

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ**

МДК.02.01 Технология разработки программного обеспечения

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.02.01 Технология разработки программного обеспечения» 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.02.01 Технология разработки программного обеспечения» 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 для составления технических заданий, спецификаций, руководств пользователя и отчетов о разработке программного обеспечения.	Назначение и виды технологий разработки ПО; основные этапы жизненного цикла программного обеспечения (ЖЦ ПО); методы сбора и анализа требований заказчика.
ОК 4 ОК 5	Применять мультимедийные технологии и специализированное ПО для визуализации проектирования (построение UML-диаграмм, создание макетов интерфейса) и демонстрации функционала разработанного продукта.	Состав, структуру и принципы работы CASE-средств (средств автоматизированного проектирования) и сред быстрой разработки (RAD); основы языка моделирования UML.
ОК 9 ОК 10	Обрабатывать экономическую и статистическую информацию для расчета трудоемкости, стоимости и оценки качества разработки, используя средства пакета прикладных программ.	Методы оценки сложности и качества программного обеспечения (метрики); базовые принципы управления ИТ-проектами, планирования и учета ресурсов.
ПК 5.1 ПК 5.2 ПК 5.6 ПК 6.3	Собирать исходные данные (требования, стандарты, регламенты) для разработки проектной документации на информационную систему. Разрабатывать проектную документацию (Техническое задание, Технический проект) в строгом соответствии с требованиями заказчика и стандартами ЕСПД (ГОСТ). Разрабатывать подсистемы безопасности (разграничение прав доступа, шифрование, защита от несанкционированного доступа) в соответствии с техническим заданием. Производить разработку модулей в соответствии с техническим заданием и корпоративными стандартами.	Инструментальные средства разработки; современные методологии разработки; принципы объектно-ориентированного программирования и построения архитектуры ИС; стандарты оформления программной документации.

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	120
в том числе:	
– теоретическое обучение	50
– практические занятия <i>(если предусмотрено)</i>	66
– самостоятельная работа	
– консультации	
– промежуточная аттестация – <i>(дифференцируемый зачет)</i>	4

2.2 Тематический план и содержание учебной дисциплины «МДК.02.01 технология разработки программного обеспечения»

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 и стандартизация требований к программному обеспечению	Содержание учебного материала: Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями. Современные принципы и методы разработки программных приложений. Методы организации работы в команде разработчиков. Системы контроля версий. Основные подходы к интегрированию программных модулей. Стандарты кодирования.	16	ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10
	Практическое занятие №1: «Анализ предметной области» Практическое занятие №2: «Разработка и оформление технического задания» Практическое занятие №3: «Построение архитектуры программного средства» Практическое занятие №4: «Изучение работы в системе контроля версий»	22	ПК 5.1 ПК 5.2 ПК 5.6 ПК 6.3
	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. Описание и анализ требований. Диаграммы IDEF	Содержание учебного материала: Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения	16	ОК 1 ОК 2 ОК 4
	Практическое занятие №5: «Построение диаграммы Вариантов использования и диаграммы. Последовательности»	22	ОК 5 ОК 9
	Практическое занятие 6: «Построение диаграммы Кооперации и диаграммы Развертывания»		ОК 10
	Практическое занятие №7: «Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов»		ПК 5.1 ПК 5.2 ПК 5.6
	Практическое занятие №8: «Построение диаграммы компонентов» Практическое занятие №9: «Построение диаграмм потоков данных»		ПК 6.3
Тема 1.3 Оценка качества программных средств	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
	Содержание учебного материала: Цели и задачи и виды тестирования. Стандарты качества программной документации. Меры и метрики. Тестовое покрытие. Тестовый сценарий, тестовый пакет. Анализ спецификаций. Верификация и аттестация программного обеспечения	18	ОК 1 ОК 2 ОК 4 ОК 5
	Практическое занятие №10: «Разработка тестового сценария» Практическое занятие №11: «Оценка необходимого количества тестов» Практическое занятие №12: «Разработка тестовых пакетов» Практическое занятие №13: «Оценка программных средств с помощью метрик» Практическое занятие №14: «Инспекция программного кода на предмет соответствия стандартам кодирования»	22	ОК 9 ОК 10 ПК 5.1 ПК 5.2 ПК 5.6 ПК 6.3
	Самостоятельная работа обучающихся	Не предусмотрено	
Промежуточная аттестация - дифференцированный зачет	Лабораторное занятие	Не предусмотрено	
		4	

ИТОГО часов		120	
Теоретические занятия		50	
Практические занятия		66	
Самостоятельная работа		0	
дифференцированный зачет		4	

3 Условия реализации учебной дисциплины

3.1 Для реализации программы учебной дисциплины «МДК.02.01 Технология разработки программного обеспечения» 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.02.01 Технология разработки программного обеспечения»

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.02.01 Технология разработки программного обеспечения» 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.02.01 Технология разработки программного обеспечения» 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 5.1 ПК 5.2 ПК 5.6 ПК 6.3	31	Знать понятие требований к ПО, их классификацию, методы сбора, анализа и документирования (ГОСТ 19, ГОСТ 34).
	32	Знать унифицированный язык моделирования UML (диаграммы вариантов использования, классов, последовательностей, деятельности), а также методологию IDEF0.
	33	Знать метрики оценки качества ПО, цели и виды тестирования, методы расчета трудоемкости и стоимости разработки.
	34	Знать инструментальные средства разработки: CASE-средства, среды программирования (IDE), системы контроля версий (Git), а также стандарты и принципы построения архитектуры ИС.
	У1	Уметь собирать исходные данные, обрабатывать требования и разрабатывать проектную документацию (Техническое задание, Технический проект) в соответствии с требованиями заказчика.
	У2	Уметь применять UML и IDEF для построения моделей предметной области, моделирования бизнес-процессов, архитектуры системы и диаграмм компонентов.
	У3	Уметь проводить инспекцию кода, разрабатывать тестовые сценарии, оценивать качество, проводить верификацию и аттестацию программного продукта.

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 и стандартизация требований к программному обеспечению	31	Способность понимать классификацию требований, методологии и стандарты разработки (ГОСТ 19, ГОСТ 34).	<i>Устный опрос (п 5.1 вопросы 1-5) Реферат (п 5.2 темы 1-15) Практическая работа №1-4 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	32	Способность знать принципы работы систем контроля версий и интегрированных сред разработки (IDE).		
	У1	Способность собирать исходные данные, проводить анализ предметной области и разрабатывать Техническое задание.		
Тема №. 1.2. Описание и анализ требований. Диаграммы IDEF	33	Способность знать унифицированный язык моделирования UML и методологию IDEF0, их состав и структуру.	<i>Устный опрос (п 5.1 вопросы 6-10) Реферат (п 5.2 темы 1-15) Практическая работа №5-9 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У2	Способность строить UML-диаграммы и IDEF-модели для описания требований.		
	У3	Способность применять инструментальные CASE-средства для визуализации архитектуры будущей системы.		
Тема №. 1.3 Оценка качества программных средств	34	Способность знать метрики качества, виды тестирования, цели верификации и аттестации, а также тестовую документацию.	<i>Устный опрос (п 5.1 вопросы 11-18) Реферат (п 5.2 темы 1-15) Практическая работа №10-14 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У4	Способность разрабатывать тест-кейсы, проводить функциональное тестирование, анализировать ошибки и оценивать качество программного кода.		
	У5	Способность оформлять отчетную документацию по тестированию и проводить инспекцию (ревизию) кода на соответствие стандартам.		

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. Понятия требований, классификация, уровни требований.
2. Методологии и стандарты, регламентирующие работу с требованиями.
3. Современные принципы и методы разработки программных приложений.
4. Методы организации работы в команде разработчиков.
5. Системы контроля версий.
6. Основные подходы к интегрированию программных модулей.
7. Стандарты кодирования.
8. Техническое задание.
9. Архитектура программного средства.
10. Описание требований: унифицированный язык моделирования - краткий словарь.
11. Диаграммы UML.
12. Описание и оформление требований (спецификация).
13. Анализ требований и стратегии выбора решения.
14. Цели и задачи и виды тестирования.
15. Стандарты качества программной документации.
16. Меры и метрики.
17. Тестовое покрытие. Тестовый сценарий, тестовый пакет.
18. Анализ спецификаций.

5.2 Темы рефератов

Тема 1.1. Основные понятия и стандартизация требований к программному обеспечению

1. Требования к ПО: уровни, виды и классификация.
2. Методологии и стандарты сбора и анализа требований (IEEE 830, ГОСТ 19).
3. Каскадная (Waterfall) и гибкая (Agile/Scrum) модели разработки ПО: сравнительный анализ.
4. Экстремальное программирование (XP) и его методы (парное программирование, TDD).
5. Системы контроля версий (Git, SVN): назначение и принципы работы в команде.
6. Подходы к интегрированию программных модулей (Big Bang, Bottom-up, Top-down).
7. Стандарты кодирования (Code Conventions) и их роль в сопровождении кода.
8. Организация работы команды разработчиков: роли и зоны ответственности.

Тема 1.2. Описание и анализ требований. Диаграммы IDEF

9. Унифицированный язык моделирования (UML): история, назначение, краткий словарь.
10. Виды диаграмм UML и их применение в проектировании.
11. Диаграмма вариантов использования (Use Case): построение и описание сценариев.

12. Диаграммы последовательности и кооперации: отражение динамики взаимодействия объектов.
13. Диаграммы деятельности и состояний: моделирование бизнес-процессов.
14. Диаграмма классов: статическая структура информационной системы.
15. Методология функционального моделирования IDEF0: основные блоки и стрелки.
16. Диаграммы потоков данных (DFD): назначение и правила построения.
17. Анализ требований и стратегии выбора решения (оценка рисков, прототипирование).

Тема 1.3. Оценка качества программных средств

18. Цели, задачи и виды тестирования программного обеспечения.
19. Верификация и валидация: в чем разница?
20. Меры и метрики качества программного продукта (плотность дефектов, покрытие кода).
21. Тест-кейс и чек-лист: разработка, структура и применение.
22. Тестовые пакеты и тестовые сценарии: организация процесса тестирования.
23. Анализ спецификаций и тестирование требований (тест-дизайн).
24. Отладка программного обеспечения: инструменты и методы.
25. Инспекция кода (Code Review) как метод контроля качества.

5.3 Примеры практических работ

Тема 1.1. Основные понятия и стандартизация требований

Практическое занятие №1: «Анализ предметной области»

Цель: Научиться выделять границы системы и собирать исходные данные.

Пример выполнения:

1. **Опишите процесс:** Книжный магазин принимает заказы от покупателей, проверяет наличие книг на складе, оформляет доставку.
2. **Выделите действующих лиц (актеров):** Покупатель, Продавец-кассир, Кладовщик, Администратор.
3. **Опишите процессы:** Покупатель ищет книгу -> Оформляет заказ -> Оплачивает -> Получает. Продавец проверяет наличие -> Формирует чек -> Отдает книгу.
4. **Результат (в отчете):** Схема бизнес-процесса текстом или в Word.

Практическое занятие №2: «Разработка и оформление технического задания»

Цель: Уметь разрабатывать ТЗ по стандартам (ГОСТ 19.201-78).

Пример выполнения (фрагмент отчета):

- **1. Общие сведения:** Полное наименование системы - «ИС Книжный магазин».
- **2. Назначение разработки:** Автоматизация учета книг, заказов и продаж.
- **3. Требования к программе:**
 - Функциональные: поиск по названию/автору, корзина, оформление заказа, отчеты по продажам.
 - Надежности: время отклика не более 2 секунд, резервное копирование БД.
- **4. Стадии и этапы:** Разработка, тестирование, внедрение, сопровождение.

Практическое занятие №3: «Построение архитектуры программного средства»

Цель: Выбрать и описать архитектуру.

Пример выполнения:

- **Описываем трехуровневую архитектуру (N-tier):**
 - Клиентский уровень (UI) — веб-браузер или Windows Forms.
 - Уровень бизнес-логики (Backend) — сервер приложений (C# / Java).
 - Уровень данных — СУБД (MS SQL Server / PostgreSQL).
- Приводим схему (текстовое описание связей).
-

Практическое занятие №4: «Изучение работы в системе контроля версий»

Цель: Освоить базовые команды Git.

Пример выполнения (скриншоты терминала + текст):

1. git init — создание локального репозитория.
2. git add . — добавление файлов в индекс.
3. git commit -m "Инициализация проекта" — фиксация изменений.
4. git branch dev — создание ветки для разработки.
5. git merge dev — слияние веток.
6. *Результат:* Описание, зачем нужны ветки и как ими пользоваться.

Тема 1.2. Описание и анализ требований. Диаграммы IDEF

Практическое занятие №5: «Построение диаграмм Вариантов использования и диаграммы Последовательности»

Пример выполнения (текстовое описание для отчета):

- **Use Case:** Актёр «Покупатель» -> Прецедент «Оформить заказ». Актёр «Администратор» -> Прецедент «Управление каталогом книг».
- **Sequence (Последовательность):** Покупатель -> Нажимает "Купить" -> Система запрашивает данные у модуля "Корзина" -> Система отправляет запрос в БД -> БД возвращает подтверждение -> Выводится окно "Заказ оформлен".

Практическое занятие №6: «Построение диаграмм Кооперации и диаграммы Развертывания»

Пример выполнения:

- **Кооперация (Communication):** Показать, как объекты связываются друг с другом (Покупатель <-1-> Каталог <-1-> Заказ).
- **Развертывание (Deployment):** Физическое размещение: Сервер БД (Node 1) <-> Сервер приложений (Node 2) <-> Клиентские ПК (Node 3).

Практическое занятие №7: «Построение диаграмм Деятельности, диаграммы Состояний и диаграммы Классов»

Пример выполнения:

- **Деятельности (Activity):** Описываем алгоритм покупки: Выбор книги -> Проверка наличия -> Если есть: Оплата -> Завершение. Если нет: Сообщение об отсутствии.
- **Состояний (State):** Жизненный цикл заказа: Новый -> Оплачен -> Отправлен -> Доставлен -> Закрыт.
- **Классов (Class):** Класс «Книга»: Атрибуты (Название, Автор, Цена, ISBN). Методы (Добавить, Изменить). Класс «Заказ»: Атрибуты (Дата, Статус). Метод (РассчитатьИтог).

Практическое занятие №8: «Построение диаграммы Компонентов»

Пример выполнения: Показать логические модули: UI.dll (интерфейс), Logic.dll (бизнес-логика), DataAccess.dll (доступ к данным), BookDatabase.mdf (файл БД).

Практическое занятие №9: «Построение диаграмм потоков данных»

Пример выполнения: Построить модель IDEF0.

- Вход: Заказ покупателя.
- Управление: Правила продаж и законодательство.
- Механизм (Исполнитель): Продавец и ИС.
- Выход: Чек и запись в БД.

Тема 1.3. Оценка качества программных средств

Практическое занятие №10: «Разработка тестового сценария»

Пример выполнения: Написать **Тест-кейс** на проверку авторизации.

- ID: TC-001
- Заголовок: Проверка входа с валидными данными.
- Предусловия: Пользователь зарегистрирован в системе.
- Шаги: 1. Открыть форму входа. 2. Ввести логин "user". 3. Ввести пароль "123". 4. Нажать "Войти".
- Ожидаемый результат: Открывается главное окно системы.

Практическое занятие №11: «Оценка необходимого количества тестов»

Пример выполнения: Используем метод **граничных значений** для поля «Возраст» (допустимо 1-120).

- Создаем 6 тестов: -1, 0, 1, 119, 120, 121.
- Используем **классы эквивалентности**: берем 1 значение из валидного диапазона (например, 50) и 1 из невалидного (например, -100). *Результат:* Расчет общего количества тест-кейсов.

Практическое занятие №12: «Разработка тестовых пакетов»

Пример выполнения: Группируем тест-кейсы в логические блоки:

- **Test Suite 1:** Пакет тестов авторизации.
- **Test Suite 2:** Пакет тестов каталога.
- **Test Suite 3:** Пакет тестов оформления заказа.

Практическое занятие №13: «Оценка программных средств с помощью метрик»

Пример выполнения: Рассчитаем **плотность дефектов**. В коде 5000 строк (5 KLOC). Найдено 10 багов. Плотность = $10 / 5 = 2$ бага на 1000 строк. Также считаем **покрытие требований**: 8 требований из 10 покрыты тестами = 80%.

Практическое занятие №14: «Инспекция программного кода на предмет соответствия стандартам кодирования»

Пример выполнения: Проверяем код по чек-листу: правильность именования (CamelCase для переменных), наличие комментариев, отсутствие "магических чисел", правильные отступы. Исправляем найденные ошибки.

6. Оценочные средства для проведения промежуточной аттестации

6.1 Вопросы для собеседования при проведении дифференцированного зачета

1. **Что такое жизненный цикл программного обеспечения (ЖЦ ПО)?** Перечислите основные этапы и дайте краткую характеристику каждому.
2. **В чем разница между каскадной (Waterfall) и гибкой (Agile) методологиями разработки?** Приведите примеры задач, для которых подходит каждая из них.
3. **Что такое техническое задание (ТЗ)?** Перечислите основные разделы ТЗ согласно ГОСТ 19.201-78 и объясните, почему оно является ключевым документом для разработчика.
4. **Что такое система контроля версий (VCS)?** Какую роль играет Git в командной

разработке? Опишите базовый цикл работы с Git (init, add, commit, push).

5. **Какие бывают виды требований к ПО?** Поясните разницу между функциональными и нефункциональными требованиями на конкретном примере.
 6. **Что такое интегрированная среда разработки (IDE)?** Назовите не менее двух известных сред и перечислите их основные инструменты (отладчик, автодополнение и др.).
-

2. Тема № 1.2. Описание и анализ требований. Диаграммы IDEF

7. **Что такое UML?** Перечислите основные виды диаграмм UML, используемых при проектировании архитектуры.
 8. **Для чего нужна диаграмма вариантов использования (Use Case)?** Назовите основные элементы (актёр, прецедент, связи) и приведите пример для интернет-магазина.
 9. **Что такое диаграмма классов?** Опишите её основные компоненты (класс, атрибуты, методы, отношения) и её роль в ООП.
 10. **Что такое методология IDEF0?** Из каких основных элементов состоит функциональная модель (блоки, дуги, стрелки)? Как она помогает анализировать бизнес-процессы?
 11. **Чем отличаются диаграммы деятельности (Activity Diagram) от диаграмм последовательности (Sequence Diagram)?** В каких случаях используется каждая из них?
 12. **Какие CASE-средства (средства автоматизированного проектирования) вы знаете?** Какие функции они выполняют при построении моделей и генерации кода?
-

3. Тема № 1.3. Оценка качества программных средств

13. **Что такое качество программного обеспечения?** Перечислите основные характеристики качества, согласно стандарту ISO/IEC 9126.
14. **Что такое верификация и валидация?** В чем их принципиальное различие и где каждый процесс применяется на этапе разработки?
15. **Что такое аттестация (acceptance testing) программного обеспечения?** Как она связана с требованиями заказчика?

16. **Что такое тестирование?** Назовите основные уровни тестирования (модульное, интеграционное, системное) и их цели.
17. **Что такое тест-кейс и чек-лист?** Из каких элементов состоит тест-кейс? Как эти документы используются для контроля качества?
18. **Что такое метрики качества ПО?** Приведите примеры метрик (например, плотность дефектов, покрытие кода) и объясните, как они помогают анализировать процесс разработки.

6.2 Тестовые материалы

Вариант 1

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

1. Как называется основной документ, в котором фиксируются цели, требования и ограничения на разработку программного продукта?
А) Тест-план
Б) Техническое задание (ТЗ)
В) Руководство пользователя
Г) Отчет о тестировании
2. Какая диаграмма UML предназначена для отображения обмена сообщениями между объектами во времени (динамики процесса)?
А) Диаграмма классов
Б) Диаграмма последовательности
В) Диаграмма вариантов использования
Г) Диаграмма компонентов
3. Что означает аббревиатура CI/CD в современной разработке?
А) Центральный интерфейс / Центральные данные
Б) Непрерывная интеграция / Непрерывная доставка
В) Компьютерная интеграция / Компьютерная документация
Г) Контроль исполнения / Контроль данных

Часть 2. Установление соответствия (вопросы 4–6)

4. Установите соответствие между стандартом и его предназначением.
Стандарт: 1. ГОСТ 19 (ЕСПД), 2. ГОСТ 34, 3. IEEE 830
Назначение: А. Стандарты на разработку автоматизированных систем, Б. Единая система программной документации, В. Стандарт на спецификацию требований к ПО (SRS).
5. Установите соответствие между диаграммой UML и её основным назначением.
Диаграмма: 1. Use Case (Вариантов использования), 2. Class (Классов), 3. Activity (Деятельности)
Назначение: А. Описание статической структуры (классы, атрибуты, методы), Б. Описание бизнес-процессов и логики действий, В. Описание функциональных требований (кто и что делает в системе).
6. Установите соответствие между видом тестирования и его целью.
Вид: 1. Регрессионное, 2. Нагрузочное, 3. Тестирование безопасности
Цель: А. Проверка производительности при большом количестве пользователей, Б. Проверка, что новые изменения не сломали старый функционал, В. Поиск уязвимостей и защита от

несанкционированного доступа.

Часть 3. Установление последовательности (вопросы 7–9)

7. Расставьте в правильной последовательности этапы жизненного цикла ПО по каскадной модели (Waterfall):
 1. Тестирование
 2. Проектирование
 3. Анализ требований
 4. Сопровождение
 5. Разработка (кодирование)
8. Расставьте в правильном порядке базовый цикл работы с системой контроля версий Git:
 1. Commit (фиксация изменений)
 2. Push (отправка на удаленный сервер)
 3. Init (инициализация репозитория)
 4. Add (добавление файлов в индекс)
9. Расставьте в правильной последовательности жизненный цикл дефекта (Bug Lifecycle):
 1. Fixed (Исправлен)
 2. New (Обнаружен)
 3. Closed (Закрыт)
 4. In Progress (Взяли в работу)
 5. Retest (Перепроверка)

Часть 4. Открытые вопросы (вопросы 10–12)

10. Перечислите не менее трех свойств, которыми должна обладать качественная архитектура программного средства.
11. Объясните разницу между верификацией и валидацией. Приведите примеры.
12. Какова главная цель технического задания (ТЗ) и какие основные разделы оно должно содержать?

Вариант 2

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

1. Какая методология разработки предполагает работу короткими итерациями (спринтами) с постоянной обратной связью от заказчика?
 - А) Каскадная (Waterfall)
 - Б) Спиральная
 - В) Agile/Scrum
 - Г) V-образная
2. Какова главная цель системы контроля версий (Git)?
 - А) Написание кода
 - Б) Отслеживание изменений в файлах и совместная работа команды
 - В) Автоматическая сборка проекта
 - Г) Покрытие кода тестами
3. Что такое тест-кейс?
 - А) Список найденных багов
 - Б) Пошаговая инструкция с ожидаемым результатом для проверки конкретной функции
 - В) Общее описание того, что нужно протестировать без конкретных шагов
 - Г) Отчет о проделанной работе

Часть 2. Установление соответствия (вопросы 4–6)

4. Установите соответствие между типом требования и его примером.
Тип: 1. Функциональное, 2. Нефункциональное, 3. Ограничение
Пример: А. Система должна выдавать ответ не дольше 1 секунды, Б. Система должна производить расчет налогов, В. Разработка должна вестись только на языке C#.
5. Установите соответствие между элементом методологии IDEF0 и его значением.
Элемент: 1. Блок, 2. Стрелка Входа, 3. Стрелка Управления
Значение: А. Данные или предметы, которые обрабатываются, Б. Правила, стандарты или инструкции, регулирующие процесс, В. Сама функция или процесс.
6. Установите соответствие между метрикой качества ПО и её определением.
Метрика: 1. Плотность дефектов, 2. Покрытие кода, 3. Цикломатическая сложность
Определение: А. Процент исполненного кода во время тестов, Б. Количество багов на 1000 строк кода, В. Сложность логики программы (количество независимых путей).

Часть 3. Установление последовательности (вопросы 7–9)

7. Расставьте в правильном порядке этапы работы с требованиями:
1. Специфицирование (оформление требований)
 2. Сбор (выявление) требований
 3. Верификация (проверка корректности)
 4. Анализ требований
8. Расставьте в правильном порядке этапы процесса тестирования:
1. Выполнение тестов
 2. Разработка тест-кейсов
 3. Составление отчетов о найденных дефектах
 4. Планирование и оценка рисков
9. Расставьте в правильной последовательности шаги построения диаграммы классов UML:
1. Определение отношений между классами
 2. Выделение классов из описания предметной области
 3. Описание атрибутов и методов классов
 4. Построение графической схемы

Часть 4. Открытые вопросы (вопросы 10–12)

10. Опишите принципиальные отличия каскадной модели (Waterfall) от гибкой методологии (Agile).
11. Что такое матрица трассировки (Traceability Matrix) и какую роль она играет в обеспечении качества?
12. Перечислите не менее трех техник тест-дизайна «черного ящика» и кратко раскройте суть одной из них на примере.

6.3 Ключи к тестовым заданиям для проведения промежуточной аттестации

Вариант 1:

1. Б (Техническое задание)
2. Б (Диаграмма последовательности)
3. Б (Непрерывная интеграция / Непрерывная доставка)
4. 1-Б, 2-А, 3-В
5. 1-В, 2-А, 3-Б

6. 1-Б, 2-А, 3-В
7. 3, 2, 5, 1, 4
8. 3, 4, 1, 2
9. 2, 4, 1, 5, 3
10. *Пример ответа:* Модульность (разделение на независимые компоненты), слабая связность (минимизация зависимостей между модулями), инкапсуляция (сокрытие внутренней реализации), масштабируемость, расширяемость.
11. *Пример ответа:* Верификация — «правильно ли мы строим продукт» (соответствие кода ТЗ и стандартам). Валидация — «тот ли продукт мы строим» (соответствие потребностям и ожиданиям пользователя). Например, разработчик сделал синюю кнопку вместо зеленой по макету — ошибка верификации. А если заказчик хотел кнопку зеленую, а пользователю нужна была красная — ошибка валидации.
12. *Пример ответа:* Цель ТЗ — зафиксировать договоренности между заказчиком и разработчиком, чтобы избежать споров и определить объем работ. Основные разделы: Общие сведения, Назначение разработки, Требования к программе, Требования к программной документации, Стадии и этапы разработки, Порядок контроля и приемки.

Ответ может быть сформулирован в другой близкой по смыслу формулировке

Вариант 2:

1. В (Agile/Scrum)
2. Б (Отслеживание изменений в файлах и совместная работа команды)
3. Б (Пошаговая инструкция с ожидаемым результатом)
4. 1-Б, 2-А, 3-В
5. 1-В, 2-А, 3-Б
6. 1-Б, 2-А, 3-В
7. 2, 4, 1, 3
8. 4, 2, 1, 3
9. 2, 3, 1, 4
10. *Пример ответа:* Waterfall — последовательная модель, планирование всех этапов сразу, изменения вносить дорого. Agile — итеративная модель, разработка короткими спринтами, требования могут меняться, постоянная связь с заказчиком.
11. *Пример ответа:* Матрица трассировки — таблица, которая связывает каждое требование ТЗ с конкретными тест-кейсами. Роль: она показывает, что ни одно требование не осталось непроверенным, помогает понять, какие тесты сломаются при изменении определенного требования.
12. *Пример ответа:* Техники: Эквивалентное разбиение (деление данных на классы), Анализ граничных значений (проверка данных на границах диапазонов), Таблица решений (проверка комбинаций условий). Например, при проверке поля «Возраст» (допустимые значения 0-120), граничными значениями будут: -1, 0, 1, 119, 120, 121. Это частые места ошибок.

Ответ может быть сформулирован в другой близкой по смыслу формулировке

6.4 Ключи к вопросам для собеседования при проведении дифференцированного зачета

Вопрос 1: ЖЦ ПО — период от появления идеи до снятия продукта с эксплуатации. Этапы: анализ требований, проектирование, разработка, тестирование, внедрение, сопровождение.

Вопрос 2: Waterfall — последовательная линейная модель (сначала всё проектируем, потом пишем код). Agile — итеративная модель (разработка идёт короткими спринтами с постоянной обратной связью).

Вопрос 3: ТЗ — документ, фиксирующий цели, требования, технические характеристики и этапы разработки. Разделы: общие сведения, назначение, требования к системе, требования к программной документации, этапы и контроль.

Вопрос 4: VCS — система для отслеживания изменений в файлах. Git позволяет работать нескольким людям одновременно. Цикл: git init (создать), git add (добавить в индекс), git commit (зафиксировать), git push (отправить на сервер).

Вопрос 5: Функциональные — что система делает (например, оформление заказа). Нефункциональные — как система это делает (скорость, безопасность, удобство).

Вопрос 6: IDE — программа для разработки (Visual Studio, PyCharm, Eclipse). Инструменты: редактор кода, компилятор/интерпретатор, отладчик, автодополнение.

Вопрос 7: UML — унифицированный язык моделирования. Диаграммы: вариантов использования, классов, последовательностей, деятельности, состояний.

Вопрос 8: Use Case — показывает функциональные требования. Элементы: Актёр (пользователь), прецедент (действие), связи. Пример: клиент (актёр) -> "Оформить заказ" (прецедент).

Вопрос 9: Диаграмма классов — статическая модель. Класс содержит имя, атрибуты (данные) и методы (функции). Отношения: наследование, ассоциация, агрегация.

Вопрос 10: IDEF0 — методология функционального моделирования. Состоит из блоков (работ/функций) и стрелок (входы, выходы, управление, механизмы).

Вопрос 11: Activity — показывает поток управления между шагами. Sequence — показывает обмен сообщениями между объектами во времени.

Вопрос 12: CASE-средства: Rational Rose, StarUML, Microsoft Visio. Позволяют строить диаграммы, анализировать архитектуру и автоматически генерировать код.

Вопрос 13: Качество ПО — степень соответствия продукта требованиям. Характеристики: функциональность, надёжность, удобство, эффективность, сопровождаемость, переносимость.

Вопрос 14: Верификация — «правильно ли мы строим продукт» (соответствие ТЗ). Валидация — «тот ли продукт мы строим» (соответствие потребностям пользователя).

Вопрос 15: Аттестация — тестирование системы на соответствие бизнес-требованиям заказчика перед сдачей в эксплуатацию.

Вопрос 16: Тестирование — процесс проверки. Уровни: модульное (проверка функций), интеграционное (проверка связи модулей), системное (полная проверка системы).

Вопрос 17: Тест-кейс — документ с шагами и ожидаемым результатом (ID, предусловия, шаги, ожидание). Чек-лист — упрощённый список пунктов для быстрой проверки.

Вопрос 18: Метрики — измеримые показатели. Плотность дефектов (багов на строку кода), покрытие тестами (какой % кода проверен). Помогают оценить качество и скорость разработки.

Все ответы могут быть сформулированы в другой близкой по смыслу формулировке

6.5 Ключи на вопросы для устного опроса

1. Понятия требований, классификация, уровни требований.

- **Понятие:** Требование — это описание того, что система должна делать (функциональность) и как она должна это делать (ограничения, качество).
- **Уровни:** Бизнес-требования (цели организации), Пользовательские требования (задачи пользователей), Функциональные и нефункциональные требования (конкретные характеристики ПО).
- **Классификация:** Функциональные (что делает система), Нефункциональные (как делает: скорость, безопасность, удобство), Ограничения (технологии, законодательство), Атрибуты качества.

2. Методологии и стандарты, регламентирующие работу с требованиями.

- **Методологии:** Каскадная (Waterfall) — последовательная; Итеративная (Agile, Scrum, Канбан) — гибкая, с короткими спринтами.
- **Стандарты:** ГОСТ 19 (ЕСПД — Единая система программной документации), ГОСТ 34 (комплекс стандартов на АС), а также международные стандарты IEEE 830 (SRS — спецификация требований) и ISO/IEC/IEEE 29148.

3. Современные принципы и методы разработки программных приложений.

- Современные принципы: **Инкапсуляция, наследование, полиморфизм** (ООП). Разделение ответственности (Single Responsibility), модульность, слабое связывание.
- Методы: Итеративная разработка, прототипирование, RAD (быстрая разработка), экстремальное программирование (XP, включающее парное программирование и TDD), рефакторинг, непрерывная интеграция.

4. Методы организации работы в команде разработчиков.

- Распределение ролей: Тимлид, разработчик, тестировщик, аналитик, Product Owner.
- Методы: **Scrum** (спринты, Daily Stand-up, ретроспективы), **Kanban** (визуальные доски, ограничение работы в процессе), **Парное программирование** (два разработчика за одним компьютером), Code Review (взаимная проверка кода).

5. Системы контроля версий.

- Назначение: Отслеживание изменений в коде, возврат к предыдущим версиям, работа нескольких разработчиков одновременно, разрешение конфликтов.
- Типы: Централизованные (SVN) и Децентрализованные/Распределенные (Git).
- Git: основные команды (init, add, commit, push, pull, branch, merge). Gitlab/GitHub — хостинги для удаленного хранения кода.

6. Основные подходы к интегрированию программных модулей.

- **Большого взрыва (Big Bang):** Все модули собираются сразу в конце. Рискованно (много конфликтов).
- **Нисходящее (Top-Down):** Сначала тестируются верхние уровни с заглушками.
- **Восходящее (Bottom-Up):** Сначала интегрируются и тестируются нижние уровни, затем они собираются в более крупные модули.
- **Непрерывная интеграция (CI):** Постоянное слияние кода всех разработчиков в основную ветку несколько раз в день, с автоматической сборкой и тестами.

7. Стандарты кодирования.

- Единые правила написания кода для всей команды:
- Правила именования переменных и функций (смысловые названия, CamelCase или snake_case).
- Форматирование (отступы, переносы строк), комментарии (описание сложных алгоритмов и логики), ограничение сложности функций (не больше 10-20 строк, одна функция — одно действие), единый стиль работы с ошибками.

8. Техническое задание (ТЗ).

- Основной документ, согласно которому разрабатывается ПО. Устанавливает требования и ограничения.
- Основные разделы по ГОСТ 19.201-78: Общие сведения, Назначение разработки, Требования к программе (функциональные и надежности), Требования к программной документации, Техничко-экономические показатели, Стадии и этапы разработки, Порядок контроля и приемки.

9. Архитектура программного средства.

- Это высокоуровневая структура системы: набор компонентов, их взаимодействие и принципы, по которым они строятся.
- Типовые архитектуры: Клиент-серверная, Многоуровневая (N-tier), Микросервисная, Монолитная.
- В ООП часто используется паттерн **MVC** (Model-View-Controller) — разделение данных, интерфейса и логики.

10. Описание требований: унифицированный язык моделирования - краткий словарь.

- **UML (Unified Modeling Language)** — язык графического описания (визуального моделирования) систем.
- **Краткий словарь:**
 - **Актёр** — внешний пользователь системы.
 - **Вариант использования (Use Case)** — функция, которую система предоставляет актёру.
 - **ЫКласс** — описание группы объектов с общими свойствами (атрибутами) и поведением (методами).
 - **Объект** — конкретный экземпляр класса.
 - **Стереотип** — расширение стандартного элемента для уточнения смысла.

11. Диаграммы UML.

- **Диаграмма вариантов использования** — показывает, кто и зачем использует систему (требования).
- **Диаграмма классов** — статическая структура системы (классы, атрибуты, методы, связи).
- **Диаграмма последовательности** — показывает обмен сообщениями между объектами во времени (динамика).
- **Диаграмма деятельности (Activity)** — описывает бизнес-процесс и логику действий.
- **Диаграмма состояний (State)** — описывает жизненный цикл объекта (переходы между состояниями).

12. Описание и оформление требований (спецификация).

- Спецификация требований (SRS) — это официальный текст, фиксирующий все функциональные и нефункциональные требования.
- Документ оформляется в виде таблиц, списков, текстов и приложений. Для описания используются шаблоны: "Система должна...", "Пользователь должен иметь возможность...". Также применяются **Пользовательские истории (User Stories)** в формате: «Как [роль], я хочу [функция], чтобы [ценность]».

13. Анализ требований и стратегии выбора решения.

- **Анализ требований** — проверка на полноту, непротиворечивость, тестируемость, осуществимость и отсутствие двусмысленности.
- **Стратегии выбора:** Метод взвешенных критериев (сравнение вариантов по бальной шкале), выбор готового решения (COTS) против собственной разработки, оценка рисков, анализ сложности и трудозатрат (оценка по точкам или экспертным путем), прототипирование (построение быстрого макета).

14. Цели и задачи и виды тестирования.

- **Цели:** Выявление дефектов, снижение рисков, подтверждение соответствия требованиям, повышение уверенности в качестве.
- **Задачи:** Разработка тест-кейсов, исполнение тестов, регистрация и анализ ошибок.
- **Виды:** По уровню (модульное, интеграционное, системное, приемочное), по степени автоматизации (ручное, автоматизированное), по целям (функциональное, нефункциональное: нагрузочное, стрессовое, безопасность, удобство использования).

15. Стандарты качества программной документации.

- Регулируются ГОСТами серии 19 (ЕСПД), 34.
- Основные стандарты для документации: ГОСТ 19.301 (Программа и методика испытаний), ГОСТ 19.505 (Руководство оператора), ГОСТ 19.402 (Описание программы).
- Требования к качеству: полнота, актуальность, понятность, однозначность, отсутствие противоречий, соответствие самому коду и последней версии продукта.

16. Меры и метрики.

- **Меры (Measures):** Конкретные измеренные значения (например, количество строк кода, количество найденных дефектов).
- **Метрики (Metrics):** Относительные показатели, вычисляемые на основе мер для оценки качества и процесса.

- Примеры метрик: Плотность дефектов (дефекты на 1000 строк кода), коэффициент покрытия тестами (в %), цикломатическая сложность (сложность логики), время на исправление дефекта, процент автоматизированных тестов.

17. Тестовое покрытие. Тестовый сценарий, тестовый пакет.

- **Тестовое покрытие** — процент функциональности или кода, проверенный тестами (покрытие требований, покрытие операторов/ветвей кода).
- **Тестовый сценарий (Test Scenario)** — это идея/гипотеза о том, что нужно проверить (например, "Проверить вход в систему").
- **Тестовый пакет (Test Suite / Test Pack)** — набор (набор) тест-кейсов, объединенных общим функционалом, целью или модулем (например, "Пакет тестов авторизации").

18. Анализ спецификаций.

- Это этап тестирования, который начинается до написания кода (Static Testing).
- Аналитик и тестировщик внимательно читают ТЗ, ищут логические противоречия, неоднозначные формулировки, упущенные граничные условия (например, "Введите число" — а что, если введут 0 или буквы?).
- По итогам анализа создается **Матрица трассировки (Traceability Matrix)** — таблица, где каждое требование связывается с конкретными тест-кейсами. Это помогает убедиться, что ни одно требование не осталось непроверенным.

Все ответы могут быть сформулированы в другой близкой по смыслу формулировке