

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Дворникова Татьяна Александровна
Должность: врио проректора по учебной работе
Дата подписания: 25.03.2025 г.
Уникальный программный ключ:
81e4612224154ba92346993825fd4bfa9a28f49d

Федеральное государственное бюджетное учреждение
высшего образования
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»**
(Финансовый университет)

Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: arz9recw0otDu9s+VbQvRwk2PT8mAJQS
Дата: 25.03.2025 г.

Р. И. Горохова, В. И. Долгов, Е. П. Догадина
Алгоритмы и структуры данных в языке Python

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

38.03.05 - Бизнес-информатика

Образовательная программа:

Цифровые технологии в управлении земельными ресурсами и объектами
недвижимости

Профиль

Цифровые технологии в управлении земельными ресурсами и объектами
недвижимости

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол №55 от 20.05.2025)*

Одобрено

*Кафедра информационных технологий
(протокол №04 от 19.05.2025)*

Москва 2025

Содержание

Наименование разделов РПД		Стр.
1.	Наименование дисциплины	5
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	5
3.	Место дисциплины в структуре образовательной программы	6
4.	Объем дисциплины(модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	6
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	7
5.1.	Содержание дисциплины	7
5.2.	Учебно-тематический план	10
5.3.	Содержание семинаров, практических занятий	11
6.	Перечень учебно-методического обеспечения для	14

	самостоятельной работы обучающихся по дисциплине	
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	14
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	15
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	18
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	24
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	26
10.	Методические указания для обучающихся по освоению дисциплины	27
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных	27

	справочных систем	
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	28

1. Наименование дисциплины

«Алгоритмы и структуры данных в языке Python».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикатор достижения компетенции	Результаты обучения
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации	Знать: процессы сбора, обработки и интерпретации информации Уметь: описывать, описывает состав и структуру требуемых данных и информации
		Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности	Знать: принципы описания закономерностей и вариабельности Уметь: обосновывать сущность происходящего
		Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих	Знать: подходы к выполнению классификации Уметь: выполнять классификацию и оценивать полноту выполнения классификации

		групп, оценивает полноту результатов классификации, показывает прикладное назначение классификацио нных групп	
		Грамотно, логично, аргументирован о формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	Знать: методы аргументации и оценкам Уметь: формировать суждения и оценки
		Аргументирован но и логично представляет свою точку зрения посредством и на основе системного описания	Знать: подходы к описанию собственного мнения Уметь: представлять свою точку зрения

3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к «Циклу математики и информатики»

4. Объем дисциплины(модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 1 (в часах)
Общая трудоемкость дисциплины	3/108	108
Контактная работа - Аудиторные занятия	50	50
<i>Лекции</i>	16	16
<i>Семинары, практические занятия</i>	34	34
Самостоятельная работа	58	58
Вид текущего контроля	Контрольная работа;	Контрольная работа
Вид промежуточной аттестации	Зачет;	Зачет

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в программирование на Python

Общая информация о языке Python . История языка программирования, его связь с другими языками программирования, распространенность Python и основные сферы его применения. Знакомство с первыми примерами кода на Python. Философия Python.

Базовая информация о языке Python . Основные типы данных. Основные числовые типы данных и операции над ними. Математические операции над числовыми типами данных. Преобразование типов данных. Переменные и специфика их объявления. Статическая и динамическая типизация. Работа с переменными. Управление памятью и сборка мусора в Python . Именованые переменных.

Работа со строками: создание строк, специальные символы. Индексирование строк, получение срезов строк. Основные функции для работы со строками. Вывод на экран (работа с функцией print) и

форматирование строк. Различные подходы к форматированию строк, форматирование с помощью f-строк. Расширенное форматирование в Python.

Тема 2. Управляющие конструкции, списки и кортежи

Управляющие конструкции в Python . Булев тип: объявление и операции. Операции сравнения в Python . Условные операторы в Python . Реализация задачи case в Python .

Циклы в Python: while, for. Специфика циклов в Python . Функции range и enumerate и их использование в циклах.

Списки и кортежи в Python . Специфика списков и их отличие от массивов. Создание списка, оперирование вложенными списками, копирование списков, операции над списками: индексация и срезы; изменение списка; поиск, сортировка и обход; изменение списка. Кортежи в Python : синтаксис, специфика использования.

Тема 3. Словари, множества и выражения-генераторы

Словари Python . Словари: семантика, синтаксис создания, операции над словарями, перебор элементов словаря.

Множества в Python . Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python .

Выражения-генераторы в Python . Выражения-генераторы для списков: семантика и синтаксис. Пример: задача приведения списка к "плоскому" виду. Выражения-генераторы для множеств и словарей. Кейсы использования и производительность решений с использованием выражений-генераторов.

Тема 4. Функции в Python

Функции в Python : общая семантика. Создание функции и ее вызов. Расположение определений функций. Анонимные функции в Python . Необязательные параметры функций и сопоставление по ключам. Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции. Аннотации и документирование функций. Глобальные и локальные переменные.

Подход : map, filter, reduce. Реализация функций map, filter, reduce в Python. Итераторы в Python , итерируемый тип данных.

Тема 5. Работа с файлами и обработка исключительных ситуаций

Обработка исключений в Python : кейсы для использования. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert.

Работа с файлами в Python . Концепция файла в современных ОС и языках программирования. Операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов.

Сохранение объектов в файл с помощью модуля pickle и shelve . Модуль CSV .

Тема 6. Модули и пакеты

Модули и пакеты в Python : подход к структурированию программного кода с помощью модулей и пакетов. Синтаксис импортирования в Python . Создание и работа с пакетами в Python . Повторная загрузка модулей.

Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.

Тема 7. Введение в объектно-ориентированное программирование

Предпосылки и история появления ООП. Объекты и классы в ООП. Принципы и основные механизмы ООП. Логика работы абстракции, инкапсуляции, наследования и полиморфизма.

Python как объектно-ориентированный язык программирования. Базовые возможности ООП в Python : создание классов и объектов; наследование и полиморфизм; функция super(); проверка принадлежности к классу. Базовые типы в Python.

Тема 8. Объектно-ориентированное программирование в Python

Методы классов и статические переменные и методы в Python .
 Управление доступом к атрибутам класса в Python . Динамические операции с атрибутами и интроспекция в Python . Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов. Декораторы в Python: использование и создание собственных декораторов.

5.2. Учебно-тематический план

Очная форма обучения

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа		Семинары, практические занятия	Самостоятельная работа
			Общая, в т.ч.:	Лекции		
1	Введение в программирование на Python	12	6	2	4	6
2	Управляющие конструкции, списки и кортежи	12	6	2	4	6
3	Словари, множества и выражения-генераторы	14	6	2	4	8
4	Функции в Python	14	6	2	4	8
5	Работа с файлами	14	6	2	4	8

	и и обработ ка исключ ительны х ситуаци й					
6	Модули и пакеты	12	6	2	4	6
7	Введени е в объектн о- ориенти рованно е програм мирован ие	14	6	2	4	8
8	Объектн о- ориенти рованно е програм мирован ие в Python	16	8	2	6	8
	Итого	108	50	16	34	58

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).

	notebook: принципы работы и применения.	
Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Словари, множества и выражения-генераторы	Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python. Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Функции в Python	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции. Реализация функций map, filter, reduce в Python. Итераторы в Python, итерируемый тип данных. Модуль itertools.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Работа с файлами и обработка исключительных ситуаций	Исключения. Инструкция try...except...else...finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert. Работа с файлами в Python, операции с файлами:	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).

	открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов.	
Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super().	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Объектно-ориентированное программирование в Python	Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов. Декораторы в Python: использование и создание собственных декораторов.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в программирование на Python	Среда программирования. Использование документации.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Функции в Python	Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции. Функции-генераторы и выражения-генераторы в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Работа с файлами и обработка исключительных ситуаций	Работа с текстовыми, бинарными, табличными файлами. Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Модули и пакеты	Написание и запуск скриптов на Python.	Индивидуальное выполнение заданий с

	Установка модулей из глобального репозитория.	использованием Jupyter Notebook.
Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python. Реализация декораторов с параметрами в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания контрольной работы

1. В строке содержащей последовательность слов, разделенных запятыми удалить все нечетные слова. Ответ представить в виде строки. Пример: строка 'SIX,SEVEN,EIGHT,NINE,TEN' будет преобразована в: 'SIX,EIGHT,TEN'.
2. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку 'Eeny, meeny, miney, мое'.
3. Создать иерархию классов для фруктов, продающихся в магазине. Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее 2х атрибутов и 2х методов. Реализовать механизм автоматического подсчета количества всех созданных фруктов и автоматического присвоения каждому фрукту уникального идентификатора. Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать работу созданного механизма.
4. Реализовать декоратор с именем not_none, который генерирует исключительную ситуацию, если декорируемая функция вернула значения None.

5. Используя генератор словарей (и не используя код вне него) инвертировать словарь, т.е. сделать ключи словаря, его значениями и наоборот. Значения, которые в исходном словаре повторяются не добавлять в итоговый словарь. Пример: {'a':1, 'b':3, 'c':4, 'd':3} -> {1:'a', 4:'c'}.

6. Реализовать декоратор с именем print_type, выводящий на печать тип значения, возвращаемого декорируемой функцией.

Примерные вопросы контрольной работы

1. Переменные. Правила именования переменных.
2. Операции над основными числовыми типами данных.
3. Динамическая типизация в Python.
4. Преобразование типов в Python.
5. Создание строк в Python.
6. Операции над булевыми переменными.
7. Операторы сравнения и логические операторы.
8. Инструкция if...else.
9. Организация и пример цикла while в Python.
10. Организация и пример цикла for в Python.
11. Списки. Создание, основные операции.
12. Основные методы списка.
13. Кортежи. Создание, основные методы и операции.
14. Выражения-генераторы для списков в Python.
15. Словари. Создание, основные операции.
16. Операции над словарями в Python.
17. Методы для работы со словарями.
18. Множества. Создание, основные методы и операции.
19. Необязательные параметры функций в Python.

20. Заpackовка и распаковка параметров в Python.
21. Аннотации и документирование функций.
22. Исключения. Обработка исключений.
23. Пользовательские исключения.
24. Создание и вызов функции.
25. Передача аргументов функцию.
26. Функции-генераторы.
27. Лямбда-функции.
28. Модули. Инструкции import и from.
29. Базовые принципы объектно-ориентированного программирования.
30. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
31. Создание объектов в Python.
32. Управление доступом к атрибутам класса в Python.
33. Выполнение интроспекции в Python.
34. Замыкания в Python.
35. Конструктор и деструктор.
36. Наследование.
37. Абстрактные методы класса.
38. Статические методы класса.
39. Свойства класса.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2.
Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

1) Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: процессы сбора, обработки и интерпретации информации

Уметь: описывать описывает состав и структуру требуемых данных и информации

Типовые контрольные задания

1. Выполните обработку текстового файла, в котором зафиксированы факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы. На основе полученных данных составьте CSV файл, хранящий полученные результаты.

2. Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы.

2) Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы описания закономерностей и вариабельности

Уметь: обосновывать сущность происходящего

Типовые контрольные задания

1. Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями.
2. Определите количество элементов списка, значения которых совпадают с ключами заданного словаря и количество элементов списка, значения которых равны значениям в словаре.

3) Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: подходы к выполнению классификации

Уметь: выполнять классификацию и оценивать полноту выполнения классификации

Типовые контрольные задания

1. Опишите подходы используемые в Python для группировки функций, ориентированных на обработку однотипных объектов и решение близких задач.
2. Выполните классификацию с использованием бинарного дерева. Оцените количество операций и глубину дерева.

4) Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методы аргументации и оценкам

Уметь: формировать суждения и оценки

Типовые контрольные задания

1. Сформируйте собственные суждения о применимости Python для решения задач машинного обучения.
2. Выполните задачу поиска простых чисел двумя способами: с использованием циклов и с использованием рекурсивной функции. Оцените эффективность работы каждого способа.

5) Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: подходы к описанию собственного мнения

Уметь: представлять свою точку зрения

Типовые контрольные задания

1. Сформулируйте задачи по обработке массивов, которые вы сможете решить на основе языка Python
2. Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.

Примеры практико-ориентированных заданий

1. Выполните обработку текстового файла, в котором зафиксированы факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы. На основе полученных данных составьте CSV файл, хранящий полученные результаты.
2. Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы.

3. Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями.
4. Определите количество элементов списка, значения которых совпадают с ключами заданного словаря и количество элементов списка, значения которых равны значениям в словаре.
5. Опишите подходы используемые в Python для группировки функций, ориентированных на обработку однотипных объектов и решение близких задач.
6. Выполните классификацию с использованием бинарного дерева. Оцените количество операций и глубину дерева.
7. Сформируйте собственные суждения о применимости Python для решения задач машинного обучения.
8. Выполните задачу построения бинарного дерева двумя способами: на основе списков и на основе узлов и ссылок. Оцените эффективность работы каждого способа.
9. Сформулируйте задачи по обработке массивов, которые вы сможете решить на основе языка Python.
10. Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.
11. Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные.
12. Напишите скрипт на Python обрабатывающий текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV.
13. Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python.
14. Используя программу Jupyter Notebook составьте код на Python, определяющий существование треугольника по трем заданным длинам сторон. Если существует, то определить его тип. Используйте библиотеку math.

15. Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных.
16. Выберите библиотеку Python для хэширования данных. Выполните программный код быстрого поиска текстовых строк.
17. Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python.
18. Реализуйте программу на Python которой в качестве аргумента командной строки передается имя CSV-файла, в первом столбце находятся числа, которые необходимо отсортировать. Программа создает новый файл, в котором первый столбец отсортирован.

Примерные вопросы для подготовки к Зачету

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.
2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.
3. Организация ветвлений в Python. Инструкция if...else. Множественный выбор.
4. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.
5. Циклы в Python, работа и устройство цикла while. Организация бесконечного цикла, аварийное завершение цикла.
6. Списки в Python. Обращение к элементам списка и создание срезов. Обход списка и поиск элементов в списке.
7. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков.
8. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python. Операции с представлениями словарей.
9. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей.

Операции поэлементного извлечения из словаря и их использование.

10. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.

11. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.

12. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.

13. Функции стандартной библиотеки для работы с контейнерами.

14. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции.

15. Функция. Получение информации о функции. Способы передачи параметров при вызове функции.

16. Передача переменного количества параметров (именованных и не именованных) в функции Python.

17. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.

18. Вложенные функции и замыкания, специфика реализации в Python.

19. Функции высшего порядка и декораторы в Python.

20. Концепция map/filter/reduce. Реализация map/filter/reduce в Python и пример их использования.

21. Итераторы в Python: встроенные итераторы, создание собственных итераторов, типичные способы обхода итераторов и принцип их работы. Встроенные функции для работы с итераторами и возможности модуля itertools.

22. Функции-генераторы и выражения-генераторы: создание и применение в Python.

23. Синтаксис и семантика обработки исключительных ситуаций в Python. Создание пользовательских исключений и инструкция assert.

24. Базовые операции для работы с файлами в Python. Использование инструкции with ... as на примере работы с файлами.
25. CSV-файлы. Методы и функции работы с CSV-файлами.
26. Бинарные файлы. Методы и функции работы с бинарными файлами.
27. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля.
28. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.
29. Концепция класса и объекта. Принципы и механизмы ООП.
30. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python.
31. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.
32. Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python.
33. Динамические операции с атрибутами и интроспекция в Python.
34. Специальные методы для использования пользовательских классов со стандартными операторами и функциями. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов.
35. Функции высшего порядка и декораторы в Python: использование и создание собственных декораторов.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Нормативно-правовые акты:

- не предусмотрены

Основная литература:

1. Программирование. Основы Python для инженеров [Электронный ресурс] : учебное пособие для вузов / Никитина Т. П., Королев Л. В. ; Королев Л. В. 2-е изд., стер. Санкт-Петербург : Лань , 2025 156 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/454463> ISBN 978-5-507-50668-2
2. Практикум по программированию на языке Python : учебное пособие для студентов, обучающихся по направлениям "Прикладная математика и информатика", "Прикладная информатика", "Информационная безопасность", "Бизнес-информатика" (программа подготовки бакалавра) / Р.И. Горохова, Е.П. Догадина, В.И. Долгов, В.И. Макрушин ; Финуниверситет, Департамент анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Москва : Финуниверситет , 2023 1 файл (6,93 Мб) Свободный доступ из сети Интернет (чтение, печать, копирование) (дата обращения 09.10.2023) + Текст : электронный
3. Алгоритмы и структуры данных на Python : Учебное пособие / С.А. Чернышев Электрон. дан. Москва : КноРус , 2024 326 с. Режим доступа: book.ru Internet access <https://book.ru/book/949701> ISBN 978-5-406-11683-8

Дополнительная литература:

1. Математика на Python : Учебник / С.Я. Криволапов, М.Б. Хрипунова Электрон. дан. Москва : КноРус , 2025 455 с. Режим доступа: book.ru Internet access <https://book.ru/book/955467> ISBN 978-5-406-13759-8
2. Гуриков , Сергей Ростиславович Основы алгоритмизации и программирования на Python : Учебное пособие / Московский технический университет связи и информатики 1 Москва : ООО "Научно-издательский центр ИНФРА-М" , 2025 343 с. (Высшее образование) ВО - Бакалавриат <https://znanium.ru/catalog/document?id=453296> ISBN 978-5-16-020255-6 ISBN 978-5-16-102278-8 (электр. издание)
3. Разработка приложений с графическим пользовательским интерфейсом на языке Python [Электронный ресурс] : учебное пособие для вузов / Букунов С. В., Букунова О. В. ; Букунова О. В.

Санкт-Петербург : Лань , 2023 88 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/292856> ISBN 978-5-507-45191-3

4. Языки программирования. Python: решение сложных задач [Электронный ресурс] : учебное пособие для вузов / Борзунов С. В.,Кургалин С. Д. Санкт-Петербург : Лань , 2023 192 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/319394> ISBN 978-5-507-45923-0

5. Федоров , Дмитрий Юрьевич Программирование на python : учебное пособие для вузов / Д. Ю. Федоров. 6-е изд. , пер. и доп Электрон. дан. Москва : Юрайт , 2025 187 с (Высшее образование) URL: <https://urait.ru/bcode/556864> (дата обращения: 08.04.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/556864> ISBN 978-5-534-19666-5 : 829.00

6. Практические работы по информатике и основам искусственного интеллекта [Электронный ресурс] : учебное пособие для вузов / Галыгина Л. В.,Галыгина И. В. ; Галыгина И. В. Санкт-Петербург : Лань , 2023 256 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/352268> ISBN 978-5-507-47802-6

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)

2. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>

3. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>

4. •Электронно-библиотечная система BOOK.RU <http://www.book.ru>

5. •Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

6. •Электронно-библиотечная система Znanium <http://www.znaniy.ru/>

7. •Электронно-библиотечная система издательства Лань
<https://e.lanbook.com/>
8. Справка по сервисам Google - https://support.google.com/docs?hl=ru&p=about_forms#topic=1360904
9. Python Documentation <http://python.org/doc/>
10. Python Standard Library <https://docs.python.org/2/library/>

10. Методические указания для обучающихся по освоению дисциплины

Студентам при подготовке следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе".

Лекционные занятия проводятся в соответствии с тематическим планом, при изложении материала рекомендуется использовать презентации в среде PowerPoint , программный код из Jupyter Notebook и материалы по теме лекции.

В ходе интерактивных занятий следует проводить разбор конкретных примеров программного кода из Jupyter Notebook .

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

- Комплект лицензионного программного обеспечения:

1. Python 3.8
2. Kaspersky
3. Пакет офисных программ

- Современные профессиональные базы данных и информационные справочные системы:

1. Язык программирования Python 3. <https://pythonworld.ru/>
2. Размещение ЭУК: <https://www.campus.fa.ru/>
3. VK-звонки
4. VK Mail
5. Реестр УММ

- Сертифицированные программные и аппаратные средства защиты информации

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)
2. **Помещение для самостоятельной работы** обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде Университета.

3. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)