

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунов Федор Владимирович

Должность: Ректор

Дата подписания: 24.09.2026 14:09:47

Уникальный программный ключ:

fd935d10451b860e912264c037818448451b607f94388008e39877e6bcb55

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ»
«УНИВЕРСИТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ»
Факультет экономики, управления и юриспруденции**

Кафедра управления и бизнес-информатики



УТВЕРЖДАЮ

Проректор по учебно-методической работе

Г.П. Узунова / Г.П. Узунова

«02» февраля 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

PYTHON И ЕГО БИБЛИОТЕКА

Направление подготовки

09.03.02 «Информационные системы и технологии»

Профиль

«Специалист информационных систем и технологий»

Квалификация выпускника: бакалавр

Для всех
форм обучения

Симферополь, 2026 г.

1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Средства оценивания в ходе текущего контроля:

- устные опросы в ходе семинарских занятий;
- рефераты;
- тестирование;
- практические задания, выполняемые в ходе семинарского (практического) занятия или рекомендуемые для самостоятельной работы.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ПК-1	ПК-1. Способен к разработке и отладке программного кода	ПК-1.1. Основы математической логики и алгоритмизации. Методы формализации задач. Принципы и методы проектирования алгоритмов. Основные языки программирования ПК-1.2. Писать программный код на различных языках программирования. Определять и манипулировать данными в базах данных. Писать и оформлять код в соответствии с установленными стандартами. Комментировать и документировать код. ПК-1.3. Владеть: Средствами и инструментами для формализации и алгоритмизации задач. Метаподходами и шаблонами для проектирования алгоритмов. Инструментами для работы с базами данных.

1.1 Вопросы к текущему контролю

1. Какие встроенные типы данных относятся к изменяемым (mutable), а какие — к неизменяемым (immutable) в Python? Приведите примеры.
2. В чем разница между операторами `is` и `==` при сравнении объектов?
3. Объясните принцип динамической типизации в Python. Что такое утиная типизация (*Duck Typing*)?
4. Как работает механизм управления памятью в Python? Что такое счетчик ссылок и сборщик мусора (*Garbage Collector*)?
5. Для чего используется конструкция `if __name__ == "__main__":`?
6. Чем отличаются байтовые строки (bytes) от строк текста (str) и как происходит их кодирование/декодирование?
7. В каких случаях целесообразно использовать кортежи (tuple) вместо списков (list)?
8. Чем отличается множество (set) от списка с точки зрения производительности поиска элемента? Какой метод объединения множеств удаляет дубликаты?
9. Напишите пример генератора словаря (*dictionary comprehension*) для создания таблицы квадратов чисел от 1 до 10.
10. Назовите три основных инструмента функционального программирования в Python из стандартной библиотеки (map, filter, lambda или functools.reduce).
11. Что такое декоратор? Напишите простейший пример декоратора, который выводит время выполнения функции.

12. Как передавать переменное количество позиционных (*args) и именованных (**kwargs) аргументов в функцию?
13. В чем заключается проблема изменения значения аргумента по умолчанию, если он задан как изменяемый объект (например, список)?
14. Что такое магические методы (dunder methods)? Приведите примеры методов `__init__` и `__str__`.
15. Как реализуется инкапсуляция в Python? В чем разница между публичным, защищенным (`_var`) и приватным (`__var`) атрибутами класса?
16. Что такое наследование? Как вызвать конструктор родительского класса в Python 3?
17. Почему рекомендуется открывать файлы через контекстный менеджер (`with open(...) as f:`) вместо обычного вызова `open()` и `close()`?
18. В чем разница между блоками `except Exception as e:` и общим `except:`?
19. Можно ли выполнять код внутри блока `finally`, если исключение не возникло?
20. Создайте иерархию собственных исключений: базовый класс `AppError` и дочерний класс `ValidationError`. Покажите пример использования конструкции `raise ValidationError("Ошибка")`.

1.2 Темы рефератов:

1. История создания и эволюция Python: от Гвидо ван Россума до наших дней (переход с Python 2 на Python 3).
2. Философия программирования The Zen of Python: анализ принципов написания идиоматичного кода (*PEP 8*).
3. Синтаксические особенности Python: роль отступов, динамическая типизация и концепция *Duck Typing*.
4. Многопарадигменность Python: поддержка процедурного, объектно-ориентированного и функционального стилей в одном проекте.
5. Управление памятью в Python: механизм подсчета ссылок и работа сборщика мусора (*Garbage Collector*), проблема циклических ссылок.
6. Глобальная блокировка интерпретатора (*GIL*): причины возникновения, влияние на многопоточность и способы обхода ограничения.
7. Модуль `collections`: расширенные структуры данных (`namedtuple`, `deque`, `Counter`, `defaultdict`) и их практическое применение.
8. Работа с файловой системой через модули `os` и `shutil`: кроссплатформенное взаимодействие с ОС.
9. Обработка дат и времени в Python: возможности модулей `datetime` и `calendar`, часовые пояса (`pytz`, `zoneinfo`).
10. Стандартный модуль `logging`: настройка гибкой системы логирования приложений вместо использования функции `print()`.
11. Параллельное выполнение задач: сравнение многопроцессорности (`multiprocessing`) и асинхронности (`asyncio`).
12. Библиотека `NumPy`: архитектура массивов (`ndarray`) и векторизация операций как основа высокопроизводительных вычислений.
13. Библиотека `Pandas`: инструменты для очистки, трансформации и агрегации табличных данных (*DataFrame*).
14. Визуализация данных `Matplotlib` и `Seaborn`: создание статических графиков и интерактивных дашбордов.
15. Автоматизация тестирования с помощью `Pytest`: написание простых тестов, фикстуры и параметризация тест-кейсов.
16. Разработка веб-приложений на Django vs Flask: сравнительный анализ монолитных фреймворков и микрофреймворков.

17. Библиотека Requests: стандарты работы с HTTP-запросами, обработка сессий и аутентификация.
18. Автоматизация GUI и браузеров: использование библиотек Selenium или PyAutoGUI для роботизации рутинных процессов.
19. Инструменты машинного обучения Scikit-learn и TensorFlow: обзор алгоритмов классификации и регрессии в экосистеме Python.
20. Разработка настольных (Desktop) приложений: обзор библиотек Tkinter, PyQt/PySide и Kivy для создания интерфейсов.

1.3 Тестовые задания

Тест 1. Синтаксис и типы данных

Вопрос: Что произойдет при выполнении кода `x = [1, 2]; y = x; y.append(3); print(x)`?

- а) `[1, 2]`
- б) `[1, 2, 3]`
- в) Ошибка `TypeError` г) `[3]`

Правильный ответ: б) `[1, 2, 3]` (Списки — изменяемый тип, переменные ссылаются на один объект).

Тест 2. Условные операторы

Вопрос: Какой результат выведет код: `a = 10; b = "10"; print(a is b)`?

- а) `True`
- б) `False`
- в) `10` г) Ошибка сравнения типов

Правильный ответ: б) `False` (Оператор `is` проверяет идентичность объектов в памяти, а не равенство значений).

Тест 3. Циклы и генераторы

Вопрос: Какое ключевое слово используется для создания генератора вместо списка внутри квадратных скобок?

- а) `yield`
- б) `lambda`
- в) `gen`
- г) Ничего менять не нужно, это определяется автоматически

Правильный ответ: а) `yield` (или замена `[]` на `()` с выражением-генератором).

Тест 4. Аргументы функций

Вопрос: В чем опасность использования изменяемого объекта (например, списка) в качестве значения аргумента по умолчанию (`def f(arg=[])`)?

- а) Код будет работать медленнее
- б) Объект будет общим для всех вызовов функции, изменения сохранятся между ними
- в) Python выдаст ошибку `SyntaxError`
- г) Список будет принудительно скопирован при каждом вызове

Правильный ответ: б) Объект будет общим для всех вызовов функции...

Тест 5. Стандартная библиотека (модуль `os`)

Вопрос: Какой модуль стандартной библиотеки следует использовать для взаимодействия с операционной системой (работа с файлами, путями, процессами)?

- а) `sys`
- б) `os`
- в) `subprocess`
- г) `pathlib`

Правильный ответ: б) `os`

Тест 6. Работа с датой и временем

Вопрос: Какой класс из модуля `datetime` следует использовать для работы с датой и временем с учетом часового пояса?

- а) `datetime.date`

- б) `datetime.time`
- в) `datetime.datetime`
- г) `datetime.timezone`

Правильный ответ: в) `datetime.datetime` (объединяет дату, время и `tzinfo`).

Тест 7. Библиотека NumPy

Вопрос: Чем массив NumPy (`ndarray`) принципиально отличается от встроенного списка Python?

- а) Массив NumPy может хранить данные разных типов одновременно
- б) Все элементы массива NumPy должны быть одного типа, что обеспечивает высокую скорость вычислений
- в) Списки быстрее работают с линейной алгеброй
- г) Массивы NumPy неизменяемы

Правильный ответ: б) Все элементы массива NumPy должны быть одного типа...

Тест 8. Библиотека Pandas

Вопрос: Основной двумерный объект данных в библиотеке Pandas называется:

- а) `Matrix`
- б) `DataFrame`
- в) `Table`
- г) `Series`

Правильный ответ: б) `DataFrame`

Тест 9. Визуализация (Matplotlib/Seaborn)

Вопрос: Какая функция обычно используется для отображения построенного графика на экран?

- а) `plt.draw()`
- б) `plt.show()`
- в) `plt.render()`
- г) `plot.display()`

Правильный ответ: б) `plt.show()`

Тест 10. Веб-библиотеки (Requests)

Вопрос: Какой метод объекта `Response` необходимо вызвать, чтобы получить тело ответа сервера в виде словаря, если сервер вернул JSON?

- а) `response.text`
- б) `response.json()`
- в) `json.loads(response.content)`
- г) `response.parse_json()`

Правильный ответ: б) `response.json()`

Тест 11. Модуль random

Вопрос: Какую функцию использовать для случайного выбора одного элемента из последовательности `items`?

- а) `random.choice(items)`
- б) `random.sample(items, 1)[0]`
- в) `items[random.randint(0, len(items)-1)]`
- г) Все варианты верны

Правильный ответ: г) Все варианты верны (но `choice` — самый идиоматичный).

Тест 12. Регулярные выражения (re)

Вопрос: Какой флаг позволяет искать совпадение без учета регистра букв?

- а) `re.I` или `re.IGNORECASE`
- б) `re.M` или `re.MULTILINE`
- в) `re.S` или `re.DOTALL`
- г) `re.X` или `re.VERBOSE`

Правильный ответ: а) `re.I` или `re.IGNORECASE`

Тест 13. Исключения

Вопрос: Для чего предназначен блок else в конструкции try-except?

- а) Для выполнения кода, если исключение НЕ возникло
- б) Для очистки ресурсов в любом случае
- в) Для перехвата необрабатываемых исключений
- г) Это синоним блока except

Правильный ответ: а) Для выполнения кода, если исключение НЕ возникло.

Тест 14. ООП (Магические методы)

Вопрос: Какой магический метод вызывается при использовании функции len(obj)?

- а) obj.size()
- б) obj.length()
- в) obj.len()
- г) obj.size()

Правильный ответ: в) obj.len()

Тест 15. Декораторы

Вопрос: Что обязательно должна делать любая корректная реализация декоратора?

- а) Принимать только одну функцию в качестве аргумента
- б) Возвращать другую функцию (callable object)
- в) Печатать аргументы декорируемой функции
- г) Использовать библиотеку functools.wraps

Правильный ответ: б) Возвращать другую функцию (callable object).

Тест 16. Вывод данных

Вопрос: Для вывода информации в консоль используется встроенная функция _____.()

Ответ: print

Тест 17. Импорт модулей

Вопрос: Чтобы подключить сторонний функционал (например, математические функции), необходимо использовать ключевое слово _____, после которого указывается имя модуля.

Ответ: import

Тест 18. Списки (Lists)

Вопрос: Метод списка, который добавляет элемент в самый конец массива и изменяет сам список на месте, называется _____.append(item).

Ответ: .append()

Тест 19. Словари (Dictionaries)

Вопрос: Для безопасного получения значения из словаря по ключу без возникновения ошибки KeyError используется метод get(): value = my_dict.get('key', '_____'), где второе значение — это результат при отсутствии ключа.

Ответ: default (или любое значение по умолчанию)

Тест 20. Функции

Вопрос: Тело любой пользовательской функции начинается с двоеточия и обязательного отступа (обычно 4 пробела). Ключевое слово для начала определения функции — _____.

Ответ: def

Тест 21. Стандартная библиотека (Math)

Вопрос: Для использования константы числа Пи или вычисления квадратного корня нужно импортировать модуль math: import math; result = math.sqrt(16). Константа числа Пи записывается как math._____.

Ответ: pi

Тест 22. Библиотека NumPy

Вопрос: Основной объект данных в библиотеке NumPy, представляющий собой многомерную таблицу элементов одного типа, называется ndarray, а создается функцией numpy._____().

Ответ: array

Тест 23. Библиотека Pandas

Вопрос: При чтении табличных данных из текстового файла чаще всего используют функцию `pandas.read_csv()`. Чтобы сохранить обработанный DataFrame обратно в файл, используют метод `df.to_____` ('file.csv').

Ответ: csv (или `to_excel` / `to_json` в зависимости от контекста, но `to_csv` наиболее частый)

Тест 24. Обработка исключений

Вопрос: Конструкция перехвата ошибок выглядит так: `try: ... except ValueError as e: ....` Если код должен выполняться в любом случае (даже если ошибок не было), он помещается в блок _____.

Ответ: `else` (блок `finally` тоже подходит технически, но `else` выполняется только при успехе, что часто проверяется в тестах на логику потока).

Тест 25. Анонимные функции

Вопрос: Для создания короткой одноразовой функции без имени (например, внутри `sorted()`) используется ключевое слово _____.

Ответ: `lambda`

26. Установить соответствие между термином и определением:

Тест 1. Базовые структуры данных (Коллекции)

Термины:

1. Кортеж (Tuple)
2. Словарь (Dictionary)
3. Множество (Set)
4. Список (List)

Определения:

- А) Изменяемая, упорядоченная коллекция элементов произвольных типов, доступ к которым осуществляется по индексу.
- Б) Неизменяемая последовательность элементов, часто используемая для защиты данных от случайной модификации или как ключи словаря.
- В) Неупорядоченная коллекция уникальных элементов, поддерживающая математические операции (пересечение, объединение).
- Г) Неупорядоченная (до версии 3.7) или упорядоченная (с 3.7+) структура пар «ключ-значение», обеспечивающая быстрый поиск по ключу.

Правильные ответы: 1 — Б; 2 — Г; 3 — В; 4 — А.

27. Установить соответствие между термином и определением:

Стандартная библиотека Python

Термины:

1. Модуль `os`
2. Модуль `sys`
3. Модуль `datetime`
4. Модуль `random`

Определения:

- А) Инструменты для работы с датами, временем, часовыми поясами и арифметики временных интервалов.
- Б) Средства взаимодействия с операционной системой: работа с файловой системой, путями и переменными окружения.
- В) Параметры интерпретатора, аргументы командной строки (`argv`) и механизмы завершения программы (`exit()`).
- Г) Генерация псевдослучайных чисел, выбор случайных элементов из последовательностей.

Правильные ответы: 1 — Б; 2 — В; 3 — А; 4 — Г.

28. Установить соответствие между термином и определением:

Библиотеки анализа данных (Data Science)

Термины:

1. NumPy (библиотека)
2. Pandas (библиотека)
3. Matplotlib (библиотека)
4. Requests (библиотека)

Определения:

А) Основной инструмент для построения графиков, диаграмм и визуализации статистических данных.

Б) Мощный инструмент для манипулирования табличными данными (объекты Series и DataFrame), очистки CSV-файлов и агрегации.

В) Фундаментальный пакет для научных вычислений, предоставляющий объект многомерного массива ndarray и быстрые векторные операции.

Г) Простая и элегантная HTTP-библиотека для отправки запросов к веб-серверам и получения ответов.

Правильные ответы: 1 — В; 2 — Б; 3 — А; 4 — Г.

29. Установить соответствие между термином и определением:

Концепции ООП и функций

Термины:

1. Декоратор
2. Лямбда-функция
3. Dunder-метод (магический метод)
4. GIL (Global Interpreter Lock)

Определения:

А) Специальный метод класса, название которого окружено двойными подчеркиваниями (например, `__init__`, `__str__`), определяющий поведение встроенных операторов.

Б) Механизм блокировки в CPython, позволяющий выполнять только один поток Python одновременно на одном процессоре.

В) Функция без имени, ограниченная одной строкой-выражением, создаваемая с помощью ключевого слова `lambda`.

Г) Функция высшего порядка, которая принимает другую функцию, расширяет её функциональность без изменения кода и возвращает новую функцию.

Правильные ответы: 1 — Г; 2 — В; 3 — А; 4 — Б.

30. Установить соответствие между термином и определением:

Управление исключениями и контекстом

Термины:

1. Блок try-except
2. Блок finally
3. Конструкция with open(...) as f (Context Manager)
4. Ключевое слово raise

Определения:

А) Программная конструкция, гарантирующая выполнение блока кода при выходе из секции (например, закрытие файла), независимо от того, возникла ошибка или нет.

Б) Синтаксис для перехвата и обработки ошибок времени выполнения, предотвращающий аварийное завершение программы.

В) Принудительное инициирование исключения вручную внутри кода для сигнализации об ошибке логики.

Г) Менеджер контекста, который автоматически вызывает методы входа/выхода (`__enter__`, `__exit__`) для безопасного управления ресурсами (файлами, сетевыми соединениями).

Правильные ответы: 1 — Б; 2 — А; 3 — Г; 4 — В

1.4 Практические задания

Задача 1. Конвертер валют Напишите программу, которая запрашивает у пользователя сумму в рублях и курс доллара. Программа должна выводить эквивалентную сумму в долларах США, округленную до двух знаков после запятой. *Ключевые функции:* `input()`, `float()`, форматирование строк (`f-string`).

Задача 2. Площадь кольца Даны два числа: внешний радиус R и внутренний радиус r . Напишите программу вычисления площади кольца. Используйте константу π из модуля `math`. *Ключевые модули:* `import math`.

Задача 3. Анаграмма Пользователь вводит два слова. Определите, являются ли они анаграммами (состоят из одинаковых букв в разном порядке). Регистр букв не учитывается. *Ключевые методы:* `.lower()`, `sorted()`.

Задача 4. Валидатор пароля Проверьте надежность введенного пользователем пароля. Пароль считается надежным, если:

1. Длина больше 8 символов.
2. Содержит хотя бы одну цифру.
3. Содержит хотя бы одну заглавную букву. Выведите "Надежный" или "Слабый". *Ключевые методы:* `len()`, `.isdigit()`, `.isupper()`.

Задача 5. Формат даты Программа получает строку вида "дд.мм.гггг" (например, "05.12.2023"). Преобразуйте её в формат "гггг-мм-дд" (ISO 8601). *Ключевые методы:* `.split()`, `f-строки`.

Задача 6. Уникальные элементы Дан список чисел. Выведите новый список, содержащий только те элементы, которые встречаются во входном списке ровно один раз, сохраняя их исходный порядок. *Ключевые структуры:* `list`, метод `.count()` или модуль `collections.Counter`.

Задача 7. Инвертированный словарь Есть словарь `{имя: возраст}`. Создайте новый словарь, где ключом будет возраст, а значением — список имен людей этого возраста. *Ключевые структуры:* вложенные циклы, проверка наличия ключа (`in`).

Задача 8. Сортировка студентов Список содержит словари с данными студентов: `[{'name': 'Ivan', 'score': 85 }, ...]`. Отсортируйте этот список сначала по оценке (по убыванию), а при равных оценках — по имени (по возрастанию). *Ключевая функция:* `sorted(key=lambda x: (-x['score'], x['name']))`.

Задача 9. Частотный анализ текста Введите текст (строку). Подсчитайте частоту каждого символа (кроме пробелов) и выведите топ-3 самых частых символа. *Ключевые модули:* `collections.Counter`.

Задача 10. Объединение интервалов Дан список кортежей-интервалов `[(1, 3), (2, 6), (8, 10)]`. Объедините пересекающиеся интервалы. Результат для примера: `[(1, 6), (8, 10)]`. *Алгоритм:* сортировка + проход по списку.

Задача 11. Декоратор логирования Создайте декоратор `@log_call`, который перед выполнением декорируемой функции выводит её имя и аргументы, а после выполнения — результат. *Ключевое понятие:* Замыкания, `*args`, `**kwargs`.

Задача 12. Калькулятор с исключениями Реализуйте функцию деления `safe_divide(a, b)`. Она должна возвращать результат деления, но перехватывать деление на ноль (`ZeroDivisionError`) и возврат строки "Ошибка: деление на ноль". *Ключевой блок:* `try...except`.

Задача 13. Класс «Банковский счет» Создайте класс `BankAccount` со свойствами `balance` и методами `deposit(amount)` и `withdraw(amount)`. Метод снятия должен проверять наличие средств и выбрасывать исключение `ValueError`, если денег недостаточно. *Ключевое понятие:* `self`, инкапсуляция.

Задача 14. Генератор Фибоначчи Напишите генератор (функцию с `yield`), который бесконечно выдает числа последовательности Фибоначчи. *Ключевое слово:* `yield`.

Задача 15. Аннотации типов Напишите функцию обработки списка, используя аннотации типов (PEP 484). Например: `def process(data: list[int]) -> dict[str, int]:`. *Ключевое понятие:* Type Hinting.

Задача 16. Обработка лог-файла Прочитайте файл `server.log`, найдите все строки, содержащие слово "ERROR", и сохраните их в отдельный файл `errors.log`. *Ключевой контекст:* `with open(...) as f:`.

Задача 17. Анализ продаж (CSV) Используя стандартную библиотеку `csv`, прочитайте файл `sales.csv` (столбцы: Дата, Товар, Сумма). Посчитайте общую выручку по каждому товару. *Ключевой модуль:* `import csv, DictReader`.

Задача 18. Сохранение настроек (JSON) Напишите скрипт, который сохраняет текущие настройки программы (словарь Python) в файл `config.json` и умеет считывать их обратно при следующем запуске. *Ключевой модуль:* `import json, json.dump(), json.load()`.

Задача 19. Векторизация (NumPy) Дан массив температур в градусах Цельсия. Используя NumPy, переведите их в Фаренгейты по формуле $^{\circ}\text{F} = ^{\circ}\text{C} \times 1.8 + 32$ одной строчкой кода без циклов. *Ключевой код:* `np.array([...]) * 1.8 + 32`.

Задача 20. Агрегация данных (Pandas) Имеется DataFrame `df` с колонками `City` и `Sales`.

1. Удалите дубликаты городов.
2. Найдите город с максимальной суммой продаж.
3. Добавьте новую колонку `Tax`, равную 20% от суммы продаж. *Ключевые методы:* `.drop_duplicates()`, `.groupby().sum()`, `.idxmax()`, векторное присваивание `df['new'] = df['col'] * 0.2`.

2. КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОВЕДЕНИИ ТЕКУЩЕГО КОНТРОЛЯ

Вид контроля	Наименование работы	Наименование оценочных средств	Шкала оценивания
Текущий контроль	- Вопросы для обсуждения на занятиях; - Устные опросы по ранее изученному материалу; - Письменные работы: рефераты, тестовые задания; - Практические задания; - Рефераты и доклады по темам (вопросам), вынесенным на самостоятельную работу.	Оценка выступлений на практическом (семинарском) занятии, проверка заданий и аудиторных работ, устный опрос, оценивание докладов, рефератов	отлично хорошо удовлетворительно неудовлетворительно

Критерии оценивания устных ответов обучающихся

Шкала оценивания	Характеристика оценивания
отлично	оценивается ответ, который показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа.
хорошо	оценивается ответ, обнаруживающий прочные знания основных процессов

	изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.
удовлетворительно	оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа.
неудовлетворительно	оценивается ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа.

Критерии оценивания работы обучающихся на практических и семинарских занятиях

Шкала оценивания	Показатели	Критерии
Отлично	1. Полнота выполнения практического и тестового задания (полнота ответа); 2. Своевременность выполнения задания; 3. Последовательность и рациональность выполнения	Задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Дан правильный и исчерпывающий ответ на поставленные теоретические и тестовые вопросы, в которых обучающийся показал всестороннее системное знание программного материала, усвоение основной и дополнительной литературы, четкое владение понятийным аппаратом.
Хорошо	практического задания (логичность и четкость ответа); 4. Правильность ответов на вопросы; 5. Самостоятельность решения (владение дополнительным материалом); 6. Знание нормативно-законодательной базы и	Задание решено с помощью преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор формул для решения; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ. На поставленные теоретические и тестовые вопросы, при которых обучающийся показал достаточный уровень знаний основного программного материала: освоение информации лекционного курса и учебных пособий, овладение понятийным аппаратом, методикой исследований при попытке анализа различных ситуаций.

Шкала оценивания	Показатели	Критерии
Удовлетворительно	терминологии курса	Задание решено с подсказками преподавателя. Задание решено в общем виде. Обучающийся показал средний уровень знаний основного программного материала, но не мог убедительно аргументировать свой ответ, ошибся в использовании понятийного аппарата, показал недостаточные знания литературных источников.
Неудовлетворительно		Задание не решено. Обучающийся продемонстрировал значительные пробелы в знаниях основного программного материала, не аргументировал свой ответ, показал неудовлетворительные знания понятийного аппарата и специальной литературы.

Критерии оценивания рефератов

Средство контроля	Критерии оценивания	Шкала оценивания
Реферат	Выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы. Реферат раскрывает поднятую проблематику в полном объёме.	отлично
	Основные требования к реферату и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём реферата; имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы. В реферате имеются неточности и предметная область выступления раскрыта не в полной мере.	хорошо
	Имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод. В реферате не в полной степени раскрыт понятийный аппарат, имеются существенные неточности в процессе формирования выводов.	удовлетворительно
	Тема реферата не раскрыта, обнаруживается существенное непонимание проблемы. Тема реферата не раскрыта или выполнена не по существу ранее поставленного вопроса. Реферат не сдан / доклад не сделан.	неудовлетворительно

Критерии оценивания тестов

Средство контроля	Критерии оценивания – процент положительных ответов	Шкала оценивания
Тестирование	90-100	отлично
	70-89	хорошо
	40-69	удовлетворительно
	< 39	неудовлетворительно

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Средства оценивания в ходе промежуточной аттестации:

- вопросы для экзамена;
- тестовые задания к экзамену.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ПК-1	ПК-1. Способен к разработке и отладке программного кода	ПК-1.1. Основы математической логики и алгоритмизации. Методы формализации задач. Принципы и методы проектирования алгоритмов. Основные языки программирования ПК-1.2. Писать программный код на различных языках программирования. Определять и манипулировать данными в базах данных. Писать и оформлять код в соответствии с установленными стандартами. Комментировать и документировать код. ПК-1.3. Владеть: Средствами и инструментами для формализации и алгоритмизации задач. Метаподходами и шаблонами для проектирования алгоритмов. Инструментами для работы с базами данных.

3.1. Вопросы к экзамену:

- Основные парадигмы программирования, поддерживаемые Python. В чем заключается принцип *Duck Typing*?
- Динамическая типизация в Python. Как работает подсчет ссылок (*Reference Counting*) и сборщик мусора (*Garbage Collector*) для управления памятью?
- Различия между изменяемыми (*mutable*) и неизменяемыми (*immutable*) типами данных. Примеры объектов каждого типа.
- Особенности работы со строками: кодировки, методы форматирования (`%`, `.format()`, `f`-строки) и сырые строки (*raw strings*).
- Списки (*list*): основные операции, сложность доступа/изменения, отличие от кортежей (*tuple*).
- Словари (*dict*): хеш-таблицы как основа реализации, требования к ключам, методы `keys()`, `values()`, `items()`.
- Множества (*set*): математические операции над множествами и их применение для фильтрации уникальных элементов.

8. Иерархия типов чисел: `int` (произвольная точность), `float`, `complex`. Проблемы точности вещественных чисел.
9. Логический тип (`bool`) и истинность объектов в Python. Операторы сравнения и цепочки сравнений (`a < b < c`).
10. Понятие объекта первого класса (*First-class citizen*). Функции как объекты.
11. Модуль `collections`: назначение и примеры использования `namedtuple`, `deque`, `Counter`, `defaultdict`.
12. Итераторы и генераторы: протокол итерации (`__iter__`, `__next__`), отличия генератора от списка.
13. Выражения-генераторы и генераторные функции (ключевое слово `yield`). Экономия памяти при обработке больших потоков данных.
14. Работа с файловой системой через модули `os` и `pathlib`. Кроссплатформенность путей.
15. Обработка дат и времени: классы `datetime.date`, `datetime.time`, `datetime.datetime`, работа с часовыми поясами (`zoneinfo`).
16. Анонимные функции (`lambda`). Ограничения лямбда-выражений.
17. Встроенные функции высшего порядка: `map()`, `filter()`, `reduce()` (из модуля `functools`).
18. Что такое декоратор? Механизм оборачивания функций. Использование `functools.wraps` для сохранения метаданных.
19. Декораторы с параметрами. Фабрики декораторов.
20. Замыкания (*closures*): как работают свободные переменные во вложенных функциях.
21. Определение класса. Конструктор `__init__` и метод `__new__`. Атрибуты экземпляра vs атрибуты класса.
22. Инкапсуляция в Python: соглашения о публичности, защищенности (`_var`) и приватности (`__var` name mangling).
23. Наследование: одиночное и множественное. Порядок разрешения методов (*MRO*) и функция `super()`.
24. Полиморфизм в Python (утиная типизация). Абстрактные базовые классы (*ABC*, модуль `abc`).
25. Магические методы (*dunder methods*): эмуляция числовых типов (`__add__`, `__mul__`), контекста (`__enter__`, `__exit__`).
26. Перегрузка операторов. Протокол дескрипторов (*Descriptor protocol*).
27. Миксин-классы (*Mixins*) как способ организации кода.
28. Статические методы (`@staticmethod`) и методы класса (`@classmethod`). Различия в использовании.
29. Управление ссылками на объект: метод `__del__` и слабые ссылки (`weakref`).
30. Утиная типизация против проверки типов (`isinstance`, `type`). Аннотации типов (*Type Hinting*, PEP 484).
31. Система исключений Python. Базовый класс `Exception`. Стандартная иерархия встроенных ошибок.
32. Конструкция `try...except...else...finally`. Назначение каждого блока.
33. Инструкция `raise`. Создание собственных классов исключений.
34. Контекстные менеджеры (`with statement`). Протокол менеджера контекста.
35. Подавление исключений с помощью `contextlib.suppress`.
36. Структура модуля и пакета. Файл `__init__.py`. Относительный и абсолютный импорт.
37. Пути поиска модулей (`sys.path`). Переменная окружения `PYTHONPATH`.
38. Виртуальные окружения (`venv`, `virtualenv`). Изоляция зависимостей проектов.
39. Менеджер пакетов `pip`. Файлы требований (`requirements.txt`) и файлы блокировки (`poetry.lock`, `Pipfile.lock`).
40. Документация и интроспекция: `docstrings`, функция `help()`, атрибут `__doc__`, модуль `inspect`.
41. Библиотека **NumPy**: структура массива `ndarray`, векторизация операций, преимущества перед списками.
42. Индексация в NumPy: срезы, целочисленная индексация, булева маска (*fancy indexing*).
43. Библиотека **Pandas**: структуры `Series` и `DataFrame`. Отличие `label-based` (`loc`) от `position-based` (`iloc`) индексации.
44. Пропущенные данные в Pandas (`NaN`). Методы очистки данных: `.dropna()`, `.fillna()`.
45. Группировка и агрегация данных в Pandas: `.groupby()`, сводные таблицы (`pivot_table`).

46. Библиотека **Matplotlib**: архитектура (Figure, Axes). Построение базовых графиков (plot, scatter, hist).
47. Оформление графиков Matplotlib: подписи, легенды, сетки, стили (plt.style.use).
48. Библиотека **Seaborn**: преимущества перед чистым Matplotlib для статистической визуализации.
49. Символьные вычисления: обзор возможностей библиотеки **SymPy** (решение уравнений, дифференцирование).
50. Форматы сериализации данных: сравнение JSON, CSV, XML и бинарного формата Pickle. Риски безопасности при использовании pickle.loads.

3.2. Задания для экзамена:

Задача 1. Конвертер валют Напишите программу, которая запрашивает у пользователя сумму в рублях и курс доллара. Программа должна выводить эквивалентную сумму в долларах США, округленную до двух знаков после запятой. *Ключевые функции:* `input()`, `float()`, форматирование строк (f-string).

Задача 2. Площадь кольца Даны два числа: внешний радиус R и внутренний радиус r . Напишите программу вычисления площади кольца. Используйте константу π из модуля `math`. *Ключевые модули:* `import math`.

Задача 3. Анаграмма Пользователь вводит два слова. Определите, являются ли они анаграммами (состоят из одинаковых букв в разном порядке). Регистр букв не учитывается. *Ключевые методы:* `.lower()`, `sorted()`.

Задача 4. Валидатор пароля Проверьте надежность введенного пользователем пароля. Пароль считается надежным, если:

- Длина больше 8 символов.
- Содержит хотя бы одну цифру.
- Содержит хотя бы одну заглавную букву. Выведите "Надежный" или "Слабый". *Ключевые методы:* `len()`, `.isdigit()`, `.isupper()`.

Задача 5. Формат даты Программа получает строку вида "дд.мм.гггг" (например, "05.12.2023"). Преобразуйте её в формат "гггг-мм-дд" (ISO 8601). *Ключевые методы:* `.split()`, f-строки.

Задача 6. Уникальные элементы Дан список чисел. Выведите новый список, содержащий только те элементы, которые встречаются во входном списке ровно один раз, сохраняя их исходный порядок. *Ключевые структуры:* `list`, метод `.count()` или модуль `collections.Counter`.

Задача 7. Инвертированный словарь Есть словарь {имя: возраст}. Создайте новый словарь, где ключом будет возраст, а значением — список имен людей этого возраста. *Ключевые структуры:* вложенные циклы, проверка наличия ключа (`in`).

Задача 8. Сортировка студентов Список содержит словари с данными студентов: `[{'name': 'Ivan', 'score': 85}, ...]`. Отсортируйте этот список сначала по оценке (по убыванию), а при равных оценках — по имени (по возрастанию). *Ключевая функция:* `sorted(key=lambda x: (-x['score'], x['name']))`.

Задача 9. Частотный анализ текста Введите текст (строку). Подсчитайте частоту каждого символа (кроме пробелов) и выведите топ-3 самых частых символа. *Ключевые модули:* `collections.Counter`.

Задача 10. Объединение интервалов Дан список кортежей-интервалов `[(1, 3), (2, 6), (8, 10)]`. Объедините пересекающиеся интервалы. Результат для примера: `[(1, 6), (8, 10)]`. *Алгоритм:* сортировка + проход по списку.

Задача 11. Декоратор логирования Создайте декоратор `@log_call`, который перед выполнением декорируемой функции выводит её имя и аргументы, а после выполнения — результат. *Ключевое понятие:* Замыкания, `*args`, `**kwargs`.

Задача 12. Калькулятор с исключениями Реализуйте функцию деления `safe_divide(a, b)`. Она должна возвращать результат деления, но перехватывать деление на ноль

(ZeroDivisionError) и возврат строки "Ошибка: деление на ноль". *Ключевой блок:* try...except.

Задача 13. Класс «Банковский счет» Создайте класс BankAccount со свойствами balance и методами deposit(amount) и withdraw(amount). Метод снятия должен проверять наличие средств и выбрасывать исключение ValueError, если денег недостаточно. *Ключевое понятие:* self, инкапсуляция.

Задача 14. Генератор Фибоначчи Напишите генератор (функцию с yield), который бесконечно выдает числа последовательности Фибоначчи. *Ключевое слово:* yield.

Задача 15. Аннотации типов Напишите функцию обработки списка, используя аннотации типов (PEP 484). Например: def process(data: list[int]) -> dict[str, int]:. *Ключевое понятие:* Type Hinting.

Задача 16. Обработка лог-файла Прочитайте файл server.log, найдите все строки, содержащие слово "ERROR", и сохраните их в отдельный файл errors.log. *Ключевой контекст:* with open(...) as f:.

Задача 17. Анализ продаж (CSV) Используя стандартную библиотеку csv, прочитайте файл sales.csv (столбцы: Дата, Товар, Сумма). Посчитайте общую выручку по каждому товару. *Ключевой модуль:* import csv, DictReader.

Задача 18. Сохранение настроек (JSON) Напишите скрипт, который сохраняет текущие настройки программы (словарь Python) в файл config.json и умеет считывать их обратно при следующем запуске. *Ключевой модуль:* import json, json.dump(), json.load().

Задача 19. Векторизация (NumPy) Дан массив температур в градусах Цельсия. Используя NumPy, переведите их в Фаренгейты по формуле $°F = C \times 1.8 + 32$ одной строчкой кода без циклов. *Ключевой код:* np.array([...]) * 1.8 + 32.

Задача 20. Агрегация данных (Pandas) Имеется DataFrame df с колонками City и Sales.

4. Удалите дубликаты городов.
5. Найдите город с максимальной суммой продаж.
6. Добавьте новую колонку Tax, равную 20% от суммы продаж. *Ключевые методы:* .drop_duplicates(), .groupby().sum(), .idxmax(), векторное присваивание df['new'] = df['col'] * 0.2.

4. ОСНОВНЫЕ КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Шкала оценивания уровня сформированности компетенций (по пятибалльной системе) экзамен

Формируемые уровни освоения компетенций	Критерии оценивания	Шкала оценивания
Высокий уровень	Изложено правильное понимание вопроса, четко и самостоятельно дан исчерпывающий ответ, содержание раскрыто полно, профессионально, грамотно. Обучающимся усвоена взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии. Отражает успешное и систематическое применение навыков и умений по данной дисциплине в соответствии с ФГОС.	отлично

Базовый уровень	Изложено правильное понимание вопроса, дано достаточно подробное описание предмета ответа, приведены и раскрыты в тезисной форме основные понятия, относящиеся к предмету ответа. Ответ отражает полное знание учебно-программного материала, систематический характер знаний по дисциплине, а также наличие базового уровня овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС	хорошо
Пороговый уровень	Ответ отражает теоретические знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии. Данная оценка может быть выставлена обучающемуся, допустившему неточности в ответе, но обладающими необходимыми знаниями для их устранения под руководством преподавателя, отмечен начальный уровень овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС	удовлетворительно
Неудовлетворительный уровень	При ответе обучающегося обнаружено отсутствие знаний, умений и навыков и/или фрагментарные знания основного учебно-программного материала.	неудовлетворительно

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением о текущей и промежуточной аттестации обучающихся в Автономной некоммерческой организации «Образовательная организация высшего образования» «Университет экономики и управления».

Вид промежуточной аттестации – экзамен.

Форма проведения промежуточной аттестации – письменный экзамен.