

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Дворникова Татьяна Александровна
Должность: врио проректора по учебной работе
Дата подписания: 28.02.2026 09:01:17
Уникальный программный ключ:
81e4612224154ba92346993825fd4bfa9a28f49d

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

**Кафедра анализа данных и машинного обучения
Факультета информационных технологий и анализа больших данных**

УТВЕРЖДАЮ

Проректор по учебной и
методической работе

 Е.А. Каменева

«»  2024 г.

Горохова Р.И., Макрушин С.В.

Алгоритмы и структуры данных в языке Python

Рабочая программа дисциплины

для студентов, обучающихся по направлениям подготовки:

38.03.05 – Бизнес-информатика,

21.03.02 – Землеустройство и кадастры,

ОП «Цифровые технологии в управлении земельными ресурсами
и объектами недвижимости»

Рекомендовано Ученым советом

*Факультета информационных технологий и анализа больших данных
(протокол № 44 от 21.05.2024 г.)*

*Одобрено советом Кафедры анализа данных и машинного обучения
(протокол № 01 от 06.05.2024 г.)*

Москва 2024

Содержание

1. Наименование дисциплины	2
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	2
3. Место дисциплины в структуре образовательной программы	3
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	4
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	4
5.1. Содержание дисциплины.....	4
5.2. Учебно-тематический план.....	7
5.3. Содержание семинаров, практических занятий.....	8
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	10
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	10
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю.....	11
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	13
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	19
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	20
10. Методические указания для обучающихся по освоению дисциплины	22
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	22
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	23

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	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
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. 2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности. 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. 4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	Знать: процессы сбора, обработки и интерпретации информации. Уметь: описывать описывает состав и структуру требуемых данных и информации. Знать: принципы описания закономерностей и вариабельности. Уметь: обосновывать сущность происходящего. Знать: подходы к выполнению классификации. Уметь: выполнять классификацию и оценивать полноту выполнения классификации. Знать: методы аргументации и оценок. Уметь: формировать суждения и оценки.

		5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	Знать: подходы к описанию собственного мнения. Уметь: представлять свою точку зрения.
УК-12	Способность использовать прикладное программное обеспечение при решении профессиональных задач.	1. Использует основные методы и средства получения, представления, хранения и обработки данных. 2. Демонстрирует владение профессиональными пакетами прикладных программ. 3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи. 4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.	Знать: методы и средства получения, представления, хранения и обработки данных. Уметь: получать, представлять, хранить и обрабатывать данные. Знать: основные профессиональные пакеты прикладных программ. Уметь: использовать профессиональные пакеты прикладных программ. Знать: рынок прикладного ПО. Уметь: формировать критерии выбора и осуществлять выбор ПО. Знать: прикладное ПО. Уметь: решать конкретные прикладные задачи.

3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к Циклу математики и информатики по направлениям подготовки: 38.03.05 - Бизнес-информатика, 21.03.02 Землеустройство и кадастры, ОП «Цифровые технологии в управлении земельными ресурсами и объектами недвижимости».

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

Вид учебной работы по дисциплине	Всего (в з/ед. и часах)	Семестр 1 (в часах)
Общая трудоемкость дисциплины	3 зач.ед. / 108 час.	108
Контактная работа - Аудиторные занятия	50	50
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>34</i>	<i>34</i>
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. Именованные переменные.

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. Необяза-

тельные параметры функций и сопоставление по ключам. Возвращение нескольких значений из функции. Распаковка и упаковка 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. Модуль itertools. Функции-генераторы и выражения-генераторы в Python.

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

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	*Контактная работа - Аудиторная работа			Само- стоя- тельная работа	
			Общая, в т.ч.:	Лек- ции	Семинары, практиче- ские занятия		
1.	Введение в про- граммирование на Python	12	6	2	4	6	Самостоятельное решение задач (программирова- ние), выступление на семинаре
2.	Управляющие конструкции, списки и кортежи	12	6	2	4	6	Самостоятельное решение задач (программирова- ние), выступление на семинаре
3.	Словари, множе- ства и выражения- генераторы	14	6	2	4	8	Самостоятельное решение задач (программирова- ние), выступление на семинаре
4.	Функции в Python	14	6	2	4	8	Самостоятельное решение задач (программирова- ние), выступление на семинаре
5.	Работа с файлами и обработка ис- ключительных си- туаций	14	6	2	4	8	Самостоятельное решение задач (программирова- ние), выступление на семинаре
6.	Модули и пакеты	12	6	2	4	6	Самостоятельное решение задач

							(программирование), выступление на семинаре
7.	Введение в объектно-ориентированное программирование	14	6	2	4	8	Самостоятельное решение задач (программирование), выступление на семинаре
8.	Объектно-ориентированное программирование в Python	16	8	2	6	8	Самостоятельное решение задач (программирование), выступление на семинаре
	В целом по дисциплине	108	50	16	34	58	Согласно учебному плану: контрольная работа
	Итого в %		46	32	68	54	

* объем контактной работы в очно-заочной/заочной формах обучения и индивидуальных учебных планах определяется соответствующими учебными планами. Темы, реализуемые в виде контактной работы, определяются преподавателем самостоятельно, исходя из уровня их сложности.

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
1. Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
2. Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
3. 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. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)

4. 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. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
5. 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: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
6. Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
7. Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super(). <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
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. <i>Источники: 8.1; 8.2; 8.3</i>	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
1. Введение в программирование на Python	Среда программирования. Использование документации.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
2. Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
3. 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.
4. Функции программирования 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.
5. Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
6. Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
7. Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
8. 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 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. Упаковка 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 функции.
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.

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, moy'`

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.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры анализа данных и машинного обучения Факультета информационных технологий и анализа больших данных.

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**

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотношенные с индикаторами достижения компетенции	Типовые контрольные задания
Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач. (УК-1)	1. 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 состав и структуру требуемых данных и информации.	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. 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 вариабельности.	Знать: принципы описания закономерностей и вариабельности Уметь: обосновывать сущность происходящего	Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями. 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,	Знать: подходы к выполнению классификации Уметь: выполнять классификацию и оценивать полноту выполнения классификации.	Опишите подходы используемые в Python для группировки функций, ориентированных на обработку однотипных объектов и решение близких задач. Выполните классификацию с использованием бинарного дерева. Оцените количество операций и глубину дерева.

	показывает прикладное назначение классификационных групп.		
	4.Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	Знать: методы аргументации и оценок. Уметь: формировать суждения и оценки.	Сформируйте собственные суждения о применимости Python для решения задач машинного обучения. Выполните задачу построения бинарного дерева двумя способами: на основе списков и на основе узлов и ссылок. Оцените эффективность работы каждого способа.
	5.Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	Знать: подходы к описанию собственного мнения. Уметь: представлять свою точку зрения.	Сформулируйте задачи по обработке массивов, которые вы сможете решить на основе языка Python Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.
Способность использовать прикладное программное обеспечение при решении профессиональных задач (УК-12)	1.Использует основные методы и средства получения, представления, хранения и обработки данных.	Знать: методы и средства получения, представления, хранения и обработки данных. Уметь: получать, представлять, хранить и обрабатывать данные.	Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные. 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.
	2.Демонстрирует владение профессиональными пакетами прикладных программ.	Знать: основные профессиональные пакеты прикладных программ. Уметь: использовать профессиональные пакеты прикладных программ.	Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python. 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.
	3.Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	Знать: рынок прикладного ПО. Уметь: формировать критерии выборы и осуществлять выбор ПО.	Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных. Выберите библиотеку Python для хэширования данных. Выполните программный код быстрого поиска текстовых строк.
	4.Использует прикладное программное обеспечение для решения конкретных прикладных задач.	Знать: прикладное ПО. Уметь: решать конкретные прикладные задачи.	Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python. 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

Тема 1. Введение в программирование на Python

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.

Тема 2. Управляющие конструкции, списки и кортежи

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.

Тема 3. Словари, множества и выражения-генераторы

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.

Тема 4. Функции в Python

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.

Тема 5. Работа с файлами и обработка исключительных ситуаций

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.

Тема 6. Модули и пакеты

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.

Тема 7. Введение в объектно-ориентированное программирование

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.

Тема 8. Объектно-ориентированное программирование

- 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

Основная литература:

1. 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. — ЭБ Финуниверситета. — URL: <http://elib.fa.ru/fbook/books137296.pdf> (дата обращения: 23.05.2024). — Текст : электронный.
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). — ЭБС ZNANIUM. - URL: <https://znanium.com/catalog/product/1356003> (дата обращения: 23.05.2024). — Текст : электронный.

3. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-на Дону ; Таганрог : Издательство Южного федерального университета, 2017. - 107 с. – ЭБС ZNANIUM. - URL: <https://znanium.com/catalog/product/1021664> (дата обращения: 23.05.2024). – Текст : электронный.

Дополнительная литература:

4. Северенс, Ч. Введение в программирование на Python / Ч. Северенс. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 231 с. – ЭБС Университетская библиотека ONLINE. – URL: <https://biblioclub.ru/index.php?page=book&id=429184> (дата обращения: 23.05.2024). – Текст : электронный.

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

1. Pyru 1.0.9 [Электронный ресурс]: сайт. – Режим доступа: <https://pypi.python.org/pypi/pyru>.
2. Python Data Analysis Library [Электронный ресурс]: сайт. – Режим доступа: <http://pandas.pydata.org/>.
3. Python Documentation [Электронный ресурс]: сайт. – Режим доступа: <http://python.org/doc/>.
- 4 Python Standard Library [Электронный ресурс]: сайт. – Режим доступа: <https://docs.python.org/2/library/>.
5. Scikit-learn Machine Learning in Python [Электронный ресурс]: сайт. – Режим доступа: <http://scikit-learn.org>.
6. Официальный сайт продукта <https://www.python.org/>.
7. Портал Финансового университета <http://www.fa.ru/>.
8. Каталог курсов Интернет Университета Информационных Технологий <http://www.intuit.ru/>.

9. The Python Tutorial // <https://docs.python.org/3/tutorial/index.html>.
10. The Python Standard Library // <https://docs.python.org/3/library/index.html>.
11. SciPy // <http://docs.scipy.org/doc/scipy/reference/>.
12. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>.
13. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>.
14. Электронно-библиотечная система BOOK.RU <http://www.book.ru>.
15. Электронно-библиотечная система «Университетская библиотека ОН-ЛАЙН» <http://biblioclub.ru/>.
16. Электронно-библиотечная система Znanium <http://www.znanium.com>.
17. Электронно-библиотечная система издательства «ЮРАЙТ»
<https://urait.ru/>.
18. Электронно-библиотечная система издательства Проспект
<http://ebs.prospekt.org/books>.
19. Электронно-библиотечная система издательства «Лань»
<https://e.lanbook.com/>.
20. Электронная библиотека Издательского дома «Гребенников»
<https://grebennikon.ru/>.
21. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>.
22. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>.
23. Национальная электронная библиотека <http://нэб.рф/>.
24. СПАРК <https://spark-interfax.ru/>.
25. Система Thomson Reuters Eikon.
26. Бесплатный курс «Основы Python-разработки».
<https://practicum.yandex.ru/python-free/>
27. Программирование на Python. <https://stepik.org/course/67/promo>

10. Методические указания для обучающихся по освоению дисциплины

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

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

Проведение практических занятий осуществляется в компьютерных классах и включает в себя реализацию всех этапов проектирования и реализации алгоритмов.

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

11.1. Комплект лицензионного программного обеспечения:

1. Пакет офисных программ
2. Антивирус Kaspersky
3. Дистрибутив Python Anaconda (свободно распространяемое ПО)
4. Браузер Google Chrom
5. Файловый менеджер Far

11.2. Современные профессиональные базы данных и информационные справочные системы

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru>

11.3. Сертифицированные программные и аппаратные средства защиты информации – не используются.

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

Материально-техническая база Финансового университета, необходимая для осуществления образовательного процесса по данной дисциплине, включает в себя специальные помещения для проведения лекций, семинарских занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации; помещения укомплектованы специализированной мебелью и техническими средствами обучения, необходимыми для представления информации большой аудитории.

Помещения для самостоятельной работы студентов включают в себя библиотеку с читальным залом, укомплектованную в соответствии с существующими нормами необходимой учебной и учебно-методической литературой и иными материалами; медиатеку с выходом в Интернет, компьютерные классы с возможностью работы в Интернет; аудитории для консультационной деятельности.