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.06.02 ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»

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:
Старший преподаватель кафедры Информатики и
геоинформационных технологий Стрельцов А. С.

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.06.02 «Программирование на языке Python» 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	72	Зачет с оценкой
Всего	108	3	36	72	

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	**9**	**3**	**6**	ОПК-7.1 ОПК-7.3
Тема 1.1. Введение в язык программирования Python	3	1	2	
Тема 1.2. Интегрированная среда разработки	3	1	2	
Тема 1.3. Базовые элементы языка	3	1	2	

Раздел 2. Линейные алгоритмы и базовые операции	**9**	**3**	**6**	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 2.1. Функции ввода-вывода	3	1	2	
Тема 2.2. Арифметические операции и математические функции	3	1	2	
Тема 2.3. Генерация случайных чисел	3	1	2	
Раздел 3. Управляющие конструкции	**9**	**3**	**6**	ОПК-7.1 ОПК-7.3
Тема 3.1. Условные операторы	3	1	2	
Тема 3.2. Циклические конструкции	3	1	2	
Тема 3.3. Вложенные циклы и сложные алгоритмы	3	1	2	
Раздел 4. Обработка строк и текстовых данных	**9**	**3**	**6**	ОПК-7.3 ОПК-7.4
Тема 4.1. Строковый тип данных	3	1	2	
Тема 4.2. Операции со строками	3	1	2	
Тема 4.3. Форматирование и регулярные выражения	3	1	2	
Раздел 5. Структуры данных	**9**	**3**	**6**	ОПК-7.1 ОПК-7.3
Тема 5.1. Списки и операции над ними	3	1	2	
Тема 5.2. Кортежи и множества	3	1	2	
Тема 5.3. Словари и их применение	3	1	2	
Раздел 6. Функциональное программирование	**11**	**3**	**8**	ОПК-7.1 ОПК-7.3
Тема 6.1. Определение и вызов функций	5	1	4	
Тема 6.2. Расширенные возможности функций	3	1	2	
Тема 6.3. Функции высшего порядка и декораторы	3	1	2	
Раздел 7. Модульная организация программ	**15**	**5**	**10**	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 7.1. Работа с модулями	6	2	4	
Тема 7.2. Пакеты и их структура	6	2	4	
Тема 7.3. Стандартная библиотека	3	1	2	
Раздел 8. Объектно-ориентированное программирование	**17**	**5**	**12**	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 8.1. Основы ООП	6	2	4	
Тема 8.2. Наследование и полиморфизм	5	1	4	
Тема 8.3. Специальные методы и инкапсуляция	6	2	4	

Раздел 9. Итераторы и функциональные возможности	**9**	**3**	**6**	ОПК-7.1 ОПК-7.2
Тема 9.1. Протокол итерации	3	1	2	
Тема 9.2. Контейнерные типы	3	1	2	
Тема 9.3. Перечисления и именованные константы	3	1	2	
Раздел 10. Работа с файловой системой	**11**	**5**	**6**	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 10.1. Файловый ввод-вывод	3	1	2	
Тема 10.2. Операции с файлами и каталогами	5	3	2	
Тема 10.3. Сериализация данных	3	1	2	
Итого	**108**	**36**	**72**	

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: Зачет с оценкой, 1 семестр.
Зачет с оценкой проводится в устно-практической форме с использованием компьютерной техники. В начале студент получает экзаменационный билет, содержащий один теоретический вопрос и две практические задачи различного уровня сложности - базового и повышенного. После получения билета студент приступает к решению практической задачи базового уровня на компьютере в среде разработки Python, параллельно с выполнением задания готовя ответ на теоретический вопрос билета.

По завершении решения задачи базового уровня преподаватель проводит промежуточную проверку, оценивая корректность программной реализации, качество кода и соответствие решения поставленным требованиям. Затем студент устно отвечает на теоретический вопрос билета, после чего преподаватель при необходимости задает дополнительные вопросы по теме билета и смежным разделам дисциплины. После успешного прохождения промежуточной аттестации студенту выдается дополнительный бланк с практическим заданием повышенной сложности для выполнения на компьютере.

Итоговая оценка за зачет выставляется на основании комплексной проверки теоретических знаний и практических навыков программирования в соответствии с критериями оценивания, установленными в фонде оценочных средств дисциплины.

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, Зачет с оценкой
Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Вопросы/Задания:

1. Что такое PEP 8?

2. Что выведет код print(17 % 5)?

3. Как называется атрибут, значение которого является общим для всех экземпляров класса и хранится в единственном месте, в отличие от атрибута экземпляра, у которого своё значение хранится в каждом объекте отдельно?

4. Функция определена как def f(a, b=10): return a + b. Что вернёт вызов f(5)?

5. Дополните предложение термином:
Модуль _____ стандартной библиотеки Python предоставляет функции для генерации случайных чисел, случайного выбора элементов последовательности и инициализации генератора случайных чисел (seed).

6. Опишите различие позиционных и именованных аргументов функции в Python, а также поясните назначение параметров со значениями по умолчанию; приведите пример ситуации, когда порядок передачи аргументов имеет значение, а когда – нет.

7. Раскройте содержание понятия рекурсии в программировании: дайте определение, поясните обязательные компоненты корректной рекурсивной функции (базовый случай и рекурсивный переход) и опишите риск, связанный с их отсутствием.

8. Опишите механизм импорта модулей в Python: раскройте содержание понятия пространства имён (namespace), поясните различие между инструкциями import модуль и from модуль import имя.

9. Опишите, что означает конструкция if __name__ == "__main__": в Python-скрипте, и поясните, для решения какой практической задачи она обычно применяется.

10. Раскройте содержание сериализации данных в формат CSV: дайте определение, опишите структуру такого файла и поясните, для каких данных этот формат подходит лучше, а для каких – хуже по сравнению с форматом JSON.

11. Расположите этапы выполнения цикла for с функцией enumerate() при переборе списка в правильной последовательности.
1. Получение из enumerate() пары значений: индекса и соответствующего элемента списка.
2. Присваивание этих значений двум переменным цикла (например, i и элемент).
3. Создание объекта-итератора enumerate() поверх исходного списка.
4. Выполнение тела цикла с использованием текущих значений индекса и элемента.
5. Проверка, остались ли ещё элементы в списке для следующей итерации, и переход к следующей паре либо завершение цикла.

12. Расположите этапы записи и последующего чтения объекта Python в файл с использованием модуля pickle в правильной последовательности.

1. Открытие файла в бинарном режиме записи.
2. Сериализация объекта в байтовое представление функцией pickle.dump().
3. Открытие того же файла в бинарном режиме чтения.
4. Закрытие файла после записи.
5. Десериализация байтового представления обратно в объект Python функцией pickle.load().
6. Создание объекта Python, подлежащего сохранению.

13. Расположите этапы обхода списка с одновременным изменением его элементов через генератор списка (list comprehension) в правильной последовательности.

1. Последовательный перебор каждого элемента исходного списка.
2. Формирование нового списка из всех вычисленных изменённых значений.
3. Применение преобразующего выражения к текущему элементу.

14. Расположите этапы распаковки кортежа в отдельные переменные в правильной последовательности.

1. Создание кортежа с несколькими значениями.
2. Проверка соответствия количества переменных слева количеству элементов кортежа справа.
3. Запись выражения с несколькими именами переменных через запятую слева от оператора присваивания.
4. Присваивание каждой переменной соответствующего по позиции элемента кортежа.

15. Расположите этапы обработки исключения в конструкции `try/except/else/finally` при возникновении исключения, которое перехватывается вторым по счёту блоком `except`, в правильной последовательности.

1. Проверка второго блока `except` (подходит по типу).
2. Выполнение кода в блоке `try`.
3. Выполнение блока `finally`.
4. Проверка первого блока `except` (не подходит по типу).
5. Выполнение кода во втором блоке `except`.
6. Возникновение исключения.

16. Установите соответствие между интегрированной средой разработки (IDE) и её характерной особенностью

IDLE - Специализированная полнофункциональная IDE для разработки на Python с расширенными возможностями анализа кода и отладки
PyCharm - Простая среда разработки, устанавливаемая вместе со стандартным дистрибутивом Python
VS Code - Интерактивная среда, организующая код и его вывод в виде последовательности выполняемых ячеек, удобная для анализа данных
Jupyter Notebook - Универсальный, расширяемый плагинами редактор кода, поддерживающий множество языков программирования, включая Python

17. Установите соответствие между строковым методом Python и выполняемым им действием.

Split() - Заменяет все вхождения указанной подстроки на другую подстроку
join() - Возвращает индекс первого вхождения подстроки в строке или −1, если подстрока не найдена
strip() - Объединяет элементы последовательности строк в одну строку через заданный разделитель
replace() - Удаляет пробельные символы (или указанные символы) в начале и конце строки
find() - Разбивает строку на список подстрок по заданному разделителю

18. Установите соответствие между модулем стандартной библиотеки и его назначением.

Datetime - Генерация псевдослучайных чисел, выбора и перемешивания последовательностей
math - Работа с датами и временем: создание, арифметика и форматирование дат и временных интервалов
random - Математические функции и константы (тригонометрические функции, логарифмы, число π и т. п.)

19. Установите соответствие между видом наследования (или связанным механизмом ООП) и его определением.

Простое наследование - Небольшой вспомогательный класс, предназначенный для добавления конкретной дополнительной функциональности другим классам через наследование, а не для самостоятельного использования

Множественное наследование - Подкласс создаётся на основе одного родительского класса и получает все его атрибуты и методы

Миксин - Класс, задающий обязательный для реализации в подклассах интерфейс методов, но сам не предназначенный для создания экземпляров напрямую

Абстрактный класс - Подкласс создаётся сразу от нескольких родительских классов одновременно

20. Установите соответствие между функцией ввода-вывода и её особенностью.

Input() - Выводит переданные значения на экран, по умолчанию разделяя их пробелом и завершая вывод символом перевода строки

print() - Открывает файл и возвращает файловый объект для последующего чтения или записи данных.

open() - Приостанавливает выполнение программы и считывает строку текста, введённую пользователем с клавиатуры, всегда возвращая её как строку

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.

Перечень вопросов для промежуточной аттестации:

1. Что выведет следующий код: print(type(5/2))?
2. Какой оператор используется для целочисленного деления в Python?
3. Что такое PEP 8?
4. Какой метод используется для добавления элемента в конец списка?
5. Что выведет код: print("Hello"[1:4])?
6. Какое ключевое слово используется для определения функции в Python?
7. Что делает оператор continue в цикле?
8. Какой модуль используется для работы с регулярными выражениями?
9. Что означает *args в определении функции?
10. Какой метод используется для удаления пробелов в начале и конце строки?
11. Что выведет код: print(len([1, 2, [3, 4]]))?
12. Какая функция используется для генерации последовательности чисел?
13. Что такое lambda-функция в Python?
14. Какой оператор используется для проверки вхождения элемента в последовательность?
15. Что делает метод dict.get()?
16. Какое исключение возникает при делении на ноль?
17. Что означает if __name__ == "__main__":?
18. Какой метод используется для преобразования строки в список?
19. Что такое множество (set) в Python?
20. Какой оператор используется для остатка от деления?

21. Что выведет код: print(bool(""))?
22. Какой метод используется для объединения списков?
23. Что такое декоратор в Python?
24. Какой модуль используется для работы с JSON?
25. Что делает оператор with при работе с файлами?
26. Какой метод используется для получения всех ключей словаря?
27. Что означает self в методах класса?
28. Какой модуль используется для генерации случайных чисел?
29. Что выведет код: print([1, 2, 3] * 2)?
30. Какой тип данных в Python является неизменяемым (immutable)?

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.