

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ



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

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.В.11 «Высокоуровневые методы информатики и
программирования»

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль) программы бакалавриата
«Информационные системы в экономике»

ОДОБРЕНО

Ученым советом ЧУВО «ВШП»

Протокол заседания

№01-02/23 от 15 мая 2023 г.



	Документ подписан электронной цифровой подписью VSHP EDS GEN 1, уникальный ключ документа:
	8F30-29EE-EB2F-GNI5
Организация:	ЧУВО «ВШП», ИНН: 6903013604
Дата подписания:	15.05.2023
Подписал:	Аллабян М. Г.

Тверь, 2023

Рабочая программа учебной дисциплины **Б1.В.11 Высокоуровневые методы информатики и программирования**, компонента основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** направленность (профиль) **«Информационные системы в экономике»**, 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 системы в экономике.

2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Целью изучения дисциплины «Высокоуровневые методы информатики и программирования» является изучение основных принципов процедурного и модульного программирования; обучение правилам и подходам к разработке алгоритмов, кодированию и отладке программ на языке программирования C++, пригодных для практического применения, изучение принципов работы программного средства Dev C++ и его использование при решении прикладных задач.

Основными задачами изучения дисциплины являются:

- изучение основных синтаксических конструкций языка программирования C++, правил и рекомендаций построения программ на указанном языке;
- изучение возможностей среды разработки программного обеспечения Dev C++;
- привитие практических умений и навыков разрабатывать алгоритмы и программы, пригодные для практического применения;
- привития умений писать и отлаживать коды на языке программирования C++, тестировать работоспособность программы на указанном языке, навыков отладки и тестирования работоспособности программ;
- привитие умений и навыков применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении прикладных задач различных классов, их отладки и тестирования работоспособности.

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 системы в экономике.

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	<p>Знать:</p> <ul style="list-style-type: none"> - Основные источники информации для разработки программного обеспечения (далее ПО). - Методы критического анализа информации. <p>Уметь:</p> <ul style="list-style-type: none"> - Находить актуальную информацию в специализированных источниках. - Анализировать и оценивать достоверность и релевантность информации. <p>Владеть:</p> <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	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы системного подхода в разработке ПО. - Основные этапы системного анализа и проектирования. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять системный подход для анализа задач. - Разрабатывать системные решения для поставленных задач. <p>Владеть:</p> <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	<p>Знать:</p> <ul style="list-style-type: none"> - Методы декомпозиции целей на задачи. - Принципы приоритизации задач. <p>Уметь:</p> <ul style="list-style-type: none"> - Определять и формулировать задачи на основе целей проекта. - Устанавливать приоритеты для выполнения задач. <p>Владеть:</p> <ul style="list-style-type: none"> - 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	<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.

			<ul style="list-style-type: none"> - Навыками принятия оптимальных решений в рамках правовых норм.
ПК-1	Разработка и отладка программного кода	ПК-1.1 Способен разрабатывать программный код	Знать: <ul style="list-style-type: none"> - Основные языки программирования для разработки ПО. - Принципы написания качественного кода. Уметь: <ul style="list-style-type: none"> - Писать программный код на одном из изучаемых языков. - Разрабатывать алгоритмы и структуры данных для решения задач. Владеть: <ul style="list-style-type: none"> - 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	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:

	выпусков программного продукта		<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 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 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.
		<p>ПК-4.2 Способен проектировать программное обеспечение</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы проектирования ПО. - Основные подходы к архитектурному дизайну. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать архитектуру и дизайн ПО. - Использовать UML и другие инструменты для проектирования. <p>Владеть:</p> <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 5 зачетных единиц, 180 часов, включая все формы контактной и самостоятельной работы обучающихся.

Объем дисциплины по учебному плану составляет – 5 зачётных единицы = 180 академических часов.

Контактная работа обучающегося (студенты) с научно-педагогическим работником организации (всего) - 62 академических часов,

в том числе:

Лекционные занятия (Лек.) - 20 академических часов,

Практические занятия (Пр.) - 40 академических часов,

Консультации (Конс.) - 2 академических часа.

Самостоятельная работа обучающегося (студента):

Самостоятельная работа (СР) - 82 академический час,

Текущий контроль успеваемости

и промежуточно-заочная аттестация обучающегося (студента):

Часы на контроль - 36 академических часов.

Таблица 2. Объём дисциплины

№ п/п	Раздел/тема дисциплины	Семестр/курс	Виды учебной деятельности, включая самостоятельную работу обучающихся (студентов), и трудоёмкость (в ак. часах)				Коды формируемых компетенций
			Виды учебных занятий по дисциплине			Самостоятельная работа СР	
			Лек.	Пр.	Консульт		
1	Тема 1. Введение в высокоуровневые методы информатики и программирования	9 семестр/ 5 курс	2	4		8	УК-1.1, УК-1.2
2	Тема 2. Парадигмы программирования	9 семестр/ 5 курс	2	4		8	УК-2.1, УК-2.2
3	Тема 3. Объектно-ориентированное программирование (ООП)	9 семестр/ 5 курс	2	4		8	ПК-1.1, ПК-1.2
4	Тема 4. Функциональное программирование	9 семестр/ 5 курс	2	4		8	ПК-1.1, ПК-2.2
5	Тема 5. Логическое программирование	9 семестр/ 5 курс	2	4		8	ПК-1.1, ПК-2.2
6	Тема 6 Методы и паттерны программирования	9 семестр/ 5 курс	2	4		8	ПК-4.1, ПК-4.2
7	Тема 7. Генетические алгоритмы и эвристические методы	9 семестр/ 5 курс	2	4		8	ПК-2.1, ПК-2.2
8	Тема 8 Современные техники программирования	9 семестр/ 5 курс	4	4		8	ПК-3.1, ПК-3.2
9	Тема 9 Архитектурные стили и шаблоны	9 семестр/ 5 курс	2	4		8	ПК-4.1, ПК-4.2
10	Тема 10 Итоговое занятие и подготовка к экзамену	9 семестр/ 5 курс		4	2	10	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:	9 семестр/ 5 курс	62 ак. часов		82 ак. часа	-		
Часы на контроль	9 семестр/ 5 курс	36 ак. час (форма промежуточной аттестации – экзамен – 9 семестр)					
ВСЕГО ак. часов:	9 семестр/ 5 курс	180 академических часа					

5.2. Тематическое содержание дисциплины

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

Тема 1: Введение в высокоуровневые методы информатики и программирования

Содержание: Основные понятия и методы высокоуровневого программирования. Преимущества использования высокоуровневых методов по сравнению с низкоуровневыми.

Тема 2: Парадигмы программирования

Содержание: Обзор парадигм программирования: процедурное, объектно-ориентированное, функциональное, логическое программирование. Примеры языков программирования для каждой парадигмы.

Тема 3: Объектно-ориентированное программирование (ООП)

Содержание: Принципы ООП: инкапсуляция, наследование, полиморфизм, абстракция. Реализация ООП на примере языков C++, Java и Python. Примеры проектирования и реализации классов и объектов.

Тема 4: Функциональное программирование

Содержание: Основные концепции функционального программирования: чистые функции, неизменяемость данных, функции высшего порядка. Реализация на примере языка Haskell и Scala.

Тема 5: Логическое программирование

Содержание: Основы логического программирования. Языки логического программирования, такие как Prolog. Примеры задач, решаемых с помощью логического программирования.

Тема 6: Методы и паттерны программирования

Содержание: Обзор ключевых паттернов проектирования: Singleton, Factory, Observer, Decorator, Strategy и другие. Примеры применения паттернов на различных языках программирования.

Тема 7: Генетические алгоритмы и эвристические методы

Содержание: Основы генетических алгоритмов и эвристических методов. Применение генетических алгоритмов для решения сложных задач оптимизации. Примеры на языках Python и C++.

Тема 8: Современные техники программирования

Содержание: Современные техники программирования: метапрограммирование, аспектно-ориентированное программирование, программирование на основе контрактов. Примеры и реализация на различных языках программирования.

Тема 9: Архитектурные стили и шаблоны

Содержание: Основные архитектурные стили: многослойная архитектура, микросервисы, REST,

MVC и другие. Примеры их применения в различных проектах.

Тема 10: Итоговое занятие и подготовка к экзамену

Содержание: Повторение и обобщение пройденного материала. Решение типовых задач и вопросов для подготовки к экзамену. Обсуждение практических аспектов высокоуровневого программирования и системного подхода к разработке программного обеспечения.

5.2.1 Содержание практических занятий

Таблица 3

№ п/п	Наименование темы (раздела) дисциплины	Содержание практического занятия
1	Практическое занятие 1: Введение в высокоуровневые методы программирования	Задание: Ознакомиться с основными понятиями и методами высокоуровневого программирования. Выполнить сравнительный анализ высокоуровневых и низкоуровневых методов программирования. Цель: Понять преимущества и области применения высокоуровневых методов программирования.
2	Практическое занятие 2: Парадигмы программирования	Задание: Написать примеры программ на различных парадигмах программирования (процедурное, объектно-ориентированное, функциональное, логическое). Проанализировать различия и особенности каждой парадигмы. Цель: Освоить основные парадигмы программирования и понять их применение.
3	Практическое занятие 3: Объектно-ориентированное программирование (ООП)	Задание: Разработать классы и объекты с использованием принципов ООП на языках C++, Java и Python. Реализовать наследование, инкапсуляцию и полиморфизм в своих проектах. Цель: Закрепить знания и навыки в ООП, научиться проектировать и реализовывать классы и объекты.
4	Практическое занятие 4: Функциональное программирование	Задание: Написать программы, используя функциональное программирование на языках Haskell и Scala. Реализовать чистые функции, неизменяемость данных и функции высшего порядка. Цель: Освоить основные концепции функционального программирования и научиться их применять на практике.
5	Практическое занятие 5: Логическое программирование	Задание: Разработать решения задач с использованием логического программирования на языке Prolog. Создать факты, правила и запросы для решения задач. Цель: Научиться использовать логическое программирование для решения задач, требующих логических выводов.
6	Практическое занятие 6: Методы и паттерны программирования	Задание: Реализовать ключевые паттерны проектирования, такие как Singleton, Factory, Observer, Decorator, Strategy, на различных языках программирования. Проанализировать их применение в реальных проектах. Цель: Освоить паттерны проектирования и научиться их применять для решения различных задач программирования.
7	Практическое занятие 7: Генетические алгоритмы и эвристические методы	Задание: Написать программы, использующие генетические алгоритмы для решения задач оптимизации на языках Python и C++. Реализовать основные компоненты генетических алгоритмов: селекция, кроссовер, мутация. Цель: Освоить принципы генетических алгоритмов и научиться применять их для решения сложных задач оптимизации.
8	Практическое занятие 8: Современные техники программирования	Задание: Реализовать примеры программ, использующих метапрограммирование, аспектно-ориентированное программирование и программирование на основе контрактов на различных языках программирования. Цель: Изучить и освоить современные техники программирования, которые повышают гибкость и надежность программного кода.
9	Практическое занятие 9: Архитектурные стили и шаблоны	Задание: Разработать архитектуру программного проекта, используя многослойную архитектуру, микросервисы, REST или MVC. Реализовать основные компоненты и взаимодействие между ними. Цель: Освоить основные архитектурные стили и шаблоны, научиться их применять в разработке программного обеспечения.

10	Практическое занятие 10: Итоговое занятие и подготовка к экзамену	Задание: Повторение пройденного материала, решение типовых задач и вопросов для подготовки к экзамену. Обсуждение практических аспектов высокоуровневого программирования и системного подхода к разработке программного обеспечения. Цель: Обобщение и закрепление изученного материала для успешной сдачи экзамена.
----	--	--

5.2.2 Содержание самостоятельной работы

Таблица 4

№ п/п	Наименование темы (раздела) дисциплины	Содержание самостоятельной работы	Форма контроля
1	Самостоятельная работа 1: Введение в высокоуровневые методы программирования	Задание: Прочитать рекомендованные материалы по основным понятиям и методам высокоуровневого программирования. Подготовить краткий обзор преимуществ и областей применения высокоуровневых методов. Цель: Понять основные понятия и преимущества высокоуровневых методов.	Ответы на контрольные вопросы.
2	Самостоятельная работа 2: Парадигмы программирования	Задание: Изучить материалы по парадигмам программирования. Выполнить дополнительные упражнения по каждой парадигме. Подготовить сравнительный анализ парадигм. Цель: Освоить основные парадигмы программирования и их отличия.	Ответы на контрольные вопросы.
3	Самостоятельная работа 3: Объектно-ориентированное программирование (ООП)	Задание: Прочитать материалы по принципам ООП. Реализовать небольшие проекты на языках C++, Java и Python с использованием принципов ООП. Цель: Закрепить знания и навыки в ООП.	Выполнение практического задания.
4	Самостоятельная работа 4: Функциональное программирование	Задание: Изучить материалы по функциональному программированию. Написать программы, использующие функциональное программирование на языках Haskell и Scala. Цель: Освоить основные концепции функционального программирования.	Выполнение практического задания.
5	Самостоятельная работа 5: Логическое программирование	Задание: Прочитать материалы по логическому программированию. Решить задачи, используя язык Prolog. Цель: Научиться использовать логическое программирование для решения задач.	Выполнение практического задания.
6	Самостоятельная работа 6: Методы и паттерны программирования	Задание: Изучить материалы по ключевым паттернам проектирования. Реализовать паттерны проектирования на различных языках программирования. Цель: Освоить паттерны проектирования и их применение.	Выполнение практического задания.
7	Самостоятельная работа 7: Генетические алгоритмы и эвристические методы	Задание: Прочитать материалы по генетическим алгоритмам и эвристическим методам. Реализовать генетический алгоритм для решения задачи оптимизации на языке Python. Цель: Освоить принципы генетических алгоритмов и их применение.	Выполнение практического задания.
8	Самостоятельная работа 8: Современные техники программирования	Задание: Изучить материалы по метапрограммированию, аспектно-ориентированному программированию и программированию на основе контрактов. Написать программы с использованием этих техник. Цель: Изучить и освоить современные техники программирования.	Выполнение практического задания.
9	Самостоятельная работа 9:	Задание: Прочитать материалы по архитектурным стилям и шаблонам. Разработать архитектуру программного	Тест.

	Архитектурные стили и шаблоны	проекта, используя изученные стили и шаблоны. Цель: Освоить архитектурные стили и шаблоны, научиться их применять.	
10	Самостоятельная работа 10: Итоговое занятие и подготовка к экзамену	Задание: Повторение пройденного материала, решение типовых задач и вопросов для подготовки к экзамену. Подготовить доклад по одной из изученных тем. Цель: Повторение и обобщение изученного материала для успешной сдачи экзамена.	Экзамен.

6.Оценочные материалы по дисциплине

Оценочные материалы находятся в документе «Оценочные материалы по дисциплине «Высокоуровневые методы информатики и программирования».

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 описки, 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 (студента) 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 системы в экономике»), 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 Институтом основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** (направленность (профиль) «Информационные системы в экономике»), 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 системы в экономике»), 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 обучающего (бакалавра), осваивающего основную образовательную программу высшего образования в учебной группе) и условия организации обучения, изложенного в виде утверждённой в установленном Институте 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 системы в экономике»), 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. Семакин И.Г., Программирование, численные методы и математическое моделирование [Электронный ресурс] : учебное пособие / И.Г. Семакин, О.Л. Русакова, Е.Л. Тарунин, А.П. Шкарапута. - М. : КноРус, 2023. - 298 с. - ISBN 978-5-406-10904-5. - Режим доступа: <https://book.ru/book/947073>

2. Методы оптимальных решений (Экономико-математические методы и моделирование) [Электронный ресурс] : учебное пособие / С.И. Макаров, М.В. Курганова, Е.Ю. Нуйкина [и др.]

; под ред. С.И. Макарова. - М. : КноРус, 2022. - 298 с. - ISBN 978-5-406-09775-5. - Режим доступа: <https://book.ru/book/944117>

Дополнительная литература:

1. Зыков, С.В. Основы современного программирования. Разработка гетерогенных систем в Интернет-ориентированной среде [Электронный ресурс] : учебное пособие / С.В. Зыков. - Саратов : Ай Пи Эр Медиа, 2017. - 484 с. - ISBN 978-5-9908055-9-0

2. Иванова Г.С., Технология программирования [Электронный ресурс] : учебник / Г.С. Иванова. - М. : КноРус, 2022. - 333 с. - ISBN 978-5-406-10176-6. - Режим доступа: <https://book.ru/book/944682>

3. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.

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: «НТЦ 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.

<p>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</p>	<p>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)</p>
<p>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 оборудования:</p> <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. 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>
<p>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 обучающихся, с перечнем основного оборудования:</p>	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>

<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. 	
<p>Помещение для самостоятельной работы обучающихся с перечнем основного оборудования:</p> <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. 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>
<p>Помещение для практических занятий на персональных компьютерах:</p> <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; - Ноутбуки с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**



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

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.В.11 «Высокоуровневые методы информатики и программирования»

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль) программы бакалавриата

«Информационные системы в экономике»

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	<p>Знать:</p> <ul style="list-style-type: none"> - Основные источники информации для разработки программного обеспечения (далее ПО). - Методы критического анализа информации. <p>Уметь:</p> <ul style="list-style-type: none"> - Находить актуальную информацию в специализированных источниках. - Анализировать и оценивать достоверность и релевантность информации. <p>Владеть:</p> <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	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы системного подхода в разработке ПО. - Основные этапы системного анализа и проектирования. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять системный подход для анализа задач. - Разрабатывать системные решения для поставленных задач. <p>Владеть:</p> <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	<p>Знать:</p> <ul style="list-style-type: none"> - Методы декомпозиции целей на задачи. - Принципы приоритизации задач. <p>Уметь:</p> <ul style="list-style-type: none"> - Определять и формулировать задачи на основе целей проекта. - Устанавливать приоритеты для выполнения задач. <p>Владеть:</p> <ul style="list-style-type: none"> - 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	<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	ПК-1.1 Способен разрабатывать программный код	<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.
		ПК-1.2 Способен отлаживать программный код	<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	ПК-2.1 Способен проверять работоспособность программного кода	<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.

		ПК-2.2 Способен рефакторить программный код	<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	ПК-3.1 Способен интегрировать программные модули и компоненты	<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.
		ПК-3.2 Способен проверять работоспособность выпусков программного продукта	<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	ПК-4.1 Способен разрабатывать требования к программному обеспечению	<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	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы проектирования ПО. - Основные подходы к архитектурному дизайну. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать архитектуру и дизайн ПО. - Использовать UML и другие инструменты для проектирования. <p>Владеть:</p> <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)

<i>Показатель оценивания компетенций</i>	<i>Результат обучения</i>	<i>Критерии оценивания компетенций</i>
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»	
УК-1, УК-2, ПК-1, ПК-2.	Обучаемый продемонстрировал: глубокие исчерпывающие	Обучаемый продемонстрировал: твердые и достаточно	Обучаемый продемонстрировал: твердые знания и понимание	Обучаемый продемонстрировал: неправильные ответы на основные вопросы;	Тестовые задания

<p><u>ПК-3</u> <u>ПК-4</u> Знать:</p>	<p>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.</p>	<p>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.</p>	<p>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.</p>	<p>грубые ошибки в ответах; непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы; не владеет основной литературой, рекомендованной учебной программой дисциплины.</p>	
<p><u>УК-1</u> <u>УК-2</u> <u>ПК-1</u> <u>ПК-2</u> <u>ПК-3</u> <u>ПК-4</u> Уметь:</p>	<p>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</p>	<p>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</p>	<p>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</p>	<p>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 литературой, рекомендованной учебной программой дисциплины</p>	<p>Вопросы Практические задания</p>
<p><u>УК-1</u> <u>УК-2</u> <u>ПК-1</u> <u>ПК-2</u> <u>ПК-3</u> <u>ПК-4</u> Владеть:</p>	<p>Обучаемый продемонстрировал: понимание учебного материала; правильные и конкретные ответы, включая дополнительные и уточняющие вопросы;</p>	<p>Обучаемый продемонстрировал: понимание учебного материала; продемонстрировал логически последовательные, достаточно полные, верные ответы;</p>	<p>Обучаемый продемонстрировал: понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы, в том числе при наводящих вопросах</p>	<p>Обучаемый продемонстрировал: непонимание основного учебного материала; дал неправильные ответы на поставленные вопросы; не владеет основной учебной литературой, рекомендованной</p>	<p>Вопросы Практические задания</p>

	свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	самостоятельно устранил замечания по отдельным элементам; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины	учебной программой дисциплины	
--	---	---	--	-------------------------------	--

1. Оценочные материалы для самостоятельной работы обучающихся (студентов)

1.1 Доклад

Доклад позволит студентам углубиться в высокоуровневые методы информатики и программирования, а также понять их важность и применение в разработке программного обеспечения. Доклад покрывает компетенции 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.

Примерная тематика докладов:

1. **Введение в высокоуровневые методы информатики и программирования:** Основные понятия и методы высокоуровневого программирования, преимущества использования высокоуровневых методов по сравнению с низкоуровневыми.
2. **Парадигмы программирования:** Обзор парадигм программирования: процедурное, объектно-ориентированное, функциональное, логическое программирование. Примеры языков программирования для каждой парадигмы.
3. **Объектно-ориентированное программирование (ООП):** Принципы ООП: инкапсуляция, наследование, полиморфизм, абстракция. Реализация ООП на примере языков C++, Java и Python. Примеры проектирования и реализации классов и объектов.
4. **Функциональное программирование:** Основные концепции функционального программирования: чистые функции, неизменяемость данных, функции высшего порядка. Реализация на примере языка Haskell и Scala.
5. **Логическое программирование:** Основы логического программирования. Языки логического программирования, такие как Prolog. Примеры задач, решаемых с помощью логического программирования.
6. **Методы и паттерны программирования:** Обзор ключевых паттернов проектирования: Singleton, Factory, Observer, Decorator, Strategy и другие. Примеры применения паттернов на различных языках программирования.
7. **Генетические алгоритмы и эвристические методы:** Основы генетических алгоритмов и эвристических методов. Применение генетических алгоритмов для решения сложных задач оптимизации. Примеры на языках Python и C++.
8. **Современные техники программирования:** Современные техники программирования: метапрограммирование, аспектно-ориентированное программирование, программирование на основе контрактов. Примеры и реализация на различных языках программирования.
9. **Архитектурные стили и шаблоны:** Основные архитектурные стили: многослойная архитектура, микросервисы, REST, MVC и другие. Примеры их применения в различных проектах.

Цель написания докладов: Углубить понимание и критическое осмысление роли высокоуровневых методов информатики и программирования в разработке программного обеспечения, развивая аналитические и научные навыки студентов.

Структура доклада:

1. **Введение**
 - Краткое описание темы и целей доклада.
 - 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 высокоуровневых методов информатики и программирования 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 методов и парадигм программирования.
 - Анализ современных тенденций.
 - Примеры применения высокоуровневых методов программирования в реальных проектах.
3. **Аналитическая часть (20%)**
 - Глубина анализа роли высокоуровневых методов программирования в процессе разработки ПО.
 - Наличие собственных выводов и оценок.
4. **Оформление (10%)**
 - Соответствие требованиям к оформлению докладов (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 доклада должен составлять 8-12 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 +.

1. Какие преимущества имеет высокоуровневое программирование по сравнению с низкоуровневым? (УК-1.1)

- А) Позволяет работать ближе к аппаратному уровню.
- Б) Обеспечивает большую производительность.
- В) Упрощает разработку и обслуживание программного обеспечения. +
- Г) Уменьшает размер исходного кода.

2. Какая из следующих парадигм программирования является процедурной? (УК-2.1)

- А) Объектно-ориентированная.
- Б) Функциональная.

- В) Логическая.
 - Г) Процедурная. +
- 3. Какой принцип ООП позволяет скрыть детали реализации и показать только функциональность? (ПК-1.1)**
- А) Инкапсуляция. +
 - Б) Наследование.
 - В) Полиморфизм.
 - Г) Абстракция.
- 4. Какой язык программирования используется для функционального программирования? (ПК-1.1)**
- А) Java.
 - Б) C++.
 - В) Haskell. +
 - Г) Prolog.
- 5. Какое утверждение верно для логического программирования? (УК-2.1)**
- А) Программы пишутся в виде набора функций.
 - Б) Программы представляют собой набор фактов и правил. +
 - В) Программы состоят из объектов и классов.
 - Г) Программы организованы в виде процедур.
- 6. Какой паттерн проектирования используется для создания одного экземпляра класса? (ПК-4.2)**
- А) Factory.
 - Б) Singleton. +
 - В) Observer.
 - Г) Strategy.
- 7. В каком языке программирования наиболее распространено использование генетических алгоритмов? (ПК-1.1)**
- А) JavaScript.
 - Б) Python. +
 - В) SQL.
 - Г) HTML.
- 8. Что такое метапрограммирование? (УК-2.2)**
- А) Программирование на основе контрактов.
 - Б) Программирование с использованием объектов и классов.
 - В) Программирование, где программы могут порождать другие программы. +
 - Г) Программирование на основе логических выражений.
- 9. Какой архитектурный стиль характеризуется использованием небольших независимых сервисов? (ПК-4.2)**
- А) MVC.
 - Б) Микросервисы. +
 - В) Многослойная архитектура.
 - Г) REST.
- 10. В чем основное преимущество использования паттернов проектирования? (ПК-4.2)**
- А) Уменьшение объема кода.
 - Б) Повышение производительности программ.
 - В) Повышение гибкости и повторного использования кода. +
 - Г) Упрощение отладки программного обеспечения.
- 11. Что такое аспектно-ориентированное программирование? (ПК-1.1)**
- А) Программирование с использованием аспектов для улучшения модульности. +
 - Б) Программирование, основанное на функциях.
 - В) Программирование, основанное на логике.

- Г) Программирование, основанное на контрактах.

12. Какой язык программирования чаще всего используется для логического программирования? (ПК-1.1)

- А) Java.
- Б) C++.
- В) Haskell.
- Г) Prolog. +

13. Какой принцип ООП позволяет объекту менять свое поведение в зависимости от его состояния? (ПК-1.1)

- А) Инкапсуляция.
- Б) Наследование.
- В) Полиморфизм. +
- Г) Абстракция.

14. Что такое программирование на основе контрактов? (ПК-1.2)

- А) Метод, позволяющий объектам взаимодействовать между собой.
- Б) Метод, определяющий интерфейсы и соглашения для взаимодействия компонентов. +
- В) Метод, основанный на логических выражениях.
- Г) Метод, использующий функции высшего порядка.

15. В чем заключается основная цель использования генетических алгоритмов? (ПК-1.1)

- А) Оптимизация задач с использованием методов эволюции и естественного отбора. +
- Б) Улучшение производительности программного кода.
- В) Разработка программного обеспечения.
- Г) Тестирование программного кода.

16. Какой язык программирования наиболее часто используется для функционального программирования? (ПК-1.1)

- А) Prolog.
- Б) Python.
- В) Scala. +
- Г) JavaScript.

17. Какой паттерн проектирования используется для создания сложных объектов с помощью простых шагов? (ПК-4.2)

- А) Builder. +
- Б) Factory.
- В) Observer.
- Г) Strategy.

18. В чем основное преимущество использования микросервисной архитектуры? (ПК-4.2)

- А) Упрощение разработки.
- Б) Масштабируемость и независимость компонентов. +
- В) Повышение производительности.
- Г) Уменьшение объема кода.

19. Какое свойство функционального программирования обеспечивает неизменяемость данных? (ПК-1.1)

- А) Чистые функции. +
- Б) Полиморфизм.
- В) Инкапсуляция.
- Г) Наследование.

20. Что такое метапрограммирование? (ПК-1.1)

- А) Способность программы порождать другую программу. +
- Б) Использование наследования в программировании.
- В) Применение абстракций в коде.
- Г) Инкапсуляция данных в объектах.

21. Какие методы используются для поиска и критического анализа информации? (УК-1.1, УК-1.2)

- А) Чтение документации.
- Б) Использование специализированных баз данных и научных публикаций. +
- В) Просмотр обучающих видеороликов.
- Г) Чтение блогов.

22. Как системный подход помогает в решении сложных задач программирования? (УК-1.2, УК-2.1)

- А) Разделяет задачу на более мелкие части и решает их поочередно. +
- Б) Использует только одну методологию для решения.
- В) Игнорирует влияние внешних факторов.
- Г) Упрощает задачу до одной небольшой проблемы.

23. Какие из перечисленных методов используются для проверки работоспособности программного кода? (ПК-2.1)

- А) Отладка. +
- Б) Компиляция.
- В) Линковка.
- Г) Обфускация.

24. Как рефакторинг улучшает структуру и читаемость кода? (ПК-2.2)

- А) Делает код длиннее.
- Б) Убирает ненужные комментарии.
- В) Улучшает структуру и делает код более читаемым. +
- Г) Изменяет название переменных.

25. Что включает в себя процесс интеграции программных модулей? (ПК-3.1)

- А) Написание новых функций.
- Б) Сборка и тестирование взаимодействия модулей. +
- В) Удаление лишнего кода.
- Г) Создание документации.

26. Какой метод используется для проверки работоспособности всех компонентов программного продукта перед выпуском? (ПК-3.2)

- А) Юнит-тестирование.
- Б) Интеграционное тестирование.
- В) Системное тестирование. +
- Г) Статическое тестирование.

27. Какой этап разработки ПО включает определение требований к программному обеспечению? (ПК-4.1)

- А) Дизайн.
- Б) Сбор и анализ требований. +
- В) Кодирование.
- Г) Внедрение.

28. Как правовые нормы и ограничения влияют на выбор оптимальных решений в программировании? (УК-2.2)

- А) Они не имеют значения.
- Б) Определяют, какие технологии и методы могут быть использованы. +
- В) Ограничивают только финансовые аспекты.
- Г) Влияют только на этап тестирования.

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 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ

1. Какие основные преимущества высокоуровневого программирования по сравнению с низкоуровневым? (УК-1.1)

- **Правильный ответ:** Высокоуровневое программирование упрощает разработку и обслуживание программного обеспечения, позволяет сосредоточиться на логике приложения, а не на деталях реализации, и ускоряет процесс разработки за счет использования абстракций и библиотек.

2. Назовите основные парадигмы программирования. (УК-2.1)

- **Правильный ответ:** Процедурное, объектно-ориентированное, функциональное и логическое программирование.

3. Что такое инкапсуляция в объектно-ориентированном программировании? (ПК-1.1)

- **Правильный ответ:** Инкапсуляция – это принцип ООП, который позволяет скрыть детали реализации объекта и предоставить доступ только к необходимым методам и данным через публичный интерфейс.

4. Каковы основные принципы функционального программирования? (ПК-1.1)

- **Правильный ответ:** Основные принципы функционального программирования включают использование чистых функций, неизменяемость данных и функции высшего порядка.

5. В чем заключается суть логического программирования? (УК-2.1)

- **Правильный ответ:** Логическое программирование основывается на использовании фактов и правил для решения задач через логические выводы.

6. Какой паттерн проектирования используется для создания единственного экземпляра класса? (ПК-4.2)

- **Правильный ответ:** Паттерн Singleton.

7. Для чего используются генетические алгоритмы? (ПК-1.1)

- **Правильный ответ:** Генетические алгоритмы используются для оптимизации сложных задач путем применения методов эволюции и естественного отбора.

8. Что такое метапрограммирование? (УК-2.2)

- **Правильный ответ:** Метапрограммирование – это метод программирования, при котором программы могут порождать другие программы.
- 9. Каковы основные преимущества микросервисной архитектуры? (ПК-4.2)**
- **Правильный ответ:** Микросервисная архитектура обеспечивает масштабируемость, независимость компонентов и упрощение разработки и развертывания отдельных сервисов.
- 10. В чем заключается суть аспектно-ориентированного программирования? (ПК-1.1)**
- **Правильный ответ:** Аспектно-ориентированное программирование улучшает модульность кода путем разделения кода на аспекты, которые могут быть применены к различным точкам программы.
- 11. Как логическое программирование используется для решения задач? (ПК-1.1)**
- **Правильный ответ:** Логическое программирование решает задачи через формулирование фактов и правил, на основе которых делаются логические выводы.
- 12. Какие принципы ООП обеспечивают полиморфизм? (ПК-1.1)**
- **Правильный ответ:** Полиморфизм в ООП обеспечивается за счет наследования и интерфейсов, которые позволяют объектам разных классов быть обработанными одинаково.
- 13. Что такое программирование на основе контрактов? (ПК-1.2)**
- **Правильный ответ:** Программирование на основе контрактов определяет интерфейсы и соглашения для взаимодействия компонентов, обеспечивая надежность и предсказуемость поведения системы.
- 14. В чем заключается системный подход к решению задач программирования? (УК-1.2)**
- **Правильный ответ:** Системный подход включает разбиение задачи на более мелкие части, анализ взаимосвязей между ними и использование комплексного подхода к решению задачи.
- 15. Как выполняется тестирование программного кода? (ПК-2.1)**
- **Правильный ответ:** Тестирование программного кода выполняется с помощью различных методов, включая юнит-тестирование, интеграционное тестирование и системное тестирование для проверки корректности и работоспособности кода.
- 16. Что такое рефакторинг и для чего он используется? (ПК-2.2)**
- **Правильный ответ:** Рефакторинг – это процесс улучшения структуры кода без изменения его внешнего поведения, используемый для повышения читаемости и поддержки кода.
- 17. Что включает в себя процесс интеграции программных модулей? (ПК-3.1)**
- **Правильный ответ:** Процесс интеграции включает сборку программных модулей и проверку их взаимодействия для обеспечения корректной работы всей системы.
- 18. Как проводится системное тестирование программного продукта? (ПК-3.2)**
- **Правильный ответ:** Системное тестирование включает проверку работоспособности всех компонентов программного продукта в совокупности для выявления возможных ошибок и проблем.
- 19. Какие этапы включает процесс сбора и анализа требований к программному обеспечению? (ПК-4.1)**
- **Правильный ответ:** Процесс включает сбор требований от заинтересованных сторон, их анализ, документирование и согласование для обеспечения полного понимания целей и функций системы.
- 20. Как правовые нормы и ограничения влияют на выбор методов программирования? (УК-2.2)**
- **Правильный ответ:** Правовые нормы и ограничения определяют, какие технологии и методы могут быть использованы, обеспечивая соответствие законодательству и стандартам безопасности.
- 21. Каковы преимущества использования функционального программирования? (ПК-1.1)**

- **Правильный ответ:** Функциональное программирование обеспечивает предсказуемость, уменьшение побочных эффектов и улучшенную тестируемость кода.
22. **Что такое паттерн проектирования Observer? (ПК-4.2)**
- **Правильный ответ:** Паттерн Observer позволяет одному объекту уведомлять другие объекты об изменениях своего состояния, обеспечивая слабую связанность между ними.
23. **Как генетические алгоритмы применяются для оптимизации? (ПК-1.1)**
- **Правильный ответ:** Генетические алгоритмы применяют методы эволюции и естественного отбора для поиска оптимальных решений сложных задач.
24. **Что такое многослойная архитектура и как она используется? (ПК-4.2)**
- **Правильный ответ:** Многослойная архитектура разделяет систему на слои (например, презентационный, логический, и данных), что упрощает разработку, поддержку и масштабирование системы.
25. **Какие инструменты используются для отладки программного кода? (ПК-2.1)**
- **Правильный ответ:** Инструменты для отладки включают отладчики (debuggers), логирование и профайлеры, которые помогают выявлять и исправлять ошибки в коде.
26. **Какие методы используются для анализа и декомпозиции задач в программировании? (УК-1.2)**
- **Правильный ответ:** Методы анализа и декомпозиции включают диаграммы потоков данных, структурный анализ и объектно-ориентированный анализ, которые помогают разбивать задачи на управляемые части и определять их взаимосвязи.

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.

2.3 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 парадигмы программирования с их определениями (УК-2.1, ПК-1.1)

Парадигма	Определение
А - Процедурное	1 - Программирование, при котором программы пишутся в виде последовательности инструкций и процедур.
В - Объектно-ориентированное	2 - Программирование, основанное на использовании объектов и классов.
С - Функциональное	3 - Программирование, где функции являются основными строительными блоками и данные неизменяемы.
Д - Логическое	4 - Программирование, при котором программы пишутся в виде набора фактов и правил для логического вывода.

Правильный ответ: А-1, В-2, С-3, Д-4

Задача 2: Соотнесите принципы ООП с их описаниями (ПК-1.1, ПК-4.2)

Принцип	Описание
А - Инкапсуляция	1 - Скрытие деталей реализации объекта и предоставление доступа только через публичный интерфейс.
В - Наследование	2 - Возможность создавать новые классы на основе существующих.
С - Полиморфизм	3 - Способность объекта менять свое поведение в зависимости от состояния.
Д - Абстракция	4 - Определение общих характеристик для создания иерархии классов.

Правильный ответ: А-1, В-2, С-3, Д-4

Задача 3: Соотнесите паттерны проектирования с их применением (ПК-4.2, ПК-1.1)

Паттерн	Применение
---------	------------

A - Singleton	1 - Гарантирует, что у класса есть только один экземпляр и предоставляет глобальную точку доступа к нему.
B - Factory	2 - Определяет интерфейс для создания объектов, но позволяет подклассам изменять тип создаваемого объекта.
C - Observer	3 - Позволяет одному объекту уведомлять другие объекты об изменениях своего состояния.
D - Builder	4 - Создание объекта шаг за шагом.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 4: Соотнесите методы программирования с их характеристиками (УК-2.2, ПК-1.2)

Метод	Характеристика
A - Метaproграммирование	1 - Программы могут порождать другие программы.
B - Аспектно-ориентированное программирование	2 - Использование аспектов для улучшения модульности кода.
C - Программирование на основе контрактов	3 - Определение интерфейсов и соглашений для взаимодействия компонентов.
D - Логическое программирование	4 - Использование фактов и правил для решения задач через логические выводы.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 5: Соотнесите этапы разработки ПО с их описаниями (ПК-4.1, ПК-3.2)

Этап	Описание
A - Сбор и анализ требований	1 - Определение целей и функций системы путем сбора требований от заинтересованных сторон.
B - Дизайн	2 - Разработка архитектуры и дизайна системы.
C - Кодирование	3 - Написание программного кода на основе спроектированной архитектуры.
D - Системное тестирование	4 - Проверка работоспособности всех компонентов программного продукта в совокупности.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 6: Соотнесите парадигмы программирования с языками, в которых они используются (УК-1.1, ПК-1.1)

Парадигма	Язык
A - Процедурное	1 - C++

B - Объектно-ориентированное	2 - Java
C - Функциональное	3 - Haskell
D - Логическое	4 - Prolog

Правильный ответ: A-1, B-2, C-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 ЭКЗАМЕНА

1. Какие преимущества имеет высокоуровневое программирование по сравнению с низкоуровневым? (УК-1.1)

- **Правильный ответ:** Высокоуровневое программирование упрощает процесс разработки и обслуживания программного обеспечения за счет использования абстракций и библиотек. Оно позволяет сосредоточиться на логике приложения, а не на деталях реализации, что ускоряет разработку и снижает количество ошибок.

2. Опишите основные парадигмы программирования и приведите примеры языков, использующих каждую из них. (УК-2.1, ПК-1.1)

- **Правильный ответ:** Основные парадигмы программирования включают процедурное (C), объектно-ориентированное (Java, C++), функциональное (Haskell, Scala) и логическое (Prolog). Процедурное программирование основывается на последовательности инструкций, объектно-ориентированное использует объекты и классы, функциональное фокусируется на чистых функциях и неизменяемых данных, а логическое использует факты и правила для логических выводов.

3. Объясните принцип инкапсуляции в объектно-ориентированном программировании и его преимущества. (ПК-1.1)

- **Правильный ответ:** Инкапсуляция - это принцип, позволяющий скрыть детали реализации объекта и предоставить доступ к его данным и методам только через публичный интерфейс. Это повышает безопасность кода, улучшает его читаемость и облегчает поддержку.

4. Опишите основные принципы функционального программирования и их преимущества. (ПК-1.1)

- **Правильный ответ:** Основные принципы функционального программирования включают использование чистых функций, неизменяемость данных и функции высшего порядка. Преимущества включают предсказуемость, отсутствие побочных эффектов и улучшенную тестируемость кода.

5. В чем заключается суть логического программирования и в каких задачах оно наиболее эффективно? (УК-2.1, ПК-1.1)

- **Правильный ответ:** Логическое программирование основывается на использовании фактов и правил для решения задач через логические выводы. Оно эффективно в задачах,

требующих сложных логических выводов, таких как экспертные системы и автоматическое доказательство теорем.

6. Опишите паттерн Singleton и его применение в программировании. (ПК-4.2)

- **Правильный ответ:** Паттерн Singleton гарантирует, что у класса есть только один экземпляр, и предоставляет глобальную точку доступа к нему. Он применяется для управления ресурсами, такими как соединения с базой данных или настройками конфигурации.

7. Как работают генетические алгоритмы и для каких задач они используются? (ПК-1.1)

- **Правильный ответ:** Генетические алгоритмы имитируют процесс естественного отбора для решения задач оптимизации. Они используются в задачах, где традиционные методы неэффективны, таких как сложные задачи оптимизации и машинное обучение.

8. Что такое метапрограммирование и как оно применяется в программировании? (УК-2.2, ПК-1.2)

- **Правильный ответ:** Метапрограммирование - это метод программирования, при котором программы могут порождать другие программы. Оно используется для автоматизации задач, генерации кода и создания более гибких и адаптируемых систем.

9. В чем заключаются основные преимущества микросервисной архитектуры? (ПК-4.2)

- **Правильный ответ:** Микросервисная архитектура обеспечивает масштабируемость, независимость компонентов и упрощение разработки и развертывания отдельных сервисов. Это позволяет командам работать параллельно и быстрее внедрять изменения.

10. Объясните принцип аспектно-ориентированного программирования и его применение. (ПК-1.1)

- **Правильный ответ:** Аспектно-ориентированное программирование улучшает модульность кода путем разделения кода на аспекты, которые могут быть применены к различным точкам программы. Это особенно полезно для управления сквозной функциональностью, такой как логирование и обработка ошибок.

11. Как логическое программирование используется для решения задач? (ПК-1.1)

- **Правильный ответ:** Логическое программирование решает задачи через формулирование фактов и правил, на основе которых делаются логические выводы. Это особенно полезно для задач, требующих сложных логических выводов, таких как экспертные системы.

12. Какие принципы ООП обеспечивают полиморфизм и как они работают? (ПК-1.1)

- **Правильный ответ:** Полиморфизм в ООП обеспечивается за счет наследования и интерфейсов, которые позволяют объектам разных классов быть обработанными одинаково. Это позволяет использовать один интерфейс для различных действий, улучшая гибкость и расширяемость кода.

13. Что такое программирование на основе контрактов и как оно улучшает качество кода? (ПК-1.2)

- **Правильный ответ:** Программирование на основе контрактов определяет интерфейсы и соглашения для взаимодействия компонентов, обеспечивая надежность и предсказуемость поведения системы. Это улучшает качество кода за счет явного определения условий и инвариантов.

14. В чем заключается системный подход к решению задач программирования и его преимущества? (УК-1.2)

- **Правильный ответ:** Системный подход включает разбиение задачи на более мелкие части, анализ взаимосвязей между ними и использование комплексного подхода к решению задачи. Это позволяет учитывать все аспекты проблемы и находить оптимальные решения.

15. Какие методы используются для проверки работоспособности программного кода? (ПК-2.1)

- **Правильный ответ:** Методы включают юнит-тестирование, интеграционное тестирование, системное тестирование и автоматизированное тестирование. Они помогают выявить ошибки и обеспечить корректную работу кода.
- 16. Что такое рефакторинг и какие методы используются для улучшения кода? (ПК-2.2)**
- **Правильный ответ:** Рефакторинг – это процесс улучшения структуры кода без изменения его внешнего поведения. Методы включают улучшение имен переменных и функций, разбиение больших функций на более мелкие, устранение дублирования кода и улучшение структуры классов.
- 17. Как проводится интеграция программных модулей и какие проблемы могут возникнуть? (ПК-3.1)**
- **Правильный ответ:** Интеграция включает сборку программных модулей и проверку их взаимодействия для обеспечения корректной работы всей системы. Возможные проблемы включают несовместимость интерфейсов, ошибки в данных и проблемы с производительностью.
- 18. Как проводится системное тестирование и какие аспекты оно охватывает? (ПК-3.2)**
- **Правильный ответ:** Системное тестирование включает проверку работоспособности всех компонентов программного продукта в совокупности. Оно охватывает функциональность, производительность, безопасность и совместимость системы.
- 19. Какие этапы включает процесс сбора и анализа требований к программному обеспечению? (ПК-4.1)**
- **Правильный ответ:** Процесс включает сбор требований от заинтересованных сторон, их анализ, документирование и согласование. Это обеспечивает полное понимание целей и функций системы, необходимых для ее разработки.
- 20. Как правовые нормы и ограничения влияют на выбор методов программирования? (УК-2.2)**
- **Правильный ответ:** Правовые нормы и ограничения определяют, какие технологии и методы могут быть использованы, обеспечивая соответствие законодательству и стандартам безопасности. Это влияет на выбор языков программирования, библиотек и фреймворков.
- 21. Каковы преимущества использования функционального программирования и где оно наиболее эффективно? (ПК-1.1)**
- **Правильный ответ:** Функциональное программирование обеспечивает предсказуемость, уменьшение побочных эффектов и улучшенную тестируемость кода. Оно эффективно для задач, требующих параллельной обработки и математических вычислений.
- 22. Что такое паттерн проектирования Observer и его применение? (ПК-4.2)**
- **Правильный ответ:** Паттерн Observer позволяет одному объекту уведомлять другие объекты об изменениях своего состояния. Он применяется для реализации механизма подписки и уведомлений, например, в графических интерфейсах и системах событий.
- 23. Как генетические алгоритмы применяются для оптимизации задач? (ПК-1.1)**
- **Правильный ответ:** Генетические алгоритмы применяют методы эволюции и естественного отбора для поиска оптимальных решений сложных задач. Они используются в задачах оптимизации, машинного обучения и разработки искусственного интеллекта.
- 24. Что такое многослойная архитектура и как она используется в разработке ПО? (ПК-4.2)**
- **Правильный ответ:** Многослойная архитектура разделяет систему на слои (например, презентационный, логический, и данных, что упрощает разработку, поддержку и масштабирование системы. Это позволяет разделять задачи и ответственность между различными слоями, улучшая модульность и тестируемость системы.

25. Какие инструменты используются для отладки программного кода и как они работают? (ПК-2.1)

- **Правильный ответ:** Инструменты для отладки включают отладчики (debuggers), логирование и профайлеры. Отладчики позволяют пошагово выполнять код, устанавливать точки останова и анализировать значения переменных. Логирование записывает информацию о выполнении программы для последующего анализа. Профайлеры помогают выявлять узкие места в производительности кода.

26. Какие методы используются для анализа и декомпозиции задач в программировании? (УК-1.2)

- **Правильный ответ:** Методы анализа и декомпозиции включают диаграммы потоков данных, структурный анализ и объектно-ориентированный анализ. Эти методы помогают разбивать задачи на управляемые части и определять их взаимосвязи, что позволяет более эффективно решать сложные проблемы.

27. Как правовые нормы и ограничения влияют на выбор методов программирования? (УК-2.2)

- **Правильный ответ:** Правовые нормы и ограничения определяют, какие технологии и методы могут быть использованы, обеспечивая соответствие законодательству и стандартам безопасности. Это влияет на выбор языков программирования, библиотек и фреймворков, а также на способы обработки и хранения данных.

28. Как использование различных парадигм программирования может улучшить разработку программного обеспечения? (ПК-1.1, ПК-4.2)

- **Правильный ответ:** Использование различных парадигм программирования позволяет выбирать наиболее подходящие методы и инструменты для решения конкретных задач, что повышает эффективность и качество разработки. Например, объектно-ориентированное программирование хорошо подходит для моделирования реальных объектов, функциональное программирование упрощает работу с параллельными вычислениями, а логическое программирование эффективно для задач с сложными логическими выводами.

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.