



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: 5387FD2AF5B17FED816FDE9A1D4973B1
Владелец: Степанченко Алексей Леонидович
Действителен: с 21.07.2025 по 14.10.2026

«УТВЕРЖДАЮ»

Проректор
А. Л. Степанченко
16 февраля 2024 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
«Б1.О.06.03 ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность(профиль) подготовки: Прикладная информатика

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2024

Срок получения образования: 4 года

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

Москва, 2024

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	14.12.2023, № 7-23

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 - овладение 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 O-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.06.03 «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	34	74	Зачет с оценкой
Всего	108	3	34	74	

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, соотношенные 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	3	8	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	1	4	
Тема 10.3. Сериализация данных	3	1	2	
Итого	108	34	74	

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

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. Какой тип данных в Python является неизменяемым (immutable)?
 - A. list
 - B. dict
 - C. set
 - D. tuple
2. Что выведет следующий код: `print(type(5/2))`?
 - A. `<class 'int'>`
 - B. `<class 'float'>`
 - C. `<class 'double'>`
 - D. `<class 'number'>`
3. Какой оператор используется для целочисленного деления в Python?
 - A. /
 - B. //
 - C. %
 - D. div
4. Что такое PEP 8?
 - A. Версия Python
 - B. Руководство по стилю написания кода Python
 - C. Пакетный менеджер Python
 - D. Интегрированная среда разработки
5. Какой метод используется для добавления элемента в конец списка?
 - A. `add()`
 - B. `insert()`
 - C. `append()`
 - D. `push()`
6. Что выведет код: `print("Hello"[1:4])`?
 - A. Hel
 - B. ell
 - C. ello
 - D. Hell
7. Какое ключевое слово используется для определения функции в Python?
 - A. `function`
 - B. `def`
 - C. `func`
 - D. `define`
8. Что делает оператор `continue` в цикле?
 - A. Завершает выполнение цикла
 - B. Переходит к следующей итерации цикла
 - C. Приостанавливает выполнение цикла
 - D. Возвращает значение из цикла
9. Какой модуль используется для работы с регулярными выражениями?
 - A. `regex`
 - B. `regexp`
 - C. `re`

D. pattern

10. Что означает *args в определении функции?

- A. Обязательный аргумент
- B. Произвольное количество позиционных аргументов
- C. Именованный аргумент
- D. Аргумент по умолчанию

11. Какой метод используется для удаления пробелов в начале и конце строки?

- A. remove()
- B. delete()
- C. strip()
- D. clean()

12. Что выведет код: print(len([1, 2, [3, 4]]))?

- A. 4
- B. 3
- C. 2
- D. 5

13. Какая функция используется для генерации последовательности чисел?

- A. seq()
- B. sequence()
- C. range()
- D. numbers()

14. Что такое lambda-функция в Python?

- A. Функция для работы со списками
- B. Анонимная функция, определяемая в одну строку
- C. Функция для математических вычислений
- D. Функция высшего порядка

15. Какой оператор используется для проверки вхождения элемента в последовательность?

- A. has
- B. contains
- C. in
- D. exists

16. Что делает метод dict.get()?

- A. Создает новый словарь
- B. Возвращает значение по ключу, или значение по умолчанию если ключа нет
- C. Удаляет элемент из словаря
- D. Обновляет словарь

17. Какое исключение возникает при делении на ноль?

- A. ValueError
- B. TypeError
- C. ZeroDivisionError
- D. MathError

18. Что означает if __name__ == "__main__":?

- A. Начало главной функции
- B. Код выполнится только если файл запущен напрямую, а не импортирован
- C. Объявление главного модуля
- D. Проверка имени файла

19. Какой метод используется для преобразования строки в список?

- A. list()
- B. split()
- C. tolist()
- D. array()

20. Что такое множество (set) в Python?
- A. Упорядоченная коллекция элементов
 - B. Неупорядоченная коллекция уникальных элементов
 - C. Список с фиксированным размером
 - D. Словарь без значений
21. Какой оператор используется для остатка от деления?
- A. //
 - B. /
 - C. %
 - D. mod
22. Что выведет код: `print(bool(""))`?
- A. True
 - B. False
 - C. None
 - D. Error
23. Какой метод используется для объединения списков?
- A. merge()
 - B. concat()
 - C. extend()
 - D. combine()
24. Что такое декоратор в Python?
- A. Функция, которая модифицирует поведение другой функции
 - B. Оформление кода
 - C. Комментарий к функции
 - D. Тип переменной
25. Какой модуль используется для работы с JSON?
- A. jsonlib
 - B. json
 - C. simplejson
 - D. data
26. Что делает оператор with при работе с файлами?
- A. Открывает файл для записи
 - B. Автоматически закрывает файл после выполнения блока кода
 - C. Создает новый файл
 - D. Читает содержимое файла
27. Какой метод используется для получения всех ключей словаря?
- A. get_keys()
 - B. keys()
 - C. all_keys()
 - D. list_keys()
28. Что означает self в методах класса?
- A. Статический метод
 - B. Ссылка на текущий экземпляр класса
 - C. Приватная переменная
 - D. Конструктор класса
29. Какой модуль используется для генерации случайных чисел?
- A. rand
 - B. random
 - C. randint
 - D. rng
30. Что выведет код: `print([1, 2, 3] * 2)`?
- A. [2, 4, 6]

- B. [1, 2, 3, 1, 2, 3]
C. 6
D. [1, 2, 3, 2]

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: учебное пособие / К. Ф. Лысаков. - Практическое программирование на Python - Новосибирск: Новосибирский государственный университет, 2023. - 75 с. - 2227-8397. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/134584.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке
2. Сузи Р. А. Язык программирования Python: учебное пособие: учебное пособие / Р. А. Сузи. - Язык программирования Python - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 350 с. - 978-5-4497-3351-1. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/142310.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

Дополнительная литература

1. Амоа К. А. Разработка программных пакетов на языке Python: учебное пособие: 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 с. - 978-5-7731-0887-0. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/108184.html> (дата обращения: 16.03.2025). - 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

Учебная аудитория

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.