

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ



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

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.О.07 «Основы 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.О.07 Основы 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 «Основы программирования» является изучение основных принципов процедурного, модульного и объектно-ориентированного программирования; обучение правилам и подходам к разработке и отладке программного обеспечения на языке программирования C/C++ при решении задач профессиональной деятельности.

Для этого в рамках дисциплины решаются следующие задачи:

- изучение основных синтаксических конструкций языка программирования C/C++, правил и рекомендаций построения программ на указанном языке;
- изучение современных сред разработки программного обеспечения;
- привитие умений и навыков применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности, их отладки и тестирования работоспособности.

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина является компонентом обязательной части 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, УК-3, ОПК-3, ОПК-5, ОПК-6.

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"> - Источники и методы поиска информации по программированию. - Критерии оценки достоверности и релевантности информации. <p>Уметь:</p> <ul style="list-style-type: none"> - Осуществлять поиск информации в учебниках, научных статьях и интернет-ресурсах по программированию. - Анализировать и критически оценивать найденную информацию. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками использования специализированных поисковых систем и баз данных для поиска информации. - Методами критического анализа и оценки достоверности 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 программировании. - Методы системного анализа и проектирования программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять системный подход для разработки программного обеспечения. - Разрабатывать алгоритмы и программы с использованием системного подхода. <p>Владеть:</p> <ul style="list-style-type: none"> - Методами и инструментами системного анализа для разработки программного обеспечения. - Навыками проектирования программного обеспечения с применением системного подхода.
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 анализа и декомпозиции задач в программировании. - Основные этапы разработки программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Определять задачи, вытекающие из поставленной цели, и разрабатывать план их выполнения. - Структурировать задачи и определять их приоритетность. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками анализа и декомпозиции задач. - Методами планирования и структурирования задач в программировании.

		<p>УК-2.2 Способен выбирать оптимальные способы решения задач, исходя из правовых норм, ресурсов и ограничений</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Правовые нормы и ограничения в области программирования. - Методы оценки ресурсов и ограничений при разработке программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Выбирать оптимальные методы и инструменты для решения задач в программировании. - Обосновывать выбор методов и инструментов с учетом правовых норм, ресурсов и ограничений. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками оценки ресурсов и ограничений при выборе способов решения задач. - Методами обоснования выбора оптимальных решений в программировании.
УК-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	<p>УК-3.1 Способен осуществлять социальное взаимодействие в команде</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы командной работы в разработке программного обеспечения. - Основы делового общения и взаимодействия в команде. <p>Уметь:</p> <ul style="list-style-type: none"> - Эффективно взаимодействовать с членами команды при разработке программного обеспечения. - Принимать участие в обсуждениях и совместных проектах. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками командного взаимодействия в процессе разработки программного обеспечения. - Методами делового общения и взаимодействия в команде.
		<p>УК-3.2 Способен реализовывать свою роль в команде</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Различные роли и функции в команде разработки программного обеспечения. - Методы распределения обязанностей в команде. <p>Уметь:</p> <ul style="list-style-type: none"> - Определять и выполнять свою роль в команде разработки программного обеспечения. - Принимать ответственность за выполнение задач в команде. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками самоуправления и координации в рамках команды разработки. - Методами эффективного выполнения своей роли в команде.

ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.	ОПК-3.1 Способен решать стандартные задачи профессиональной деятельности с использованием информационно-коммуникационных технологий (далее ИКТ)	<p>Знать:</p> <ul style="list-style-type: none"> - Основные информационно-коммуникационные технологии, применяемые в программировании. - Возможности ИКТ для автоматизации процессов разработки программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять ИКТ для решения задач в области программирования. - Автоматизировать рутинные процессы с использованием ИКТ. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками работы с информационно-коммуникационными технологиями. - Методами автоматизации процессов разработки программного обеспечения.
		ОПК-3.2 Способен учитывать основные требования информационной безопасности при решении профессиональных задач	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы и требования информационной безопасности в программировании. - Методы обеспечения безопасности программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять требования информационной безопасности при разработке программного обеспечения. - Обеспечивать защиту данных и информации в программных продуктах. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками обеспечения информационной безопасности в процессе разработки программного обеспечения. - Методами и инструментами защиты данных и информации.
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.	ОПК-5.1 Способен устанавливать программное обеспечение для информационных систем	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы и методы установки программного обеспечения для информационных систем. - Виды и особенности программного обеспечения для информационных систем. <p>Уметь:</p> <ul style="list-style-type: none"> - Выполнять установку и настройку программного обеспечения для информационных систем. - Настраивать программные компоненты для обеспечения их корректной работы. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками инсталляции и конфигурации программного обеспечения. - Методами диагностики и устранения неисправностей при установке ПО.
		ОПК-5.2	Знать:

		Способен инсталлировать аппаратное обеспечение для автоматизированных систем	<ul style="list-style-type: none"> - Основные принципы и методы установки аппаратного обеспечения для автоматизированных систем. - Виды и характеристики аппаратного обеспечения для автоматизированных систем. <p>Уметь:</p> <ul style="list-style-type: none"> - Выполнять установку и настройку аппаратного обеспечения для автоматизированных систем. - Подключать и конфигурировать аппаратные компоненты для их корректной работы. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками инсталляции и настройки аппаратного обеспечения. - Методами диагностики и устранения неисправностей аппаратного обеспечения.
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.	ОПК-6.1 Способен разрабатывать алгоритмы для информационных систем и технологий	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы разработки алгоритмов для информационных систем. - Методы и инструменты алгоритмизации. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать алгоритмы для решения задач в области информационных систем и технологий. - Оптимизировать алгоритмы для повышения их эффективности. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками алгоритмизации и разработки алгоритмов. - Инструментами и методами для оптимизации алгоритмов.
		ОПК-6.2 Способен разрабатывать программы, пригодные для практического применения в области информационных систем и технологий	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы разработки программного обеспечения. - Этапы разработки программного обеспечения и методы тестирования. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать программы для решения практических задач в области информационных систем и технологий. - Тестировать и отлаживать программы для обеспечения их корректной работы. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками программирования и разработки программного обеспечения. - Методами и инструментами тестирования и отладки программ.

5. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

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

Общая трудоемкость учебной дисциплины составляет 6 зачетные единицы, 216 часа, включая все формы контактной и самостоятельной работы обучающихся.

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

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

в том числе:

Лекционные занятия (Лек.) - 26 академических часов,

Практические занятия (Пр.) - 44 академических часов,

Консультации (Конс.) - 2 академических часа.

Самостоятельная работа обучающегося (студента):

Самостоятельная работа (СР) - 99 академический час,

Текущий контроль успеваемости

и промежуточно-заочная аттестация обучающегося (студента):

Часы на контроль - 45 академических часов.

Таблица 2. Объём дисциплины

№ п/п	Раздел/тема дисциплины	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	
			Лек.	Пр.	Консульт	СР	
1	Тема 1. Введение в программирование	1 семестр/1 курс	2	2	-	7	УК-1.1, УК-1.2
2	Тема 2. История развития языков программирования	1 семестр/1 курс	2	2	-	8	УК-2.1
3	Тема 3. Основы алгоритмизации и блок-схемы	1 семестр/1 курс	2	4	-	8	УК-2.2, ОПК-6.1
4	Тема 4. Основы Python	1 семестр/1 курс	2	4	-	8	ОПК-3.1, ОПК-6.2
5	Тема 5. Функции и модули в Python	1 семестр/1 курс	2	2	-	8	ОПК-6.1, ОПК-6.2
6	Тема 6. Основы структур данных	1 семестр/1 курс	2	2	-	8	ОПК-6.1, ОПК-6.2
7	Тема 7. Основы C/C++	1 семестр/1 курс	2	2	-	8	ОПК-3.1, ОПК-6.1, ОПК-5.1, ОПК-5.2
8	Тема 8 Подготовка к зачету	1 семестр/1 курс	2	2	-	8	УК-1.1, УК-1.2, УК-2.1, УК-2.2, ОПК-3.1, ОПК-6.1,

							ОПК-6.2, ОПК-5.1, ОПК-5.2
9	Тема 9 Основы объектно-ориентированного программирования (ООП)	2 семестр/ 1 курс		2	-	4	ОПК-3.1, ОПК-6.2
10	Тема 10 Алгоритмы сортировки и поиска	2 семестр/ 1 курс		2	-	4	ОПК-6.1, ОПК-6.2
11	Тема 11 Разработка графического интерфейса пользователя (GUI)	2 семестр/ 1 курс	2	4	-	4	УК-3.1, УК-3.2
12	Тема 12 Работа с файлами	2 семестр/ 1 курс	2	4	-	4	ОПК-3.1, ОПК-6.2, ОПК-5.1
13	Тема 13 Основы тестирования и обработка ошибок	2 семестр/ 1 курс	2	4	-	4	ОПК-3.2, ОПК-6.2
14	Тема 14 Основы JavaScript	2 семестр/ 1 курс	2	2	-	4	ОПК-3.1, ОПК-6.1
15	Тема 15 Основы разработки веб-приложений	2 семестр/ 1 курс	2	2	-	4	ОПК-3.1, ОПК-6.2, ОПК-5.2
16	Тема 16 Подготовка к экзамену	2 семестр/ 1 курс		4	2	8	УК-1.1, УК-1.2, УК-2.1, УК-2.2, УК-3.1, УК-3.2, ОПК-3.1, ОПК-3.2, ОПК-6.1, ОПК-6.2, ОПК-5.1, ОПК-5.2
ИТОГО аудиторных часов/СР:		1-2 семестры/ 1 курс	72 ак. часов			99 ак. часа	-
Часы на контроль		1-2 семестры/ 1 курс	45 ак. час (форма промежуточной аттестации – зачет – 1 семестр форма промежуточной аттестации – экзамен – 2 семестр)				
ВСЕГО ак. часов:		1-2 семестры/ 1 курс	216 академических часов				

5.2. Тематическое содержание дисциплины

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

1 семестр

Тема 1 - Введение в программирование

Основные понятия программирования. Рассмотрение роли и значения программирования в современном мире. Введение в языки программирования, такие как Python, C/C++ и JavaScript.

Тема 2 - История развития языков программирования

Историческое развитие языков программирования. Основные этапы и ключевые языки: от машинного кода до современных высокоуровневых языков.

Тема 3 - Основы алгоритмизации и блок-схемы

Понятие и свойства алгоритмов. Изучение основных алгоритмических структур: последовательность, ветвление, цикл. Построение блок-схем для алгоритмов.

Тема 4 - Основы Python

Изучение переменных, типов данных и операторов в Python. Рассмотрение основных конструкций: ввод-вывод, условия, циклы. Написание и отладка простых программ.

Тема 5 - Функции и модули в Python

Определение и вызов функций в Python. Изучение параметров и возвращаемых значений. Использование модулей и библиотек.

Тема 6 - Основы структур данных

Изучение основных структур данных: массивы, списки, множества, словари. Операции над структурами данных в Python.

Тема 7 - Основы C/C++

Синтаксис и основные конструкции языков C и C++. Изучение переменных, типов данных и операторов. Условные операторы и циклы. Операции над структурами данных в C++.

Тема 8 - Подготовка к зачету

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

2 семестр

Тема 9 - Основы объектно-ориентированного программирования (ООП)

Основные концепции ООП: классы, объекты, наследование, полиморфизм. Примеры реализации ООП на Python и C++.

Тема 10 - Алгоритмы сортировки и поиска

Изучение основных алгоритмов сортировки: пузырьковая сортировка, быстрая сортировка, сортировка вставками. Алгоритмы поиска: линейный поиск, бинарный поиск. Реализация и сравнение алгоритмов на Python и C++.

Тема 11 - Разработка графического интерфейса пользователя (GUI)

Основы разработки GUI на Python с использованием Tkinter и на C++ с использованием Qt. Изучение элементов интерфейса: кнопки, текстовые поля, метки. Создание простых приложений с графическим интерфейсом.

Тема 12 - Работа с файлами

Операции с файлами на Python и C++: чтение, запись, удаление. Форматы данных: текстовые файлы, CSV, JSON. Обработка и анализ данных из файлов.

Тема 13 - Основы тестирования и обработка ошибок

Основы тестирования программного обеспечения: виды тестирования, написание тестов. Обработка ошибок в Python и C++.

Тема 14 - Основы JavaScript

Изучение основных конструкций языка JavaScript: переменные, операторы, функции. Работа с DOM: манипуляции элементами HTML и CSS. События и обработчики событий.

Тема 15 - Основы разработки веб-приложений

Введение в веб-технологии: HTML, CSS, JavaScript. Основы разработки фронтенда и бэкенда. Простейший веб-сервер на Python (Flask) и JavaScript (Node.js).

Тема 16 - Подготовка к экзамену

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

5.2.1 Содержание практических занятий

Таблица 3

№ п/п	Наименование темы (раздела) дисциплины	Содержание практического занятия
1 семестр		
1	Практическое занятие 1: Введение в программирование	Задание: Изучение среды разработки и написание первой программы на Python. Знакомство с синтаксисом и базовыми конструкциями языка. Цель: Ознакомиться с основами программирования и научиться писать простые программы на Python.
2	Практическое занятие 2: История развития языков программирования	Задание: Исследование истории развития различных языков программирования. Подготовка презентации о выбранном языке программирования. Цель: Понять эволюцию языков программирования и их влияние на современную информатику.
3	Практическое занятие 3: Основы алгоритмизации и блок-схемы	Задание: Создание блок-схем для различных алгоритмов. Решение задач с использованием последовательностей, ветвлений и циклов. Цель: Научиться создавать блок-схемы и разрабатывать алгоритмы.
4	Практическое занятие 4: Основы Python	Задание: Написание программ с использованием переменных, операторов и условий. Выполнение задач с вводом и выводом данных. Цель: Понять основы программирования на Python и научиться использовать базовые конструкции языка.
5	Практическое занятие 5: Функции и модули в Python	Задание: Создание функций и модулей в Python. Написание программ с использованием функций и модулей для структурирования кода. Цель: Научиться создавать и использовать функции и модули в Python.
6	Практическое занятие 6: Основы структур данных	Задание: Изучение и реализация основных структур данных: массивы, списки, множества и словари. Написание программ с использованием этих структур данных. Цель: Понять основы структур данных и научиться их использовать в программах.
7	Практическое занятие 7: Основы C/C++	Задание: Написание простых программ на C/C++. Изучение синтаксиса, переменных, операторов и циклов. Операции над структурами данных в C++. Цель: Ознакомиться с основами программирования на C/C++ и научиться писать простые программы.
8	Практическое занятие 8: Подготовка к зачету	Задание: Решение типовых задач и контрольных вопросов для подготовки к зачету. Цель: Повторить и обобщить пройденный материал для успешной сдачи зачета.
2 семестр		
9	Практическое занятие 9: Основы объектно-ориентированного программирования (ООП)	Задание: Создание классов и объектов в Python и C++. Реализация наследования и полиморфизма. Написание программ с использованием ООП. Цель: Понять основные концепции ООП и научиться их применять на практике.
10	Практическое занятие 10: Алгоритмы сортировки и поиска	Задание: Реализация различных алгоритмов сортировки и поиска на Python и C++. Сравнение эффективности различных алгоритмов. Цель: Изучить основные алгоритмы сортировки и поиска и научиться их

		реализовывать.
11	Практическое занятие 11: Разработка графического интерфейса пользователя (GUI)	Задание: Создание простых GUI-приложений с использованием Tkinter для Python и Qt для C++. Изучение основных элементов интерфейса. Цель: Научиться разрабатывать приложения с графическим интерфейсом.
12	Практическое занятие 12: Работа с файлами	Задание: Чтение, запись и удаление файлов на Python и C++. Работа с текстовыми файлами, CSV и JSON. Обработка и анализ данных из файлов. Цель: Понять основы работы с файлами и научиться выполнять операции с различными форматами данных.
13	Практическое занятие 13: Основы тестирования и обработка ошибок	Задание: Написание тестов для программ на Python и C++. Обработка ошибок и исключений. Улучшение надежности программного обеспечения. Цель: Изучить методы тестирования программ и научиться обрабатывать ошибки.
14	Практическое занятие 14: Основы JavaScript	Задание: Написание простых программ на JavaScript. Работа с DOM для манипуляций элементами HTML и CSS. Изучение событий и обработчиков событий. Цель: Ознакомиться с основами программирования на JavaScript и научиться работать с веб-страницами.
15	Практическое занятие 15: Основы разработки веб-приложений	Задание: Создание простых веб-приложений с использованием HTML, CSS, JavaScript, Flask и Node.js. Разработка фронтенда и бэкенда. Цель: Научиться разрабатывать простые веб-приложения и работать с веб-технологиями.
16	Практическое занятие 16: Подготовка к экзамену	Задание: Решение типовых задач и контрольных вопросов для подготовки к экзамену. Цель: Повторить и обобщить пройденный материал для успешной сдачи экзамена.

5.2.2 Содержание самостоятельной работы

Таблица 4

№ п/п	Наименование темы (раздела) дисциплины	Содержание самостоятельной работы	Форма контроля
1 семестр			
1	Самостоятельная работа 1: Введение в программирование	Задание: Изучить основы синтаксиса Python. Написать несколько простых программ, решающих базовые задачи.	Выполнение практического задания.
2	Самостоятельная работа 2: История развития языков программирования	Задание: Изучить историю развития языков программирования. Подготовить реферат на выбранную тему.	Подготовка реферата.
3	Самостоятельная работа 3: Основы алгоритмизации и блок-схемы	Задание: Разработать блок-схемы для нескольких простых алгоритмов. Изучить материалы по основам алгоритмизации.	Выполнение практического задания.
4	Самостоятельная работа 4: Основы Python	Задание: Написать программы с использованием переменных, операторов и условий. Выполнить задачи с вводом и выводом данных.	Выполнение практического задания.
5	Самостоятельная работа 5: Функции и модули в Python	Задание: Создать и использовать функции и модули в Python. Написать программы для структурирования кода.	Выполнение практического задания.
6	Самостоятельная работа 6: Основы структур данных	Задание: Изучить основные структуры данных: массивы, списки, множества и словари. Написать программы с использованием этих структур данных.	Выполнение практического задания.
7	Самостоятельная работа 7: Основы C/C++	Задание: Написать простые программы на C/C++. Изучить синтаксис, переменные, операторы и циклы. Операции над структурами данных в C++.	Выполнение практического задания.
8	Самостоятельная работа 8: Подготовка к зачету	Задание: Решение типовых задач и контрольных вопросов для подготовки к зачету.	Тест.
2 семестр			
9	Самостоятельная работа 9: Основы объектно-ориентированного	Задание: Изучить концепции ООП: классы, объекты, наследование, полиморфизм. Написать программы с использованием ООП на Python и C++.	Выполнение практического задания.

	программирования (ООП)		
10	Самостоятельная работа 10: Алгоритмы сортировки и поиска	Задание: Реализовать различные алгоритмы сортировки и поиска на Python и C++. Сравнить эффективность различных алгоритмов.	Выполнение практического задания.
11	Самостоятельная работа 11: Разработка графического интерфейса пользователя (GUI)	Задание: Создать простые GUI-приложения с использованием Tkinter для Python и Qt для C++. Изучить основные элементы интерфейса.	Выполнение практического задания.
12	Самостоятельная работа 12: Работа с файлами	Задание: Выполнить операции с файлами на Python и C++: чтение, запись, удаление. Работа с текстовыми файлами, CSV и JSON.	Выполнение практического задания.
13	Самостоятельная работа 13: Основы тестирования и обработка ошибок	Задание: Написать тесты для программ на Python и C++. Обработать ошибки и исключения в программах.	Выполнение практического задания.
14	Самостоятельная работа 14: Основы JavaScript	Задание: Написать простые программы на JavaScript. Работать с DOM для манипуляции элементами HTML и CSS. Изучить события и обработчики событий.	Выполнение практического задания.
15	Самостоятельная работа 15: Основы разработки веб-приложений	Задание: Создать простые веб-приложения с использованием HTML, CSS, JavaScript, Flask и Node.js. Разработать фронтенд и бэкенд.	Выполнение практического задания.
16	Самостоятельная работа 16: Подготовка к экзамену	Задание: Решение типовых задач и контрольных вопросов для подготовки к экзамену. Повторение пройденного материала.	Экзамен

6. Оценочные материалы по дисциплине

Оценочные материалы находятся в документе «Оценочные материалы по дисциплине «Основы 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 приобретает 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. Иванова Г.С., Программирование [Электронный ресурс] : учебник / Г.С. Иванова. - М. : КноРус, 2022. - 426 с. - ISBN 978-5-406-09829-5. - Режим доступа: <https://book.ru/book/943869>
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>

Дополнительная литература:

1. Гордиенко А.П., Функциональное программирование [Электронный ресурс] : учебник / А.П. Гордиенко. - М. : КноРус, 2022. - 277 с. - ISBN 978-5-406-08432-8. - Режим доступа: <https://book.ru/book/942660>

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> <ul style="list-style-type: none"> - Столы для обучающихся; 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>

<ul style="list-style-type: none"> - 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.О.07 «Основы 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, УК-3, ОПК-3, ОПК-5, ОПК-6.

Код компетенции	Наименование компетенции	Индекс и наименование индикатора содержания компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Способен осуществлять поиск и критический анализ информации	Знать: <ul style="list-style-type: none"> - Источники и методы поиска информации по программированию. - Критерии оценки достоверности и релевантности информации. Уметь: <ul style="list-style-type: none"> - Осуществлять поиск информации в учебниках, научных статьях и интернет-ресурсах по программированию. - Анализировать и критически оценивать найденную информацию. Владеть: <ul style="list-style-type: none"> - Навыками использования специализированных поисковых систем и баз данных для поиска информации. - Методами критического анализа и оценки достоверности информации.
		УК-1.2 Способен применять системный подход для решения поставленных задач	Знать: <ul style="list-style-type: none"> - Принципы системного подхода в программировании. - Методы системного анализа и проектирования программного обеспечения. Уметь: <ul style="list-style-type: none"> - Применять системный подход для разработки программного обеспечения. - Разрабатывать алгоритмы и программы с использованием системного подхода. Владеть: <ul style="list-style-type: none"> - Методами и инструментами системного анализа для разработки программного обеспечения. - Навыками проектирования программного обеспечения с применением системного подхода.
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих	УК-2.1 Способен определять круг задач в рамках поставленной цели	Знать: <ul style="list-style-type: none"> - Методы анализа и декомпозиции задач в программировании. - Основные этапы разработки программного обеспечения. Уметь: <ul style="list-style-type: none"> - Определять задачи, вытекающие из поставленной цели, и разрабатывать план их выполнения. - Структурировать задачи и определять их приоритетность.

	правовых норм, имеющих ресурсы и ограничений		<p>Владеть:</p> <ul style="list-style-type: none"> - Навыками анализа и декомпозиции задач. - Методами планирования и структурирования задач в программировании.
		<p>УК-2.2 Способен выбирать оптимальные способы решения задач, исходя из правовых норм, ресурсов и ограничений</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Правовые нормы и ограничения в области программирования. - Методы оценки ресурсов и ограничений при разработке программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Выбирать оптимальные методы и инструменты для решения задач в программировании. - Обосновывать выбор методов и инструментов с учетом правовых норм, ресурсов и ограничений. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками оценки ресурсов и ограничений при выборе способов решения задач. - Методами обоснования выбора оптимальных решений в программировании.
УК-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	<p>УК-3.1 Способен осуществлять социальное взаимодействие в команде</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы командной работы в разработке программного обеспечения. - Основы делового общения и взаимодействия в команде. <p>Уметь:</p> <ul style="list-style-type: none"> - Эффективно взаимодействовать с членами команды при разработке программного обеспечения. - Принимать участие в обсуждениях и совместных проектах. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками командного взаимодействия в процессе разработки программного обеспечения. - Методами делового общения и взаимодействия в команде.
		<p>УК-3.2 Способен реализовывать свою роль в команде</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Различные роли и функции в команде разработки программного обеспечения. - Методы распределения обязанностей в команде. <p>Уметь:</p> <ul style="list-style-type: none"> - Определять и выполнять свою роль в команде разработки программного обеспечения. - Принимать ответственность за выполнение задач в команде. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками самоуправления и координации в рамках команды разработки. - Методами эффективного выполнения своей роли в команде.

ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.	ОПК-3.1 Способен решать стандартные задачи профессиональной деятельности с использованием информационно-коммуникационных технологий (далее ИКТ)	<p>Знать:</p> <ul style="list-style-type: none"> - Основные информационно-коммуникационные технологии, применяемые в программировании. - Возможности ИКТ для автоматизации процессов разработки программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять ИКТ для решения задач в области программирования. - Автоматизировать рутинные процессы с использованием ИКТ. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками работы с информационно-коммуникационными технологиями. - Методами автоматизации процессов разработки программного обеспечения.
		ОПК-3.2 Способен учитывать основные требования информационной безопасности при решении профессиональных задач	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы и требования информационной безопасности в программировании. - Методы обеспечения безопасности программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Применять требования информационной безопасности при разработке программного обеспечения. - Обеспечивать защиту данных и информации в программных продуктах. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками обеспечения информационной безопасности в процессе разработки программного обеспечения. - Методами и инструментами защиты данных и информации.
ОПК-5	Способен установить программное и аппаратное обеспечение для информационных и автоматизированных систем.	ОПК-5.1 Способен установить программное обеспечение для информационных систем	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы и методы установки программного обеспечения для информационных систем. - Виды и особенности программного обеспечения для информационных систем. <p>Уметь:</p> <ul style="list-style-type: none"> - Выполнять установку и настройку программного обеспечения для информационных систем. - Настраивать программные компоненты для обеспечения их корректной работы. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками инсталляции и конфигурации программного обеспечения. - Методами диагностики и устранения неисправностей при установке ПО.
		ОПК-5.2	Знать:

		Способен инсталлировать аппаратное обеспечение для автоматизированных систем	<ul style="list-style-type: none"> - Основные принципы и методы установки аппаратного обеспечения для автоматизированных систем. - Виды и характеристики аппаратного обеспечения для автоматизированных систем. <p>Уметь:</p> <ul style="list-style-type: none"> - Выполнять установку и настройку аппаратного обеспечения для автоматизированных систем. - Подключать и конфигурировать аппаратные компоненты для их корректной работы. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками инсталляции и настройки аппаратного обеспечения. - Методами диагностики и устранения неисправностей аппаратного обеспечения.
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.	ОПК-6.1 Способен разрабатывать алгоритмы для информационных систем и технологий	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы разработки алгоритмов для информационных систем. - Методы и инструменты алгоритмизации. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать алгоритмы для решения задач в области информационных систем и технологий. - Оптимизировать алгоритмы для повышения их эффективности. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками алгоритмизации и разработки алгоритмов. - Инструментами и методами для оптимизации алгоритмов.
		ОПК-6.2 Способен разрабатывать программы, пригодные для практического применения в области информационных систем и технологий	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы разработки программного обеспечения. - Этапы разработки программного обеспечения и методы тестирования. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать программы для решения практических задач в области информационных систем и технологий. - Тестировать и отлаживать программы для обеспечения их корректной работы. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками программирования и разработки программного обеспечения. - Методами и инструментами тестирования и отладки программ.

КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

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

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

		(вопроса); недостаточное полное владение литературой, рекомендованной учебной программой дисциплины
	Владеть	Обучающийся понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы при устранении неточностей и ошибок в решениях в ходе защиты задания (проекта, портфолио) при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

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

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

КРИТЕРИИ И ПРОЦЕДУРЫ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ

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

- тестирование.

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

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

Критерии оценивания результата обучения по дисциплине (модулю)

Результат обучения по дисциплине (модулю)	ШКАЛА ОЦЕНИВАНИЯ				Процедуры оценивания
	«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»	
<u>УК-1</u> <u>УК-2</u> <u>УК-3</u> <u>ОПК-3</u> <u>ОПК-5</u> <u>ОПК-6</u> Знать:	Обучаемый 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	Обучаемый 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	Обучаемый 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	Обучаемый продемонстрировал: неправильные ответы на основные вопросы; грубые ошибки в ответах; непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы; не владеет основной литературой, рекомендованной учебной программой дисциплины.	Тестовые задания

	программой дисциплины.	замечаний по отдельным вопросам; достаточное владение литературой.	литературой, рекомендованной учебной программой дисциплины.		
<u>УК-1</u> <u>УК-2</u> <u>УК-3</u> <u>ОПК-3</u> <u>ОПК-5</u> <u>ОПК-6</u> Уметь:	Обучаемый продемонстрировал: понимание учебного материала, содержательные, полные, правильные и конкретные ответы на все поставленные вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание учебного материала; логически последовательные и правильные ответы на основные задания/вопросы, включая дополнительные; устранение замечаний по отдельным элементам задания; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание основного учебного материала; правильные, без грубых ошибок, ответы на основные вопросы, включая дополнительные, при устранении, при наводящих вопросах преподавателя, замечаний по отдельным элементам задания; недостаточное полное владение литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: непонимание основного учебного материала; не дал правильные ответы на основные вопросы, включая дополнительные; не устранил, при наводящих вопросах преподавателя, замечания и грубые ошибки по вопросу; не владеет основной учебной литературой, рекомендованной учебной программой дисциплины	Вопросы Практические задания
<u>УК-1</u> <u>УК-2</u> <u>УК-3</u> <u>ОПК-3</u> <u>ОПК-5</u> <u>ОПК-6</u> Владеть:	Обучаемый продемонстрировал: понимание учебного материала; правильные и конкретные ответы, включая уточняющие вопросы; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание учебного материала; продемонстрировал логически последовательные, достаточно полные, верные ответы; самостоятельно устранил замечания по отдельным элементам; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы, в том числе при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: непонимание основного учебного материала; дал неправильные ответы на поставленные вопросы; не владеет основной учебной литературой, рекомендованной учебной программой дисциплины	Вопросы Практические задания

1. Оценочные материалы для самостоятельной работы обучающихся (студентов)

1.1 Реферат

1 семестр

Реферат позволит студентам углубиться в основные концепции и методы программирования, а также понять их важность и применение в современных информационных системах. Реферат покрывает компетенции УК-1.1, УК-1.2, УК-2.1, УК-2.2, УК-3.1, УК-3.2, ОПК-3.1, ОПК-3.2, ОПК-5.1, ОПК-5.2, ОПК-6.1, ОПК-6.2.

Примерная тематика рефератов:

1. **История и развитие языков программирования:** Проанализируйте историческое развитие языков программирования, основные этапы и ключевые языки от машинного кода до современных высокоуровневых языков.
2. **Основы алгоритмизации и блок-схемы:** Рассмотрите понятие и свойства алгоритмов, основные алгоритмические структуры: последовательность, ветвление, цикл. Построение блок-схем для алгоритмов.
3. **Изучение языка программирования Python:** Изучите переменные, типы данных и операторы в Python. Рассмотрите основные конструкции: ввод-вывод, условия, циклы. Написание и отладка простых программ.
4. **Функции и модули в Python:** Определение и вызов функций в Python, изучение параметров и возвращаемых значений, использование модулей и библиотек.
5. **Основы структур данных:** Изучение основных структур данных: массивы, списки, множества, словари. Операции над структурами данных в Python.
6. **Изучение языка программирования C/C++:** Синтаксис и основные конструкции языков C и C++. Изучение переменных, типов данных и операторов, условные операторы и циклы, операции над структурами данных.

1.2 Реферат

2 семестр

Реферат позволит студентам углубиться в основные концепции и методы программирования, а также понять их важность и применение в современных информационных системах. Реферат покрывает компетенции УК-1.1, УК-1.2, УК-2.1, УК-2.2, УК-3.1, УК-3.2, ОПК-3.1, ОПК-3.2, ОПК-5.1, ОПК-5.2, ОПК-6.1, ОПК-6.2.

Примерная тематика рефератов:

1. **Основы объектно-ориентированного программирования (ООП):** Основные концепции ООП: классы, объекты, наследование, полиморфизм. Примеры реализации ООП на Python и C++.
2. **Алгоритмы сортировки и поиска:** Изучение основных алгоритмов сортировки: пузырьковая сортировка, быстрая сортировка, сортировка вставками. Алгоритмы поиска: линейный поиск, бинарный поиск.
3. **Разработка графического интерфейса пользователя (GUI):** Основы разработки GUI на Python с использованием Tkinter и на C++ с использованием Qt. Изучение элементов интерфейса: кнопки, текстовые поля, метки.
4. **Работа с файлами:** Операции с файлами на Python и C++: чтение, запись, удаление. Форматы данных: текстовые файлы, CSV, JSON. Обработка и анализ данных из файлов.
5. **Основы тестирования и обработка ошибок:** Основы тестирования программного обеспечения: виды тестирования, написание тестов. Обработка ошибок в Python и C++.
6. **Основы JavaScript и разработка веб-приложений:** Изучение основных конструкций языка JavaScript: переменные, операторы, функции. Работа с DOM: манипуляции элементами HTML и CSS. Основы разработки фронтенда и бэкенда.

Цель написания рефератов: Углубить понимание и критическое осмысление роли программирования в современных информационных системах, развивая аналитические и научные навыки студентов.

Структура реферата:

1. **Введение**
 - Краткое описание темы и целей реферата.
 - Актуальность темы.
2. **Основная часть**
 - Теоретические основы темы.
 - История и эволюция (если применимо).
 - Применение в современной экономике/бизнесе.
 - Примеры и кейсы.
 - Проблемы и вызовы.
 - Перспективы и тенденции развития.
3. **Заключение**
 - Выводы по результатам исследования.
 - Значение 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 этапов развития программирования.
 - Анализ современных тенденций.
 - Примеры применения программирования в информационных системах.
3. **Аналитическая часть (20%)**
 - Глубина анализа роли программирования в информационных системах.
 - Наличие собственных выводов и оценок.
4. **Оформление (10%)**
 - Соответствие требованиям к оформлению рефератов (шрифт, отступы, заголовки и т.д.).
 - Корректное оформление ссылок и списка литературы.
5. **Язык и стиль (10%)**
 - Грамотность и точность изложения.
 - Научный стиль текста.

Требования к объему:

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

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

2.1 Тестовые задания для текущего контроля успеваемости в виде ЗНАНИЙ

В тестовом задании вопросы, которые имеют закрытый характер.

Правильные ответы выделены знаком +.

1 семестр

1. Какой язык программирования является интерпретируемым? (УК-1.1)
 - C++.
 - Python. +
 - Java.
 - Assembly.
2. Какая функция в Python используется для вывода информации на экран? (УК-1.1)
 - print(). +
 - display().
 - echo().
 - output().
3. Какой тип данных используется для хранения целых чисел в Python? (УК-1.2)
 - float.
 - int. +
 - str.
 - bool.
4. Какой оператор используется для проверки условий в Python? (УК-1.2)
 - if. +
 - for.
 - while.
 - def.
5. Какой символ используется для комментариев в Python? (УК-1.2)
 - //
 - /*
 - # +
 - <!--
6. Как называется метод, который позволяет объединить два списка в Python? (ОПК-6.1)
 - append().
 - extend(). +
 - insert().
 - add().
7. Какой из следующих методов используется для добавления элемента в конец списка в Python? (ОПК-6.1)
 - append(). +
 - insert().
 - extend().
 - pop().
8. Что означает аббревиатура ООП? (ОПК-6.1)
 - Объектно-ориентированное программирование. +
 - Основы операторного программирования.
 - Общие оптимизационные принципы.
 - Операторная обработка данных.
9. Какой из следующих языков является объектно-ориентированным? (ОПК-6.2)
 - Python. +
 - HTML.
 - CSS.
 - SQL.

10. Какой принцип ООП позволяет создавать новый класс на основе существующего? (ОПК-6.2)
- Инкапсуляция.
 - Наследование. +
 - Полиморфизм.
 - Абстракция.
11. Какой из следующих алгоритмов является алгоритмом сортировки? (УК-1.1)
- Линейный поиск.
 - Быстрая сортировка. +
 - Двоичный поиск.
 - Хеширование.

Критерии оценки результатов теста

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

2.2 Тестовые задания для текущего контроля успеваемости в виде ЗНАНИЙ

В тестовом задании вопросы, которые имеют закрытый характер.

Правильные ответы выделены знаком +.

2 семестр

1. Какая библиотека используется для создания GUI-приложений на Python? (ОПК-5.1)
 - NumPy.
 - Tkinter. +
 - Pandas.
 - Matplotlib.
2. Как называется метод для открытия файла в Python? (ОПК-5.2)
 - read().
 - write().
 - open(). +

- close().
- 3. Какой режим открытия файла используется для записи данных в Python? (ОПК-5.2)
 - 'r'.
 - 'w'. +
 - 'a'.
 - 'b'.
- 4. Как называется метод для чтения данных из файла в Python? (ОПК-5.2)
 - read(). +
 - write().
 - open().
 - close().
- 5. Какой из следующих операторов используется для обработки исключений в Python? (ОПК-5.2)
 - if.
 - while.
 - try. +
 - for.
- 6. Какой метод используется для написания тестов в Python? (ОПК-3.1)
 - assert(). +
 - test().
 - verify().
 - check().
- 7. Как называется метод для выполнения тестов в Python? (ОПК-3.1)
 - run().
 - execute().
 - test(). +
 - check().
- 8. Какой язык программирования используется для работы с DOM? (УК-3.2)
 - HTML.
 - CSS.
 - JavaScript. +
 - Python.
- 9. Что означает DOM? (УК-3.2)
 - Data Object Model.
 - Document Object Model. +
 - Dynamic Object Model.
 - Design Object Model.
- 10. Какой из следующих фреймворков используется для разработки веб-приложений на Python? (ОПК-6.1)
 - React.
 - Angular.
 - Flask. +
 - Vue.js.
- 11. Какой из следующих фреймворков используется для разработки серверных приложений на JavaScript? (ОПК-6.2)
 - Flask.
 - Node.js. +
 - Django.
 - Ruby on Rails.

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.

2.3 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ 1 семестр

1. Что такое язык программирования? (УК-1.1)
 - **Правильный ответ:** Язык программирования — это формальный язык, предназначенный для написания компьютерных программ, которые определяют поведение вычислительной системы.
2. Какова роль Python в программировании? (УК-1.1)
 - **Правильный ответ:** Python является высокоуровневым интерпретируемым языком программирования, широко используемым для веб-разработки, анализа данных, автоматизации и научных исследований благодаря его простоте и читаемости.
3. Какие типы данных поддерживаются в Python? (УК-1.2)
 - **Правильный ответ:** Python поддерживает такие типы данных, как целые числа (int), числа с плавающей запятой (float), строки (str), логические значения (bool), списки (list), кортежи (tuple), множества (set) и словари (dict).
4. Что такое условный оператор и как он используется в Python? (УК-1.2)
 - **Правильный ответ:** Условный оператор (if) используется для выполнения блока кода, если заданное условие истинно. Он позволяет реализовать ветвление в программе.
5. Какой символ используется для комментариев в Python? (УК-1.2)
 - **Правильный ответ:** Для комментариев в Python используется символ #.
6. Что такое список в Python и как к нему обращаться? (ОПК-6.1)
 - **Правильный ответ:** Список (list) в Python — это упорядоченная изменяемая коллекция элементов. К элементам списка можно обращаться по их индексам, начиная с нуля.
7. Как добавить элемент в конец списка в Python? (ОПК-6.1)
 - **Правильный ответ:** Для добавления элемента в конец списка в Python используется метод append().

8. Что такое объектно-ориентированное программирование (ООП)? (ОПК-6.1)
 - **Правильный ответ:** Объектно-ориентированное программирование (ООП) — это парадигма программирования, основанная на концепции объектов, которые являются экземплярами классов и могут содержать данные и методы.
9. Какой принцип ООП позволяет одному методу работать с разными типами данных? (ОПК-6.2)
 - **Правильный ответ:** Полиморфизм — это принцип ООП, позволяющий одному методу работать с разными типами данных и обеспечивающий возможность использования единого интерфейса для объектов различных классов.
10. Что такое быстрая сортировка и как она работает? (УК-1.1)
 - **Правильный ответ:** Быстрая сортировка (QuickSort) — это алгоритм сортировки, который выбирает опорный элемент и разделяет массив на две части: элементы меньше опорного и элементы больше опорного, затем рекурсивно сортирует обе части таким же образом.
11. Какой алгоритм поиска является наиболее эффективным для отсортированного массива? (УК-1.1)
 - **Правильный ответ:** Двоичный поиск (Binary Search) является наиболее эффективным для отсортированного массива, так как его временная сложность составляет $O(\log n)$.

2.4 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ

2 семестр

1. Как создать окно в Tkinter? (ОПК-5.1)
 - **Правильный ответ:** Для создания окна в Tkinter используется класс Tk(), например, создается объект root класса Tk.
2. Как добавить кнопку в окно Tkinter? (ОПК-5.1)
 - **Правильный ответ:** Для добавления кнопки в окно Tkinter используется класс Button(), создается объект кнопки и указывается ее родительский элемент и текст.
3. Как открыть файл для чтения в Python? (ОПК-5.2)
 - **Правильный ответ:** Для открытия файла для чтения в Python используется функция open() с режимом 'r'.
4. Как записать данные в файл в Python? (ОПК-5.2)
 - **Правильный ответ:** Для записи данных в файл в Python используется метод write() объекта файла.
5. Как прочитать все строки из файла в Python? (ОПК-5.2)
 - **Правильный ответ:** Для чтения всех строк из файла в Python используется метод readlines() объекта файла.
6. Как обрабатывать исключения в Python? (ОПК-5.2)
 - **Правильный ответ:** Для обработки исключений в Python используется конструкция try-except, где в блоке try размещается код, который может вызвать исключение, а в блоке except — код обработки исключения.
7. Что такое модуль unittest в Python? (ОПК-3.1)
 - **Правильный ответ:** Модуль unittest в Python используется для написания и выполнения тестов. Он предоставляет классы и методы для создания тестовых случаев, тестовых наборов и запуска тестов.
8. Какую роль выполняет JavaScript в веб-разработке? (УК-3.2)
 - **Правильный ответ:** JavaScript используется для создания динамических и интерактивных веб-страниц, позволяет манипулировать DOM и обрабатывать события на клиентской стороне.
9. Что такое DOM? (УК-3.2)

- **Правильный ответ:** DOM (Document Object Model) — это объектная модель документа, представляющая структуру HTML или XML документа в виде дерева узлов, определяющих структуру документа.
- 10. Какой фреймворк используется для создания веб-приложений на Python? (ОПК-6.1)
 - **Правильный ответ:** Flask — это популярный фреймворк для создания веб-приложений на Python, который обеспечивает простоту и гибкость в разработке веб-сервисов.
- 11. Какой фреймворк используется для создания серверных приложений на JavaScript? (ОПК-6.2)
 - **Правильный ответ:** Node.js — это фреймворк, позволяющий создавать серверные приложения на JavaScript, обеспечивая высокую производительность и масштабируемость.

Критерии оценки ответов на вопросы

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

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

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

Задача 1. Соотнесите языки программирования с их описаниями (УК-1.1)

Чтобы определить правильное соответствие, необходимо понимать основные языки программирования и их описания.

Понятие	Определение
---------	-------------

A - Python	1 - Высокоуровневый интерпретируемый язык, широко используемый для веб-разработки и анализа данных
B - C++	2 - Компилируемый язык, использующийся для системного и прикладного программирования
C - JavaScript	3 - Язык сценариев, используемый для создания интерактивных веб-страниц
D - Java	4 - Объектно-ориентированный язык программирования, часто используемый для разработки корпоративных приложений

Правильный ответ: A-1, B-2, C-3, D-4

Задача 2. Соотнесите типы данных Python с их примерами (УК-1.2)

Чтобы определить правильное соответствие, необходимо понимать типы данных Python и их примеры.

Понятие	Определение
A - int	1 - 42
B - str	2 - "Привет, мир!"
C - bool	3 - True/False
D - float	4 - 3.14

Правильный ответ: A-1, B-2, C-3, D-4

Задача 3. Соотнесите конструкции Python с их описанием (ОПК-6.1)

Чтобы определить правильное соответствие, необходимо понимать конструкции Python и их описания.

Понятие	Определение
A - if	1 - Условный оператор
B - for	2 - Цикл с известным числом итераций
C - while	3 - Цикл с условием
D - def	4 - Определение функции

Правильный ответ: A-1, B-2, C-3, D-4

Задача 4. Соотнесите основные концепции ООП с их определениями (ОПК-6.1)

Чтобы определить правильное соответствие, необходимо понимать основные концепции ООП и их определения.

Понятие	Определение
A - Инкапсуляция	1 - Скрытие данных и методов внутри объекта
B - Наследование	2 - Создание нового класса на основе существующего

С - Полиморфизм	3 - Возможность использовать единый интерфейс для объектов различных классов
D - Абстракция	4 - Представление сущности через ее ключевые характеристики

Правильный ответ: A-1, B-2, C-3, D-4

Задача 5. Соотнесите методы работы с файлами в Python с их функциями (ОПК-5.2)

Чтобы определить правильное соответствие, необходимо понимать методы работы с файлами в Python и их функции.

Понятие	Определение
A - open()	1 - Открытие файла для чтения или записи
B - read()	2 - Чтение данных из файла
C - write()	3 - Запись данных в файл
D - close()	4 - Закрытие файла

Правильный ответ: A-1, B-2, C-3, D-4

Задача 6. Соотнесите термины тестирования в программировании с их описанием (ОПК-3.1)

Чтобы определить правильное соответствие, необходимо понимать термины тестирования в программировании и их описание.

Понятие	Определение
A - Юнит-тестирование	1 - Тестирование отдельных компонентов программы
B - Интеграционное тестирование	2 - Тестирование взаимодействия между компонентами
C - Регрессионное тестирование	3 - Тестирование для выявления ошибок после изменений в коде
D - Системное тестирование	4 - Тестирование всей системы в целом

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 ЗАЧЕТА 1 семестр

1. Вопрос: Что такое язык программирования? Приведите примеры языков программирования. (УК-1.1)
 - Пример правильного ответа: Язык программирования — это формальный язык, предназначенный для написания компьютерных программ. Примеры языков: Python, C++, Java, JavaScript.
2. Вопрос: Что такое условный оператор и как он используется в Python? (УК-1.2)
 - Пример правильного ответа: Условный оператор (if) используется для выполнения блока кода, если заданное условие истинно. Пример: `if x > 0: print("Положительное число")`.
3. Вопрос: Какой символ используется для комментариев в Python? (УК-1.2)
 - Пример правильного ответа: Для комментариев в Python используется символ #.
4. Вопрос: Что такое список в Python и как к нему обращаться? (ОПК-6.1)
 - Пример правильного ответа: Список (list) в Python — это упорядоченная изменяемая коллекция элементов. К элементам списка можно обращаться по их индексам, начиная с нуля. Пример: `myList[0]` для обращения к первому элементу списка.
5. Вопрос: Что такое объектно-ориентированное программирование (ООП)? Приведите примеры классов и объектов. (ОПК-6.1)
 - Пример правильного ответа: Объектно-ориентированное программирование (ООП) — это парадигма программирования, основанная на концепции объектов, которые являются экземплярами классов и могут содержать данные и методы. Пример: класс `Car` и объект `myCar`.
6. Вопрос: Как реализовать наследование в Python? Приведите пример. (ОПК-6.2)
 - Пример правильного ответа: Наследование в Python реализуется с помощью указания родительского класса в круглых скобках после имени нового класса. Пример: `class ElectricCar(Car)`.
7. Вопрос: Что такое быстрая сортировка и как она работает? (УК-1.1)
 - Пример правильного ответа: Быстрая сортировка (QuickSort) — это алгоритм сортировки, который выбирает опорный элемент и разделяет массив на две части: элементы меньше опорного и элементы больше опорного, затем рекурсивно сортирует обе части.
8. Вопрос: Как открыть файл для чтения в Python? (ОПК-5.2)
 - Пример правильного ответа: Для открытия файла для чтения в Python используется функция `open()` с режимом 'r'. Пример: `file = open('file.txt', 'r')`.
9. Вопрос: Как записать данные в файл в Python? (ОПК-5.2)
 - Пример правильного ответа: Для записи данных в файл в Python используется метод `write()`. Пример: `file.write("Текст")`.

Критерии оценивания для зачета (зачет/незачет)

1. **Понимание основных понятий и терминов**
 - **Зачет:** Студент правильно использует основные понятия и термины, объясняет их в контексте.
 - **Незачет:** Студент не знает или неправильно использует основные понятия и термины, не может объяснить их.
2. **Точность и полнота ответов на теоретические вопросы**
 - **Зачет:** Ответы на большинство вопросов полные и точные, основные аспекты тем раскрыты.

- **Незачет:** Ответы неполные, содержат значительные ошибки или отсутствуют важные аспекты тем.
3. **Способность к применению знаний на практике**
 - **Зачет:** Студент правильно решает практические задачи, выполняет SQL-запросы без ошибок.
 - **Незачет:** Студент допускает ошибки при решении практических задач, не может правильно выполнить SQL-запросы.
 4. **Качество выполнения практических заданий**
 - **Зачет:** Практические задания выполнены корректно, все необходимые элементы присутствуют.
 - **Незачет:** Практические задания выполнены некорректно, содержат ошибки или неполные.

Окончательное решение по зачету:

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

3.2 Вопросы для проведения промежуточной аттестации в форме ЭКЗАМЕНА 2 семестр

1. Вопрос: Какова роль Python в программировании? (УК-1.1)
 - Ответ: Python является высокоуровневым интерпретируемым языком программирования, широко используемым для веб-разработки, анализа данных, автоматизации, научных исследований и других областей благодаря его простоте и читаемости.
2. Вопрос: Какие простые типы данных поддерживаются в Python? Приведите примеры. (УК-1.2)
 - Ответ: Python поддерживает следующие типы данных:
 - Целые числа (int), например, 42.
 - Числа с плавающей запятой (float), например, 3.14.
 - Строки (str), например, "Привет, мир!".
 - Логические значения (bool), например, True и False.
3. Вопрос: Как добавить элемент в конец списка в Python? (ОПК-6.1)
 - Ответ: Для добавления элемента в конец списка в Python используется метод `append()`. Пример: `myList.append(5)`.
4. Вопрос: Что такое объектно-ориентированное программирование (ООП)? Приведите примеры классов и объектов. (ОПК-6.1)
 - Ответ: Объектно-ориентированное программирование (ООП) — это парадигма программирования, основанная на концепции объектов, которые являются экземплярами классов и могут содержать данные и методы. Пример: класс `Car` и объект `myCar` — экземпляр класса `Car`.
5. Вопрос: Как реализовать наследование в Python? Приведите пример. (ОПК-6.2)
 - Ответ: Наследование в Python реализуется с помощью указания родительского класса в круглых скобках после имени нового класса. Пример: `class ElectricCar(Car)`. Наследование позволяет новому классу (подклассу) наследовать атрибуты и методы родительского класса.
6. Вопрос: Что такое полиморфизм в ООП и как он используется? (ОПК-6.2)
 - Ответ: Полиморфизм — это принцип ООП, позволяющий использовать единый интерфейс для объектов различных классов. Он позволяет одному и тому же методу выполнять разные действия в зависимости от класса объекта. Пример: метод `move()` может быть реализован по-разному в классах `Car` и `Bicycle`, но вызываться одинаково.

7. Вопрос: Что такое быстрая сортировка и как она работает? (УК-1.1)
 - Ответ: Быстрая сортировка (QuickSort) — это алгоритм сортировки, который выбирает опорный элемент и разделяет массив на две части: элементы меньше опорного и элементы больше опорного, затем рекурсивно сортирует обе части. Алгоритм продолжает деление до тех пор, пока не останутся только отсортированные части.
8. Вопрос: Как открыть файл для чтения в Python? (ОПК-5.2)
 - Ответ: Для открытия файла для чтения в Python используется функция `open()` с режимом `'r'`. Файл открывается, и его содержимое может быть считано методами `read()`, `readline()` или `readlines()`.
9. Вопрос: Как записать данные в файл в Python? (ОПК-5.2)
 - Ответ: Для записи данных в файл в Python используется метод `write()` объекта файла. Пример: метод `write()` записывает строку в файл.
10. Вопрос: Как обрабатывать исключения в Python? (ОПК-5.2)
 - Ответ: Для обработки исключений в Python используется конструкция `try-except`, где в блоке `try` размещается код, который может вызвать исключение, а в блоке `except` — код обработки исключения. Пример: обработка исключения деления на ноль с помощью `try` и `except`.
11. Вопрос: Какие основные конструкции управления потоком исполнения существуют в Python? (УК-1.2)
 - Ответ: В Python существуют следующие основные конструкции управления потоком исполнения:
 - Условные операторы: `if`, `elif`, `else`.
 - Циклы: `for`, `while`.
 - Операторы прерывания циклов: `break`, `continue`, `pass`.
12. Вопрос: Какие функции ввода-вывода существуют в Python и как они используются? (ОПК-5.2)
 - Ответ: Основные функции ввода-вывода в Python включают:
 - `input()`: используется для ввода данных от пользователя. Пример: `input("Введите ваше имя")`.
 - `print()`: используется для вывода данных на экран. Пример: `print("Привет, мир!")`.
13. Вопрос: Как добавить кнопку в окно Tkinter? (ОПК-5.1)
 - Ответ: Для добавления кнопки в окно Tkinter используется класс `Button()`, создается объект кнопки с указанием ее родительского элемента и текста. Кнопка добавляется в окно и выполняет заданное действие при нажатии.
14. Вопрос: Что такое функции и как они используются в Python? (ОПК-6.1)
 - Ответ: Функции в Python — это блоки кода, которые выполняют определенные задачи и могут быть вызваны в разных частях программы. Функции определяются с помощью ключевого слова `def`, за которым следует имя функции и параметры. Функции помогают структурировать код, делают его более читаемым и повторно используемым.
15. Вопрос: Что такое модули и как они используются в Python? (ОПК-6.2)
 - Ответ: Модули в Python — это файлы с кодом, которые могут быть импортированы и использованы в других программах. Модули помогают организовать и повторно использовать код. Импорт модулей осуществляется с помощью ключевого слова `import`, после которого указывается имя модуля.
16. Вопрос: Как выполнять отладку программ в Python? (ОПК-3.1)
 - Ответ: Отладка программ в Python включает использование встроенных функций и модулей, таких как `print()` для вывода промежуточных результатов и модуль `pdb` для пошагового выполнения кода. Отладка помогает выявлять и исправлять ошибки в программе.

17. Вопрос: Что такое списки в Python и как выполнять операции с ними? (ОПК-6.1)
- Ответ: Списки в Python — это упорядоченные коллекции элементов, которые могут содержать любые типы данных. Операции с списками включают добавление, удаление, изменение и доступ к элементам. Списки поддерживают методы `append()`, `extend()`, `insert()`, `remove()`, `pop()` и другие для манипулирования данными.
18. Вопрос: Как реализовать и использовать классы в Python? (ОПК-6.2)
- Ответ: Классы в Python определяются с помощью ключевого слова `class`, за которым следует имя класса и его атрибуты и методы. Объекты создаются из классов путем вызова класса с использованием скобок. Классы помогают организовать данные и поведение в одном месте, что упрощает разработку и поддержку программ.
19. Вопрос: Как обрабатывать ошибки и исключения в Python? (ОПК-3.1)
- Ответ: Ошибки и исключения в Python обрабатываются с помощью конструкции `try-except`. В блоке `try` размещается код, который может вызвать исключение, а в блоке `except` — код обработки исключения. Также можно использовать `finally` для выполнения кода независимо от того, произошло исключение или нет.

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.