

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ**

*МДК.02.02 Инструментальные средства 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.02.02 Инструментальные средства 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.02.02 Инструментальные средства 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	Использовать современные IDE для разработки, отладки и тестирования кода. Применять системы контроля версий для командной работы.	Назначение и классификацию инструментальных средств разработки. Архитектуру и функциональные возможности популярных IDE.
ОК 4 ОК 5	Использовать средства автоматической сборки и управления зависимостями. Проводить статический и динамический анализ кода с помощью специализированных инструментов.	Принципы работы систем контроля версий и основы Git. Инструменты для автоматизации сборки и CI/CD.
ОК 9 ОК 10	Создавать и использовать UML-диаграммы в CASE-средствах. Работать с базами данных через инструменты администрирования.	Средства профилирования и отладки Инструменты для тестирования Средства моделирования и документирования
ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6	Выбирать и настраивать инструментарий для конкретного этапа разработки. Интегрировать инструменты в процесс разработки для повышения производительности. Автоматизировать рутинные задачи с помощью скриптов и инструментов сборки. Оценивать эффективность инструментов и обосновывать выбор. Документировать конфигурации и процедуры работы с инструментами.	Стандарты и регламенты использования инструментальных средств в команде. Инструменты для управления требованиями и задачами. Средства непрерывной интеграции и доставки. Инструменты для анализа кода. Методологии DevOps и инструментарий для контейнеризации.

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	87
в том числе:	
– теоретическое обучение	34
– практические занятия (<i>если предусмотрено</i>)	50
– самостоятельная работа	
– консультации	
– промежуточная аттестация – (<i>дифференцируемый зачет</i>)	3

2.2 Тематический план и содержание учебной дисциплины «МДК.02.02 Инструментальные средства 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	Содержание учебного материала: Назначение и функции Инструментальных средств разработки программного обеспечения. Основные понятия: программа, программное обеспечение, задачи и приложения. Технологические и функциональные задачи, группы компьютерных пользователей, сопровождение программ. Классификация инструментальных средств разработки ПО.	8	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
	Практическая работа	Не предусмотрено	
Тема 1.2. Защита программного обеспечения	Содержание учебного материала: Защита ПО. Виды воздействий, методы защиты программных продуктов. Правовая защита, Авторское право. Методы маркетинга ПП.	6	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
	Практическая работа	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 1.3. Понятие технологии разработки программного обеспечения.	Содержание учебного материала: Стадии жизненного цикла ПО: моделирование, анализ требований, анализ и проектирование, кодирование, тестирование, отладка, установка и сопровождение. Модели и технологии разработки ПП. Использование инструментальных средств при проектировании программного обеспечения методами – индивидуальный, командный, модель зрелости возможностей	8	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5 ПК 3.6
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
	Практическая работа	Не предусмотрено	
Тема 1.4. Пользовательский интерфейс	Содержание учебного материала: Понятие пользовательского интерфейса. Инструментальные средства создания интерфейса пользователя. Принципы построения интерфейсов. Требования, предъявляемые к стандартному графическому интерфейсу пользователя	2	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5 ПК 3.6
	Практическая работа №1: Изучение видов интерфейсов ПП. Проектирование пользовательского интерфейса.	7	
	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 1.5. Общая характеристика инструментальных средств разработки программных продуктов	Содержание учебного материала: Стили и языки программирования. Понятие модели. Структурный подход к проектированию. CASE-средства. Функциональные возможности и характеристика. Примеры CASE-технологии	4	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6
	Практическая работа №2: 1. Анализ инструментов разработки ПО, CASE – технологий 2. Разработка UI с помощью диаграмм	7	
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
Тема 1.6. Разработка Web-приложений	Содержание учебного материала: Web-технологии. Языки создания web-приложений. Основы HTML. Основные объекты Понятие верстки. CMS - системы	6	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6
	Практическая работа №3: Создание HTML-страниц Практическая работа №4: Проектирование и создание макета сайта Практическая работа №4: Создание приложений с элементами управления HTML. Практическая работа №5 : Использование таблиц стилей для оформления HTML-страниц	36	
	Самостоятельная работа обучающихся	Не предусмотрено	
	Лабораторное занятие	Не предусмотрено	
Промежуточная аттестация - дифференцированный зачет		3	

1	2	3	4
ИТОГО часов		87	
Теоретические занятия		34	
Практические занятия		50	
Самостоятельная работа			
Дифференцированный зачет		3	

3 Условия реализации учебной дисциплины

3.1 Для реализации программы учебной дисциплины «МДК.02.02 Инструментальные средства 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, 2024 .- 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, 2025. — 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.jip.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, тестирование

- 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 компьютерных сетей, принципы пакетной передачи данных, 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 автоматизированной	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; Промежуточная аттестация: тестирование

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.02.02 Инструментальные средства 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.02.02 Инструментальные средства 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.02.02 «Инструментальные средства 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	31	Назначение и классификация инструментальных средств разработки ПО (IDE, системы контроля версий, сборщики, отладчики)
ОК 5 ОК 9	32	Архитектуру и функциональные возможности популярных IDE (Visual Studio, IntelliJ IDEA, VS Code, PyCharm)
ОК 10 ПК 3.1	33	Принципы работы систем контроля версий (Git, SVN) и основы CI/CD (Jenkins, GitLab CI)
ПК 3.2 ПК 3.3	34	Инструменты для тестирования, профилирования и статического анализа кода (JUnit, Selenium, SonarQube, Checkstyle)
ПК 3.4 ПК 3.5	35	Инструменты для документирования, моделирования и управления задачами (Enterprise Architect, Jira, PlantUML)
ПК 3.6	У1	Использовать современные IDE для разработки, отладки и рефакторинга кода
	У2	Применять Git для управления версиями и совместной разработки
	У3	Использовать средства автоматической сборки и управления зависимостями (Maven, Gradle, npm)
	У4	Интегрировать инструменты в процесс разработки, настраивать их для повышения производительности

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	31	Способность классифицировать инструментальные средства разработки ПО по назначению	<i>Реферат (п 5.2 темы 1-10)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	32	Способность описывать архитектуру и функционал современных IDE		
	У1	Способность использовать IDE для написания и отладки кода		
Тема 1.2. Защита программного обеспечения	33	Способность объяснять принципы работы Git и CI/CD	<i>Устный опрос (п 5.1 вопросы 1-3) Реферат (п 5.2 темы 11-20)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У2	Способность применять Git для управления версиями в команде		
Тема 1.3. Понятие технологии разработки программного обеспечения.	34	Способность описывать инструменты сборки (Maven, Gradle, npm)	<i>Устный опрос (п 5.1 вопросы 4-7) Реферат (п 5.2 темы 21-30)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У3	Способность настраивать сборку проекта с использованием систем автоматизации		
Тема 1.4. Пользовательский интерфейс	35	Способность перечислять инструменты для тестирования (JUnit, Selenium, Postman) и статического анализа (SonarQube)	<i>Устный опрос (п 5.1 вопросы 8-11) Практическая работа №1 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У4	Способность выполнять тестирование и анализ кода с использованием специализированных инструментов		
Тема 1.5. Общая характеристика инструментальных средств разработки программных продуктов	31	Способность классифицировать инструментальные средства разработки ПО по назначению	<i>Устный опрос (п 5.1 вопросы 12-15) Практическая работа №2 (п 5.3)</i>	<i>Устный опрос (п 6.1 вопросы 1-18) Тест вариант №1-2 (п 6.2)</i>
	У1	Способность использовать IDE для написания и отладки кода		

Тема 1.6. Разработка Web- приложени й	У2	Способность применять Git для управления версиями в веб-проектах	<i>Устный опрос (п 5.1 вопросы 16-18)</i>	<i>Устный опрос (п 6.1 вопросы 1-18)</i>
	У4	Способность использовать инструменты для тестирования веб-приложений (Selenium, Postman)	<i>Практическая работа №3-5 (п 5.3)</i>	<i>Тест вариант №1-2 (п 6.2)</i>

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 программное обеспечение (ПО)? Каково его основное назначение?
2. Назовите и охарактеризуйте три основных класса программного обеспечения (системное, прикладное, инструментальное).
3. Что такое программа, алгоритм и данные? Как они связаны между собой?
4. Что такое жизненный цикл программного обеспечения? Перечислите его основные стадии.
5. Дайте определение следующим понятиям: требование, спецификация, архитектура, реализация, тестирование.
6. Каковы основные цели и задачи защиты программного обеспечения?
7. Что такое несанкционированное использование и распространение ПО? Какие существуют правовые механизмы борьбы с этим (лицензии, авторское право)?
8. Что такое программные вирусы, трояны, шпионское ПО? Каковы основные методы защиты от них?
9. Объясните, для чего используются следующие механизмы защиты: лицензионные ключи, аппаратные ключи, обфускация кода.
10. Что такое уязвимость программного обеспечения и патч? Почему важно регулярно обновлять ПО? Что подразумевается под технологией разработки программного обеспечения?
12. Опишите суть каскадной модели жизненного цикла ПО ("водопад"). Каковы ее сильные и слабые стороны?
13. В чем заключаются основные принципы гибких методологий разработки (Agile), таких как Scrum или Kanban?
14. Что такое система контроля версий (например, Git) и какую роль она играет в современной разработке?
15. Что такое тестирование ПО? Какие виды тестирования вы знаете (модульное, интеграционное, системное)?
16. Что такое пользовательский интерфейс (UI)? Почему его качество критически важно для успеха программного продукта?
17. Чем отличается пользовательский интерфейс (UI) от пользовательского опыта (UX)?
18. Назовите основные принципы и правила хорошего дизайна пользовательского интерфейса.

5.2 Темы рефератов

Тема 1.1. Основные понятия

1. Классификация и назначение инструментальных средств разработки программного обеспечения.
2. Современные интегрированные среды разработки (IDE): обзор и сравнительный анализ.

3. CASE-средства: роль и место в процессе проектирования программных продуктов.
4. Инструментальные средства для работы с базами данных (ERwin, PowerDesigner и др.).
5. Инструменты автоматизации сборки и управления зависимостями проектов.
6. Средства документирования и генерации отчетности в разработке ПО.
7. Эволюция языков программирования и инструментальных средств разработки.
8. Системы контроля версий (Git, SVN) как обязательный инструмент современной разработки.
9. Группы пользователей программного обеспечения и их влияние на выбор средств разработки.
10. Технологические и функциональные задачи, решаемые с помощью инструментальных средств.

Тема 1.2. Защита программного обеспечения

11. Правовая защита программного обеспечения: авторское право и смежные права.
12. Основные методы защиты программных продуктов от несанкционированного доступа.
13. Криптографические методы защиты информации в программных продуктах.
14. Аппаратные и программные методы защиты от взлома и пиратства.
15. Виды лицензий на программное обеспечение (Freeware, Shareware, Open Source) и их особенности.
16. Методы маркетинга программных продуктов и их влияние на рынок ПО.
17. Анализ уязвимостей программного обеспечения: методы тестирования на проникновение.
18. Технические и организационные меры защиты программного обеспечения на предприятии.
19. Электронные подписи и сертификаты как инструмент защиты ПО и данных.
20. Вредоносное ПО: классификация и методы борьбы с ним.

Тема 1.3. Понятие технологии разработки программного обеспечения

21. Жизненный цикл программного обеспечения: стадии и модели.
22. Каскадная модель разработки ПО: преимущества и ограничения.
23. Спиральная модель разработки: управление рисками и итерациями.
24. Гибкие методологии (Agile, Scrum, Kanban): практика применения в ИТ-командах.
25. Индивидуальная и командная разработка ПО: подходы и особенности.
26. Модель зрелости возможностей (СММІ) и её уровни.
27. Технология прототипирования в разработке программного обеспечения.
28. Инструменты и методы анализа требований и проектирования архитектуры.
29. Отладка, тестирование и сопровождение программных продуктов: инструментарий.
30. Влияние инструментальных средств на качество и сроки разработки ПО.

5.3 Примеры практических работ

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: Изучение видов интерфейсов ПП. Проектирование пользовательского интерфейса

Цель:

Изучить основные виды пользовательских интерфейсов (GUI, CLI, веб-интерфейс, мобильный) и освоить базовые принципы проектирования UI с использованием инструментов прототипирования.

Задание:

1. Изучить классификацию интерфейсов (графический, командная строка, голосовой, жестовый и т.д.).
2. Для заданной предметной области (например, «Онлайн-магазин», «Библиотека», «Система управления задачами») разработать концепцию пользовательского интерфейса:
 - Определить целевую аудиторию и сценарии использования.
 - Выбрать подходящий тип интерфейса и обосновать выбор.
 - Нарисовать эскиз (wireframe) главного экрана и одного диалогового окна.
 - Создать прототип в Figma, Balsamiq или [Draw.io](https://draw.io) (или на бумаге).
3. Описать требования к интерфейсу (цветовая схема, шрифты, расположение элементов, навигация).

4. Подготовить отчёт.

Содержание отчёта:

- Описание предметной области и целевой аудитории.
- Выбор типа интерфейса с обоснованием.
- Эскизы (скриншоты) главного экрана и диалога.
- Описание навигации и взаимодействия.
- Выводы.

Контрольные вопросы:

1. Какие виды пользовательских интерфейсов вы знаете?
2. В чём разница между GUI и веб-интерфейсом?
3. Какие инструменты используются для создания прототипов?
4. Какие требования предъявляются к удобству интерфейса?

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: Принципы построения интерфейсов. Требования к графическому интерфейсу

Цель:

Изучить основные принципы построения качественных пользовательских интерфейсов и научиться проверять интерфейс на соответствие стандартам и требованиям.

Задание:

1. Изучить принципы построения интерфейсов (простота, понятность, обратная связь, последовательность, доступность, привычность).
2. Выбрать существующее приложение (например, интернет-банк, мобильное приложение, веб-сервис).
3. Провести анализ его интерфейса по чек-листу:
 - Соответствие принципам (оценить каждый принцип по 5-балльной шкале с комментариями).
 - Выявить недостатки.
 - Предложить улучшения (нарисовать или описать, как должно быть).
4. Оформить отчёт в виде анализа с рекомендациями.

Содержание отчёта:

- Описание выбранного приложения.
- Чек-лист с оценками и комментариями.
- Выявленные недостатки с иллюстрациями.
- Предложения по улучшению (эскизы или описание).
- Выводы.

Контрольные вопросы:

1. Назовите основные принципы хорошего дизайна UI.
2. Что такое доступность (accessibility) интерфейса?
3. Как проверить интерфейс на удобство?
4. Какие стандарты регулируют разработку пользовательских интерфейсов?

ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: Создание HTML-страниц

Цель:

Освоить базовые конструкции HTML для создания веб-страниц: структура документа, заголовки, абзацы, списки, таблицы, гиперссылки, изображения.

Задание:

Создать HTML-страницу с личной информацией или визиткой, которая содержит:

- Заголовок страницы (<title>).
- Заголовки разных уровней (<h1>–<h3>).
- Абзацы текста (<p>).

- Нумерованный и маркированный списки (`<ol>`, `<ul>`).
- Таблицу (например, расписание, список навыков или контакты).
- Гиперссылки (на соцсети, портфолио).
- Изображение (можно заглушку).

Порядок выполнения:

1. Создать файл `index.html` в любом редакторе (VS Code, Notepad++).
2. Написать базовую структуру: `<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`.
3. Добавить все требуемые элементы.
4. Проверить страницу в браузере (открыть файл).
5. Внести корректировки при необходимости.

Содержание отчёта:

- Листинг HTML-кода (можно скриншот или вставить текст).
- Скриншот страницы в браузере.
- Краткое описание каждого элемента.

Контрольные вопросы:

1. Что такое HTML и для чего он используется?
2. Какие теги используются для создания списков и таблиц?
3. Как вставить изображение на страницу?
4. Чем отличается `<ol>` от `<ul>`?

ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: Проектирование и создание макета сайта

Цель:

Научиться создавать многостраничный макет сайта с использованием HTML и базовых принципов вёрстки (div-контейнеры, семантические теги).

Задание:

Создать многостраничный сайт-визитку из 3–4 страниц (Главная, Обо мне, Услуги/Портфолио, Контакты):

1. Продумать структуру сайта (меню, шапка, основное содержимое, подвал).
2. Для каждой страницы создать отдельный HTML-файл.
3. Использовать семантические теги: `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<footer>`.
4. Создать единое меню на всех страницах (ссылки друг на друга).

Порядок выполнения:

1. Спроектировать структуру на бумаге (карта сайта).
2. Создать папку проекта.
3. В папке создать файлы: `index.html`, `about.html`, `services.html`, `contacts.html`.
4. В каждом файле прописать базовую структуру и контент.
5. Сделать меню (список ссылок) в `<nav>` и разместить его на всех страницах.
6. Проверить работу ссылок (переходы между страницами).
7. Оформить отчёт.

Содержание отчёта:

- Описание структуры сайта (карта сайта или схема).
- Скриншоты всех страниц.
- Листинг HTML-кода (или ссылка на проект).
- Выводы.

Контрольные вопросы:

1. Какие семантические теги HTML5 вы использовали?
2. Для чего нужен `<nav>`?
3. Как организовать навигацию между страницами?
4. Чем отличается `<div>` от `<section>`?

ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: Использование таблиц стилей для оформления HTML-страниц

Цель:

Освоить базовые приёмы CSS для оформления веб-страниц: работа с цветами, шрифтами, фонами, отступами, рамками, позиционированием.

Задание:

Взять HTML-страницы из работы №4 (или создать новую) и оформить их с помощью CSS:

1. Подключить внешний файл стилей style.css.
2. Задать:
 - Цвет фона и текста для всего сайта.
 - Шрифт (подключить Google Fonts).
 - Стили для заголовков, абзацев, ссылок.
 - Оформить таблицу (цвет, границы).
 - Добавить отступы и рамки для блоков.
 - Сделать меню горизонтальным (flexbox).
3. Добавить hover-эффекты для ссылок.

Порядок выполнения:

1. Создать файл style.css в той же папке.
2. Подключить его в <head> всех страниц: <link rel="stylesheet" href="style.css">.
3. Прописать базовые стили для тегов (body, h1–h3, p, a, ul, table).
4. Создать классы для оформления блоков (например, .container, .card, .menu).
5. Применить стили к элементам.
6. Проверить в браузере и доработать.

Содержание отчёта:

- Скриншот страниц с новым оформлением.
- Листинг CSS-кода.
- Краткое описание использованных приёмов.

Контрольные вопросы:

1. Какие способы подключения CSS вы знаете?
2. Как задать цвет фона и текста?
3. Что такое селекторы и какие они бывают?
4. Как работает flexbox для создания меню?

ПРАКТИЧЕСКАЯ РАБОТА №6

Тема: Создание сайта с помощью CMS WordPress

Цель:

Освоить базовые навыки работы с системой управления контентом WordPress: установка, выбор темы, создание страниц, публикация контента.

Задание:

1. Установить WordPress локально (OpenServer, XAMPP или на хостинге).
2. Выбрать и установить тему (можно бесплатную из каталога).
3. Создать структуру сайта:
 - Главная страница.
 - Страница «О нас».
 - Страница «Услуги».
 - Страница «Контакты».
4. Настроить меню и добавить страницы в меню.
5. Добавить несколько записей (новостей или статей) в блог.
6. Настроить виджеты (сайдбар, подвал).
7. Сделать скриншоты и подготовить отчёт.

Порядок выполнения:

1. Установить локальный сервер (OpenServer или XAMPP).
2. Скачать WordPress и установить его.
3. Зайти в админ-панель (/wp-admin).
4. Выбрать тему (Внешний вид → Темы → Добавить новую → установить).
5. Создать страницы (Страницы → Добавить новую).
6. Настроить меню (Внешний вид → Меню).
7. Создать рубрику и добавить записи.
8. Настроить виджеты (Внешний вид → Виджеты).
9. Проверить отображение сайта на фронтенде.
10. Оформить отчёт.

Содержание отчёта:

- Скриншоты админ-панели (страницы, меню, виджеты).
- Скриншоты готового сайта (главная, страницы, записи).
- Краткое описание выполненных действий.
- Выводы (что освоено, какие трудности возникли).

Контрольные вопросы:

1. Что такое CMS и зачем она нужна?
2. В чём преимущества WordPress?
3. Как создать страницу и добавить её в меню?
4. Чем отличается страница от записи в WordPress?
5. Какие плагины могут понадобиться для расширения функционала?

6. Оценочные средства для проведения промежуточной аттестации

6.1 Вопросы для собеседования при проведении дифференцированного зачета

Тема 1.1. Основные понятия

1. Дайте определение понятиям «программа», «программное обеспечение» и «приложение». В чем их принципиальное отличие?
2. Перечислите основные технологические и функциональные задачи, которые решаются с помощью инструментальных средств разработки ПО.
3. По какому принципу классифицируются инструментальные средства разработки программного обеспечения? Приведите примеры основных групп.

Тема 1.2. Защита программного обеспечения

4. Какие виды воздействий на программное обеспечение вы знаете? Опишите механизм каждого из них.
5. Перечислите основные методы защиты программных продуктов. В чем разница между аппаратными и программными методами?
6. Что такое авторское право в контексте ПО? Какие методы маркетинга применяются для продвижения программных продуктов?

Тема 1.3. Понятие технологии разработки программного обеспечения

7. Опишите основные стадии жизненного цикла программного обеспечения (от моделирования до сопровождения).
8. Сравните две модели разработки ПО: индивидуальную и командную. В чем их специфика?
9. Что такое «модель зрелости возможностей» (СММ)? Назовите уровни зрелости и их значение.

Тема 1.4. Пользовательский интерфейс

10. Дайте определение понятию «пользовательский интерфейс». Какие инструментальные средства используются для его создания?
11. Перечислите основные принципы построения эффективных пользовательских интерфейсов.
12. Какие требования предъявляются к стандартному графическому интерфейсу пользователя (GUI)?

Тема 1.5. Общая характеристика инструментальных средств разработки

13. Что такое «модель» в контексте проектирования ПО? Какие подходы к проектированию вы знаете?
14. Что такое CASE-средства? Перечислите их функциональные возможности и характеристики.
15. Приведите примеры известных CASE-технологий. Для каких задач они применяются?

Тема 1.6. Разработка Web-приложений

16. Охарактеризуйте современные Web-технологии. Какие языки используются для создания Web-приложений?
17. Что такое HTML и «верстка»? Какие основные объекты HTML вы знаете?
18. Для чего нужны CMS-системы? Приведите примеры и опишите принцип их работы.

6.2 Тестовые материалы

Вариант №1

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

1. Что из перечисленного относится к системному программному обеспечению?

- А) Текстовый редактор
- Б) Операционная система
- В) Электронная таблица
- Г) Графический редактор

2. Какой класс ПО предназначен для создания, отладки и тестирования программ?

- А) Системное
- Б) Прикладное
- В) Инструментальное
- Г) Сервисное

3. Что такое алгоритм?

- А) Готовая программа
- Б) Последовательность действий для решения задачи
- В) Набор данных
- Г) Инструкция пользователя

Часть 2. Установление соответствия (вопросы 4–6)

4. Установите соответствие между классом ПО и его примером.

Класс ПО	Пример
----------	--------

1. Системное	А. Microsoft Office
2. Прикладное	Б. Windows 11
3. Инструментальное	В. Visual Studio

Ответ запишите в виде сочетания цифр и букв (например, 1-Б, 2-А, 3-В).

5. Установите соответствие между механизмом защиты и его описанием.

Механизм защиты	Описание
1. Лицензионный ключ	А. Физическое устройство для активации ПО
2. Аппаратный ключ	Б. Уникальный код для проверки легальности использования
3. Обфускация кода	В. Затруднение обратной разработки и анализа кода

6. Установите соответствие между стадией жизненного цикла ПО и её содержанием.

Стадия ЖЦ ПО	Содержание
1. Анализ требований	А. Написание программного кода
2. Реализация	Б. Выявление потребностей заказчика
3. Сопровождение	В. Исправление ошибок и обновление ПО

Часть 3. Установление последовательности (вопросы 7–9)

7. Расставьте в правильной последовательности стадии каскадной модели (Waterfall):

1. Тестирование
2. Проектирование
3. Сопровождение
4. Реализация
5. Анализ требований
6. Внедрение

Ответ: _____ (цифры через запятую)

8. Расставьте в правильном порядке этапы работы с Git в команде:

1. Создание новой ветки (branch)
2. Фиксация изменений (commit)
3. Отправка в удалённый репозиторий (push)
4. Создание Pull Request
5. Слияние веток (merge)

9. Установите последовательность действий при защите ПО от несанкционированного использования:

1. Выбор механизма защиты (ключи, обфускация)
2. Анализ угроз и рисков
3. Интеграция защиты в продукт
4. Оценка эффективности защиты

Часть 4. Открытые вопросы (вопросы 10–12)

10. Опишите разницу между системным, прикладным и инструментальным программным обеспечением. Приведите по одному примеру для каждого класса.

11. Какие правовые и технические методы защиты программного обеспечения вы знаете? Назовите не менее трёх и кратко опишите каждый.

12. Что такое система контроля версий (например, Git) и какую роль она играет в современной разработке ПО?

Вариант №2

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

1. Что из перечисленного относится к прикладному программному обеспечению?

А) Компилятор

- Б) Операционная система
- В) Текстовый редактор
- Г) Отладчик

2. Какой класс ПО включает в себя компиляторы, отладчики и системы контроля версий?

- А) Системное
- Б) Прикладное
- В) Инструментальное
- Г) Базовое

3. Что такое программа?

- А) Алгоритм, записанный на языке программирования
- Б) Последовательность действий
- В) Набор данных
- Г) План работы

Часть 2. Установление соответствия (вопросы 4–6)

4. Установите соответствие между инструментом и его назначением.

Инструмент	Назначение
1. Git	А. Интегрированная среда разработки
2. Visual Studio	Б. Система контроля версий
3. JUnit	В. Фреймворк для модульного тестирования

5. Установите соответствие между термином и его определением.

Термин	Определение
1. Архитектура ПО	А. Описание того, что система должна делать
2. Требование	Б. Фундаментальная организация системы
3. Патч	В. Исправление, устраняющее уязвимость или ошибку

6. Установите соответствие между методом защиты и его описанием.

Метод защиты	Описание
1. Лицензирование	А. Физическое устройство для активации ПО
2. Аппаратная защита	Б. Заключение лицензионного соглашения с пользователем
3. Патчинг	В. Установка исправлений для устранения уязвимостей

Часть 3. Установление последовательности (вопросы 7–9)

7. Расставьте в правильной последовательности этапы разработки ПО:

1. Тестирование
2. Сбор требований
3. Внедрение
4. Сопровождение
5. Проектирование
6. Реализация

8. Расставьте в правильном порядке этапы обработки найденной уязвимости в ПО:

1. Уведомление разработчика о найденной уязвимости
2. Воспроизведение и анализ уязвимости
3. Разработка патча
4. Тестирование патча
5. Выпуск и распространение обновления

9. Установите последовательность действий при использовании Git:

1. Клонирование репозитория
2. Создание новой ветки
3. Внесение изменений и коммит
4. Отправка изменений (push)

5. Создание Pull Request и слияние

Часть 4. Открытые вопросы (вопросы 10–12)

10. В чём разница между каскадной (Waterfall) и гибкой (Agile) моделью разработки ПО? Назовите основные преимущества и недостатки каждой.

11. Какие виды вредоносного программного обеспечения вы знаете и как от них защищаться? Назовите не менее трёх видов и методы защиты.

12. Что такое обфускация кода и для каких целей она применяется при защите программного обеспечения?

6.3 Ключи к тестовым заданиям для проведения промежуточной аттестации

Вариант №1

Часть 1 (выбор ответа):

1 – Б, 2 – В, 3 – Б

Часть 2 (соответствие):

4 – 1-Б, 2-А, 3-В

5 – 1-Б, 2-А, 3-В

6 – 1-Б, 2-А, 3-В

Часть 3 (последовательность):

7 – 5, 2, 4, 1, 6, 3

8 – 1, 2, 3, 4, 5

9 – 2, 1, 3, 4

Вариант №2

Часть 1 (выбор ответа):

1 – В, 2 – В, 3 – А

Часть 2 (соответствие):

4 – 1-Б, 2-А, 3-В

5 – 1-Б, 2-А, 3-В

6 – 1-Б, 2-А, 3-В

Часть 3 (последовательность):

7 – 2, 5, 6, 1, 3, 4

8 – 1, 2, 3, 4, 5

9 – 1, 2, 3, 4, 5

Эталонные ответы на открытые вопросы

Вариант №1

10.

- Системное ПО (ОС Windows, Linux) – управляет ресурсами компьютера, обеспечивает работу других программ.

- Прикладное ПО (Microsoft Office, 1C) – решает конкретные задачи пользователя.

- Инструментальное ПО (Visual Studio, Git) – используется разработчиками для создания, отладки и тестирования программ.

11.

- Правовые: авторское право, лицензии (EULA), патенты – юридически защищают интеллектуальную собственность.

- Технические: лицензионные ключи (код активации), аппаратные ключи (USB-токен), обфускация кода (затруднение анализа), шифрование данных.

12.

Система контроля версий – инструмент для отслеживания изменений в коде. Роль: хранение истории, совместная работа, ветвление и слияние, резервное копирование, Code Review.

Ответ может быть сформулирован в другой близкой по смыслу формулировке

Вариант №2

10.

- Каскадная модель – строгая последовательность этапов, возврат назад сложен. **Плюсы:** дисциплина, предсказуемость. **Минусы:** жёсткость, результат виден только в конце.

- Agile – итеративная разработка, быстрая реакция на изменения. **Плюсы:** гибкость, вовлечение заказчика. **Минусы:** сложность управления, не всегда подходит для крупных проектов.

11.

- Вирусы – самокопируются, внедряются в файлы.
- Трояны – маскируются под полезные программы.
- Шпионское ПО – собирает конфиденциальную информацию.

Методы защиты: антивирусы, брандмауэры, регулярные обновления, обучение пользователей, резервное копирование.

12.

Обфускация кода – преобразование исходного кода в трудночитаемую форму (переименование переменных, добавление мусорного кода). Цель: затруднить обратную разработку, анализ и несанкционированное изменение программы.

Ответ может быть сформулирован в другой близкой по смыслу формулировке

6.4 Ключи к вопросам для собеседования при проведении дифференцированного зачета

- **Вопрос 1:** Программа — набор инструкций для ЭВМ. ПО — совокупность программ, документации и данных. Приложение — прикладная программа для решения конкретной задачи пользователя.
- **Вопрос 2:** Автоматизация процессов, отладка и тестирование, управление версиями, документирование, работа с базами данных, создание интерфейсов.
- **Вопрос 3:** По назначению (средства анализа, проектирования, разработки, тестирования), по уровням (низкоуровневые и высокоуровневые), по типу лицензий.
- **Вопрос 4:** Вирусы (разрушение данных), взлом (несанкционированный доступ), логические бомбы (срабатывание по условию), перехват трафика.
- **Вопрос 5:** Программные (шифрование, лицензирование, привязка к железу) и аппаратные (ключи, донглы, платы).
- **Вопрос 6:** Авторское право защищает код как литературное произведение. Маркетинг: демо-версии, Open Source, freemium, реклама, корпоративные продажи.
- **Вопрос 7:** Моделирование -> Анализ требований -> Анализ и проектирование -> Кодирование -> Тестирование -> Отладка -> Установка -> Сопровождение.
- **Вопрос 8:** Индивидуальный — один человек проходит все этапы (гибко, но ограниченно). Командный — распределение ролей (эффективно для больших проектов).
- **Вопрос 9:** CMMI — модель зрелости возможностей. Уровни от 1 (нестабильный процесс) до 5 (оптимизируемый процесс).

- **Вопрос 10:** Интерфейс — способ взаимодействия пользователя с системой. Инструменты: Visual Studio, Delphi, Figma, Qt Designer.
- **Вопрос 11:** Принципы: ориентированность на пользователя, интуитивность, обратная связь, отмена действий, минимизация ошибок.
- **Вопрос 12:** Единый стиль, логичное расположение элементов, читабельность текста, отсутствие перегрузки, поддержка мыши и клавиатуры.
- **Вопрос 13:** Модель — упрощенное описание системы. Подходы: структурный (процедурный) и объектно-ориентированный.
- **Вопрос 14:** CASE — средства автоматизированного проектирования. Функции: визуальное моделирование (UML), генерация кода, управление требованиями, документирование.
- **Вопрос 15:** Примеры: Rational Rose, StarUML, BPwin, ERwin, Visual Paradigm. Применяются для анализа требований, построения архитектуры и БД.
- **Вопрос 16:** Web-технологии: HTML, CSS, JavaScript, серверные языки (PHP, Python, Java). Используются для создания сайтов и веб-сервисов.
- **Вопрос 17:** HTML — язык разметки гипертекста. Верстка — расположение элементов на странице. Основные объекты: заголовки, абзацы, изображения, ссылки, формы.
- **Вопрос 18:** CMS — система управления содержимым. Позволяет редактировать контент без знания кода (WordPress, Joomla, 1С-Битрикс).

6.5 Ключи на вопросы для устного опроса

1. Что такое программное обеспечение (ПО)? Каково его основное назначение?

Программное обеспечение (ПО) – это совокупность программ, данных и документации, обеспечивающих функционирование компьютерной системы и решение задач пользователя.

Основное назначение ПО:

- Преобразование аппаратных ресурсов компьютера в функциональные возможности для пользователя.
- Автоматизация процессов обработки информации.
- Управление работой технических устройств.
- Обеспечение интерфейса между человеком и машиной.

2. Назовите и охарактеризуйте три основных класса программного обеспечения.

Класс	Характеристика	Примеры
ПО Системное	Обеспечивает работу компьютера как единого целого, управляет аппаратными ресурсами и предоставляет интерфейс для прикладного ПО.	ОС (Windows, Linux), драйверы, утилиты, компиляторы.
ПО Прикладное	Предназначено для решения конкретных задач пользователя (офисные, бухгалтерские, графические, игры).	Microsoft Office, 1С, Adobe Photoshop, браузеры.
Инструментальное ПО	Используется разработчиками для создания, отладки, тестирования и сопровождения программ.	IDE (Visual Studio, IntelliJ IDEA), Git, компиляторы, отладчики, системы CI/CD.

3. Что такое программа, алгоритм и данные? Как они связаны между собой?

- **Алгоритм** – последовательность чётко определённых действий, ведущих к решению задачи. Это «рецепт» решения.
- **Программа** – алгоритм, записанный на языке программирования, понятном компьютеру (исполняемый код).
- **Данные** – информация, которая обрабатывается программой в соответствии с алгоритмом.

Связь: Программа реализует алгоритм и оперирует данными. Без алгоритма программа бессмысленна, без данных – бесполезна. Триада «алгоритм – программа – данные» – основа любой информационной системы.

4. Что такое жизненный цикл программного обеспечения? Перечислите его основные стадии.

Жизненный цикл ПО – это последовательность всех этапов, которые проходит ПО от момента возникновения идеи до полного вывода из эксплуатации.

Основные стадии (по стандарту ISO/IEC 12207):

1. **Анализ и сбор требований** – выявление потребностей заказчика.
2. **Проектирование** – разработка архитектуры, структуры, интерфейсов.
3. **Реализация (кодирование)** – написание кода.
4. **Тестирование** – проверка на соответствие требованиям и выявление ошибок.
5. **Внедрение** – установка и запуск в эксплуатацию.
6. **Сопровождение** – исправление ошибок, обновления, доработки.
7. **Вывод из эксплуатации** – завершение использования и архивация.

5. Дайте определение следующим понятиям: требование, спецификация, архитектура, реализация, тестирование.

Термин	Определение
--------	-------------

Требование	Описание функциональных или нефункциональных характеристик системы, которым она должна соответствовать.
Спецификация	Формальный документ, детально описывающий требования, структуру и поведение системы.
Архитектура	Фундаментальная организация системы, представленная в виде её компонентов, их связей и принципов проектирования.
Реализация	Процесс написания программного кода на основе проектной документации (кодирование).
Тестирование	Процесс проверки программного продукта на соответствие требованиям, поиск ошибок и оценка качества.

6. Каковы основные цели и задачи защиты программного обеспечения?

Цели защиты ПО:

- Обеспечение целостности, конфиденциальности и доступности ПО и данных.
- Защита интеллектуальной собственности разработчика.
- Предотвращение несанкционированного использования, копирования и распространения.

Задачи защиты:

- Аутентификация пользователей (проверка подлинности).
- Контроль доступа к функциям и данным.
- Шифрование данных при хранении и передаче.
- Обнаружение и устранение уязвимостей.
- Мониторинг и аудит действий пользователей.
- Защита от вредоносного ПО и атак.

7. Что такое несанкционированное использование и распространение ПО? Какие существуют правовые механизмы борьбы с этим?

Несанкционированное использование – эксплуатация ПО без получения необходимых лицензий или с нарушением условий лицензионного соглашения. **Несанкционированное распространение** – копирование и передача ПО третьим лицам без разрешения правообладателя.

Правовые механизмы защиты:

- **Авторское право** – защищает исходный код, интерфейсы и документацию.
- **Лицензионные соглашения (EULA)** – определяют условия использования ПО.
- **Патентное право** – защищает алгоритмы и технические решения.
- **Законодательство об ответственности** – предусматривает штрафы и уголовную ответственность за пиратство.

8. Что такое программные вирусы, трояны, шпионское ПО? Каковы основные методы защиты от них?

Тип угрозы	Характеристика
Вирус	Вредоносная программа, способная самокопироваться и внедряться в исполняемые файлы.
Троян	Маскируется под полезное ПО, но выполняет скрытые вредоносные действия (кража данных, удаленный доступ).
Шпионское ПО (Spyware)	Собирает конфиденциальную информацию о пользователе без его ведома.

Методы защиты:

- Установка и регулярное обновление антивирусного ПО.
- Использование брандмауэров (межсетевых экранов).
- Регулярная установка обновлений безопасности (патчей).
- Обучение пользователей правилам цифровой гигиены (не открывать подозрительные письма и ссылки).
- Использование систем обнаружения вторжений (IDS/IPS).

- Регулярное резервное копирование данных.

9. Объясните, для чего используются следующие механизмы защиты: лицензионные ключи, аппаратные ключи, обфускация кода.

Механизм	Назначение
Лицензионные ключи	Уникальные коды, активирующие ПО и проверяющие легальность использования. Позволяют контролировать количество установок.
Аппаратные ключи (dongles)	Физические устройства (USB-ключи), подключаемые к компьютеру. Без них ПО не запускается. Используются для дорогого профессионального ПО.
Обфускация кода	Преобразование исходного кода в трудночитаемую форму (переименование переменных, добавление мусорного кода). Затрудняет обратную разработку и анализ, не влияя на функциональность.

10. Что такое уязвимость программного обеспечения и патч? Почему важно регулярно обновлять ПО?

- **Уязвимость** – ошибка или недостаток в ПО, который может быть использован злоумышленником для нарушения безопасности (например, переполнение буфера, SQL-инъекция, неверная проверка прав).
- **Патч** – исправление, выпускаемое разработчиком для устранения конкретной уязвимости или ошибки.

Почему важно обновляться:

- Устранение известных уязвимостей, которыми могут воспользоваться злоумышленники.
- Повышение стабильности и производительности системы.
- Добавление новых функций (в некоторых случаях).
- Соответствие нормативным требованиям (например, PCI DSS требует установки критических патчей в течение 30 дней).

11. Что подразумевается под технологией разработки программного обеспечения?

Технология разработки ПО – это совокупность методов, инструментов, процессов и практик, используемых для создания программных продуктов. Она охватывает все этапы жизненного цикла: от анализа требований до сопровождения.

Компоненты технологии разработки:

- Методологии (Waterfall, Agile, Scrum, Kanban).
- Инструментальные средства (IDE, Git, CI/CD, тестовые фреймворки).
- Процессы управления (управление требованиями, изменениями, конфигурациями).
- Стандарты и регламенты (код-стайл, Code Review, документирование).

12. Опишите суть каскадной модели жизненного цикла ПО. Каковы её сильные и слабые стороны?

Каскадная модель (Waterfall) – линейная последовательная модель, в которой каждый следующий этап начинается только после полного завершения предыдущего.

Этапы: Требования → Проектирование → Реализация → Тестирование → Внедрение → Сопровождение.

Сильные стороны:

- Простота управления (четкие этапы, понятные сроки).
- Дисциплина и строгость (все документы готовятся заранее).
- Хорошо подходит для проектов с устойчивыми требованиями (например, госзаказы).

Слабые стороны:

- Жёсткость – возврат на предыдущий этап крайне сложен и дорог.
- Проблема «информационного вакуума» – заказчик видит результат только в конце, риски несоответствия высоки.
- Не подходит для сложных и быстро меняющихся проектов.

13. В чём заключаются основные принципы гибких методологий разработки (Agile), таких как Scrum или Kanban?

Agile-манифест (ключевые ценности):

- Люди и взаимодействие важнее процессов и инструментов.
- Работающий продукт важнее исчерпывающей документации.
- Сотрудничество с заказчиком важнее согласования условий контракта.
- Готовность к изменениям важнее следования плану.

Принципы Agile (кратко):

- Итеративная разработка (короткие циклы – спринты 2–4 недели).
- Постоянная обратная связь от заказчика.
- Приоритет – работающий код с минимально необходимым функционалом (MVP).
- Команда самоорганизуется и принимает коллективные решения.
- Регулярная рефлексия и улучшение процессов (ретроспективы).

Scrum – фреймворк с ролями (Product Owner, Scrum Master, команда), артефактами (бэклог, спринт) и событиями (планирование, ежедневные стендапы, ретроспектива). **Kanban** – метод визуализации задач на доске с ограничением WIP (work in progress), без фиксированных итераций.

14. Что такое система контроля версий (например, Git) и какую роль она играет в современной разработке?

Система контроля версий (VCS) – инструмент, отслеживающий изменения в исходном коде и документах, позволяющий возвращаться к предыдущим состояниям и работать в команде.

Роль в разработке:

- **Хранение истории** – все изменения сохраняются и комментируются.
- **Совместная работа** – несколько разработчиков могут работать одновременно над одним проектом без конфликтов (через ветки).
- **Ветвление и слияние** – возможность разрабатывать новые функции изолированно, а затем интегрировать их.
- **Резервное копирование** – код хранится в центральной репозитории (GitHub, GitLab, Bitbucket).
- **Code Review** – возможность проверки изменений перед интеграцией (Pull Request / Merge Request).

15. Что такое тестирование ПО? Какие виды тестирования вы знаете?

Тестирование ПО – процесс проверки программы на соответствие требованиям, выявление дефектов и оценка качества.

Основные виды тестирования:

Вид	Описание
Модульное (Unit)	Проверка отдельных компонентов (функций, методов) в изоляции. Используются фреймворки JUnit, NUnit, pytest.
Интеграционное	Проверка взаимодействия между модулями (например, API-вызовы).
Системное	Проверка всей системы в целом на соответствие функциональным и нефункциональным требованиям.
Функциональное	Проверка, что система выполняет заявленные функции.
Нагрузочное	Проверка производительности под нагрузкой.
Приёмочное (UAT)	Проверка заказчиком перед вводом в эксплуатацию.
Регрессионное	Повторная проверка уже работавших функций после изменений, чтобы убедиться, что новые исправления не сломали старое.

16. Что такое пользовательский интерфейс (UI)? Почему его качество критически важно для успеха программного продукта?

Пользовательский интерфейс (UI, User Interface) – это совокупность графических элементов, экранов, меню, кнопок и способов взаимодействия пользователя с системой.

Почему UI важен:

- **Первое впечатление** – пользователь судит о качестве всего продукта по интерфейсу.
- **Удовольствие от работы** – интуитивный, приятный интерфейс повышает продуктивность и удовлетворённость.
- **Снижение ошибок** – хороший UI предотвращает неправильные действия пользователя.
- **Конкурентное преимущество** – пользователь выбирает между продуктами, руководствуясь удобством.
- **Лояльность** – неудобный интерфейс – главная причина отказа от продукта.

17. Чем отличается пользовательский интерфейс (UI) от пользовательского опыта (UX)?

UI (User Interface)	UX (User Experience)
Внешний вид – как это выглядит.	Ощущение – как это работает и какие эмоции вызывает.
Фокус на элементах: кнопки, шрифты, цвета, расположение.	Фокус на процессе: удобство, скорость, логичность, удовлетворённость.
Ответственность дизайнера: сделать красиво и понятно.	Ответственность аналитика: спроектировать путь пользователя.
Пример: кнопка «Купить» должна быть зелёной и заметной.	Пример: пользователь должен потратить не более 2 кликов на оформление заказа.

Кратко: UI – это оболочка, UX – это то, как пользователь чувствует себя во время работы с оболочкой. Хороший UI без UX – красиво, но неудобно. Хороший UX без UI – удобно, но некрасиво.

18. Назовите основные принципы и правила хорошего дизайна пользовательского интерфейса.

1. **Простота** – не перегружать экран элементами; минимализм.
2. **Понятность** – все элементы должны быть очевидны и интуитивны (например, иконки с подписями).
3. **Иерархия** – важное должно быть заметно (размер, цвет, положение).
4. **Последовательность** – единые правила навигации и стиля по всему приложению.
5. **Обратная связь** – реагировать на действия пользователя (подсветка кнопок, анимация, сообщения об успехе/ошибке).
6. **Доступность** – учёт людей с ограничениями (контрастность, шрифты, поддержка экранных дикторов).
7. **Привычность** – использовать стандартные паттерны (кнопка «Назад» слева, логотип ведёт на главную и т.д.).
8. **Толерантность к ошибкам** – разрешать отмену действий («Ctrl+Z»), подтверждать опасные операции.
9. **Адаптивность** – интерфейс должен корректно отображаться на разных экранах (десктоп, планшет, смартфон).

Все ответы могут быть сформулированы в другой близкой по смыслу формулировке

