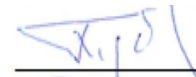


УТВЕРЖДАЮ

Директор филиала


Т.А. Хубаев
« 30 » июня 2023 г.

**Приложение к рабочей программе дисциплины
Алгоритмы и структуры данных в языке Python**

Направление подготовки 38.03.05 «Бизнес-информатика»,
образовательная программа «Цифровая трансформация управления
бизнесом»,
профиль «ИТ- менеджмент в бизнесе»

2023

(год утверждения программы)

Одобрено кафедрой «Корпоративные инфокоммуникационные системы»

Протокол от «22» июня 2023 г. № 12.

Содержание

1. Содержание семинаров, практических занятий.....	3
2. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	6
2.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	6
2.2.Перечень вопросов, заданий, тем для подготовки к текущему контролю	7
3. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	11
4. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	19
5.Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	20
6. Методические указания для обучающихся по освоению дисциплины.....	20
7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)	23
7.1 Комплект лицензионного программного обеспечения	23
7.2 Современные профессиональные базы данных, и информационные справочные системы	23
7.3 Сертифицированные программные и аппаратные средства защиты информации	23
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	24

1. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях, рекомендуемые источники из разделов 6,7 (указывается раздел и порядковый номер источника)	Формы Проведения занятий
Тема 1. Введение в программирование на Python	1. Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения Рекомендуемые источники: 4.1–4, 5.1-6	Устный опрос, дискуссия
Тема 2. Управляющие конструкции, списки и кортежи	1 Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 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. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 4. Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с

		кратким словесным описанием
Тема 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.. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 6. Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 7. Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super(). Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
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	Интерактивная форма, практикум по решению задач по тематике занятия в

	классов Рекомендуемые источники: 4.1–4, 5.1-6	индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 9. Введение в функциональное программирование	Элементы функционального программирования в Python: функции – граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 10. Функциональное программирование в Python	Реализация функций map, filter, reduce в Python. Итераторы в Python, итерируемый тип данных. Модуль itertools. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
Тема 11. Структуры данных: массивы, стеки, очереди, списки.	Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с динамическими массивами. Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием

Тема 12 Алгоритмы поиска и сортировки	Реализация на Python простых методов сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками). Реализация на Python эффективных методов сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. Рекомендуемые источники: 4.1–4, 5.1-6	Интерактивная форма, практикум по решению задач по тематике занятия в индивидуально и сдача преподавателю в конце занятия с кратким словесным описанием
---	---	---

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

2.1. 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
Тема 1. Введение в программирование на Python	Среда программирования. Использование документации.	Индивидуальное выполнение заданий с использованием IDE
Тема 2. Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Индивидуальное выполнение заданий с использованием IDE
Тема 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 IDE
Тема 4. Функции	Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции.	Индивидуальное выполнение заданий с использованием IDE
Тема 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 IDE
Тема 6. Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Индивидуальное выполнение заданий с использованием IDE

Тема 7. Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Индивидуальное выполнение заданий с использованием IDE
Тема 8. Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Индивидуальное выполнение заданий с использованием IDE
Тема 9. Введение в функциональное программирование	Реализация декораторов с параметрами в Python.	Индивидуальное выполнение заданий с использованием IDE
Тема 10. Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Индивидуальное выполнение заданий с использованием IDE
Тема 11. Структуры данных: массивы, стеки, очереди, списки.	Реализация различных вариантов связанных списков на Python.	Индивидуальное выполнение заданий с использованием IDE
Тема 12. Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Индивидуальное выполнение заданий с использованием IDE

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

Перечень примерных вопросов для поведения опроса, дискуссий

1. Python - мультипарадигмальный язык программирования
2. Итераторы для словарей
3. Ключи и значения
4. Список построений
5. Синтаксис понимания
6. Синтаксис нарезки списка
7. Изменение списка
8. Анонимные функции
9. Перебор словарей
10. Понимание понимания
11. Введение в Python

12. Строки, списки и кортежи
13. Словари и наборы
14. Условное выполнение и циклы
15. Понимание
16. Функции
17. Модули
18. Области и пространства имен
19. Промежуточный Python
20. Обработка файлов
21. Объектно-ориентированное программирование
22. Итератор, генератор, декораторы
23. Лямбда-выражения
24. Регулярное выражение
25. В чем разница между списком и кортежем?
26. Как выполняется интерполяция строк?
27. Что такое декоратор?
28. Определите класс car с двумя атрибутами: color и speed. Затем создайте экземпляр и верните speed
29. В чем разница между методами экземпляра, класса и статическими методами в Python?
30. Переменные в Python передаются по ссылке или по значению?
31. Как объединить списки в Python?
32. В чем разница между списками и массивами?
33. Как объединить два массива?
34. Назовите изменяемые и неизменяемые объекты
35. Какая разница между словарями и JSON?
36. В чем разница между pass, continue и break?
37. Преобразуйте следующий цикл for в генератор списков (list comprehension)
38. Как перевести строку в верхний/нижний регистр?

39. Приведите пример генератора словарей (dict comprehension)
40. Как выполняется обработка исключений в Python?
41. Каковы преимущества использования Python?
42. Как писать комментарии на Python?
43. Как прокомментировать несколько строк в Python?

Примерный перечень практических задач

1. Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python
2. Выберите библиотеку Python для хэширования данных в целях быстрого поиска текстовых строк
3. Реализуйте программу на Python которой в качестве аргумента командной строки передается имя CSV-файла, в первом столбце которого находятся числа, которые необходимо отсортировать. Программа создает новый файл, в котором первый столбец отсортирован
4. Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями
5. Опишите подходы используемых в Python для группировки функций ориентированных на обработку однотипных объектов и решение близких задач
6. Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.

Пример варианта контрольной работы (1 семестр)

1. Составить в интерпретаторе Python программу которая:
 - просит пользователя ввести целое число и присваивает его переменной num
 - просит пользователя ввести множитель для возведения в степень и присваивает его переменной step
 - проверяет истинность условия, что введенное пользователем целое число num меньше 100

- если это условие ИСТИННО, то необходимо возвести число num в степень step (напоминаю, возведение в степень осуществляется оператором **) и присвоить результат переменной result.

Результат вывести на печать, если результат проверки «ложь», то вывести на печать сообщение: "Введенное вами число > 100"

2. Напишите программу, находящую коэффициенты многочлена n-й степени, проходящего через n+1 точку:

Программа должна принимать на вход в первой строке натуральное число M ($M=n+1$) после этого в цикле M строк. Каждая из M строк содержит действительных числа: x_i, y_i - координаты M точек, через которые проходит график многочлена.

На выходе программа должна выдавать через пробел n+1 ($n+1=M$) коэффициент многочлена: $c_0, c_1, c_2, \dots, c_n$.

Пример варианта контрольной работы (2 семестр)

1. Напишите программу, которая будет решать систему линейных уравнений вида:

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases}$$

На вход программа должна принимать 2 строки (используйте стандартную функцию input() Python). Каждая строка содержит 3 действительных числа через пробел: a_{i1}, a_{i2}, b_i , где i - номер строки.

На выходе программа должна выдавать значения x и y в одну строку через пробел.

2. Напишите функцию f2, которая имеет 2 параметра: первый (p1) – ссылка на функцию с 1 аргументом, которая возвращает число, второй (p2) – число, которое по умолчанию равно 1. Функция f2 выводит на экран значение p1 в точке p2.

3. Создайте класс Товар, в котором имеются атрибуты:

- name – название товара (закрытый);
- count – количество;
- price – цена (закрытый).

Создайте конструктор класса.

Создайте метод discount, который уменьшает цену на количество процентов, заданных в качестве параметра метода.

Создайте экземпляр класса Товар (name – Тетради, count – 100, price – 85).

Уменьшите количество товара на 25 (используем атрибут), а цену – на 10% (вызовите метод).

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях

кафедры «Корпоративные инфокоммуникационные системы».

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

Перечень компетенций, формируемых в процессе освоения дисциплины содержится в разделе 2 «Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы».

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

Примерный перечень вопросов для подготовки к зачету

1. Базовые числовые типы int и float.
2. Встроенные функции и модули для работы с числами.
3. Генераторы списков.
4. Интерпретация.
5. Использование документации.
6. Компиляция.
7. Кортежи. Создание кортежа.
8. Литералы строк.
9. Место языка Python среди других языков программирования.
10. Методы для работы с множествами.
11. Методы для работы со словарями.
12. Методы кортежей.
13. Методы списков: добавление и удаление элементов, поиск элемента.
14. Методы списков: сортировка, преобразование списка в строку.
15. Многомерные списки.
16. Модули datetime и calendar.
17. Операторы для работы с числовыми объектами.

18. Операции над кортежами.
19. Операции над множествами: объединение, пересечение, разность, добавление и удаление элемента.
20. Операции над списками: доступ к элементу по индексу, извлечение среза.
21. Операции над списками: конкатенация, повторение, проверка вхождения.
22. Операции над строками: доступ по индексу, получение подстроки.
23. Операции над строками: конкатенация, повторение.
24. Операции над строками: повторение, проверка вхождения.
25. Особенности языка программирования Python.
26. Перебор элементов словаря.
27. Перебор элементов списка.
28. Получение текущей даты и времени.
29. Понятие языка программирования.
30. Преобразование и смешивание в выражениях значений разных типов.
31. Проверка наличия элемента во множестве.
32. Работа в интерактивном режиме интерпретатора.
33. Работа с датой и временем.
34. Работа с числами.
35. Регулярные выражения.
36. Синтаксис и семантика языка.
37. Смешанный подход.
38. Создание копии списка, полная и поверхностная копии списка.
39. Создание множества.
40. Создание словаря. Операции над словарями.
41. Создание списка.
42. Специальные символы.

43. Способы реализации языков.
44. Сравнение множеств.
45. Среда программирования.
46. Строки.
47. Установка Python.
48. Форматирование даты и времени.
49. Форматирование строк.
50. Форматы чисел.
51. Функции и методы для работы со строками.
52. Числовые литералы.

Теоретические вопросы для подготовки к экзамену

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. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.
4. Списки в Python. Обращение к элементам списка и создание срезов. Обход списка и поиск элементов в списке. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков.

Тема 3. Словари, множества и выражения-генераторы

5. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python. Операции с представлениями словарей.
6. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.
7. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.

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

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

10. Функции стандартной библиотеки для работы с контейнерами.

Тема 4. Функции

11. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции. Получение информации о функции. Способы передачи параметров при вызове функции.

12. Передача переменного количества параметров (именованных и не именованных) в функции Python. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.

Тема 5. Работа с файлами и обработка исключительных ситуаций

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

14. Базовые операции для работы с файлами в Python. Использование инструкции with ... as на примере работы с файлами.

Тема 6. Модули и пакеты

15. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

Тема 7-8. Объектно-ориентированное программирование

16. Концепция класса и объекта. Принципы и механизмы ООП.

17. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python.

18. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.

19. Методы классов и статические переменные и методы в

Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.

Тема 9-10. Функциональное программирование

20. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка элементов функционального программирования в Python.

21. Концепция «функции – граждане первого класса» в языке программирования, поддержка этой концепции в Python. Специфика лямбдафункций в Python их возможности и ограничения. Типичные сценарии использования лямбда-функций в Python.

22. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.

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

24. Функции высшего порядка и декораторы в Python.

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

26. 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.

27. Функции генераторы и выражения генераторы: создание и применение в Python.

Тема 11. Структуры данных: массивы, стеки, очереди, списки.

28. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работа с array в Python.

29. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.

30. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.

31. Связанные списки: однонаправленные и двунаправленные – принцип реализации. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве.

Тема 12. Алгоритмы поиска и сортировки

32. Алгоритм обменной сортировки, сложность сортировки и возможности по ее улучшению.

33. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.

34. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве. Сложность поиска в отсортированном и не отсортированном массиве.

35. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.

36. Алгоритм быстрой сортировки, сложность сортировки и возможности по ее улучшению.

37. Алгоритм сортировки слиянием, сложность сортировки.

Пример экзаменационного билета

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов. (20 баллов)
2. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению. (20 баллов).
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 2 объекта) и продемонстрировать работу созданного механизма (20 баллов)

Примеры оценочных средств для проверки индикаторов достижения компетенций, формируемых дисциплиной

Наименовани	Наименование	Результаты	Типовые контрольные
-------------	--------------	------------	---------------------

е компетенции	индикатора достижения компетенции	обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	задания
	1. Использует основные методы и средства получения, представления, хранения и обработки данных.	1. Знать: методы и средства получения, представления, хранения и обработки данных. Уметь: получать, представлять, хранить и обрабатывать данные.	Практико-ориентированные задания Задание: Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные.
	2. Демонстрирует владение профессиональным и пакетами прикладных программ.	2. Знать: основные профессиональные пакеты прикладных программ. Уметь: использовать профессиональные пакеты прикладных программ.	Практико-ориентированные задания Задание: Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python
	3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	3.Знать: рынок прикладного ПО. Уметь: формировать критерии выборы и осуществлять выбор ПО.	Практико-ориентированные задания Задание: Выберите библиотеку Python для хэширования данных в целях быстрого поиска текстовых строк
	4. Использует прикладное программное обеспечение для решения конкретных прикладных задач	4.Знать: прикладное ПО. Уметь: решать конкретные прикладные задачи	Практико-ориентированные задания Задание: Реализуйте программу на Python которой в качестве аргумента командной строки передается имя CSVфайла, в первом столбце которого находятся числа, которые

			необходимо отсортировать. Программа создает новый файл, в котором первый столбец отсортирован
	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации	1.Знать: процессы сбора, обработки и интерпретации информации Уметь: описывать описывает состав и структуру требуемых данных и информации	Практико-ориентированные задания Задание: 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	2.Знать: принципы описания закономерностей и вариабельности Уметь: обосновывать сущность происходящего	Практико-ориентированные задания Задание: Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями.
	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.	3.Знать: подходы к выполнению классификации Уметь: выполнять классификацию и оценивать полноту выполнения классификации.	Практико-ориентированные задания Задание: Опишите подходы используемых в Python для группировки функций ориентированных на обработку однотипных объектов и решение близких задач
	4.Грамотно, логично, аргументировано	Знать: методы аргументации и оценкам. Уметь:	Практико-ориентированные задания Задание:

	формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	формировать суждения и оценки.	Сформируйте собственные суждения о применимости Python для решения задач машинного обучения.
	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.	Практико-ориентированные задания Задание: Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.

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

Основная литература

1. - Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — ISBN 978-5-16-017142-5. - URL: <https://znanium.com/catalog/product/1356003> — Режим доступа: Электронно-библиотечная система Znanium.com — Текст: электронный.
2. - Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва: Юрайт, 2021. — 210 с. — (Высшее образование). — ISBN 978-5-534-14638-7. —URL: <https://ez.el.fa.ru:2428/bcode/478098> — Режим доступа: Электронно-библиотечная система Юрайт. — Текст: электронный.

Дополнительная литература:

3. Жуков, Р. А. Язык программирования Python: практикум: учебное пособие / Р.А. Жуков. — Москва: ИНФРА-М, 2022. — 216 с. - ISBN

978-5-16-016971-2. - URL: <https://znanium.com/catalog/product/1689648>.

– Режим доступа: Электронно-библиотечная система Znanium.com –
Текст: электронный.

4. - Чернышев, С. А. Основы программирования на python: учебное пособие для вузов / С. А. Чернышев. — Москва: Юрайт, 2021. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. —URL: <https://ez.el.fa.ru:2428/bcode/477353>. – Режим доступа: Электронно-библиотечная система Юрайт. – Текст: электронный

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

1. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
2. Электронно-библиотечная система Znanium <http://www.znanium.com>
3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.biblio-online.ru>
5. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com>

6. Методические указания для обучающихся по освоению дисциплины

Основными формами обучения студентов являются лекции, семинары и практические занятия, самостоятельная работа и консультации.

Лекции. В курсе лекций по дисциплине излагается предусмотренный рабочей программой учебный материал с учетом действующих законодательных и нормативных актов. Для лучшего усвоения изучаемой дисциплины студентам рекомендуется конспектировать лекции. Лекционный материал делится на части рассматриваемые вопросы.

Семинары и практические занятия. Важную роль в закреплении теоретических знаний играют семинары и практические занятия. Они проводятся по большинству изучаемых тем. Магистрантам рекомендуется изучая теоретический материал по соответствующей теме: изучить материалы лекций, просмотреть основную и дополнительную литературу, рекомендованную преподавателем, готовиться к опросу, проблемной дискуссии, осуществлять подготовку докладов (рефератов) и презентаций по изучаемым темам.

Консультации. Консультации студентов проводятся преподавателем в соответствии с объемами, установленными в учебном плане. На консультациях в индивидуальном порядке рассматриваются вопросы, возникшие у студентов в ходе самостоятельной работы (подготовки к экзамену). Таким образом консультации дополняют учебный процесс и даже их небольшой объем является весьма эффективным для решения возникающих вопросов. Готовясь в беседе с преподавателем на консультации студент должен уметь четко сформулировать свой вопрос, выделить его среди других, показать общее знание материала.

Самостоятельная работа. Самостоятельная работа студентов является важной частью процесса обучения. Целью самостоятельной работы студентов является формирование у студентов способности к саморазвитию, творческому применению полученных знаний, формирование умения использовать нормативную, правовую, справочную документацию и специальную литературу. Самостоятельная работа содержит в себе различные виды и формы работ. В ходе изучения дисциплины предусмотрены следующие обязательные формы:

- изучение отдельных теоретических вопросов темы, подготовка докладов (рефератов), презентаций по изучаемым темам, решение практико-ориентированных заданий;
- выполнение контрольной работы.

При освоении отдельных теоретических вопросов, подготовке

докладов (рефератов), презентаций по изучаемым темам, решении практико-ориентированных задач необходимо изучение лекций, материалов и литературы, предложенной преподавателем, Интернет-ресурсов, сайтов международных профессиональных организаций, а также осуществление работы с основной и дополнительной литературой.

Методические рекомендации для обучающихся по выполнению контрольной работы.

Контрольная работа – вид самостоятельной практической и исследовательской работы магистрантов, с целью углубления и закрепления теоретических знаний.

Целью выполнения контрольной работы является закрепление теоретических знаний и практических навыков студентов по дисциплине

Контрольная работа по дисциплине выполняется по вариантам.

В процессе выполнения контрольной работы студенту рекомендуется выполнить следующие виды работ:

- отобрать источники, собрать и проанализировать информацию по проблеме;
- систематизировать и проанализировать собранную информацию по проблеме;
- представить освещение вопроса с собственным заключением.

Контрольная работа выполняется студентом под руководством преподавателя кафедры «Корпоративные инфокоммуникационные системы» самостоятельно.

Требования к оформлению контрольной работы.

Контрольная работа выполняется на компьютере (гарнитура TimesNewRoman, шрифт 14) через 1,5 интервала с полями: верхнее, нижнее - 2; правое - 3; левое - 1,5. Отступ первой строки абзаца - 1,25. Нумерация страниц внизу в центре. Таблицы и рисунки встраиваются в текст работы или выносятся в Приложение.

Титульный лист является первой страницей и заполняется по строго определенным правилам.

Список использованных источников составляет одну из частей работы, отражающей самостоятельную творческую работу автора, позволяет судить о степени фундаментальности работы.

В тексте обязательны ссылки на литературные источники, лучше всего постраничные.

Кроме того, полезным будет использование электронных ресурсов системы «Консультант Плюс» и интернет-сайтов www.iasb.ru, www.gaap.ru.

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

7.1 Комплект лицензионного программного обеспечения

- 1) Антивирусная защита KasperskySecurity для виртуальных и облачных сред;
- 2) Windows, Microsoft Office.
- 3) IDEPython (свободно распространяемое ПО).

7.2 Современные профессиональные базы данных, и информационные справочные системы

1. Информационно-правовая система «Гарант»: <https://www.garant.ru>
2. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
3. Большая Российская энциклопедия: <https://bigenc.ru/>

7.3 Сертифицированные программные и аппаратные средства защиты информации

Не используются

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

Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой магистратуры, оснащенная оборудованием и техническими средствами обучения.

362002, Республика Северная Осетия-Алания, г. Владикавказ, ул. Молодежная, д. 7, ауд. № 47.

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол компьютерный – 18 шт.

Стол (студенческий) двухместный – 6 шт.

Стулья – 26 шт.

Доска для маркера – 1 шт.

Шкаф для документов – 1 шт.

Технические средства обучения:

Компьютер в сборе – 19 шт.,

Мультимедиа-проектор – 1 шт.,

Экран настенный – 1 шт.,

Перечень лицензионного программного обеспечения:

1) Программы для ЭВМ KasperskySecurity для виртуальных и облачных сред;

2) Продукты компании Microsoft, включая ОС WindowsOffice.

Учебная аудитория оснащена компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Финансового университета.

Помещения для самостоятельной работы обучающихся:

362002, Республика Северная Осетия-Алания, г. Владикавказ, ул. Молодежная, д.7, каб. № 72.

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол (студенческий) двухместный – 13 шт.

Стулья – 27 шт.

Доска меловая – 1 шт.

Шкаф для документов – 1 шт.

Технические средства обучения:

Компьютер в сборе – 10 шт.,

Мультимедиа-проектор – 1 шт.,

Экран настенный – 1 шт.

Перечень лицензионного программного обеспечения:

1) Программы для ЭВМ KasperskySecurity для виртуальных и облачных сред;

2) Продукты компании Microsoft, включая ОС WindowsOffice.

362002, Республика Северная Осетия-Алания, г. Владикавказ, ул. Молодежная, д.7, каб. № 55. Читальный зал.

Специализированная мебель:

Столы – 20 шт.

Стулья – 40 шт.

Шкаф для книг – 4 шт.

Стеллажи книжные – 13 шт.

Стеллажи выставочные – 4 шт.

Компьютер в сборе – 6 шт.

Телевизор – 1 шт.

Перечень лицензионного программного обеспечения:

1) Программы для ЭВМ KasperskySecurity для виртуальных и облачных сред;

2) Продукты компании Microsoft, включая ОС WindowsOffice.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и

обеспечением доступа в электронную информационно-образовательную среду Финансового университета.