МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информатики и геоинформационных технологий

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5BFEE1D286E9C95A0C0D2C611E434BD5
Владелец: Степанченко Алексей Леонидович
Действителен: с 23.09.2026 по 17.12.2027

«УТВЕРЖДАЮ»
Проректор
А. Л. Степанченко
15 мая 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ) ПРОГРАММИРОВАНИЕ И ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «Б1.О.06.04 ОСНОВЫ ПРОГРАММИРОВАНИЯ»

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) подготовки: Разработка программного обеспечения для обработки и анализа пространственных данных

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2026

Срок получения образования: 4 года

Объем: в зачетных единицах: 3 з.е.
в академических часах: 108 ак.ч.

Москва, 2026

Разработчики:

Преподаватель кафедры Информатики и геоинформационных технологий Махкамбоев А. И.

Доцент кафедры Информатики и геоинформационных технологий, к.н., доцент Заблоцкий В. Р.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 922

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	05.11.2025, № 12-25

Содержание

1. Цель и задачи освоения дисциплины (модуля)

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

3. Место дисциплины в структуре ОП

4. Объем дисциплины и виды учебной работы

5. Содержание дисциплины
   5.1. Разделы, темы дисциплины и виды занятий
   5.2. Содержание разделов, тем дисциплины и формы текущего контроля

6. Оценочные материалы текущего контроля

7. Оценочные материалы промежуточной аттестации

8. Порядок проведения промежуточной аттестации

9. Материально-техническое и учебно-методическое обеспечение дисциплины
   9.1. Перечень основной и дополнительной учебной литературы
   9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
   9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
   9.4. Оборудование и технические средства обучения

10. Методические указания по освоению дисциплины (модуля)

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - овладевание основными принципами программирования на высокоуровневом языке Python и прикладными аспектами его применения.

Задачи изучения дисциплины:

- Ознакомление студентов с основами языка Python;
- Изучение базовых алгоритмов и простейших структур данных;
- Получение практики применения базовых возможностей и библиотек языка для решения прикладных задач.

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования

Знать:

ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости.

Уметь:

ОПК-7.1/Ум1 Анализировать техническое задание и выделять функциональные требования к программному продукту, выполнять декомпозицию сложных задач на подзадачи, выбирать оптимальные алгоритмические решения с учетом специфики языка Python, оценивать временную и пространственную сложность разрабатываемых алгоритмов, обосновывать выбор программных решений для конкретной практической задачи.

Владеть:

ОПК-7.1/Нв1 Навыками системного анализа предметной области и формализации требований, методами алгоритмического проектирования и выбора структур данных в Python, инструментами профилирования и оценки производительности Python-приложений, навыками технико-экономического обоснования выбранных программных решений.

ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации

Знать:

ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутые структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации О-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения.

Уметь:

ОПК-7.2/Ум1 Разрабатывать эффективные алгоритмические решения с учетом ограничений по времени и памяти, выбирать подходящие структуры данных на основе анализа требований задачи, оценивать и сравнивать эффективность различных алгоритмических подходов, реализовывать классические алгоритмы и адаптировать их под конкретные задачи, применять встроенные структуры данных Python и библиотеки collections для оптимизации решений.

Владеть:

ОПК-7.2/Нв1 Навыками разработки и реализации эффективных алгоритмов на языке Python, техниками оптимизации алгоритмов и выбора оптимальных структур данных, методами профилирования и анализа производительности алгоритмов, инструментами визуализации алгоритмов и структур данных.

ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

Знать:

ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

Уметь:

ОПК-7.3/Ум1 Писать читаемый и эффективный код на языке Python с соблюдением принципов DRY и KISS, структурировать программу с использованием функций, классов и модулей, документировать код с использованием строк документации и комментариев, аннотацию типов для повышения надежности кода, использовать инструменты автоматической проверки качества кода (pylint, flake8, black).

Владеть:

ОПК-7.3/Нв1 Навыками разработки модульных Python-приложений с четким разделением ответственности, инструментами рефакторинга и оптимизации кода, методами написания самодокументируемого кода, навыками работы с системами контроля версий для командной разработки.

ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях

Знать:

ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.

Уметь:

ОПК-7.4/Ум1 Использовать встроенные и внешние отладчики для поиска и устранения ошибок, разрабатывать unit-тесты с использованием модулей unittest и pytest, применять методы логирования для диагностики проблем в работе программы, проводить профилирование кода и выявлять узкие места производительности, обрабатывать исключения и создавать информативные сообщения об ошибках.

Владеть:

ОПК-7.4/Нв1 Навыками систематической отладки и рефакторинга программного кода, техниками написания тестируемого кода и создания тестовых сценариев, инструментами автоматизированного тестирования и непрерывной интеграции, методами оптимизации производительности Python-приложений, навыками использования систем контроля версий для отслеживания и исправления ошибок.

3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.О.06.04 «Основы программирования» относится к обязательной части образовательной программы и изучается в семестре(ах): 1.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
1 семестр	108	3	36	36	Экзамен (36)
Всего	108	3	36	36	36

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

Наименование раздела, темы	Всего	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соотнесенные с результатами освоения программы
Раздел 1. Основы программирования и среда разработки	**10**	**5**	**5**	ОПК-7.1 ОПК-7.3
Тема 1.1. Введение в язык программирования Python	4	2	2	
Тема 1.2. Интегрированная среда разработки	4	2	2	
Тема 1.3. Базовые элементы языка	2	1	1	

Раздел 2. Линейные алгоритмы и базовые операции	**6**	**3**	**3**	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 2.1. Функции ввода-вывода	2	1	1	
Тема 2.2. Арифметические операции и математические функции	2	1	1	
Тема 2.3. Генерация случайных чисел	2	1	1	
Раздел 3. Управляющие конструкции	**6**	**3**	**3**	ОПК-7.1 ОПК-7.3
Тема 3.1. Условные операторы	2	1	1	
Тема 3.2. Циклические конструкции	2	1	1	
Тема 3.3. Вложенные циклы и сложные алгоритмы	2	1	1	
Раздел 4. Обработка строк и текстовых данных	**6**	**3**	**3**	ОПК-7.3 ОПК-7.4
Тема 4.1. Строковый тип данных	2	1	1	
Тема 4.2. Операции со строками	2	1	1	
Тема 4.3. Форматирование и регулярные выражения	2	1	1	
Раздел 5. Структуры данных	**6**	**3**	**3**	ОПК-7.1 ОПК-7.3
Тема 5.1. Списки и операции над ними	2	1	1	
Тема 5.2. Кортежи и множества	2	1	1	
Тема 5.3. Словари и их применение	2	1	1	
Раздел 6. Функциональное программирование	**6**	**3**	**3**	ОПК-7.1 ОПК-7.3
Тема 6.1. Определение и вызов функций	2	1	1	
Тема 6.2. Расширенные возможности функций	2	1	1	
Тема 6.3. Функции высшего порядка и декораторы	2	1	1	
Раздел 7. Модульная организация программ	**10**	**5**	**5**	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 7.1. Работа с модулями	4	2	2	
Тема 7.2. Пакеты и их структура	4	2	2	
Тема 7.3. Стандартная библиотека	2	1	1	
Раздел 8. Объектно-ориентированное программирование	**10**	**5**	**5**	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 8.1. Основы ООП	4	2	2	
Тема 8.2. Наследование и полиморфизм	2	1	1	
Тема 8.3. Специальные методы и инкапсуляция	4	2	2	

Раздел 9. Итераторы и функциональные возможности	**6**	**3**	**3**	ОПК-7.1 ОПК-7.2
Тема 9.1. Протокол итерации	2	1	1	
Тема 9.2. Контейнерные типы	2	1	1	
Тема 9.3. Перечисления и именованные константы	2	1	1	
Раздел 10. Работа с файловой системой	**6**	**3**	**3**	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 10.1. Файловый ввод-вывод	2	1	1	
Тема 10.2. Операции с файлами и каталогами	2	1	1	
Тема 10.3. Сериализация данных	2	1	1	
Итого	**72**	**36**	**36**	

5.2. Содержание разделов, тем дисциплин и формы текущего контроля

Раздел 1. Основы программирования и среда разработки

Тема 1.1. Введение в язык программирования Python

История развития языка Python. Области применения. Сравнительный анализ с другими языками программирования. Версии Python и их особенности.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 1.2. Интегрированная среда разработки

Установка и настройка Python. Знакомство со средой IDLE. Альтернативные IDE: PyCharm, VS Code, Jupyter Notebook. Настройка рабочего окружения.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 1.3. Базовые элементы языка

Структура программы. Правила именования идентификаторов. Комментарии и документирование кода. Основные типы данных: числовые, логические, строковые. Динамическая типизация. Преобразование типов.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 2. Линейные алгоритмы и базовые операции

Тема 2.1. Функции ввода-вывода

Функция input() и ее особенности. Функция print() и форматированный вывод. Управляющие последовательности. Работа с потоками ввода-вывода.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 2.2. Арифметические операции и математические функции

Оператор присваивания. Арифметические операторы. Приоритет операций. Встроенные математические функции. Модуль math и его возможности.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 2.3. Генерация случайных чисел

Модуль random. Генерация случайных целых и вещественных чисел. Случайный выбор из последовательности. Инициализация генератора случайных чисел.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 3. Управляющие конструкции

Тема 3.1. Условные операторы

Операторы сравнения и логические операторы. Конструкция if-elif-else. Тернарный условный оператор. Вложенные условия.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 3.2. Циклические конструкции

Цикл while и его особенности. Цикл for и итерирование. Функции range() и enumerate(). Операторы break, continue, else в циклах.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 3.3. Вложенные циклы и сложные алгоритмы

Организация вложенных циклов. Оптимизация циклических конструкций. Примеры реализации классических алгоритмов.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 4. Обработка строк и текстовых данных

Тема 4.1. Строковый тип данных

Создание и инициализация строк. Специальные символы и экранирование. Строки Unicode. Неизменяемость строк.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 4.2. Операции со строками

Конкатенация и репликация строк. Индексация и срезы. Методы строк: поиск, замена, разделение, объединение. Проверка свойств строк.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 4.3. Форматирование и регулярные выражения

Методы форматирования: %, format(), f-строки. Модуль re. Синтаксис регулярных выражений. Поиск и замена по шаблону.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 5. Структуры данных

Тема 5.1. Списки и операции над ними

Создание и инициализация списков. Методы списков. Многомерные списки. Генераторы списков и списковые включения. Сортировка и поиск элементов.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 5.2. Кортежи и множества

Особенности кортежей. Распаковка кортежей. Создание множеств. Операции над множествами: объединение, пересечение, разность.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 5.3. Словари и их применение

Создание словарей. Методы словарей. Итерирование по словарям. Генераторы словарей. Вложенные словари. OrderedDict, defaultdict, Counter.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 6. Функциональное программирование

Тема 6.1. Определение и вызов функций

Синтаксис определения функций. Параметры и аргументы. Позиционные и именованные аргументы. Значения параметров по умолчанию.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 6.2. Расширенные возможности функций

Функции с переменным числом параметров (*args, **kwargs). Область видимости переменных. Замыкания. Рекурсия. Вложенные функции.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 6.3. Функции высшего порядка и декораторы

Lambda-функции. Функции map(), filter(), reduce(). Функции-генераторы и yield. Декораторы функций. Параметризованные декораторы.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 7. Модульная организация программ

Тема 7.1. Работа с модулями

Концепция модульности. Создание модулей. Инструкции import и from. Пространства имен. Повторная загрузка модулей.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 7.2. Пакеты и их структура

Создание пакетов. Файл __init__.py. Абсолютный и относительный импорт. Пути поиска модулей. Переменная sys.path.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 7.3. Стандартная библиотека

Обзор основных модулей стандартной библиотеки. Работа с датой и временем. Математические и статистические модули. Системные модули.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 8. Объектно-ориентированное программирование

Тема 8.1. Основы ООП

Концепция классов и объектов. Определение классов. Атрибуты и методы. Конструктор init и деструктор del. Атрибуты класса и экземпляра.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 8.2. Наследование и полиморфизм

Простое и множественное наследование. Переопределение методов. Функции super() и isinstance(). Абстрактные классы и методы. Миксины.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 8.3. Специальные методы и инкапсуляция

Магические методы. Перегрузка операторов. Свойства (property). Дескрипторы. Приватные и защищенные атрибуты. Декораторы классов.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 9. Итераторы и функциональные возможности

Тема 9.1. Протокол итерации

Итерируемые объекты и итераторы. Методы iter и next. Создание пользовательских итераторов. Функция iter().

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 9.2. Контейнерные типы

Протокол контейнеров. Реализация последовательностей. Методы getitem, setitem, delitem. Эмуляция словарей.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 9.3. Перечисления и именованные константы

Модуль enum. Создание перечислений. Атрибуты и методы перечислений. Функциональный API. IntEnum и Flag.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 10. Работа с файловой системой

Тема 10.1. Файловый ввод-вывод

Открытие файлов и режимы доступа. Контекстный менеджер with. Чтение и запись текстовых файлов. Работа с бинарными файлами. Кодировки.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 10.2. Операции с файлами и каталогами

Модули os и os.path. Создание, переименование, удаление файлов и каталогов. Обход дерева каталогов. Модуль pathlib.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 10.3. Сериализация данных

Модуль pickle для сохранения объектов. Форматы JSON и CSV. Модули json и csv. Работа с конфигурационными файлами. Модуль configparser.

Текущий контроль

Вид (форма) контроля
Тестирование

6. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:
- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Экзамен, 1 семестр.

7. Оценочные материалы текущего контроля

Раздел 1. Основы программирования и среда разработки
Контролируемые ИДК: ОПК-7.1 ОПК-7.3

Тема 1.1. Введение в язык программирования Python
Форма контроля/оценочное средство: Тестирование

Тема 1.2. Интегрированная среда разработки
Форма контроля/оценочное средство: Тестирование

Тема 1.3. Базовые элементы языка
Форма контроля/оценочное средство: Тестирование

Раздел 2. Линейные алгоритмы и базовые операции
Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3

Тема 2.1. Функции ввода-вывода
Форма контроля/оценочное средство: Тестирование

Тема 2.2. Арифметические операции и математические функции
Форма контроля/оценочное средство: Тестирование

Тема 2.3. Генерация случайных чисел
Форма контроля/оценочное средство: Тестирование

Раздел 3. Управляющие конструкции
Контролируемые ИДК: ОПК-7.1 ОПК-7.3

Тема 3.1. Условные операторы
Форма контроля/оценочное средство: Тестирование

Тема 3.2. Циклические конструкции
Форма контроля/оценочное средство: Тестирование

Тема 3.3. Вложенные циклы и сложные алгоритмы
Форма контроля/оценочное средство: Тестирование

Раздел 4. Обработка строк и текстовых данных
Контролируемые ИДК: ОПК-7.3 ОПК-7.4

Тема 4.1. Строковый тип данных
Форма контроля/оценочное средство: Тестирование

Тема 4.2. Операции со строками
Форма контроля/оценочное средство: Тестирование

Тема 4.3. Форматирование и регулярные выражения
Форма контроля/оценочное средство: Тестирование

Раздел 5. Структуры данных
Контролируемые ИДК: ОПК-7.1 ОПК-7.3

Тема 5.1. Списки и операции над ними
Форма контроля/оценочное средство: Тестирование

Тема 5.2. Кортежи и множества
Форма контроля/оценочное средство: Тестирование

Тема 5.3. Словари и их применение
Форма контроля/оценочное средство: Тестирование

Раздел 6. Функциональное программирование
Контролируемые ИДК: ОПК-7.1 ОПК-7.3

Тема 6.1. Определение и вызов функций
Форма контроля/оценочное средство: Тестирование

Тема 6.2. Расширенные возможности функций
Форма контроля/оценочное средство: Тестирование

Тема 6.3. Функции высшего порядка и декораторы
Форма контроля/оценочное средство: Тестирование

Раздел 7. Модульная организация программ
Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Тема 7.1. Работа с модулями
Форма контроля/оценочное средство: Тестирование

Тема 7.2. Пакеты и их структура
Форма контроля/оценочное средство: Тестирование

Тема 7.3. Стандартная библиотека
Форма контроля/оценочное средство: Тестирование

Раздел 8. Объектно-ориентированное программирование
Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3

Тема 8.1. Основы ООП
Форма контроля/оценочное средство: Тестирование

Тема 8.2. Наследование и полиморфизм
Форма контроля/оценочное средство: Тестирование

Тема 8.3. Специальные методы и инкапсуляция
Форма контроля/оценочное средство: Тестирование

Раздел 9. Итераторы и функциональные возможности
Контролируемые ИДК: ОПК-7.1 ОПК-7.2

Тема 9.1. Протокол итерации
Форма контроля/оценочное средство: Тестирование

Тема 9.2. Контейнерные типы
Форма контроля/оценочное средство: Тестирование

Тема 9.3. Перечисления и именованные константы
Форма контроля/оценочное средство: Тестирование

Раздел 10. Работа с файловой системой
Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Тема 10.1. Файловый ввод-вывод
Форма контроля/оценочное средство: Тестирование

Тема 10.2. Операции с файлами и каталогами
Форма контроля/оценочное средство: Тестирование

Тема 10.3. Сериализация данных
Форма контроля/оценочное средство: Тестирование

8. Оценочные материалы промежуточной аттестации

1 семестр, Экзамен
Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Вопросы/Задания:

1. Как называется встроенная функция Python, возвращающая одновременно индекс и значение элемента при итерировании по последовательности в цикле for?

2. Дан список chisla = [3, 7, 1, 9, 4]. Какое значение вернёт выражение sorted(chisla)[2]?

3. Функция определена как def f(*args, **kwargs): return len(args), len(kwargs). Что вернёт вызов f(1, 2, 3, x=4, y=5)?

4. Дополните предложение термином:

Специальный метод класса _____, вызываемый автоматически при создании нового экземпляра, используется для инициализации его атрибутов.

5. Какая функция позволяет проверить, что объект относится к заданному классу (или его подклассу), и часто применяется вместе с наследованием и полиморфизмом?

6. Дайте определение динамической типизации в Python и объясните, чем она отличается от статической типизации; приведите пример, иллюстрирующий эту особенность.

7. Раскройте содержание понятия области видимости (scope) переменных в Python: опишите правило LEGB и поясните, что такое замыкание (closure).

8. Опишите принцип работы функций-генераторов и оператора yield в Python, поясните их отличие от обычных функций, возвращающих значение через return.

9. Дайте определение простого и множественного наследования классов в Python, опишите назначение функции super() и поясните, для чего используются абстрактные классы.

10. Опишите механизм декораторов функций в Python: дайте определение, поясните общий принцип их работы и приведите пример типовой задачи, для решения которой применяются декораторы.

11. Расположите этапы обработки текстового файла с использованием контекстного менеджера with в правильной последовательности.

1. Автоматическое закрытие файла при выходе из блока with (в том числе при возникновении исключения).
2. Открытие файла функцией open() с указанием пути и режима доступа.
3. Построчное чтение содержимого файла или чтение его целиком.
4. Обработка прочитанных данных внутри блока with.
5. Создание объекта файла, связанного с именем, указанным после ключевого слова as.

12. Расположите этапы выполнения генератора списков (list comprehension) вида [x**2 for x in range(5) if x % 2 == 0] в правильной последовательности.

1. Проверка условия x % 2 == 0 для отбора подходящих элементов.
2. Последовательный перебор значений x из range(5).
3. Вычисление выражения x**2 для каждого отобранного элемента и добавление результата в новый список.

13. Расположите этапы поиска значения по шаблону в тексте с использованием модуля re в правильной последовательности.

1. Определение строки шаблона (регулярного выражения), описывающей искомую последовательность символов.
2. Импорт модуля re.
3. Извлечение найденного текста (группы совпадения) из результата поиска.
4. Компиляция шаблона функцией re.compile() (либо прямая передача шаблона в функцию поиска).
5. Проверка, было ли найдено совпадение.
6. Вызов функции поиска (например, re.search() или re.findall()) для применения шаблона к исходному тексту.

14. Расположите этапы создания и импорта пользовательского пакета Python в правильной последовательности.

1. Создание директории пакета, содержащей файлы модулей.
2. Импорт нужных имён из пакета инструкцией import или from ... import ... в другом файле программы.

3. Добавление в директорию пакета файла __init__.py.
4. Определение функций и классов внутри модулей пакета.

15. Расположите этапы обхода дерева каталогов файловой системы с использованием модуля os в правильной последовательности.

1. Получение списка всех файлов и подкаталогов текущего каталога при обходе.
2. Определение корневого каталога, с которого начинается обход.
3. Рекурсивный переход в каждый найденный подкаталог и повторение обхода для него.
4. Обработка (например, вывод пути или фильтрация по расширению) каждого найденного файла.

16. Установите соответствие между структурой данных Python и её ключевым свойством.

Список (list) - Неизменяемая упорядоченная последовательность элементов
Кортеж (tuple) - Изменяемая неупорядоченная коллекция уникальных элементов
Множество (set) - Изменяемая коллекция пар «ключ – значение» с уникальными ключами
Словарь (dict) - Изменяемая упорядоченная последовательность элементов, допускающая дубликаты

17. Установите соответствие между операцией над множествами и её результатом.

Объединение множеств (|) - Содержит только элементы, входящие одновременно в оба множества
Пересечение множеств (&) - Содержит элементы первого множества, не входящие во второе
Разность множеств (-) - Содержит все элементы, входящие хотя бы в одно из множеств

18. Установите соответствие между магическим методом класса и ситуацией, в которой он автоматически вызывается интерпретатором.

__init__ - Вызывается при преобразовании объекта в строку или при выводе объекта функцией print()
__str__ - Вызывается для получения длины (числа элементов) объекта
__len__ - Вызывается при сравнении двух объектов
__eq__ - Вызывается при обращении к элементу объекта по индексу или ключу в квадратных скобках
__getitem__ - Вызывается при создании нового экземпляра класса для инициализации его атрибутов

19. Установите соответствие между модулем стандартной библиотеки Python и его назначением.

Os.path - Работа с регулярными выражениями для поиска и замены текста по шаблону
re - Генерация случайных чисел и случайный выбор элементов из последовательности
random - Чтение и запись конфигурационных файлов в формате секций и пар «ключ – значение»
configparser - Работа с путями файловой системы (проверка существования, объединение и разбор путей)

20. Установите соответствие между специализированной структурой из модуля collections и её особенностью.

Counter - Словарь, автоматически создающий значение по умолчанию при обращении к отсутствующему ключу вместо возбуждения исключения
defaultdict - Словарь, явно гарантирующий сохранение порядка добавления пар «ключ – значение»
OrderedDict - Словарь для подсчёта количества вхождений каждого уникального элемента коллекции

9. Материально-техническое и учебно-методическое обеспечение дисциплины

9.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Лысаков К. Ф. Практическое программирование на Python: учебное пособие / К. Ф. Лысаков. - Практическое программирование на Python - Новосибирск: Новосибирский государственный университет, 2023. - 75 c. - 2227-8397. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/134584.html - Режим доступа: по подписке

2. Сузи Р. А. Язык программирования Python: учебное пособие / Р. А. Сузи. - Язык программирования Python - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 350 c. - 978-5-4497-3351-1. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/142310.html - Режим доступа: по подписке

Дополнительная литература

1. Амоа К. А. Разработка программных пакетов на языке Python: учебное пособие / К. А. Амоа, Н. А. Рындин, Ю. С. Скворцов. - Разработка программных пакетов на языке Python - Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2020. - 61 c. - 978-5-7731-0887-0. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/108184.html - Режим доступа: по подписке

9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

Не используются.

Ресурсы «Интернет»

1. https://sdo.miigaik.ru/ - Система дистанционного обучения МИИГАиК
2. https://docs.python.org/3/index.html - Техническая документация по языку программирования Python

9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения
(обновление производится по мере появления новых версий программы)

1. Anaconda;
2. PyCharm;
3. Microsoft Visual Studio;

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Не используется.

9.4. Оборудование и технические средства обучения

Компьютерный класс

Маркерная доска - 1 шт.
Моноблок - 96 шт.
Проектор - 8 шт.
Автоматизированное рабочее место (АРМ) - 40 шт.
Плоттер - 1 шт.
Сканер - 1 шт.
Интерактивная панель - 1 шт.

10. Методические указания по освоению дисциплины (модуля)

С целью успешного освоения данного предмета студенту необходимо выполнять домашние работы по указанным преподавателям темам и представленным на занятиях материалам. Для более успешного выполнения работы студенту предлагается использование учебно-методических материалов, соответствующих указанной теме.

Самостоятельная работа студентов описывается и регулируется методическими рекомендациями по дисциплине.

Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами

При необходимости обучения студентов-инвалидов и лиц с ограниченными возможностями здоровья аудиторные занятия могут быть заменены или дополнены изучением полнотекстовых лекций, презентаций, видео- и аудиоматериалов в электронной информационно-образовательной среде (ЭИОС) университета. Индивидуальные задания подбираются в адаптированных к ограничениям здоровья формах (письменно или устно, в форме презентаций). Выбор методов обучения зависит от их доступности для инвалидов и лиц с ограниченными возможностями здоровья.

Рабочая программа дисциплины для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.