного относится к инструментам автоматизации тестирования?
- А) Баг-трекер Jira (для ручной работы)
 - Б) Пошаговый отладчик в среде разработки
 - В) Специализированные средства, эмулирующие действия пользователя (например, Selenium)
 - Г) Текстовый редактор для написания документации

Часть 2. Установление соответствия (вопросы 4–6)

4. Установите соответствие между понятием и его определением.
Понятие: 1. Отказ (Failure), 2. Дефект (Bug), 3. Ошибка (Error)
Определение: А. Ошибка, допущенная человеком в коде или логике, Б. Недочет в самом продукте, обнаруженный при проверке, В. Проявление дефекта в реальной работе системы (сбой, вылет).
5. Установите соответствие между ролью в команде и её функцией.
Роль: 1. Тест-лид, 2. Тест-аналитик, 3. Автоматизатор (QA Automation)
Функция: А. Пишет тест-планы, распределяет задачи, отвечает за процесс, Б. Анализирует требования и разрабатывает тест-кейсы, В. Пишет скрипты для автотестов и поддерживает их.
6. Установите соответствие между видом тестирования и его задачами.
Вид: 1. Тестирование пользовательского интерфейса (UI), 2. Приемочное тестирование, 3. Интеграционное тестирование
Задача: А. Проверка интерфейса на соответствие макетам и удобство использования, Б. Проверка готовности системы к эксплуатации заказчиком, В. Проверка корректности взаимодействия модулей между собой.

Часть 3. Установление последовательности (вопросы 7–9)

7. Расставьте в правильной последовательности этапы планирования тестирования (создания тест-плана):
- 1. Определение критериев начала и окончания тестирования
 - 2. Определение целей, объема и рисков
 - 3. Разработка стратегии и выбор инструментов
 - 4. Согласование сроков и распределение ресурсов
8. Расставьте в правильной последовательности этапы проведения нагрузочного тестирования:
- 1. Анализ полученных результатов и метрик
 - 2. Подготовка тестовой среды и генераторов нагрузки
 - 3. Постановка целей и требований к производительности
 - 4. Проведение тестов с увеличением нагрузки
9. Расставьте в правильном порядке этапы работы с требованиями (покрытие требований):
- 1. Составление матрицы трассировки требований
 - 2. Написание тест-кейсов на каждое требование
 - 3. Анализ требований и выявление функциональности
 - 4. Выполнение тест-кейсов и оценка процента покрытия

Часть 4. Открытые вопросы (вопросы 10–12)

10. Опишите разницу между ручным и автоматизированным тестированием. Перечислите плюсы и минусы автоматизации.
11. Что такое «Покрытие требований» (Requirements Coverage) и как оно рассчитывается?
12. Назовите основные принципы тестирования (минимум 3) и раскройте суть любого одного из них на примере.

6.3 Ключи к тестовым заданиям для проведения промежуточной аттестации

Вариант 1:

1. Б (белого ящика)
2. В (убедиться, что изменения не сломали старый функционал)
3. Б (XP - экстремальное программирование)
4. 1-Б, 2-А, 3-В
5. 1-В, 2-А, 3-Б
6. 1-А, 2-Б, 3-В
7. 3, 2, 4, 1, 5
8. 2, 3, 4, 1
9. 2, 1, 3, 4
10. Тестирование — процесс проверки соответствия продукта требованиям. Техники: Классы эквивалентности, Анализ граничных значений, Таблица решений.
11. Верификация — "правильно ли строим продукт" (соответствие ТЗ). Валидация — "тот ли продукт строим" (соответствие потребностям пользователя).
12. Исследовательское тестирование — одновременное изучение, проектирование и выполнение тестов без жестких сценариев. Применяется при недостатке документации или для поиска скрытых багов.

Вариант 2:

1. Б (Анализ граничных значений)
2. В (Тестирование восстановления)
3. В (Selenium и аналоги)
4. 1-В, 2-Б, 3-А
5. 1-А, 2-Б, 3-В
6. 1-А, 2-Б, 3-В
7. 2, 3, 1, 4
8. 3, 2, 4, 1
9. 3, 1, 2, 4
10. Ручное — человек выполняет кейсы. Автоматическое — программа (скрипты). Плюсы авто: скорость, переиспользуемость, отсутствие усталости. Минусы: дорого писать, дорого поддерживать при изменении кода.
11. Покрытие требований — процент требований, проверенных хотя бы одним тест-кейсом. Рассчитывается как отношение количества покрытых требований к общему числу требований.
12. Принципы: Исчерпывающее тестирование невозможно, Кластеризация дефектов (80/20), Раннее тестирование. Суть "раннего тестирования": искать баги на этапе анализа требований, а не на этапе релиза, так как это в разы дешевле.

6.4 Ключи на вопросы для собеседования при проведении дифференцированного зачета

Тема 1.1. Основные понятия организации и тестирования

1. Дайте определение термину «Тестирование программного обеспечения». Чем тестирование отличается от отладки?

Ответ: Тестирование ПО — это процесс проверки соответствия программного продукта требованиям и выявления дефектов (багов). Главное отличие: **тестирование** — это процесс *поиска* ошибок (чтобы сказать, что программа работает неправильно). **Отладка** — это процесс *исправления* ошибок (поиск причины и написание правильного кода). Тестировщик находит баг и оформляет его, а отладчик (программист) его чинит.

2. Что такое «дефект», «ошибка» и «отказ»? Опишите их взаимосвязь.

Ответ:

- **Ошибка (Error)** — это действие или ошибка человека (программиста) в коде.
- **Дефект (Bug/Defect)** — это недочет в самом продукте (например, неправильно работает кнопка).
- **Отказ (Failure)** — это проявление дефекта во время работы системы (система упала, выдала сбой).
- **Связь:** Ошибка в коде -> приводит к дефекту в программе -> дефект вызывает отказ в работе у пользователя.

3. Какова цель и основные задачи тестирования на разных этапах жизненного цикла разработки ПО?

Ответ: Главная цель — повышение качества продукта и предотвращение дорогостоящих ошибок.

Задачи: на этапе анализа требований — проверка их полноты и непротиворечивости; на этапе разработки — проверка промежуточных модулей; на этапе внедрения — проверка готовности системы к эксплуатации. Роль тестировщика в команде — гарант качества (QA), связующее звено между заказчиком и разработчиками.

Тема 1.2. Методы, виды и принципы тестирования программных продуктов

4. Перечислите основные виды тестирования по уровню проведения.

Ответ:

- **Модульное (Unit)** — тестируется отдельная функция или метод.
- **Интеграционное (Integration)** — проверяется взаимодействие нескольких модулей между собой.
- **Системное (System)** — полное тестирование всей системы целиком.
- **Приемочное (Acceptance)** — тестирование системы заказчиком на соответствие его бизнес-требованиям.

5. Объясните разницу между функциональным и нефункциональным тестированием.

Ответ: **Функциональное** тестирование проверяет, *что* делает система (выполняет ли она заявленные функции: кнопка сохраняет файл, калькулятор считает). **Нефункциональное** проверяет, *как* она это делает (быстро ли, надежно ли, безопасно ли). К нефункциональным относятся: нагрузочное, стрессовое, тестирование производительности, безопасности (Security Testing) и удобства использования (Usability).

6. Что такое «тест-дизайн»? Опишите техники (классы эквивалентности, граничные значения).

Ответ: Тест-дизайн — это процесс создания тестовых данных и сценариев.

- **Классы эквивалентности** — деление входных данных на группы, внутри которых поведение системы одинаково (например, для возраста 0-120 лет: класс "невалидный до 0", класс "валидный 0-120", класс "невалидный выше 120").
- **Граничные значения** — проверка данных на границах этих диапазонов (например, 0, 1, 119, 120, 121). Это самые частые места ошибок.

7. Что такое «регрессионное тестирование»? Когда и зачем оно проводится?

Ответ: Регрессионное тестирование — это повторная проверка уже работающего функционала после внесения изменений в код или исправления багов. Проводится для того, чтобы убедиться, что новое исправление не сломало старые функции.

8. Что такое позитивное и негативное тестирование?

Ответ: **Позитивное** — проверка системы с корректными (валидными) данными (система должна выдать правильный результат). **Негативное** — проверка с некорректными (невалидными) данными (например, вместо числа вводим буквы, вводим пустую строку). Система должна выдать понятную ошибку, а не зависнуть или упасть.

9. Что такое тест-план, тест-кейс и чек-лист? В чем разница?

Ответ:

- **Тест-план** — это документ, описывающий весь процесс тестирования (что тестируем, кто тестирует, когда, какие ресурсы нужны).
- **Тест-кейс** — это подробная пошаговая инструкция (Шаги, Предусловия, Ожидаемый результат,

Фактический результат). Его удобно автоматизировать.

- **Чек-лист** — это упрощенный список пунктов (например, "проверить поле Имя: пустое, с цифрами"). Он меньше по объему и подходит для быстрого ручного тестирования.

Тема 1.3. Автоматизация тестирования. Инструментарий тестирования

10. Каковы плюсы и минусы автоматизации тестирования?

Ответ: **Плюсы:** экономия времени при регрессионном тестировании, возможность повторять тысячи тестов быстро, исключение человеческого фактора (усталость, невнимательность). **Минусы:** высокая стоимость написания и поддержки автотестов (при изменении кода автотесты часто ломаются и их нужно обновлять), невозможность проверить визуальную эстетику или юзабилити. Автоматизация целесообразна для стабильного, часто тестируемого функционала.

11. Что такое баг-трекер? Опишите жизненный цикл дефекта (Bug Lifecycle).

Ответ: Баг-трекер (Jira, Bugzilla) — это система для регистрации, отслеживания и управления ошибками. Жизненный цикл обычно такой: **New (Обнаружен)** -> **Open/In Progress (Взяли в работу)** -> **Fixed (Исправлен)** -> **Retest (Перепроверка)** -> **Closed (Закрыт)**. Если баг не подтвердился, он переходит в статус **Rejected (Отклонён)**.

12. Какие инструменты автоматизации тестирования вы знаете?

Ответ:

- **Postman / SoapUI** — инструменты для тестирования API (интерфейсов взаимодействия систем).
- **Selenium / Cypress** — инструменты для автоматизации тестирования веб-интерфейсов (эмулируют действия пользователя в браузере).
- **TestRail / Jira + Zephyr** — системы управления тестами (хранение тест-кейсов и результатов).
- **JMeter** — для нагрузочного тестирования.

13. Как собирается и обрабатывается статистика по дефектам? Какие метрики качества ПО существуют?

Ответ: Данные о багах берутся из баг-трекера. Основные метрики: **Плотность дефектов** (количество найденных багов на 1000 строк кода), **Покрытие тестами** (какой процент функционала проверен), **Уровень дефектности** (критичность оставшихся багов перед релизом). На основе статистики строятся графики, анализируется тренд (количество багов уменьшается или растет).

Вопросы на профессиональные компетенции (ПК) и документацию

14. Представьте, что вам выдали техническое задание на разработку ИС. Какие исходные данные необходимо собрать для составления тестовой документации?

Ответ: Для начала работы тестировщику нужны: 1) **Техническое задание (ТЗ)** на систему; 2) **Требования заказчика** (в виде текста или User Story); 3) **Макеты интерфейса** (визуальные прототипы, как должно выглядеть окно); 4) **Схемы баз данных** (чтобы понять, какие поля заполняются); 5) **Бизнес-логика** (правила, по которым программа должна считать). На основе этого создается тест-план и собираются тест-данные.

15. Опишите порядок действий при обнаружении критического дефекта в модуле безопасности системы.

Ответ: 1) Немедленно зарегистрировать баг в баг-трекере (Jira) с максимальным приоритетом (Blocker/Critical). 2) Приложить точные шаги воспроизведения, скриншоты, логи и версию среды (ОС, браузер). 3) Провести тестирование смежных модулей (чтобы понять масштаб бедствия). 4) Немедленно уведомить тимлида разработчиков и тестировщиков (обычно голосом или в рабочем чате, так как критический баг останавливает релиз). 5) После исправления провести повторный (регрессионный) тест безопасности всего модуля, и только потом закрыть баг.

Все ответы могут быть сформулированы в другой близкой по смыслу формулировке

6.5 Ключи на вопросы для устного опроса

1. Что такое тестирование программного обеспечения и каковы его основные цели?

Ответ: Тестирование — это процесс проверки программного продукта на соответствие требованиям, выявления дефектов и оценки качества. **Основные цели:**

- Нахождение ошибок (багов) до того, как их увидит пользователь.
- Снижение рисков выпуска некачественного продукта.
- Оценка соответствия требованиям заказчика (валидация) и технической документации (верификация).
- Улучшение качества продукта (поиск и исправление дефектов).

2. Объясните разницу между верификацией и валидацией в контексте тестирования ПО.

Ответ:

- **Верификация (Verification)** — это проверка того, «*правильно ли мы строим продукт*». Тестировщик проверяет, соответствует ли система техническому заданию, спецификациям и стандартам разработки (нет ли пропущенных полей, правильно ли реализованы функции по ТЗ).
- **Валидация (Validation)** — это проверка того, «*тот ли продукт мы строим*». Проверяется, соответствует ли система реальным ожиданиям и потребностям конечного пользователя. (Пример: заказчик просил зеленую кнопку, разработчик сделал синюю — это ошибка верификации. А если заказчик просил зеленую, разработчик сделал зеленую, но пользователь хотел оранжевую — это ошибка валидации).

3. Что такое ошибка (defect/bug), отказ (failure) и инцидент (incident) в тестировании?

Ответ:

- **Ошибка (Error/Defect/Bug):** недочет в программном коде, допущенный разработчиком.
- **Отказ (Failure):** проявление этой ошибки в реальности. Например, программа вылетает или выдает неверный результат.
- **Инцидент (Incident):** любое отклонение ожидаемого поведения системы от фактического (сбой, зависание, неверный текст). Инцидент может быть вызван не только багом, но и ошибкой пользователя или сбоем оборудования.

4. Какие основные принципы тестирования вы знаете? Опишите не менее трех.

Ответ: Основные принципы тестирования (стандарт ISTQB):

1. **Исчерпывающее тестирование невозможно.** Нельзя проверить все возможные комбинации данных и действий, поэтому тестирование всегда основывается на оценке рисков.
2. **Кластеризация дефектов.** Как правило, 80% ошибок находится в 20% модулей системы. Усилия сосредотачиваются там, где чаще всего происходят падения.
3. **Парадокс пестицида.** Если повторять одни и те же тесты, они перестают находить новые ошибки. Тест-кейсы нужно постоянно пересматривать и обновлять.
4. **Раннее тестирование.** Тестировать нужно начинать как можно раньше (на этапе анализа требований), чтобы исправление ошибок стоило дешевле.
5. **Заблуждение об отсутствии ошибок.** Отсутствие найденных багов не гарантирует качество продукта (например, если программа работает, но делает не то, что нужно пользователю).

5. Что такое тестовый случай (test case) и из каких элементов он состоит?

Ответ: Тест-кейс — это формализованное описание одного шага или набора шагов для проверки конкретной функции. **Элементы тест-кейса:**

- Уникальный ID (ТС-001).
- Название (кратко о том, что проверяем).
- Предусловия (что должно быть настроено/введено ДО начала проверки).
- Шаги (пошаговая инструкция действий).
- Ожидаемый результат (что должна вывести система).
- Фактический результат (заполняется после выполнения: соответствует или не соответствует).
- Статус (Passed / Failed / Blocked).
- Приоритет (High, Medium, Low).

6. Что такое тест-план и какую роль он играет в процессе тестирования?

Ответ: Тест-план — это официальный документ, описывающий весь объем работ по тестированию. В нем указываются: цели, объем (что тестируем, а что нет), стратегия (какие методы используем), сроки, распределение ресурсов, люди, инструменты, критерии начала и окончания тестирования, а также риски. **Роль:** это "дорожная карта", которая помогает планировать работу, согласовывать ее с заказчиком и избегать срыва сроков.

7. Объясните понятия "покрытие требований" (requirements coverage) и "покрытие кода" (code coverage).

Ответ:

- **Покрытие требований** — процентное соотношение требований (пунктов ТЗ), которые проверены хотя бы одним тест-кейсом.
- **Покрытие кода** — метрика, показывающая, какой процент программного кода был исполнен во время тестов (например, покрытие операторов, ветвей, условий). Стопроцентное покрытие кода встречается редко, и оно не гарантирует отсутствие багов, но показывает, что базовые логические пути не пропущены.

8. Какие основные роли в команде тестирования вы знаете?

Ответ:

- **QA Engineer (Тестировщик)** — ручное выполнение тест-кейсов и поиск багов.
- **QA Automation Engineer (Автоматизатор)** — пишет скрипты для автотестов.
- **Test Analyst (Тест-аналитик)** — анализирует требования, придумывает тест-кейсы, занимается тест-дизайном.
- **Test Lead (Тест-лид)** — руководит командой, пишет тест-планы, распределяет задачи, взаимодействует с менеджером проекта.
- **QA Manager** — отвечает за качество процесса в целом.

9. Что такое жизненный цикл дефекта (bug lifecycle)?

Ответ: Это путь, который проходит ошибка от момента её обнаружения до закрытия. Стандартный цикл:

1. **New (Открыт)** — баг заведен.
2. **Open/In Progress** — разработчик взял в работу.
3. **Fixed (Исправлен)** — разработчик поставил статус, что баг исправлен.
4. **Retest (Перепроверка)** — тестировщик проверяет фикс.
5. **Closed (Закрыт)** — бага нет, тест пройден.

Дополнительные статусы: **Rejected (Отклонен)** — не баг/воспроизводится неправильно; **Reopened (Переоткрыт)** — баг остался или появился снова; **Deferred (Отложен)** — будут исправлять в следующем релизе.

10. Как документация связана с процессом тестирования?

Ответ: Тестирование полностью завязано на документации. Исходные данные берутся из ТЗ (технического задания) и **требований**. На их основе пишется **Тест-план**. Затем создается **Тест-кейсы** и **Чек-листы**. В процессе работы заполняются **Баг-репорты**. По завершению составляется **Отчет о тестировании** со статистикой. Документация фиксирует как результаты, так и ответственность за качество.

11. Какие основные уровни тестирования вы знаете? Опишите каждый из них.

Ответ:

- **Модульное (Unit)** — проверка отдельных функций/методов кода разработчиками.
- **Интеграционное (Integration)** — проверка взаимодействия отдельных модулей и API между собой.
- **Системное (System)** — проверка всей системы целиком (функциональность, производительность, безопасность).
- **Приемочное (Acceptance)** — финальная проверка заказчиком на соответствие его бизнес-требованиям (альфа/бета тестирование).

12. В чем разница между функциональным и нефункциональным тестированием?

Ответ:

- **Функциональное** проверяет «Что делает система?» (есть ли нужные функции, правильно ли они работают).
- **Нефункциональное** проверяет «Как система делает это?» (скорость, устойчивость, безопасность, удобство использования).

13. Что такое регрессионное тестирование и когда оно проводится?

Ответ: Регрессионное тестирование — это повторное выполнение уже существующих тестов. Оно проводится **после каждого обновления кода (фикса бага или добавления новой функции)**, чтобы убедиться, что новые изменения не сломали старый, работающий функционал. Это самый объемный этап тестирования, поэтому его часто автоматизируют.

14. Объясните разницу между тестированием "черного ящика" и "белого ящика".

Ответ:

- **Черный ящик** — тестировщик не видит внутренний код и структуру системы. Он проверяет только внешнее поведение (вводит данные и смотрит на результат). Это метод тестировщиков-специалистов.
- **Белый ящик** — тестировщик (чаще разработчик) видит код, структуру циклов и условий. Он проверяет логику работы программы (покрытие веток и условий).

15. Какие методы тестирования "черного ящика" вы знаете? Опишите не менее двух методов.

Ответ:

- **Эквивалентное разбиение (Equivalence Partitioning)** — делит входные данные на группы (валидные и невалидные), берется по одному представителю из каждой группы (например, если пароль до 10 символов, тестируем один из 0-10 и один из 11+).
- **Анализ граничных значений (Boundary Value Analysis)** — проверяются данные на границах диапазонов (0, 1, 9, 10, 11). Это самые частые места ошибок.
- **Таблица решений (Decision Table)** — для проверки сложных бизнес-логик с комбинациями разных условий (Если А и Б, то В).
- **Проверка состояний и переходов (State Transition)** — когда объект может менять состояния (например, заказ: Новый -> Оплачен -> Отправлен), проверяются корректные и некорректные переходы.

16. Что такое тестирование пользовательского интерфейса (UI testing)?

Ответ: Это проверка визуальной и интерактивной части программы (кнопки, поля, меню).

Проверяется: соответствие интерфейса дизайн-макету, читаемость текста, правильность расположения элементов, реакция на клики/наведение мыши, адаптивность под разные разрешения экрана, корректное отображение ошибок (например, красная подсветка невалидного поля).

17. Какие виды нефункционального тестирования вы знаете? (например, нагрузочное, безопасность)

Ответ:

- **Нагрузочное (Performance/Load)** — проверка скорости работы и поведения под нагрузкой (много пользователей).
- **Стрессовое (Stress)** — проверка поведения системы при экстремальных нагрузках (сбои, восстановление).
- **Тестирование безопасности (Security)** — проверка уязвимостей (SQL-инъекции, взлом пароля, доступ к чужим данным).
- **Тестирование надежности (Reliability)** — работа без сбоев длительное время.
- **Тестирование удобства использования (Usability)** — насколько понятно, удобно и интуитивно ли взаимодействовать с системой.
- **Тестирование совместимости (Compatibility)** — работа на разных ОС, браузерах, версиях устройств.

18. Что такое исследовательское тестирование (exploratory testing)?

Ответ: Это подход, при котором тестировщик **одновременно изучает систему, проектирует тесты и выполняет их на лету**, без предварительно написанных жестких тест-кейсов. Тестировщик двигается по системе как исследователь, пытаясь найти скрытые баги, основываясь на своей интуиции и

предыдущем опыте. Особенно полезно на ранних этапах, когда документация ещё не готова, или для проверки сложных, малоизученных модулей.

Все ответы могут быть сформулированы в другой близкой по смыслу формулировке