

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**



**Частное учреждение высшего образования
«Высшая школа предпринимательства (институт)»
(ЧУВО «ВШП»)**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.В.04 «Программная инженерия»**

Направление подготовки: 09.03.02 Информационные системы и технологии

**Направленность (профиль) программы бакалавриата
«Информационные технологии»**

ОДОБРЕНО

Ученым советом ЧУВО «ВШП»

Протокол заседания

№01-02/24 от 15 мая 2025 г.

УТВЕРЖДАЮ

ЧУВО «ВШП»

5» мая 2025 г.

Аллабян М.Г.



Тверь, 2025

Рабочая программа учебной дисциплины **Б1.В.04 Программная инженерия**, компонента основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** направленность (профиль) **«Информационные технологии»**, направлена на обеспечение у обучающегося способности осуществлять профессиональную деятельность в соответствующей области и сферах профессиональной деятельности, в том числе на их практическую подготовку с учётом рабочей программы воспитания и календарного плана воспитательной работы Частном учреждении высшего образования **«Высшая школа предпринимательства (институт)»** (далее — **ЧУВО «ВШП»**).

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

Настоящая рабочая программа учебной дисциплины устанавливает требования к результатам обучения студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей и студентов направления подготовки 09.03.02 Информационные системы и технологии.

Программа учебной дисциплины разработана в соответствии с ФГОС ВО, утвержденного приказом Минобрнауки России от 19.09.2017 № 926 «Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии», основной профессиональной образовательной программой высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии, направленность (профиль) Информационные технологии.

2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Целью изучения дисциплины «Программная инженерия» является формирование у студентов теоретических и практических навыков по изучению и использованию современных технологий разработки программного обеспечения в соответствии с международными стандартами обучения программной инженерии.

Основными задачами изучения дисциплины являются формирование у студентов умений и навыков по проблемам оценки требований, проектирования, разработки, качества, повышения надежности и документирования программного обеспечения, а также по вопросам управления коллективной разработкой программного обеспечения.

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина является компонентом части, формируемой участниками образовательных отношений Блока 1 основной профессиональной образовательной программы высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии, направленность (профиль) — Информационные технологии.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ В РАМКАХ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций.

В результате освоения учебной дисциплины обучающийся должен демонстрировать следующие результаты обучения: УК-1, УК-2, ПК-1, ПК-2, ПК-3, ПК-4.

Таблица 1. Результаты обучения

Код компетенции	Наименование компетенции	Индекс и наименование индикатора содержания компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Способен осуществлять поиск и критический анализ информации	Знать: - Основные источники информации для разработки программного обеспечения (далее ПО). - Методы критического анализа информации. Уметь: - Находить актуальную информацию в специализированных источниках. - Анализировать и оценивать достоверность и релевантность информации. Владеть: - Навыками эффективного поиска информации.

			- Навыками критического мышления при анализе информации.
		УК-1.2 Способен применять системный подход для решения поставленных задач	Знать: - Принципы системного подхода в разработке ПО. - Основные этапы системного анализа и проектирования. Уметь: - Применять системный подход для анализа задач. - Разрабатывать системные решения для поставленных задач. Владеть: - Навыками системного анализа проблем. - Навыками интеграции различных элементов системы в единое целое.
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Способен определять круг задач в рамках поставленной цели	Знать: - Методы декомпозиции целей на задачи. - Принципы приоритизации задач. Уметь: - Определять и формулировать задачи на основе целей проекта. - Устанавливать приоритеты для выполнения задач. Владеть: - Навыками декомпозиции целей на задачи. - Навыками управления задачами и ресурсами.
		УК-2.2 Способен выбирать оптимальные способы решения задач, исходя из правовых норм, ресурсов и ограничений	Знать: - Основные правовые нормы в области информационных технологий (далее ИТ). - Принципы оценки и управления ресурсами. Уметь: - Оценивать доступные ресурсы и ограничения. - Выбирать наилучшие способы решения задач с учетом правовых норм. Владеть: - Навыками оценки ресурсов и ограничений. - Навыками принятия оптимальных решений в рамках правовых норм.
ПК-1	Разработка и отладка программного кода	ПК-1.1 Способен разрабатывать программный код	Знать: - Основные языки программирования для разработки ПО. - Принципы написания качественного кода. Уметь: - Писать программный код на одном из изучаемых языков. - Разрабатывать алгоритмы и структуры данных для решения задач. Владеть: - Навыками написания и документирования кода. - Навыками использования инструментов разработки.

		ПК-1.2 Способен отлаживать программный код	Знать: <ul style="list-style-type: none"> - Основные методы и инструменты отладки кода. - Принципы устранения ошибок и багов в коде. Уметь: <ul style="list-style-type: none"> - Использовать отладчики и логи для выявления ошибок. - Исправлять выявленные ошибки и оптимизировать код. Владеть: <ul style="list-style-type: none"> - Навыками отладки и тестирования кода. - Навыками применения инструментов для отладки.
ПК-2	Проверка работоспособности и рефакторинг кода программного обеспечения	ПК-2.1 Способен проверять работоспособность программного кода	Знать: <ul style="list-style-type: none"> - Принципы и методы тестирования кода. - Основные виды тестов (модульные, интеграционные, системные). Уметь: <ul style="list-style-type: none"> - Писать и выполнять тесты для проверки кода. - Использовать тестовые фреймворки для автоматизации тестирования. Владеть: <ul style="list-style-type: none"> - Навыками написания и выполнения тестов. - Навыками применения инструментов для тестирования.
		ПК-2.2 Способен рефакторить программный код	Знать: <ul style="list-style-type: none"> - Принципы и методы рефакторинга. - Основные техники улучшения структуры кода. Уметь: <ul style="list-style-type: none"> - Анализировать код на предмет улучшений. - Применять техники рефакторинга для повышения качества кода. Владеть: <ul style="list-style-type: none"> - Навыками рефакторинга кода. - Навыками оценки и улучшения структуры кода.
ПК-3	Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта	ПК-3.1 Способен интегрировать программные модули и компоненты	Знать: <ul style="list-style-type: none"> - Принципы интеграции модулей и компонентов. - Методы и инструменты для интеграции. Уметь: <ul style="list-style-type: none"> - Интегрировать модули и компоненты в единое приложение. - Проверять корректность интеграции и совместимость компонентов. Владеть: <ul style="list-style-type: none"> - Навыками интеграции программных модулей. - Навыками использования инструментов для интеграции.
		ПК-3.2 Способен проверять работоспособность выпусков программного продукта	Знать: <ul style="list-style-type: none"> - Методы и инструменты для проверки выпусков ПО. - Принципы контроля качества выпусков. Уметь: <ul style="list-style-type: none"> - Проводить тестирование и верификацию выпусков ПО.

			<ul style="list-style-type: none"> - Оценивать стабильность и производительность программного продукта. Владеть: <ul style="list-style-type: none"> - Навыками тестирования и верификации выпусков ПО. - Навыками применения инструментов контроля качества.
ПК-4	Разработка требований и проектирование программного обеспечения	ПК-4.1 Способен разрабатывать требования к программному обеспечению	Знать: <ul style="list-style-type: none"> - Принципы и методы анализа требований. - Техники документирования требований. Уметь: <ul style="list-style-type: none"> - Собирать и анализировать требования к ПО. - Документировать требования в виде спецификаций. Владеть: <ul style="list-style-type: none"> - Навыками анализа и сбора требований. - Навыками создания и поддержки документации по требованиям.
		ПК-4.2 Способен проектировать программное обеспечение	Знать: <ul style="list-style-type: none"> - Принципы и методы проектирования ПО. - Основные подходы к архитектурному дизайну. Уметь: <ul style="list-style-type: none"> - Разрабатывать архитектуру и дизайн ПО. - Использовать UML и другие инструменты для проектирования. Владеть: <ul style="list-style-type: none"> - Навыками проектирования программного обеспечения. - Навыками применения инструментов для проектирования.

5. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5.1. Объем дисциплины в зачетных единицах с указанием количества академических часов

Общая трудоемкость учебной дисциплины составляет 6 зачетных единиц, 216 часов, включая все формы контактной и самостоятельной работы обучающихся.

Объем дисциплины по учебному плану составляет –
6 зачётных единицы = 216 академических часов.

Контактная работа обучающегося (студенты) с научно-педагогическим работником организации (всего) - 72 академических часов,

в том числе:

Лекционные занятия (Лек.) - 20 академических часов,
Практические занятия (Пр.) - 50 академических часов,
Консультации (Конс.) - 2 академических часа.

Самостоятельная работа обучающегося (студента):

Самостоятельная работа (СР) - 99 академический час,

Текущий контроль успеваемости

и промежуточно-заочная аттестация обучающегося (студента):

Часы на контроль - 45 академических часов.

Таблица 2. Объём дисциплины

№ п/п	Раздел/тема дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (студентов), и трудоёмкость (в ак. часах)				Коды формируемых компетенций	
		Виды учебных занятий по дисциплине			Самосто ятельна я работа		
		Лек.	Пр.	Конс.	СР		
5 семестр / 3курс							
1	Тема 1. Введение в программную инженерию	1	2	-	8	УК-1.1, УК-1.2	
2	Тема 2. Модели жизненного цикла программного обеспечения	1	4	-	8	УК-2.1, УК-2.2	
3	Тема 3. Анализ требований	2	2	-	8	ПК-4.1, ПК-4.2	
4	Тема 4. Проектирование программного обеспечения	1	4	-	8	ПК-4.2, УК-1.2	
5	Тема 5. Разработка программного обеспечения	2	4	-	8	ПК-1.1, ПК-1.2, ПК-3.1	
6	Тема 6. Тестирование программного обеспечения	2	2	-	8	ПК-2.1, ПК-2.2, ПК-3.2	
7	Тема 7. Управление конфигурацией и версиями	1	4	-	8	ПК-3.1, ПК-3.2	
8	Тема 8. Подготовка к зачету		4		9	УК-1.1, УК-1.2, УК-2.1, УК-2.2, ПК-1.1, ПК-1.2, ПК-2.1, ПК-2.2, ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2	
6 семестр / 3 курс							
9	Тема 9. Управление проектами разработки ПО	2	2	-	4	УК-2.1, УК-2.2	
10	Тема 10. Методы и средства обеспечения качества ПО	2	4	-	4	ПК-2.1, ПК-2.2, ПК-4.2	
11	Тема 11. Рефакторинг и оптимизация кода	1	2	-	4	ПК-2.2, ПК-3.2	
12	Тема 12. Безопасность программного обеспечения	1	2	-	4	УК-1.1, ПК-2.1	
13	Тема 13. Документирование программного обеспечения	1	2	-	4	УК-1.2, ПК-4.1	
14	Тема 14. Выпуск и сопровождение программного обеспечения	1	4	-	4	ПК-3.1, ПК-3.2	
15	Тема 15. Инновации и современные тенденции в программной инженерии	2	4	-	4	УК-1.1, УК-2.2	
16	Тема 16. Итоговое занятие и подготовка к экзамену		4	2	8	УК-1.1, УК-1.2, УК-2.1, УК-2.2, ПК-1.1, ПК-1.2, ПК-2.1, ПК-2.2, ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2	
ИТОГО аудиторных часов/СР:		5-6 семестры/ 3 курс	72 ак. часов			99 ак. часа	-
Часы на контроль		5-6 семестры/ 3 курс	9 ак. час (форма промежуточной аттестации – зачет – 5 семестр) 36 ак. час (форма промежуточной аттестации – экзамен – 6 семестр)				
ВСЕГО ак. часов:		5-6 семестры/ 3 курс	216 академических часов				

5.2. Тематическое содержание дисциплины

* количество академических часов и виды занятий представлены в таблице № 2.

5 семестр

Тема 1: Введение в программную инженерию

Изучение основ программной инженерии, её целей и задач. Обзор жизненного цикла программного обеспечения (SDLC).

Тема 2: Модели жизненного цикла программного обеспечения

Изучение различных моделей SDLC: водопадная модель, спиральная модель, Agile, Scrum и их применение.

Тема 3: Анализ требований

Процесс сбора, анализа и документирования требований. Методы и техники анализа требований. Создание спецификаций требований.

Тема 4: Проектирование программного обеспечения

Принципы и методы проектирования ПО. Создание архитектурных схем. Использование UML для проектирования.

Тема 5: Разработка программного обеспечения

Изучение методологий разработки. Работа с языками программирования (Python, Java, JS, C++). Основы командной разработки.

Тема 6: Тестирование программного обеспечения

Типы тестирования: модульное, интеграционное, системное, приемочное. Методологии тестирования. Инструменты для автоматизированного тестирования.

Тема 7: Управление конфигурацией и версиями

Основы управления конфигурацией. Системы контроля версий: Git, SVN. Практики управления версиями.

Тема 8: Подготовка к зачету

Повторение и обобщение пройденного материала. Решение типовых задач и вопросов для зачета.

6 семестр

Тема 9: Управление проектами разработки ПО

Основы управления проектами. Роли и обязанности участников команды. Планирование и оценка проектов.

Тема 10: Методы и средства обеспечения качества ПО

Процессы обеспечения качества. Стандарты и модели качества. Инструменты и методы для оценки качества.

Тема 11: Рефакторинг и оптимизация кода

Методы рефакторинга кода. Инструменты и техники оптимизации производительности программного обеспечения.

Тема 12: Безопасность программного обеспечения

Основы безопасности ПО. Методы и инструменты для защиты программных систем. Обнаружение и предотвращение уязвимостей.

Тема 13: Документирование программного обеспечения

Типы документации. Создание и поддержка пользовательской и технической документации. Стандарты документирования.

Тема 14: Выпуск и сопровождение программного обеспечения

Подготовка приложения к публикации. Создание подписанного релиза. Выпуск и сопровождение ПО.

Тема 15: Инновации и современные тенденции в программной инженерии

Изучение новых технологий и методологий. Влияние инноваций на процессы разработки ПО.

Тема 16: Итоговое занятие и подготовка к экзамену

Повторение и обобщение пройденного материала. Решение типовых задач и вопросов для подготовки к экзамену.

5.2.1 Содержание практических занятий**Таблица 3**

№ п/п	Наименование темы (раздела) дисциплины	Содержание практического занятия
5 семестр		
1	Практическое занятие 1: Введение в программную инженерию	Задание: Обзор жизненного цикла программного обеспечения на конкретных примерах, обсуждение роли и задач программной инженерии. Цель: Понять основные этапы SDLC и их значение.
2	Практическое занятие 2: Модели жизненного цикла ПО	Задание: Сравнение различных моделей SDLC и их применение на примере реальных проектов, разработка краткого плана проекта с использованием одной из моделей. Цель: Изучить и сравнить различные модели жизненного цикла ПО, научиться применять их на практике.
3	Практическое занятие 3: Анализ требований	Задание: Сбор и документирование требований для простого проекта, создание спецификации требований. Цель: Научиться собирать, анализировать и документировать требования.
4	Практическое занятие 4: Проектирование программного обеспечения	Задание: Создание архитектурной схемы и диаграммы UML для проекта, обсуждение различных подходов к проектированию. Цель: Изучить основы проектирования ПО, научиться использовать UML.
5	Практическое занятие 5: Разработка программного обеспечения	Задание: Написание простого модуля на одном из языков программирования (Python, Java, JS, C++), обсуждение особенностей командной разработки. Цель: Научиться основам разработки ПО, понять особенности командной работы.
6	Практическое занятие 6: Тестирование программного обеспечения	Задание: Написание и выполнение модульных тестов для разработанного модуля, использование автоматизированных тестовых фреймворков. Цель: Изучить основные методы и инструменты тестирования.
7	Практическое занятие 7: Управление конфигурацией и версиями	Задание: Использование системы контроля версий Git для управления проектом, выполнение типовых операций с версиями. Цель: Научиться управлять конфигурацией и версиями ПО.
8	Практическое занятие 8: Подготовка к зачету	Задание: Решение типовых задач и вопросов для подготовки к зачету, обсуждение ключевых тем курса. Цель: Повторение и обобщение пройденного материала для успешной сдачи зачета.
6 семестр		
9	Практическое занятие 9: Управление проектами разработки ПО	Задание: Разработка плана проекта с использованием методологии Scrum, определение ролей и обязанностей участников команды. Цель: Изучить основы управления проектами разработки ПО, научиться применять Scrum на практике.

10	Практическое занятие 10: Методы и средства обеспечения качества ПО	Задание: Проведение анализа качества проекта с использованием стандартных инструментов, разработка плана по улучшению качества. Цель: Изучить методы и средства обеспечения качества ПО, научиться проводить анализ и улучшение качества.
11	Практическое занятие 11: Рефакторинг и оптимизация кода	Задание: Проведение рефакторинга и оптимизации кода простого проекта, обсуждение различных техник оптимизации. Цель: Научиться рефакторингу и оптимизации кода.
12	Практическое занятие 12: Безопасность программного обеспечения	Задание: Анализ уязвимостей в проекте и внедрение методов защиты, использование инструментов для обнаружения и предотвращения уязвимостей. Цель: Изучить основы безопасности ПО, научиться защищать программные системы.
13	Практическое занятие 13: Документирование программного обеспечения	Задание: Создание технической документации для проекта, обсуждение стандартов и методов документирования. Цель: Научиться создавать и поддерживать документацию ПО.
14	Практическое занятие 14: Выпуск и сопровождение программного обеспечения	Задание: Подготовка проекта к выпуску, создание подписанного релиза, разработка плана сопровождения. Цель: Изучить процессы выпуска и сопровождения ПО.
15	Практическое занятие 15: Инновации и современные тенденции в программной инженерии	Задание: Анализ современных тенденций и технологий в разработке ПО, обсуждение влияния инноваций на процессы разработки. Цель: Понять влияние инноваций на процессы разработки ПО.
16	Практическое занятие 16: Итоговое занятие и подготовка к экзамену	Задание: Решение типовых задач и вопросов для подготовки к экзамену, обсуждение практических аспектов разработки ПО. Цель: Повторение и обобщение пройденного материала для успешной сдачи экзамена.

5.2.2 Содержание самостоятельной работы

Таблица 4

№ п/п	Наименование темы (раздела) дисциплины	Содержание самостоятельной работы	Форма контроля
5 семестр			
1	Самостоятельная работа 1: Изучение литературы по программной инженерии	Задание: Прочитать рекомендованные главы из учебников и статьи по программной инженерии, подготовить краткий отчет по ключевым аспектам. Цель: Углубление теоретических знаний по программной инженерии.	Подготовить реферат
2	Самостоятельная работа 2: Выполнение упражнений по анализу требований	Задание: Самостоятельное выполнение упражнений по сбору и анализу требований для заданного проекта, подготовка спецификации требований. Цель: Закрепление навыков анализа и документирования требований.	Выполнение практического задания.
3	Самостоятельная работа 3: Проектирование UML-диаграмм	Задание: Создание UML-диаграмм для различных случаев использования, подготовка отчета по выполненным диаграммам. Цель: Изучение UML и проектирование ПО.	Выполнение практического задания.
4	Самостоятельная работа 4: Практическое программирование	Задание: Разработка небольших модулей на выбранном языке программирования, подготовка кода и отчета по выполненной работе. Цель: Закрепление навыков программирования.	Выполнение практического задания.
5	Самостоятельная работа 5: Написание модульных тестов	Задание: Самостоятельное написание модульных тестов для разработанных модулей, подготовка отчетов по результатам тестирования. Цель: Изучение методов тестирования ПО.	Выполнение практического задания.
6	Самостоятельная работа 6: Использование	Задание: Практическая работа с Git для управления версиями проекта, подготовка отчета по выполненным операциям.	Выполнение практического задания.

	системы контроля версий	Цель: Закрепление навыков управления конфигурацией и версиями.	
7	Самостоятельная работа 7: Обобщение текущего пройденного материала	Задание: Решение типовых задач и контрольных вопросов для подготовки к зачету, подготовка краткого отчета. Цель: Повторение пройденного материала.	Тест.
6 семестр			
8	Самостоятельная работа 8: Изучение литературы по управлению проектами	Задание: Прочитать рекомендованные главы из учебников и статьи по управлению проектами, подготовить краткий отчет по ключевым аспектам. Цель: Углубление теоретических знаний по управлению проектами.	Ответы на контрольные вопросы.
9	Самостоятельная работа 9: Разработка плана проекта	Задание: Самостоятельная разработка плана проекта с использованием методологии Scrum, подготовка отчетов по выполненной работе. Цель: Закрепление навыков управления проектами.	Выполнение практического задания.
10	Самостоятельная работа 10: Проведение анализа качества ПО	Задание: Самостоятельное проведение анализа качества ПО с использованием стандартных инструментов, подготовка отчета по результатам анализа. Цель: Изучение методов и средств обеспечения качества.	Выполнение практического задания.
11	Самостоятельная работа 11: Рефакторинг и оптимизация кода	Задание: Самостоятельное проведение рефакторинга и оптимизации кода проекта, подготовка отчета по выполненной работе. Цель: Закрепление навыков рефакторинга и оптимизации.	Выполнение практического задания.
12	Самостоятельная работа 12: Анализ уязвимостей и внедрение методов защиты	Задание: Самостоятельный анализ уязвимостей и внедрение методов защиты в проект, подготовка отчета по результатам работы. Цель: Изучение основ безопасности ПО.	Выполнение практического задания.
13	Самостоятельная работа 13: Создание технической документации	Задание: Самостоятельное создание технической документации для проекта, подготовка отчетов по выполненной работе. Цель: Закрепление навыков документирования ПО.	Выполнение практического задания.
14	Самостоятельная работа 14: Подготовка проекта к выпуску	Задание: Самостоятельная подготовка проекта к выпуску и разработка плана сопровождения, подготовка отчетов по выполненной работе. Цель: Изучение процессов выпуска и сопровождения ПО.	Выполнение практического задания.
15	Самостоятельная работа 15: Анализ современных тенденций и технологий	Задание: Самостоятельный анализ современных тенденций и технологий в разработке ПО, подготовка отчета по результатам анализа. Цель: Понимание влияния инноваций на процессы разработки ПО.	Выполнение практического задания.
16	Самостоятельная работа 16: Обобщение пройденного материала	Задание: Решение типовых задач и контрольных вопросов для подготовки к экзамену, подготовка краткого отчета. Цель: Повторение пройденного материала.	Экзамен

6.Оценочные материалы по дисциплине

Оценочные материалы находятся в документе «Оценочные материалы по дисциплине «Программная инженерия».

7.Методические материалы для обучающихся по освоению дисциплины (модуля)

А) Рекомендации обучающемуся (студенту) по работе с конспектом после лекции

Какими бы замечательными качествами в области методики ни обладал лектор, какое бы большое значение на занятиях ни уделял лекции слушатель, глубокое понимание материала достигается только путем самостоятельной работы над ним. Самостоятельную работу следует

начинать с доработки конспекта, желательно в тот же день, пока время не стерло содержание лекции из памяти (через 10 часов после лекции в памяти остается не более 30-40 % материала). С целью доработки необходимо в первую очередь прочитать записи, восстановить текст в памяти, а также исправить опiski, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее прочитать материал по рекомендуемой литературе, разрешая в ходе чтения, возникшие ранее затруднения, вопросы, а также дополнения и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. В ходе доработки конспекта углубляются, расширяются и закрепляются знания, а также дополняется, исправляется и совершенствуется конспект. Подготовленный конспект и рекомендуемая литература используется при подготовке к практическому занятию. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений и формул, к решению примеров, задач, к ответам на вопросы, предложенные в конце лекции преподавателем или помещенные в рекомендуемой литературе. Примеры, задачи, вопросы по теме являются материалом самоконтроля. Непременным условием глубокого усвоения учебного материала является знание основ, на которых строится изложение материала. Обычно преподаватель напоминает, какой ранее изученный материал и в какой степени требуется подготовить к очередному занятию. Эта рекомендация, как и требование систематической и серьезной работы над всем лекционным курсом, подлежит безусловному выполнению. Потери логической связи как внутри темы, так и между ними приводит к негативным последствиям: материал учебной дисциплины перестает основательно восприниматься, а творческий труд подменяется утомленным переписыванием. Обращение к ранее изученному материалу не только помогает восстановить в памяти известные положения, выводы, но и приводит разрозненные знания в систему, углубляет и расширяет их. Каждый возврат к старому материалу позволяет найти в нем что-то новое, переосмыслить его с иных позиций, определить для него наиболее подходящее место в уже имеющейся системе знаний. Неоднократное обращение к пройденному материалу является наиболее рациональной формой приобретения и закрепления знаний. Очень полезным, но, к сожалению, еще мало используемым в практике самостоятельной работы, является предварительное ознакомление с учебным материалом. Даже краткое, беглое знакомство с материалом очередной лекции дает многое. Обучающиеся (студенты) получают общее представление о её содержании и структуре, о главных и второстепенных вопросах, о терминах и определениях. Все это облегчает работу на лекции и делает ее целеустремленной.

Б) Рекомендации обучающемуся (студенту) по подготовке к занятиям семинарского типа

Обучающийся (студент) должен чётко уяснить, что именно с лекции начинается его подготовка к лабораторному/ практическому/ семинарскому/ методическому/ клиническому практическому занятию. Вместе с тем, лекция лишь организует мыслительную деятельность, но не обеспечивает глубину усвоения программного материала. При подготовке к такому виду занятий можно выделить 2 этапа:

- 1-й - организационный,
- 2-й - закрепление и углубление теоретических знаний.

На первом этапе обучающийся (студент) планирует свою самостоятельную работу, которая включает:

- уяснение задания на самостоятельную работу;
- подбор рекомендованной литературы;
- составление плана работы, в котором определяются основные пункты предстоящей подготовки.

Составление плана дисциплинирует и повышает организованность в работе. Второй этап включает непосредственную подготовку обучающегося (студента) к занятию. Начинать надо с изучения рекомендованной литературы. Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой

обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы обучающийся (студент) должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. Заканчивать подготовку следует составлением плана (перечня основных пунктов) по изучаемому материалу (вопросу). Такой план позволяет составить концентрированное, сжатое представление по изучаемым вопросам. В процессе подготовки к семинарскому занятию рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю. Идя на консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения. В начале семинарского занятия обучающиеся (студенты) под руководством преподавателя более глубоко осмысливают теоретические положения по теме занятия, раскрывают и объясняют основные явления и факты. В процессе творческого обсуждения и дискуссии вырабатываются умения и навыки использовать приобретенные знания для решения практических задач.

В) Рекомендации по самостоятельной работе обучающегося (студента) над изучаемым материалом

Успешное освоение данного курса базируется на рациональном сочетании нескольких видов учебной деятельности - лекций, семинарских занятий, самостоятельной работы. При этом самостоятельную работу следует рассматривать одним из главных звеньев полноценного высшего образования, на которую отводится значительная часть учебного времени.

Самостоятельная работа студентов складывается из следующих составляющих:

- работа с основной и дополнительной литературой, с материалами интернета и конспектами лекций;
- внеаудиторная подготовка к контрольным работам, выполнение докладов, рефератов и курсовых работ;
- выполнение самостоятельных практических работ;
- подготовка к экзаменам (зачетам) непосредственно перед ними.

Для правильной организации работы необходимо учитывать порядок изучения разделов курса, находящихся в строгой логической последовательности. Поэтому хорошее усвоение одной части дисциплины является предпосылкой для успешного перехода к следующей. Задания, проблемные вопросы, предложенные для изучения дисциплины, в том числе и для самостоятельного выполнения, носят междисциплинарный характер и базируются, прежде всего, на причинно-следственных связях между компонентами окружающего нас мира. В течение семестра, необходимо подготовить рефераты (проекты) с использованием рекомендуемой основной и дополнительной литературы и сдать рефераты для проверки преподавателю. Важным составляющим в изучении данного курса является решение ситуационных задач и работа над проблемно-аналитическими заданиями, что предполагает знание соответствующей научной терминологии и т.д.

Для лучшего запоминания материала целесообразно использовать индивидуальные особенности и разные виды памяти: зрительную, слуховую, ассоциативную. Успешному запоминанию также способствует приведение ярких свидетельств и наглядных примеров. Учебный материал должен постоянно повторяться и закрепляться.

При выполнении докладов, творческих, информационных, исследовательских проектов особое внимание следует обращать на подбор источников информации и методику работы с ними.

Для успешной сдачи экзамена (зачета) рекомендуется соблюдать следующие правила:

1. Подготовка к экзамену (зачету) должна проводиться систематически, в течение всего семестра.
2. Интенсивная подготовка должна начаться не позднее, чем за месяц до экзамена.
3. Время непосредственно перед экзаменом (зачетом) лучше использовать таким

образом, чтобы оставить последний день свободным для повторения курса в целом, для систематизации материала и доработки отдельных вопросов.

На экзамене высокую оценку получают студенты, использующие данные, полученные в процессе выполнения самостоятельных работ, а также использующие собственные выводы на основе изученного материала.

Учитывая значительный объем теоретического материала, студентам рекомендуется регулярное посещение и подробное конспектирование лекций.

8. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- задания для выполнения, а также инструкция о порядке проведения промежуточной аттестации оформляются увеличенным шрифтом;

- задания для выполнения на промежуточной аттестации зачитываются ассистентом;

- письменные задания выполняются на бумаге, надиктовываются ассистенту;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство;

в) для глухих и слабослышащих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- промежуточно-заочная аттестация проводится в письменной форме;

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования;

- по желанию студента промежуточно-заочная аттестация может проводиться в письменной форме;

д) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по желанию студента промежуточно-заочная аттестация проводится в устной форме.

Примечание:

а) Для обучающегося (бакалавра), осваивающего учебную дисциплину, обязательный компонент основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии (направленность (профиль) «Информационные технологии»), форма обучения — очно-заочная), одобренной на заседании Учёного совета образовательной организации, утверждённой ректором Частного образовательного учреждения высшего образования «Высшая школа предпринимательства», по индивидуальному учебному плану (при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра)), Институт:**

- разрабатывает, согласовывает с участниками образовательных отношений и утверждает в установленном порядке согласно соответствующему локальному нормативному акту **индивидуальный учебный план** конкретного обучающегося (бакалавра) (*учебный план, обеспечивающий освоение конкретной основной образовательной программы высшего образования на основе индивидуализации её содержания с учётом особенностей и образовательных потребностей конкретного обучающегося (бакалавра)*);

- устанавливает для конкретного обучающегося (бакалавра) по индивидуальному учебному плану **одинаковые дидактические единицы** — элементы содержания учебного материала, изложенного в виде утверждённой в установленном образовательной организацией порядке согласно соответствующему локальному нормативному акту рабочей программы учебной дисциплины, обязательного компонента разработанной и реализуемой Институтom основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** (направленность (профиль) «Информационные технологии»), форма обучения — очно-заочная), как и для обучающегося (бакалавра), осваивающего основную образовательную программу высшего образования в учебной группе;

- определяет в индивидуальном учебном плане конкретного обучающегося (бакалавра) **объём учебной дисциплины** с указанием количества академических часов/ ЗЕТ, выделенных на его контактную работу (групповую и (или) индивидуальную работу) с руководящими и (или) научно-педагогическими работниками, реализующими основную образовательную программу высшего образования;

- определяет в индивидуальном учебном плане конкретного обучающегося (бакалавра) количество академических часов/ ЗЕТ по учебной дисциплине, выделенных на его самостоятельную работу (*при необходимости*).

б) Для обучающегося (бакалавра) с ограниченными возможностями здоровья и инвалида, осваивающего учебную дисциплину, обязательный компонент основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии (направленность (профиль) «Информационные технологии»), форма обучения — очно-заочная), одобренной на заседании Учёного совета образовательной организации, утверждённой ректором Частного образовательного учреждения высшего образования «Высшая школа предпринимательства», (при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий)), Институт:

- разрабатывает, согласовывает с участниками образовательных отношений и утверждает в установленном порядке согласно соответствующему локальному нормативному акту **индивидуальный учебный план** конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья/ инвалида (*при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий)*) (*учебный план, обеспечивающий освоение конкретной основной образовательной программы высшего образования на основе индивидуализации её содержания с учётом особенностей и образовательных потребностей конкретного обучающегося (бакалавра)*);

- устанавливает для конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья содержание образования (**одинаковые дидактические единицы** — элементы содержания учебного материала, как и для обучающегося (бакалавра), осваивающего основную образовательную программу высшего образования в учебной группе) и условия организации обучения, изложенного в виде утверждённой в установленном Институтom порядке согласно соответствующему локальному нормативному акту рабочей программы учебной дисциплины, обязательного компонента разработанной и реализуемой им адаптированной основной профессиональной образовательной программы высшего образования - программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** (направленность (профиль) «Информационные технологии»), форма обучения — очно-заочная),

а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида (для конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья/инвалида *(при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий))*);

- определяет в индивидуальном учебном плане конкретного обучающегося бакалавра с ограниченными возможностями здоровья/инвалида *(при наличии факта зачисления такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий))* **объём учебной дисциплины** с указанием количества академических часов/ ЗЕТ, выделенных на его контактную работу (групповую и (или) индивидуальную работу) с руководящими и (или) научно-педагогическими работниками, реализующими основную образовательную программу высшего образования;

- определяет в индивидуальном учебном плане конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья/инвалида *(при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий))* количество академических часов/ ЗЕТ по учебной дисциплине, выделенных на его самостоятельную работу *(при необходимости)*.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

9.1 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Ратушняк Г.Я., Технологии разработки и проектирования информационных систем. Часть 1 [Электронный ресурс] : учебное пособие / Г.Я. Ратушняк, А.Л. Золкин. - М. : Русайнс, 2022. - 201 с. - ISBN 978-5-4365-9893-2. - Режим доступа: <https://book.ru/book/945635>

2. Ратушняк Г.Я., Технологии разработки и проектирования информационных систем. Часть 2 [Электронный ресурс] : учебное пособие / Г.Я. Ратушняк, А.Л. Золкин. - М. : Русайнс, 2022. - 349 с. - ISBN 978-5-466-01439-6. - Режим доступа: <https://book.ru/book/946232>

Дополнительная литература:

1. Письменский Г.И., Основы проектирования электронной информационно-образовательной среды [Электронный ресурс] : монография / Г.И. Письменский, В.В. Киселев, Л.В. Неровный, С.В. Сафонова. - М. : Русайнс, 2022. - 102 с. - ISBN 978-5-466-02119-6. - Режим доступа: <https://book.ru/book/947240>

9.2 Используемое программное обеспечение (комплект лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства в соответствии с п.4.3.2. ФГОС ВО 09.03.02):

1. Microsoft Windows 11 Pro или аналогичная ОС, включая дистрибутивы Linux, например Debian, Ubuntu, OpenSuse, в том числе отечественного производства, например ОС Astra Linux Common Edition (Разработчик: АО «НПО РусБИТех»), ОС «РОСА» (Разработчик: «НТИЦ ИТ РОСА»).

2. Microsoft Office 365 или аналогичный офисный пакет, например OpenOffice, LibreOffice, ONLYOFFICE, в том числе отечественного производства, например МойОфис (Разработчик: ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»).

3. Adobe Reader или аналогичный просмотрщик PDF, например Okular, Foxit Reader, в том числе отечественного производства, например Окуляр ГОСТ (Разработчик: ООО «Лаборатория 50»).

4. Google Chrome или аналогичный веб-браузер, например Microsoft Edge, Mozilla Firefox, в том числе отечественного производства, например Яндекс.Браузер (Разработчик: ООО «ЯНДЕКС»).

5. Microsoft Visual Studio Code или аналогичная IDE, например Sublime Text, Eclipse, в том

числе отечественного производства

6. PyCharm / IntelliJ IDEA / CLion / WebStorm либо аналогичная IDE полного стека, в том числе отечественного производства

7. MySQL CE 8.0 / MySQL Workbench или аналогичные СУБД, например MS SQL, PostgreSQL, в том числе отечественного производства

8. Android Studio или аналогичная IDE для разработки мобильных приложений, в том числе отечественного производства

9. Figma или аналогичное ПО для подготовки макетов, например Penpot, Lunacy, в том числе отечественного производства.

9.3 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля) (в соответствии с п.4.3.4. ФГОС ВО 09.03.02)

1. Электронно-библиотечная система BOOK.RU [Электронный ресурс]. - Режим доступа: <https://book.ru/>

9.4 Базы данных, информационно-справочные и поисковые системы (в соответствии с п.4.3.4. ФГОС ВО 09.03.02)

1. КонсультантПлюс: справочно-поисковая система [Электронный ресурс]. - <http://www.consultant.ru>

2. Мировая цифровая библиотека: <http://wdl.org/ru>

3. Научная электронная библиотека «Scopus»: <https://www.scopus.com>

4. Научная электронная библиотека ScienceDirect: <http://www.sciencedirect.com>

5. Научная электронная библиотека «eLIBRARY»: <https://elibrary.ru>

6. Портал «Гуманитарное образование» <http://www.humanities.edu.ru>

7. Федеральный портал «Российское образование» <http://www.edu.ru>

8. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» <http://school-collection.edu.ru>

9. Поисковые системы Yandex, Rambler и др.

10. Электронная библиотека Российской Государственной Библиотеки (РГБ): <http://elibrary.rsl.ru>

11. Электронно-библиотечная система <http://www.sciteclibrary.ru>

10. Материально-техническое обеспечение дисциплины

Помещения для проведения всех видов работы, предусмотренных учебным планом, укомплектованы необходимым оборудованием и техническими средствами обучения.

Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы, с указанием перечня основного оборудования, учебно-наглядных пособий и используемого программного обеспечения	Адрес (местоположение) помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом (в случае реализации образовательной программы в сетевой форме дополнительно указывается наименование организации, с которой заключен договор)
Специализированная многофункциональная учебная аудитория для проведения учебных занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающихся, с перечнем основного	170001, Россия, город Тверь, улица Спартака, дом 26а

оборудования: <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	
Специализированная многофункциональная учебная аудитория для проведения учебных занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающийся, с перечнем основного оборудования: <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	170001, Россия, город Тверь, улица Спартак, дом 26а
Помещение для самостоятельной работы обучающихся с перечнем основного оборудования: <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	170001, Россия, город Тверь, улица Спартак, дом 26а
Помещение для практических занятий на персональных компьютерах: <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; 	170001, Россия, город Тверь, улица Спартак, дом 26а

- | | |
|--|--|
| <ul style="list-style-type: none">- Стул педагогического работника;- Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;- Ноутбуки с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;- Маркерная или меловая доска;- Проектор. | |
|--|--|

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**



**Частное учреждение высшего образования
«Высшая школа предпринимательства (институт)»
(ЧУВО «ВШП»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
по дисциплине
Б1.В.04 «Программная инженерия»**

**Направление подготовки: 09.03.02 Информационные системы и технологии
Направленность (профиль) программы бакалавриата
«Информационные технологии»**

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ В РАМКАХ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций.

В результате освоения учебной дисциплины обучающийся должен демонстрировать следующие результаты обучения: УК-1, УК-2, ПК-1, ПК-2, ПК-3, ПК-4.

Код компетенции	Наименование компетенции	Индекс и наименование индикатора содержания компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Способен осуществлять поиск и критический анализ информации	Знать: <ul style="list-style-type: none"> - Основные источники информации для разработки программного обеспечения (далее ПО). - Методы критического анализа информации. Уметь: <ul style="list-style-type: none"> - Находить актуальную информацию в специализированных источниках. - Анализировать и оценивать достоверность и релевантность информации. Владеть: <ul style="list-style-type: none"> - Навыками эффективного поиска информации. - Навыками критического мышления при анализе информации.
		УК-1.2 Способен применять системный подход для решения поставленных задач	Знать: <ul style="list-style-type: none"> - Принципы системного подхода в разработке ПО. - Основные этапы системного анализа и проектирования. Уметь: <ul style="list-style-type: none"> - Применять системный подход для анализа задач. - Разрабатывать системные решения для поставленных задач. Владеть: <ul style="list-style-type: none"> - Навыками системного анализа проблем. - Навыками интеграции различных элементов системы в единое целое.
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Способен определять круг задач в рамках поставленной цели	Знать: <ul style="list-style-type: none"> - Методы декомпозиции целей на задачи. - Принципы приоритизации задач. Уметь: <ul style="list-style-type: none"> - Определять и формулировать задачи на основе целей проекта. - Устанавливать приоритеты для выполнения задач. Владеть: <ul style="list-style-type: none"> - Навыками декомпозиции целей на задачи. - Навыками управления задачами и ресурсами.

		<p>УК-2.2</p> <p>Способен выбирать оптимальные способы решения задач, исходя из правовых норм, ресурсов и ограничений</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Основные правовые нормы в области информационных технологий (далее ИТ). - Принципы оценки и управления ресурсами. <p>Уметь:</p> <ul style="list-style-type: none"> - Оценивать доступные ресурсы и ограничения. - Выбирать наилучшие способы решения задач с учетом правовых норм. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками оценки ресурсов и ограничений. - Навыками принятия оптимальных решений в рамках правовых норм.
ПК-1	Разработка и отладка программного кода	<p>ПК-1.1</p> <p>Способен разрабатывать программный код</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Основные языки программирования для разработки ПО. - Принципы написания качественного кода. <p>Уметь:</p> <ul style="list-style-type: none"> - Писать программный код на одном из изучаемых языков. - Разрабатывать алгоритмы и структуры данных для решения задач. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками написания и документирования кода. - Навыками использования инструментов разработки.
		<p>ПК-1.2</p> <p>Способен отлаживать программный код</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Основные методы и инструменты отладки кода. - Принципы устранения ошибок и багов в коде. <p>Уметь:</p> <ul style="list-style-type: none"> - Использовать отладчики и логи для выявления ошибок. - Исправлять выявленные ошибки и оптимизировать код. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками отладки и тестирования кода. - Навыками применения инструментов для отладки.
ПК-2	Проверка работоспособности и рефакторинг кода программного обеспечения	<p>ПК-2.1</p> <p>Способен проверять работоспособность программного кода</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы тестирования кода. - Основные виды тестов (модульные, интеграционные, системные). <p>Уметь:</p> <ul style="list-style-type: none"> - Писать и выполнять тесты для проверки кода. - Использовать тестовые фреймворки для автоматизации тестирования. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками написания и выполнения тестов. - Навыками применения инструментов для тестирования.

		<p>ПК-2.2</p> <p>Способен рефакторить программный код</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы рефакторинга. - Основные техники улучшения структуры кода. <p>Уметь:</p> <ul style="list-style-type: none"> - Анализировать код на предмет улучшений. - Применять техники рефакторинга для повышения качества кода. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками рефакторинга кода. - Навыками оценки и улучшения структуры кода.
ПК-3	Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта	<p>ПК-3.1</p> <p>Способен интегрировать программные модули и компоненты</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы интеграции модулей и компонентов. - Методы и инструменты для интеграции. <p>Уметь:</p> <ul style="list-style-type: none"> - Интегрировать модули и компоненты в единое приложение. - Проверять корректность интеграции и совместимость компонентов. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками интеграции программных модулей. - Навыками использования инструментов для интеграции.
		<p>ПК-3.2</p> <p>Способен проверять работоспособность выпусков программного продукта</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Методы и инструменты для проверки выпусков ПО. - Принципы контроля качества выпусков. <p>Уметь:</p> <ul style="list-style-type: none"> - Проводить тестирование и верификацию выпусков ПО. - Оценивать стабильность и производительность программного продукта. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками тестирования и верификации выпусков ПО. - Навыками применения инструментов контроля качества.
ПК-4	Разработка требований и проектирование программного обеспечения	<p>ПК-4.1</p> <p>Способен разрабатывать требования к программному обеспечению</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы анализа требований. - Техники документирования требований. <p>Уметь:</p> <ul style="list-style-type: none"> - Собирать и анализировать требования к ПО. - Документировать требования в виде спецификаций. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками анализа и сбора требований. - Навыками создания и поддержки документации по требованиям.

		ПК-4.2 Способен проектировать программное обеспечение	Знать: <ul style="list-style-type: none"> - Принципы и методы проектирования ПО. - Основные подходы к архитектурному дизайну. Уметь: <ul style="list-style-type: none"> - Разрабатывать архитектуру и дизайн ПО. - Использовать UML и другие инструменты для проектирования. Владеть: <ul style="list-style-type: none"> - Навыками проектирования программного обеспечения. - Навыками применения инструментов для проектирования.
--	--	--	---

КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

(признак, на основании которого, проводится оценка по выбранному показателю)

Показатель оценивания компетенций	Результат обучения	Критерии оценивания компетенций
Высокий уровень (отлично)	Знать	Обучающийся продемонстрировал: глубокие исчерпывающие знания и понимание учебного материала; содержательные, полные, правильные и конкретные ответы на все вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.
	Уметь	Обучающийся продемонстрировал: понимание учебного материала; умение свободно решать практические задания (ситуационные задачи), которые следует выполнить или описание результата, который нужно получить и др.; логически последовательные, содержательные, полные, правильные и конкретные ответы (решения) на все поставленные задания (вопросы), включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.
	Владеть	Обучающийся продемонстрировал: понимание учебного материала; умение свободно решать комплексные практические задания (решения задач по нестандартным ситуациям); логически последовательные, полные, правильные и конкретные ответы в ходе защиты задания, включая дополнительные уточняющие вопросы (задания); свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.
Средний уровень (хорошо)	Знать	Обучающийся продемонстрировал: твердые и достаточно полные знания учебного материала; правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении замечаний по отдельным вопросам; достаточное владение литературой, рекомендованной учебной программой дисциплины
	Уметь	Обучающийся продемонстрировал: понимание учебного материала; логически последовательные, правильные и конкретные ответы (решения) на основные задания (вопросы), включая дополнительные; устранение замечаний по отдельным элементам задания (вопроса); владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины
	Владеть	Обучающийся продемонстрировал: понимание учебного материала; продемонстрировал логически последовательные, достаточно полные, правильные ответы, включая

		дополнительные; самостоятельно устранил замечания по отдельным элементам задания (вопроса); владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины
Достаточный уровень (удовлетворительно)	Знать	Обучающийся продемонстрировал: твердые знания и понимание основного учебного материала; правильные, без грубых ошибок, ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины
	Уметь	Обучающийся продемонстрировал: понимание основного учебного материала; правильные, без грубых ошибок, ответы (решения) на основные задания (вопросы), включая дополнительные, устранение, при наводящих вопросах преподавателя, замечаний по отдельным элементам задания (вопроса); недостаточно полное владение литературой, рекомендованной учебной программой дисциплины
	Владеть	Обучающийся понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы при устранении неточностей и ошибок в решениях в ходе защиты задания (проекта, портфолио) при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

При проведении промежуточной аттестации в ЧУВО «ВШП» используются традиционные формы аттестации:

Форма промежуточной аттестации	Шкала оценивания
ЗАЧЕТ	«зачтено», «незачтено»
ЭКЗАМЕН	«отлично», «хорошо», «удовлетворительно», «неудовлетворительно»

КРИТЕРИИ И ПРОЦЕДУРЫ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ

Для оценивания результатов обучения в виде **ЗНАНИЙ** используются следующие процедуры и технологии:

- тестирование.

Для оценивания результатов обучения в виде **УМЕНИЙ и ВЛАДЕНИЙ** используются следующие процедуры и технологии:

- устный или письменный ответ на вопрос.
- практические задания, включающие одну или несколько задач (вопросов) в виде краткой формулировки действий (комплекса действий), которые следует выполнить, или описать результат, который нужно получить.

Критерии оценивания результата обучения по дисциплине (модулю)

Результат обучения по дисциплине (модулю)	ШКАЛА ОЦЕНИВАНИЯ				Процедуры оценивания
	«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»	
<u>УК-1</u> <u>УК-2</u> <u>ПК-1</u> <u>ПК-2</u> <u>ПК-3</u> <u>ПК-4</u>	Обучаемый продемонстрировал: глубокие исчерпывающие знания и понимание	Обучаемый продемонстрировал: твердые и достаточно полные знания учебного	Обучаемый продемонстрировал: твердые знания и понимание основного учебного материала;	Обучаемый продемонстрировал: неправильные ответы на основные вопросы; грубые ошибки в ответах; непонимание	Тестовые задания

<u>Знать:</u>	учебного материала; содержательные, полные, правильные и конкретные ответы на все вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.	материала; правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении замечаний по отдельным вопросам; достаточное владение литературой.	правильные, без грубых ошибок, ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины.	сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы; не владеет основной литературой, рекомендованной учебной программой дисциплины.	
<u>УК-1.</u> <u>УК-2.</u> <u>ПК-1.</u> <u>ПК-2.</u> <u>ПК-3.</u> <u>ПК-4.</u> <u>Уметь:</u>	Обучаемый продемонстрировал: понимание учебного материала, содержательные, полные, правильные и конкретные ответы на все поставленные вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание учебного материала; логически последовательные, правильные и конкретные ответы на основные задания/вопросы, включая дополнительные; устранение замечаний по отдельным элементам задания; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание основного учебного материала; правильные, без грубых ошибок, ответы на основные вопросы, включая дополнительные, при устранении, при наводящих вопросах преподавателя, замечаний по отдельным элементам задания; недостаточное полное владение литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: непонимание основного учебного материала; не дал правильные ответы на основные вопросы, включая дополнительные; не устранил, при наводящих вопросах преподавателя, замечания и грубые ошибки по вопросу; не владеет основной учебной литературой, рекомендованной учебной программой дисциплины	Вопросы Практические задания
<u>УК-1.</u> <u>УК-2.</u> <u>ПК-1.</u> <u>ПК-2.</u> <u>ПК-3.</u> <u>ПК-4.</u> <u>Владеть:</u>	Обучаемый продемонстрировал: понимание учебного материала; правильные и конкретные ответы, включая дополнительные и уточняющие вопросы; свободное владение основной и	Обучаемый продемонстрировал: понимание учебного материала; продемонстрировал логически последовательные, достаточно полные, верные ответы; самостоятельно устранил замечания по	Обучаемый продемонстрировал: понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы, в том числе при наводящих вопросах преподавателя; недостаточно полное владение	Обучаемый продемонстрировал: непонимание основного учебного материала; дал неправильные ответы на поставленные вопросы; не владеет основной учебной литературой, рекомендованной учебной программой дисциплины	Вопросы Практические задания

	дополнительной литературой, рекомендованной учебной программой дисциплины	отдельным элементам; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	литературой, рекомендованной учебной программой дисциплины		
--	---	--	--	--	--

1. Оценочные материалы для самостоятельной работы обучающихся (студентов)

1.1 Реферат 5 семестр

Реферат позволит студентам углубиться в историю и развитие программной инженерии, а также понять важность различных этапов жизненного цикла программного обеспечения. Реферат покрывает компетенции УК-1.1, УК-1.2, УК-2.1, УК-2.2, ПК-1.1, ПК-4.1.

Примерная тематика рефератов:

1. История и эволюция программной инженерии: Исследуйте, как развивалась программная инженерия с течением времени и какие ключевые этапы можно выделить.
2. Сравнение моделей жизненного цикла программного обеспечения: Проанализируйте основные модели SDLC, такие как водопадная модель, спиральная модель, Agile и Scrum.
3. Важность анализа требований в разработке ПО: Рассмотрите методы и техники анализа требований и их роль в успешной разработке программного обеспечения.
4. Применение UML в проектировании ПО: Изучите основные виды UML-диаграмм и их использование в проектировании программных систем.
5. Методологии разработки программного обеспечения: Сравните различные методологии разработки ПО, такие как TDD, BDD, и их применение на практике.
6. Тестирование программного обеспечения: Обсудите различные виды тестирования, их цели и методы проведения, а также важность автоматизированного тестирования.

1.2 Реферат 6 семестр

Реферат позволит студентам углубиться в историю и развитие программной инженерии, а также понять важность различных этапов жизненного цикла программного обеспечения. Реферат покрывает компетенции УК-1.1, УК-1.2, УК-2.1, УК-2.2, ПК-1.1, ПК-4.1.

Примерная тематика рефератов:

1. Управление конфигурацией и версиями ПО: Рассмотрите системы контроля версий, такие как Git и SVN, и их роль в управлении проектами разработки ПО.
2. Рефакторинг и оптимизация кода: Изучите методы и инструменты для рефакторинга кода и оптимизации производительности ПО.
3. Обеспечение безопасности программного обеспечения: Исследуйте методы и инструменты для защиты программных систем от уязвимостей и атак.
4. Документирование программного обеспечения: Рассмотрите типы документации и их роль в процессе разработки и сопровождения программного обеспечения.
5. Инновации в программной инженерии: Проанализируйте современные тенденции и инновации в программной инженерии, такие как DevOps, машинное обучение и блокчейн.
6. Управление проектами разработки ПО: Обсудите основные аспекты управления проектами разработки ПО, такие как планирование, оценка рисков и управление командой.

Цель написания рефератов: Углубить понимание и критическое осмысление различных аспектов программной инженерии, развивая аналитические и научные навыки студентов.

Структура реферата:

1. **Введение**
 - Краткое описание темы и целей реферата.
 - Актуальность темы.

2. Основная часть

- Теоретические основы темы.
- История и эволюция (если применимо).
- Применение в современной экономике/бизнесе.
- Примеры и кейсы.
- Проблемы и вызовы.
- Перспективы и тенденции развития.

3. Заключение

- Выводы по результатам исследования.
- Значение программной инженерии для современных информационных систем.

4. Список использованных источников

- Перечень использованной литературы и интернет-ресурсов.

Критерии оценивания:

1. Структура и логика изложения (20%)

- Четкая структура работы (введение, основная часть, заключение).
- Логичность и последовательность изложения материала.

2. Содержание (40%)

- Полнота раскрытия темы.
- Описание основных этапов развития программной инженерии.
- Анализ современных тенденций.
- Примеры применения программной инженерии в информационных системах.

3. Аналитическая часть (20%)

- Глубина анализа роли программной инженерии в информационных системах.
- Наличие собственных выводов и оценок.

4. Оформление (10%)

- Соответствие требованиям к оформлению рефератов (шрифт, отступы, заголовки и т.д.).
- Корректное оформление ссылок и списка литературы.

5. Язык и стиль (10%)

- Грамотность и точность изложения.
- Научный стиль текста.

Требования к объему:

Объем реферата должен составлять 10-15 страниц печатного текста (шрифт Times New Roman, размер 12, интервал 1.5, поля 2 см со всех сторон).

2. Оценочные материалы для оценки текущей аттестации обучающихся (студентов)

2.1 Тестовые задания для текущего контроля успеваемости в виде ЗНАНИЙ

В тестовом задании вопросы, которые имеют закрытый характер.

Правильные ответы выделены знаком +.

5 семестр

1. Какие основные этапы жизненного цикла программного обеспечения? (УК-1.1)

- Планирование, дизайн, разработка, маркетинг.
- Планирование, анализ требований, проектирование, разработка, тестирование, сопровождение. +
- Анализ, тестирование, выпуск, реклама.
- Планирование, дизайн, тестирование, поддержка.

2. Что такое модель жизненного цикла программного обеспечения? (УК-1.2)

- Процесс создания, использования и сопровождения программного обеспечения. +
- Набор инструментов для разработки программного обеспечения.

- Система управления проектами.
 - Методология управления командой разработчиков.
3. Какие существуют модели жизненного цикла программного обеспечения? (ПК-4.1)
 - Водопадная модель, спиральная модель, Agile, Scrum. +
 - Модель звезды, каскадная модель, TDD, BDD.
 - Модель дерева, Agile, Scrum, DDD.
 - Модель реки, спиральная модель, Kanban, Lean.
 4. В чем отличие водопадной модели от Agile? (ПК-4.2)
 - Водопадная модель линейная, Agile итеративная. +
 - Водопадная модель более гибкая, чем Agile.
 - Agile используется только для крупных проектов.
 - Agile более строгая, чем водопадная модель.
 5. Какие основные методы сбора требований? (УК-2.1)
 - Анализ конкурентов, исследование рынка, прогнозирование.
 - Чтение блогов, просмотр видеороликов, участие в семинарах.
 - Интервью, наблюдение, анализ документации, анкетирование. +
 - Использование шаблонов, чтение новостей, анализ данных.
 6. Что такое спецификация требований? (УК-2.2)
 - План тестирования программного обеспечения.
 - Код программы, написанный на языке программирования.
 - Диаграмма, отображающая архитектуру системы.
 - Документ, описывающий функциональные и нефункциональные требования к системе. +
 7. Какие типы диаграмм используются в UML? (ПК-4.2)
 - Диаграммы функций, диаграммы характеристик, диаграммы компонентов.
 - Диаграммы классов, диаграммы прецедентов, диаграммы последовательности. +
 - Диаграммы событий, диаграммы данных, диаграммы интерфейсов.
 - Диаграммы потоков, диаграммы процессов, диаграммы блоков.
 8. Что такое архитектурная схема? (ПК-4.1)
 - Инструкция по использованию системы.
 - План проекта.
 - Описание процесса тестирования.
 - Высокоуровневое представление структуры системы. +
 9. Каковы основные компоненты программы на языке Java? (ПК-1.1)
 - Страницы, разделы, кнопки, ссылки.
 - Классы, методы, поля, конструкторы. +
 - Узлы, вершины, рёбра, графы.
 - Таблицы, строки, столбцы, ключи.
 10. Какие принципы объектно-ориентированного программирования? (ПК-1.1)
 - Модуляция, функционализация, процессинг, иерархия.
 - Компиляция, интерпретация, выполнение, отладка.
 - Инкапсуляция, наследование, полиморфизм, абстракция. +
 - Логика, структура, процедура, объект.
 11. Что такое модульное тестирование? (ПК-2.1)
 - Тестирование производительности системы.
 - Тестирование безопасности приложения.
 - Тестирование интерфейсов пользователя.
 - Тестирование отдельных модулей программы на уровне исходного кода. +
 12. Какие инструменты используются для автоматизированного тестирования? (ПК-2.1)
 - Git, SVN, Mercurial.
 - JUnit, Selenium, TestNG. +
 - Word, Excel, PowerPoint.

- Photoshop, Illustrator, CorelDRAW.
13. Что такое система контроля версий? (ПК-3.1)
- Система для управления изменениями в коде и координации работы команды разработчиков. +
 - Инструмент для создания графических интерфейсов.
 - Программа для управления базами данных.
 - Сервис для автоматической генерации кода.
14. Какие операции можно выполнять с помощью Git? (ПК-3.1)
- Коммит, пуш, пул, мердж. +
 - Кодирование, декодирование, шифрование, дешифрование.
 - Сканирование, копирование, печать, сохранение.
 - Запуск, остановка, перезагрузка, обновление.

Критерии оценки результатов теста

1. **"Неудовлетворительно" (0-39%)**
 - Студент ответил правильно на менее 40% вопросов.
 - Значительные пробелы в знаниях по большинству тем.
 - Неправильное понимание ключевых понятий и принципов.
 - Неспособность применить теоретические знания на практике.
2. **"Удовлетворительно" (40-59%)**
 - Студент ответил правильно на 40-59% вопросов.
 - Основные понятия и принципы поняты частично, есть ошибки в ответах.
 - Знания по большинству тем на базовом уровне, недостаточная глубина понимания.
 - Частичная способность применять теоретические знания на практике, нужны дополнительные разъяснения.
3. **"Хорошо" (60-79%)**
 - Студент ответил правильно на 60-79% вопросов.
 - Хорошее понимание ключевых понятий и принципов, незначительные ошибки.
 - Знания по всем темам на достаточном уровне, однако есть некоторые пробелы.
 - Способность применять теоретические знания на практике, но требуется улучшение точности и уверенности.
4. **"Отлично" (80-100%)**
 - Студент ответил правильно на 80-100% вопросов.
 - Полное и правильное понимание всех ключевых понятий и принципов.
 - Глубокие знания по всем темам, минимальные или отсутствующие ошибки.
 - Высокий уровень способности применять теоретические знания на практике, демонстрация уверенности и точности в ответах.

2.2 Тестовые задания для текущего контроля успеваемости в виде ЗНАНИЙ

В тестовом задании вопросы, которые имеют закрытый характер.

Правильные ответы выделены знаком +.

6 семестр

1. Каковы основные этапы управления проектом разработки ПО? (УК-1.2)
 - Анализ, разработка, тестирование, внедрение, поддержка.
 - Сбор, анализ, проектирование, выпуск, сопровождение.
 - Инициация, планирование, исполнение, мониторинг, завершение. +
 - Дизайн, разработка, тестирование, производство, маркетинг.
2. Какие роли существуют в Scrum-команде? (ПК-3.1)
 - Менеджер проекта, дизайнер, тестировщик.
 - Архитектор, аналитик, пользователь.

- Администратор, оператор, инженер.
 - Product Owner, Scrum Master, Development Team. +
3. Что такое рефакторинг кода? (ПК-2.2)
 - Процесс тестирования производительности кода.
 - Процесс улучшения структуры существующего кода без изменения его внешнего поведения. +
 - Процесс написания нового кода.
 - Процесс документирования кода.
 4. Какие инструменты используются для рефакторинга? (ПК-2.2)
 - Word, Excel, PowerPoint.
 - IntelliJ IDEA, PyCharm, Visual Studio. +
 - Photoshop, Illustrator, CorelDRAW.
 - Git, SVN, Mercurial.
 5. Что такое безопасность программного обеспечения? (ПК-4.1)
 - Совокупность мер для защиты программного обеспечения от угроз и уязвимостей. +
 - Способность программы работать без ошибок.
 - Методика оптимизации производительности.
 - Инструмент для управления версиями.
 6. Какие методы используются для защиты программного обеспечения? (ПК-4.2)
 - Анализ, синтез, тестирование.
 - Компрессия, декомпрессия, архивирование.
 - Визуализация, моделирование, прототипирование.
 - Шифрование, аутентификация, контроль доступа. +
 7. Что такое техническая документация? (ПК-4.1)
 - Документы, описывающие структуру, функциональность и использование программного обеспечения. +
 - Книги по программированию.
 - Файлы с исходным кодом.
 - Инструкции по установке оборудования.
 8. Какие типы документации существуют? (ПК-4.1)
 - Пользовательская документация, системная документация, архитектурная документация. +
 - Литературная документация, аудиовизуальная документация, мультимедийная документация.
 - Техническая документация, графическая документация, финансовая документация.
 - Образовательная документация, юридическая документация, маркетинговая документация.
 9. Какие основные шаги при выпуске программного обеспечения? (ПК-3.2)
 - Подготовка релиза, создание установочного пакета, тестирование релиза, публикация. +
 - Дизайн, разработка, тестирование, маркетинг.
 - Исследование, анализ, проектирование, производство.
 - Планирование, разработка, тестирование, сопровождение.
 10. Какие современные тенденции влияют на процессы разработки ПО? (УК-1.1)
 - Искусственный интеллект, машинное обучение, блокчейн, DevOps. +
 - Аналоговые вычисления, механические компьютеры, перфокарты.
 - Телеграф, факс, кассетные магнитофоны.
 - Графология, гомеопатия, алхимия.

Критерии оценки результатов теста

5. **"Неудовлетворительно" (0-39%)**

- Студент ответил правильно на менее 40% вопросов.
- Значительные пробелы в знаниях по большинству тем.
- Неправильное понимание ключевых понятий и принципов.
- Неспособность применить теоретические знания на практике.

6. **"Удовлетворительно" (40-59%)**

- Студент ответил правильно на 40-59% вопросов.
- Основные понятия и принципы поняты частично, есть ошибки в ответах.
- Знания по большинству тем на базовом уровне, недостаточная глубина понимания.
- Частичная способность применять теоретические знания на практике, нужны дополнительные разъяснения.

7. **"Хорошо" (60-79%)**

- Студент ответил правильно на 60-79% вопросов.
- Хорошее понимание ключевых понятий и принципов, незначительные ошибки.
- Знания по всем темам на достаточном уровне, однако есть некоторые пробелы.
- Способность применять теоретические знания на практике, но требуется улучшение точности и уверенности.

8. **"Отлично" (80-100%)**

- Студент ответил правильно на 80-100% вопросов.
- Полное и правильное понимание всех ключевых понятий и принципов.
- Глубокие знания по всем темам, минимальные или отсутствующие ошибки.
- Высокий уровень способности применять теоретические знания на практике, демонстрация уверенности и точности в ответах.

2.3 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ

5 семестр

1. Что такое жизненный цикл программного обеспечения? (УК-1.1)

- Правильный ответ: Жизненный цикл программного обеспечения включает этапы планирования, анализа требований, проектирования, разработки, тестирования и сопровождения. Это процесс, который начинается с идеи создания программного продукта и заканчивается его выводом из эксплуатации.

2. Какие основные модели жизненного цикла программного обеспечения существуют? (ПК-4.1)

- Правильный ответ: Основные модели жизненного цикла программного обеспечения включают водопадную модель, спиральную модель, Agile и Scrum. Водопадная модель представляет собой последовательный процесс разработки, в то время как Agile и Scrum основываются на итеративном и инкрементальном подходах.

3. В чем разница между водопадной моделью и Agile? (ПК-4.2)

- Правильный ответ: Водопадная модель линейная и последовательная, каждый этап начинается только после завершения предыдущего. Agile итеративная и инкрементальная, работа над продуктом делится на небольшие итерации, что позволяет более гибко реагировать на изменения требований.

4. Какие методы сбора требований существуют? (УК-2.1)

- Правильный ответ: Основные методы сбора требований включают интервью, наблюдение, анализ документации и анкетирование. Эти методы позволяют собрать полную и точную информацию о потребностях пользователей и функциональных требованиях к системе.

5. Что такое спецификация требований? (УК-2.2)

- Правильный ответ: Спецификация требований — это документ, описывающий функциональные и нефункциональные требования к системе. Он включает

описание того, что система должна делать, а также ограничения и критерии производительности.

6. Какие типы диаграмм используются в UML? (ПК-4.2)
 - Правильный ответ: В UML используются различные типы диаграмм, такие как диаграммы классов, диаграммы прецедентов, диаграммы последовательности, диаграммы состояний и диаграммы активности. Эти диаграммы помогают визуализировать структуру и поведение системы.
7. Что такое архитектурная схема? (ПК-4.1)
 - Правильный ответ: Архитектурная схема — это высокоуровневое представление структуры системы, включающее основные компоненты и их взаимодействия. Она помогает понять, как система организована и как ее части взаимодействуют друг с другом.
8. Каковы основные компоненты программы на языке Java? (ПК-1.1)
 - Правильный ответ: Основные компоненты программы на языке Java включают классы, методы, поля и конструкторы. Классы являются основными строительными блоками, методы определяют поведение объектов, поля хранят данные, а конструкторы инициализируют объекты.
9. Какие принципы объектно-ориентированного программирования? (ПК-1.1)
 - Правильный ответ: Принципы объектно-ориентированного программирования включают инкапсуляцию, наследование, полиморфизм и абстракцию. Инкапсуляция скрывает внутренние детали объекта, наследование позволяет создавать новые классы на основе существующих, полиморфизм позволяет использовать объекты разных типов через общий интерфейс, а абстракция выделяет общие характеристики объектов.
10. Что такое модульное тестирование? (ПК-2.1)
 - Правильный ответ: Модульное тестирование — это процесс тестирования отдельных модулей программы на уровне исходного кода. Оно позволяет выявить ошибки в каждом модуле до интеграции их в общую систему, обеспечивая более качественное и надежное программное обеспечение.
11. Какие инструменты используются для автоматизированного тестирования? (ПК-2.1)
 - Правильный ответ: Для автоматизированного тестирования часто используются инструменты, такие как JUnit, Selenium и TestNG. JUnit применяется для модульного тестирования в Java, Selenium — для тестирования веб-приложений, а TestNG — для продвинутого тестирования в Java.
12. Что такое система контроля версий? (ПК-3.1)
 - Правильный ответ: Система контроля версий — это система для управления изменениями в коде и координации работы команды разработчиков. Она позволяет отслеживать изменения, возвращаться к предыдущим версиям и работать нескольким разработчикам над одним проектом. Примеры: Git, SVN, Mercurial.
13. Какие операции можно выполнять с помощью Git? (ПК-3.1)
 - Правильный ответ: С помощью Git можно выполнять операции, такие как коммит, пуш, пул и мердж. Эти операции позволяют сохранять изменения, отправлять их в удаленный репозиторий, получать обновления от других разработчиков и объединять различные ветки разработки.

Критерии оценки ответов на вопросы

- **"Отлично" (5 баллов)**
 - **Критерии:**
 - Полное и точное объяснение вопроса.
 - Ответ включает все ключевые аспекты и детали.
 - Примеры, если требуются, приведены и правильно объяснены.

- Ответ демонстрирует глубокое понимание темы.
- **"Хорошо" (4 балла)**
 - **Критерии:**
 - Корректное объяснение вопроса.
 - Ответ охватывает основные аспекты, но может отсутствовать незначительная деталь или пример.
 - Демонстрируется хорошее, но не полное понимание темы.
- **"Удовлетворительно" (3 балла)**
 - **Критерии:**
 - Общее представление о вопросе.
 - Ответ включает основные аспекты, но содержит неточности или пропуски.
 - Примеры, если требуются, могут отсутствовать или быть неверно объяснены.
 - Демонстрируется базовое понимание темы.
- **"Неудовлетворительно" (2 балла)**
 - **Критерии:**
 - Некорректное или неполное объяснение вопроса.
 - Отсутствие ключевых аспектов и деталей.
 - Примеры, если требуются, отсутствуют или приведены неверные.
 - Ответ демонстрирует недостаточное понимание темы.

2.4 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ

6 семестр

1. Каковы основные этапы управления проектом разработки ПО? (УК-1.2)
 - Правильный ответ: Основные этапы управления проектом разработки ПО включают инициацию, планирование, исполнение, мониторинг и завершение. Эти этапы помогают организовать работу команды, следить за прогрессом и обеспечивать успешное завершение проекта.
2. Какие роли существуют в Scrum-команде? (ПК-3.1)
 - Правильный ответ: В Scrum-команде существуют роли Product Owner, Scrum Master и Development Team. Product Owner отвечает за продукт и взаимодействие с заказчиком, Scrum Master помогает команде следовать Scrum-процессам, а Development Team непосредственно занимается разработкой.
3. Что такое рефакторинг кода? (ПК-2.2)
 - Правильный ответ: Рефакторинг кода — это процесс улучшения структуры существующего кода без изменения его внешнего поведения. Цель рефакторинга — сделать код более читаемым, поддерживаемым и эффективным.
4. Какие инструменты используются для рефакторинга? (ПК-2.2)
 - Правильный ответ: Для рефакторинга часто используются инструменты, такие как IntelliJ IDEA, PyCharm и Visual Studio. Эти IDE предоставляют функции для автоматического рефакторинга кода, что помогает разработчикам улучшать структуру своих программ.
5. Что такое безопасность программного обеспечения? (ПК-4.1)
 - Правильный ответ: Безопасность программного обеспечения — это совокупность мер для защиты программного обеспечения от угроз и уязвимостей. Включает методы шифрования, аутентификации, контроля доступа и другие техники для обеспечения конфиденциальности, целостности и доступности данных.
6. Какие методы используются для защиты программного обеспечения? (ПК-4.2)
 - Правильный ответ: Основные методы защиты программного обеспечения включают шифрование, аутентификацию и контроль доступа. Шифрование защищает данные, аутентификация проверяет подлинность пользователей, а контроль доступа ограничивает доступ к ресурсам.

7. Что такое техническая документация? (ПК-4.1)
- Правильный ответ: Техническая документация — это документы, описывающие структуру, функциональность и использование программного обеспечения. Включает пользовательскую документацию, системную документацию и архитектурную документацию.
8. Какие типы документации существуют? (ПК-4.1)
- Правильный ответ: Существуют различные типы документации, такие как пользовательская документация, системная документация и архитектурная документация. Пользовательская документация предназначена для конечных пользователей, системная — для разработчиков, а архитектурная описывает общую структуру системы.
9. Какие основные шаги при выпуске программного обеспечения? (ПК-3.2)
- Правильный ответ: Основные шаги при выпуске программного обеспечения включают подготовку релиза, создание установочного пакета, тестирование релиза и публикацию. Эти шаги обеспечивают готовность продукта к использованию конечными пользователями.
10. Какие современные тенденции влияют на процессы разработки ПО? (УК-1.1)
- Правильный ответ: Современные тенденции, такие как искусственный интеллект, машинное обучение, блокчейн и DevOps, значительно влияют на процессы разработки ПО. Эти технологии и подходы помогают автоматизировать процессы, улучшить качество и безопасность программного обеспечения, а также ускорить его разработку.

Критерии оценки ответов на вопросы

- **"Отлично" (5 баллов)**
 - **Критерии:**
 - Полное и точное объяснение вопроса.
 - Ответ включает все ключевые аспекты и детали.
 - Примеры, если требуются, приведены и правильно объяснены.
 - Ответ демонстрирует глубокое понимание темы.
- **"Хорошо" (4 балла)**
 - **Критерии:**
 - Корректное объяснение вопроса.
 - Ответ охватывает основные аспекты, но может отсутствовать незначительная деталь или пример.
 - Демонстрируется хорошее, но не полное понимание темы.
- **"Удовлетворительно" (3 балла)**
 - **Критерии:**
 - Общее представление о вопросе.
 - Ответ включает основные аспекты, но содержит неточности или пропуски.
 - Примеры, если требуются, могут отсутствовать или быть неверно объяснены.
 - Демонстрируется базовое понимание темы.
- **"Неудовлетворительно" (2 балла)**
 - **Критерии:**
 - Некорректное или неполное объяснение вопроса.
 - Отсутствие ключевых аспектов и деталей.
 - Примеры, если требуются, отсутствуют или приведены неверные.
 - Ответ демонстрирует недостаточное понимание темы.

2.5 Задачи на соответствие понятий для текущего контроля успеваемости в виде ВЛАДЕНИЙ

Правильные ответы расположены в таблицах друг напротив друга, во время тестирования предполагается что порядок данных в рамках каждого столбца будет случайным.

Задача 1

Соотнесите основные этапы жизненного цикла программного обеспечения с их описаниями (УК-1.1, ПК-4.1)

Этап	Описание
А. Планирование	1. Определение целей и задач проекта, а также ресурсов, необходимых для его выполнения.
В. Анализ требований	2. Сбор и документирование функциональных и нефункциональных требований к системе.
С. Проектирование	3. Создание архитектурной схемы и технической документации системы.
Д. Разработка	4. Написание и интеграция кода.

Правильный ответ: А-1, В-2, С-3, D-4

Задача 2

Соотнесите типы диаграмм UML с их назначениями (ПК-4.2)

Тип диаграммы	Назначение
А. Диаграмма классов	1. Отображение структурных элементов системы и их взаимосвязей.
В. Диаграмма прецедентов	2. Описание взаимодействия пользователей с системой.
С. Диаграмма последовательности	3. Моделирование последовательности взаимодействий между объектами.
Д. Диаграмма состояний	4. Представление состояний объекта и переходов между ними.

Правильный ответ: А-1, В-2, С-3, D-4

Задача 3

Соотнесите принципы объектно-ориентированного программирования с их описаниями (ПК-1.1)

Принцип	Описание
А. Инкапсуляция	1. Скрытие внутренних деталей объекта и предоставление интерфейса для взаимодействия с ним.
В. Наследование	2. Создание новых классов на основе существующих, с наследованием их свойств и методов.

С. Полиморфизм	3. Возможность использования объектов разных типов через общий интерфейс.
D. Абстракция	4. Выделение общих характеристик объектов для упрощения моделирования.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 4

Соотнесите типы тестирования программного обеспечения с их описаниями (ПК-2.1)

Тип тестирования	Описание
A. Модульное тестирование	1. Тестирование отдельных модулей программы на уровне исходного кода.
B. Интеграционное тестирование	2. Тестирование взаимодействия между модулями.
C. Системное тестирование	3. Полное тестирование всей системы на соответствие требованиям.
D. Приемочное тестирование	4. Оценка готовности системы к использованию конечными пользователями.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 5

Соотнесите инструменты для автоматизированного тестирования с их основными характеристиками (ПК-2.1)

Инструмент	Характеристика
A. JUnit	1. Фреймворк для модульного тестирования в Java.
B. Selenium	2. Инструмент для автоматизации тестирования веб-приложений.
C. TestNG	3. Фреймворк для продвинутого тестирования в Java.
D. Jenkins	4. Система для непрерывной интеграции и автоматизации сборки.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 6

Соотнесите понятия с их определениями (ПК-4.1)

Понятие	Определение
A. Спецификация требований	1. Документ, описывающий функциональные и нефункциональные требования к системе.
B. Архитектурная схема	2. Высокоуровневое представление структуры системы.

С. Техническая документация	3. Документы, описывающие структуру, функциональность и использование программного обеспечения.
D. Система контроля версий	4. Система для управления изменениями в коде и координации работы команды разработчиков.

Правильный ответ: А-1, В-2, С-3, D-4

Критерии оценки выполнения задач на соответствие понятий

- **Правильность соответствий:**

- **Отлично (5):** Все соответствия выполнены правильно.
- **Хорошо (4):** 1 ошибка в соответствиях.
- **Удовлетворительно (3):** 2 ошибки в соответствиях.
- **Неудовлетворительно (2):** 3 и более ошибок в соответствиях.

3. Оценочные материалы для проведения промежуточной аттестации обучающихся (студентов)

3.1 Вопросы для проведения промежуточной аттестации в форме ЗАЧЕТА 5 семестр

1. Что такое жизненный цикл программного обеспечения и какие этапы он включает? (УК-1.1)
 - Правильный ответ: Жизненный цикл программного обеспечения включает этапы планирования, анализа требований, проектирования, разработки, тестирования и сопровождения. Эти этапы охватывают весь процесс создания программного продукта от концепции до его завершения и вывода из эксплуатации.
2. Какие существуют модели жизненного цикла программного обеспечения и в чем их основные отличия? (ПК-4.1)
 - Правильный ответ: Основные модели жизненного цикла программного обеспечения включают водопадную модель, спиральную модель, Agile и Scrum. Водопадная модель линейная и последовательная, спиральная модель сочетает итеративные и последовательные подходы, Agile и Scrum являются итеративными и инкрементальными, позволяя гибко реагировать на изменения требований.
3. Какие методы сбора требований существуют и в чем их особенности? (УК-2.1)
 - Правильный ответ: Основные методы сбора требований включают интервью, наблюдение, анализ документации и анкетирование. Интервью позволяет получить информацию непосредственно от пользователей, наблюдение фиксирует реальные действия пользователей, анализ документации изучает существующие материалы, а анкетирование собирает стандартизированные ответы от группы респондентов.
4. Что такое спецификация требований и какую роль она играет в разработке ПО? (УК-2.2)
 - Правильный ответ: Спецификация требований — это документ, описывающий функциональные и нефункциональные требования к системе. Она служит основой для проектирования, разработки и тестирования системы, обеспечивая четкое понимание того, что должно быть реализовано.
5. Каковы основные принципы объектно-ориентированного программирования и как они реализуются в Java? (ПК-1.1)
 - Правильный ответ: Основные принципы объектно-ориентированного программирования включают инкапсуляцию, наследование, полиморфизм и абстракцию. В Java инкапсуляция реализуется через классы и модификаторы доступа, наследование — через ключевое слово **extends**, полиморфизм — через

- перегрузку методов и интерфейсы, абстракция — через абстрактные классы и интерфейсы.
6. Что такое модульное тестирование и какие инструменты для этого используются? (ПК-2.1)
 - Правильный ответ: Модульное тестирование — это процесс тестирования отдельных модулей программы на уровне исходного кода. Оно позволяет выявить ошибки в каждом модуле до их интеграции в общую систему. Для модульного тестирования в Java часто используется JUnit.
 7. Какие функции выполняет система контроля версий и какие операции можно выполнять с ее помощью? (ПК-3.1)
 - Правильный ответ: Система контроля версий управляет изменениями в коде и координирует работу команды разработчиков. С помощью системы контроля версий, такой как Git, можно выполнять операции коммит, пуш, пул и мердж, которые позволяют сохранять изменения, отправлять их в удаленный репозиторий, получать обновления от других разработчиков и объединять различные ветки разработки.
 8. Что такое архитектурная схема и как она помогает в разработке ПО? (ПК-4.1)
 - Правильный ответ: Архитектурная схема — это высокоуровневое представление структуры системы, включающее основные компоненты и их взаимодействия. Она помогает разработчикам понять, как система организована и как ее части взаимодействуют друг с другом, что упрощает проектирование и реализацию системы.
 9. Какие типы диаграмм используются в UML и для чего они предназначены? (ПК-4.2)
 - Правильный ответ: В UML используются различные типы диаграмм, такие как диаграммы классов (для отображения структурных элементов системы и их взаимосвязей), диаграммы прецедентов (для описания взаимодействия пользователей с системой), диаграммы последовательности (для моделирования последовательности взаимодействий между объектами) и диаграммы состояний (для представления состояний объекта и переходов между ними).
 10. Каковы основные компоненты программы на языке Java? (ПК-1.1)
 - Правильный ответ: Основные компоненты программы на языке Java включают классы, методы, поля и конструкторы. Классы являются основными строительными блоками, методы определяют поведение объектов, поля хранят данные, а конструкторы инициализируют объекты.
 11. Какие инструменты используются для автоматизированного тестирования? (ПК-2.1)
 - Правильный ответ: Для автоматизированного тестирования часто используются инструменты, такие как JUnit, Selenium и TestNG. JUnit применяется для модульного тестирования в Java, Selenium — для тестирования веб-приложений, а TestNG — для продвинутого тестирования в Java.
 12. Какие основные принципы инкапсуляции и как они реализуются в Java? (ПК-1.1)
 - Правильный ответ: Инкапсуляция скрывает внутренние детали объекта и предоставляет интерфейс для взаимодействия с ним. В Java инкапсуляция реализуется через использование классов и модификаторов доступа (private, protected, public), которые контролируют доступ к полям и методам класса.
 13. Что такое архитектурные паттерны и как они используются в разработке ПО? (ПК-4.1)
 - Правильный ответ: Архитектурные паттерны — это повторяемые решения общих архитектурных проблем в разработке ПО. Примеры включают паттерн Model-View-Controller (MVC), который разделяет логику приложения на три взаимосвязанных компонента, и паттерн многослойной архитектуры, который разделяет систему на уровни представления, бизнес-логики и данных.
 14. Как проводится модульное тестирование и какие преимущества оно дает? (ПК-2.1)

- Правильный ответ: Модульное тестирование проводится путем создания и выполнения тестов для отдельных модулей программы. Преимущества включают раннее выявление ошибок, упрощение отладки и улучшение качества кода.

Критерии оценивания для зачета (зачет/незачет)

1. Понимание основных понятий и терминов

- **Зачет:** Студент правильно использует основные понятия и термины, объясняет их в контексте.
- **Незачет:** Студент не знает или неправильно использует основные понятия и термины, не может объяснить их.

2. Точность и полнота ответов на теоретические вопросы

- **Зачет:** Ответы на большинство вопросов полные и точные, основные аспекты тем раскрыты.
- **Незачет:** Ответы неполные, содержат значительные ошибки или отсутствуют важные аспекты тем.

3. Способность к применению знаний на практике

- **Зачет:** Студент правильно решает практические задачи, выполняет SQL-запросы без ошибок.
- **Незачет:** Студент допускает ошибки при решении практических задач, не может правильно выполнить SQL-запросы.

4. Качество выполнения практических заданий

- **Зачет:** Практические задания выполнены корректно, все необходимые элементы присутствуют.
- **Незачет:** Практические задания выполнены некорректно, содержат ошибки или неполные.

Окончательное решение по зачету:

- **Зачет:** Студент демонстрирует понимание основных понятий, корректно отвечает на теоретические вопросы и выполняет практические задания.
- **Незачет:** Студент не демонстрирует понимания основных понятий, отвечает с ошибками на теоретические вопросы и/или не справляется с практическими заданиями.

3.2 Вопросы для проведения промежуточной аттестации в форме ЭКЗАМЕНА

6 семестр

1. Опишите жизненный цикл программного обеспечения. Какие основные этапы он включает и как они взаимосвязаны? (УК-1.1)
 - Правильный ответ: Жизненный цикл программного обеспечения включает этапы планирования, анализа требований, проектирования, разработки, тестирования, внедрения и сопровождения. Эти этапы взаимосвязаны: результаты одного этапа влияют на последующие, и часто итерации повторяются для улучшения и доработки системы.
2. Какие модели жизненного цикла программного обеспечения вы знаете? Охарактеризуйте основные отличия между ними. (ПК-4.1)
 - Правильный ответ: Основные модели жизненного цикла программного обеспечения включают водопадную модель, спиральную модель, Agile и Scrum. Водопадная модель линейная и последовательная, каждый этап начинается только после завершения предыдущего. Спиральная модель сочетает итеративные и последовательные подходы, добавляя управление рисками на каждом витке спирали. Agile и Scrum основываются на итеративном и инкрементальном подходах, обеспечивая гибкость и возможность адаптации к изменениям требований.
3. Какие методы сбора требований существуют и в чем их особенности? (УК-2.1)

- Правильный ответ: Основные методы сбора требований включают интервью, наблюдение, анализ документации и анкетирование. Интервью позволяет получить информацию непосредственно от пользователей, наблюдение фиксирует реальные действия пользователей, анализ документации изучает существующие материалы, а анкетирование собирает стандартизированные ответы от группы респондентов. Каждый метод имеет свои особенности и применяется в зависимости от контекста проекта и потребностей.
- 4. Что такое спецификация требований и какую роль она играет в разработке ПО? (УК-2.2)
 - Правильный ответ: Спецификация требований — это документ, описывающий функциональные и нефункциональные требования к системе. Она служит основой для проектирования, разработки и тестирования системы, обеспечивая четкое понимание того, что должно быть реализовано. Спецификация требований помогает избежать недоразумений между заказчиками и разработчиками, обеспечивая ясность и полноту описания системы.
- 5. Каковы основные принципы объектно-ориентированного программирования и как они реализуются в Java? (ПК-1.1)
 - Правильный ответ: Основные принципы объектно-ориентированного программирования включают инкапсуляцию, наследование, полиморфизм и абстракцию. В Java инкапсуляция реализуется через классы и модификаторы доступа, наследование — через ключевое слово `extends`, полиморфизм — через перегрузку методов и интерфейсы, абстракция — через абстрактные классы и интерфейсы. Эти принципы помогают создавать гибкие и легко поддерживаемые системы.
- 6. Что такое модульное тестирование и какие инструменты для этого используются? (ПК-2.1)
 - Правильный ответ: Модульное тестирование — это процесс тестирования отдельных модулей программы на уровне исходного кода. Оно позволяет выявить ошибки в каждом модуле до их интеграции в общую систему. Для модульного тестирования в Java часто используется JUnit, который предоставляет инструменты для написания и выполнения тестов, а также для создания отчетов о результатах тестирования.
- 7. Какие функции выполняет система контроля версий и какие операции можно выполнять с ее помощью? (ПК-3.1)
 - Правильный ответ: Система контроля версий управляет изменениями в коде и координирует работу команды разработчиков. С помощью системы контроля версий, такой как Git, можно выполнять операции коммит, пуш, пул и мердж, которые позволяют сохранять изменения, отправлять их в удаленный репозиторий, получать обновления от других разработчиков и объединять различные ветки разработки. Это помогает избежать конфликтов и сохранять историю изменений проекта.
- 8. Что такое архитектурная схема и как она помогает в разработке ПО? (ПК-4.1)
 - Правильный ответ: Архитектурная схема — это высокоуровневое представление структуры системы, включающее основные компоненты и их взаимодействия. Она помогает разработчикам понять, как система организована и как ее части взаимодействуют друг с другом, что упрощает проектирование и реализацию системы. Архитектурная схема также служит основой для последующего детального проектирования и документирования.
- 9. Какие типы диаграмм используются в UML и для чего они предназначены? (ПК-4.2)
 - Правильный ответ: В UML используются различные типы диаграмм, такие как диаграммы классов (для отображения структурных элементов системы и их взаимосвязей), диаграммы прецедентов (для описания взаимодействия пользователей с системой), диаграммы последовательности (для моделирования

последовательности взаимодействий между объектами) и диаграммы состояний (для представления состояний объекта и переходов между ними). Эти диаграммы помогают визуализировать структуру и поведение системы на различных уровнях абстракции.

10. Каковы основные компоненты программы на языке Java? (ПК-1.1)

- Правильный ответ: Основные компоненты программы на языке Java включают классы, методы, поля и конструкторы. Классы являются основными строительными блоками, методы определяют поведение объектов, поля хранят данные, а конструкторы инициализируют объекты. Эти компоненты используются для создания объектов и реализации функциональности программного обеспечения.

11. Какие инструменты используются для автоматизированного тестирования? (ПК-2.1)

- Правильный ответ: Для автоматизированного тестирования часто используются инструменты, такие как JUnit, Selenium и TestNG. JUnit применяется для модульного тестирования в Java, Selenium — для тестирования веб-приложений, а TestNG — для продвинутого тестирования в Java. Эти инструменты помогают автоматизировать процесс тестирования и улучшить качество программного обеспечения.

12. Какие основные принципы инкапсуляции и как они реализуются в Java? (ПК-1.1)

- Правильный ответ: Инкапсуляция скрывает внутренние детали объекта и предоставляет интерфейс для взаимодействия с ним. В Java инкапсуляция реализуется через использование классов и модификаторов доступа (`private`, `protected`, `public`), которые контролируют доступ к полям и методам класса. Это позволяет защитить данные и обеспечить целостность объекта.

13. Что такое архитектурные паттерны и как они используются в разработке ПО? (ПК-4.1)

- Правильный ответ: Архитектурные паттерны — это повторяемые решения общих архитектурных проблем в разработке ПО. Примеры включают паттерн Model-View-Controller (MVC), который разделяет логику приложения на три взаимосвязанных компонента, и паттерн многослойной архитектуры, который разделяет систему на уровни представления, бизнес-логики и данных. Использование архитектурных паттернов помогает улучшить структуру и управляемость системы.

14. Как проводится модульное тестирование и какие преимущества оно дает? (ПК-2.1)

- Правильный ответ: Модульное тестирование проводится путем создания и выполнения тестов для отдельных модулей программы. Преимущества включают раннее выявление ошибок, упрощение отладки и улучшение качества кода. Модульное тестирование помогает обеспечить надежность и устойчивость системы за счет тщательной проверки каждого модуля перед интеграцией в общий проект.

15. Какие роли существуют в Scrum-команде и какие функции они выполняют? (ПК-3.1)

- Правильный ответ: В Scrum-команде существуют роли Product Owner, Scrum Master и Development Team. Product Owner отвечает за формирование и приоритизацию бэклога продукта, взаимодействие с заказчиком и обеспечение максимальной ценности продукта. Scrum Master помогает команде следовать Scrum-процессам, устраняет препятствия и обеспечивает эффективное взаимодействие внутри команды. Development Team непосредственно занимается разработкой, тестированием и выпуском инкрементов продукта.

16. Что такое рефакторинг кода и какие инструменты используются для этого процесса? (ПК-2.2)

- Правильный ответ: Рефакторинг кода — это процесс улучшения структуры существующего кода без изменения его внешнего поведения. Цель рефакторинга — сделать код более читаемым, поддерживаемым и эффективным. Для

рефакторинга часто используются инструменты, такие как IntelliJ IDEA, PyCharm и Visual Studio, которые предоставляют функции для автоматического рефакторинга кода, что помогает разработчикам улучшать структуру своих программ.

17. Какие методы используются для защиты программного обеспечения и как они реализуются? (ПК-4.2)
- Правильный ответ: Основные методы защиты программного обеспечения включают шифрование, аутентификацию и контроль доступа. Шифрование защищает данные от несанкционированного доступа путем преобразования их в нечитаемый формат. Аутентификация проверяет подлинность пользователей с помощью паролей, биометрических данных или других средств идентификации. Контроль доступа ограничивает доступ к ресурсам системы в зависимости от прав и ролей пользователей. Эти методы помогают обеспечить безопасность и защиту данных.
18. Какие типы документации существуют в разработке программного обеспечения и для чего они предназначены? (ПК-4.1)
- Правильный ответ: Существуют различные типы документации в разработке программного обеспечения, такие как пользовательская документация, системная документация и архитектурная документация. Пользовательская документация предназначена для конечных пользователей и описывает, как использовать программное обеспечение. Системная документация предназначена для разработчиков и описывает внутреннюю структуру и функционирование системы. Архитектурная документация описывает общую структуру системы и взаимодействие между ее компонентами. Эти документы помогают в понимании, разработке и поддержке программного обеспечения.
19. Какие основные шаги при выпуске программного обеспечения и как они выполняются? (ПК-3.2)
- Правильный ответ: Основные шаги при выпуске программного обеспечения включают подготовку релиза, создание установочного пакета, тестирование релиза и публикацию. Подготовка релиза включает в себя завершение всех задач и исправление ошибок. Создание установочного пакета подразумевает сборку всех компонентов системы в единый установочный файл. Тестирование релиза включает проверку всех функций и производительности системы. Публикация означает выпуск программного обеспечения для конечных пользователей через различные каналы распространения, такие как веб-сайты или магазины приложений.
20. Какие современные тенденции влияют на процессы разработки ПО и как они изменяют подходы к разработке? (УК-1.1)
- Правильный ответ: Современные тенденции, такие как искусственный интеллект, машинное обучение, блокчейн и DevOps, значительно влияют на процессы разработки ПО. Искусственный интеллект и машинное обучение позволяют автоматизировать рутинные задачи и улучшить анализ данных. Блокчейн обеспечивает безопасные и прозрачные транзакции. DevOps интегрирует процессы разработки и эксплуатации, ускоряя выпуск новых версий и улучшая качество ПО. Эти тенденции помогают разработчикам создавать более эффективные и инновационные решения.
21. Каковы основные подходы к тестированию программного обеспечения и чем они отличаются? (ПК-2.1)
- Правильный ответ: Основные подходы к тестированию программного обеспечения включают модульное, интеграционное, системное и приемочное тестирование. Модульное тестирование проверяет отдельные модули кода, интеграционное тестирование проверяет взаимодействие между модулями,

системное тестирование оценивает систему в целом, а приемочное тестирование проверяет соответствие системы требованиям заказчика. Каждый подход имеет свои цели и методы, что позволяет выявить различные типы ошибок на разных этапах разработки.

22. В чем заключается важность управления зависимостями в разработке программного обеспечения и какие инструменты для этого используются? (ПК-3.1)
- Правильный ответ: Управление зависимостями в разработке программного обеспечения важно для обеспечения правильной работы всех компонентов системы. Инструменты, такие как Maven и Gradle для Java, позволяют автоматизировать процесс управления зависимостями, включая их установку, обновление и удаление. Это помогает избежать конфликтов версий и обеспечить стабильность системы.
23. Что такое DevOps и как этот подход изменяет процессы разработки и эксплуатации программного обеспечения? (УК-1.2)
- Правильный ответ: DevOps — это подход, интегрирующий процессы разработки и эксплуатации программного обеспечения для ускорения выпуска продуктов и улучшения их качества. DevOps включает в себя автоматизацию, мониторинг, совместную работу и постоянное улучшение процессов. Инструменты, такие как Jenkins, Docker и Kubernetes, используются для автоматизации сборки, тестирования и развертывания приложений, что позволяет сократить время разработки и повысить надежность системы.
24. Каковы основные принципы проектирования программного обеспечения и как они помогают создавать качественные продукты? (ПК-4.1)
- Правильный ответ: Основные принципы проектирования программного обеспечения включают модульность, повторное использование кода, разделение ответственности, слабую связность и высокую связность. Эти принципы помогают создавать гибкие, масштабируемые и легко поддерживаемые системы. Применение этих принципов позволяет улучшить качество кода, упростить его тестирование и отладку, а также ускорить процесс разработки.

Критерии оценки ответов на экзамене

- **"Отлично" (5 баллов)**
 - Критерии:
 - Полное и точное объяснение вопроса.
 - Ответ включает все ключевые аспекты и детали.
 - Примеры, если требуются, приведены и правильно объяснены.
 - Ответ демонстрирует глубокое понимание темы.
- **"Хорошо" (4 балла)**
 - Критерии:
 - Корректное объяснение вопроса.
 - Ответ охватывает основные аспекты, но может отсутствовать незначительная деталь или пример.
 - Демонстрируется хорошее, но не полное понимание темы.
- **"Удовлетворительно" (3 балла)**
 - Критерии:
 - Общее представление о вопросе.
 - Ответ включает основные аспекты, но содержит неточности или пропуски.
 - Примеры, если требуются, могут отсутствовать или быть неверно объяснены.
 - Демонстрируется базовое понимание темы.
- **"Неудовлетворительно" (2 балла)**
 - Критерии:

- Некорректное или неполное объяснение вопроса.
- Отсутствие ключевых аспектов и деталей.
- Примеры, если требуются, отсутствуют или приведены неверные.
- Ответ демонстрирует недостаточное понимание темы.