

Федеральное государственное образовательное 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: 9mPk22sHWrmv1OuFwUWtlwDnG2RC+BQ
Дата: 25.09.2025 г.

М.В. Смирнов

Алгоритмы и структуры данных в языке Python

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

38.03.01 - Экономика,

Образовательная программа «Финансовые инструменты и цифровые технологии
анализа»

Профиль:

«Финансовые инструменты и цифровые технологии анализа»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2025

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	7
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	7
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	8
5.1.	Содержание дисциплины	8
5.2.	Учебно-тематический план	12
5.3.	Содержание семинаров	13
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	15
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	15
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	17
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	23
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	36
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	38
10.	Методические указания для обучающихся по освоению дисциплины	39
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	39
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	40

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
ПКН-3	Способность осуществлять сбор, обработку и статистический анализ данных, применять математические методы для решения стандартных профессиональных финансово-экономических задач, интерпретировать полученные результаты	Проводит сбор, обработку и статистический анализ данных для решения финансово-экономических задач	знать: методы сбора, обработки и статистического анализа данных, применяемые при решении финансово-экономических задач уметь: выбирать и применять методы сбора, обработки и статистического анализа данных при решении финансово-экономических задач
		Формулирует математические постановки финансово-экономических задач, переходит от экономических постановок задач к математическим моделям	знать: математические понятия, используемые при построении математических моделей экономических задач уметь: формулировать математические постановки финансово-экономических задач и переходить от экономических постановок задач к математическим моделям
		Системно подходит к выбору математических методов и информационных технологий для решения конкретных финансово-экономических задач в профессиональной области	знать: особенности математических методов и информационных технологий, используемых для решения финансово-экономических задач в профессиональной области уметь: обоснованно выбирать математические методы и информационные технологии для решения конкретных финансово-экономических задач в профессиональной области
		Анализирует результаты исследования математических моделей финансово-экономических задач и делает на их	знать: экономический смысл результатов, получаемых при математическом моделировании финансово-экономических задач, и методы их интерпретации для

		основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений	формулировки рекомендаций по принятию финансово-экономических решений уметь: анализировать результаты исследования математических моделей финансово-экономических задач и делать на их основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений
УК-10	Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации	знать: статистические методы сбора и обработки данных, методики анализа и интерпретации данных уметь: структурировать и ранжировать данные, организовать процесс сбора и обработки данных, представлять данные в графической и табличной формах
		Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности	знать: методы работы с массивами данных, метрики эффективности, математические основы анализа данных уметь: определять связи и зависимости между элементами информации, использовать современные информационные технологии аналитики
		Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп	знать: подходы к выполнению классификации уметь: выполнять классификацию и оценивать полноту выполнения классификации
		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 оценивания уметь: формировать суждения и оценки
		Аргументированно и логично	знать: основы системного описания,

		представляет свою точку зрения посредством и на основе системного описания	позволяющую аргументированно и логично представлять свою точку зрения уметь: аргументировать свою точку зрения в рассуждениях, составлять системные описания объектов, процессов, явлений
УК-11	Способность к постановке целей и задач исследований, выбору оптимальных путей и методов их достижения	Аргументированно переходит от первоначальной субъективной формулировки проблемы к целостному структурированному описанию проблемной ситуации	знать: основные формулировки проблемы, технологию структурированного описания проблемной ситуации уметь: аргументированно переходить от первоначальной субъективной формулировки проблемы к целостному структурированному описанию проблемной ситуации
		Обосновывает системную формулировку цели и постановку задачи управления	знать: основные системные формулировки целей и постановки задач управления уметь: описывать и обосновывать системную формулировку проблемы
		Взвешенно и системно подходит к анализу ситуации, формулировке критериев и условий выбора	знать: основные методы анализа ситуации, технологию формирования и применения критериев оценки эффективности выбора уметь: анализировать динамику, состав и структуру информационных ресурсов, рассчитывать показатели эффективности выбора
		Критически переосмысливает свой выбор, сопоставляя с альтернативными подходами. Оценивает последствия принимаемых решений, учитывая неочевидные цепочки «последствия последствий» («причины причин») и контурные связи	знать: основные методы оценки риска при решении задач, направленных на выработку решений уметь: применять методы обработки и анализа данных и оценивать последствия принимаемых решений
		Корректно использует процедуры целеполагания, декомпозиции и агрегирования, анализа и синтеза при решении практических задач управления и подготовке	знать: процедуры целеполагания, декомпозиции и агрегирования, анализа и синтеза уметь: применять решение практических задач управления и формировать аналитические отчеты

		аналитических отчетов	
		Логично, последовательно и убедительно излагает в отчете цели, задачи, теорию и методологию исследования, результаты и выводы	знать: принципы написания научных текстов, структуру и правила оформления текстов в соответствии с нормами ГОСТ уметь: оформлять отчеты об исследовании, излагать результаты исследования
ПКП-4	Способность применять математический аппарат при разработке вычислительных алгоритмов	Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов	знать: основные математические понятия, необходимые для разработки вычислительных алгоритмов уметь: применять математический аппарат для разработки вычислительных алгоритмов
		Разрабатывает вычислительные алгоритмы для решения прикладных задач	знать: методы разработки вычислительных алгоритмов для решения прикладных задач уметь: разрабатывать вычислительные алгоритмы для решения прикладных задач
ПКП-5	Способность применять математическое моделирование для решения прикладных задач	Демонстрирует знания в области теории и методологии математического моделирования и системного анализа процессов	знать: основные методы математического моделирования и системного анализа процессов для решения прикладных задач уметь: решать прикладные задачи с помощью методов математического моделирования
		Строит математические модели, анализирует и интерпретирует результаты моделирования для решения задач прикладного характера, в том числе в части проектирования программного обеспечения	знать: математические модели и методы проектирования программного обеспечения уметь: применять математические модели и интерпретировать результаты моделирования для решения задач прикладного характера, в том числе в части проектирования

3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к «Модулю дисциплин дополнительной квалификации - программист».

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 3 (в часах)	Семестр 4 (в часах)
Общая трудоёмкость дисциплины	8/288	144	144
Контактная работа- Аудиторные занятия	100	50	50
Лекции	32	16	16
Семинары, практические занятия	68	34	34
Самостоятельная работа	188	94	94
Вид текущего контроля	Контрольная работа, Контрольная работа	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет, Зачет	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 языке Python

Общая информация о языке Python: история, связь с другими языками программирования, распространенность и сферы применения. Философия Python.

Числовые типы данных и операции над ними. Преобразование типов данных. Переменные и специфика их объявления.

Работа со строками: создание, специальные символы, индексирование, получение срезов. Основные функции для работы со строками. Ввод и вывод данных с помощью функций *input* и *print*, форматирование строк.

Тема 2. Управляющие конструкции, списки и кортежи

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

Циклы в Python: *while*, *for*. Функции *range* и *enumerate* и их использование в циклах. Бесконечный цикл.

Списки и кортежи в Python. 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. Задача приведения списка к "плоскому" виду. Выражения-генераторы для множеств и словарей. Кейсы использования и производительность решений с использованием выражений-генераторов.

Тема 4. Функции

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.

Тема 5. Обработка исключительных ситуаций

Синтаксические ошибки, логические ошибки, исключения. Обработка исключений в Python: инструкция `try ... except ... else ... finally`. Классы встроенных исключений. Иерархия исключений. Создание пользовательских исключений. Инструкция `assert`.

Тема 6. Работа с файлами

Работа с файлами в Python. Операции с файлами: открытие/закрытие файла, чтение, запись и другие методы для работы с файлами. Инструкция `with ... as` и ее использование для файлов. Рекурсивный обход каталога файловой системы.

Сохранение объектов в файл с помощью модулей `pickle` и `shelve`. Анализ и обработка данных в формате `csv`.

Тема 7. 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. Установка и обновление пакетов, менеджер пакетов `pip`.

Тема 8. Продвинутое коллекции

Модули стандартной библиотеки Python, предоставляющие дополнительные возможности по работе с коллекциями объектов. Замороженное множество `frozenset`. Модуль `collections`: Упорядоченный словарь `OrderedDict`, цепочка словарей `ChainMap`, счетчик `Counter`, словарь по умолчанию `defaultdict`, двунаправленная очередь `deque`, именованный кортеж `namedtuple`. Модуль `enum`.

Тема 9. Объектно-ориентированное программирование в Python

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.

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.

Тема 10. Введение в функциональное программирование

Парадигмы и идиомы программирования, общая концепция функционального программирования. Функциональные языки программирования.

Функции как объекты первого класса, функции высшего порядка, замыкания, функции без побочных эффектов, рекурсия. Идиомы, распространенные в функциональных языках программирования: итераторы, последовательности, ленивые вычисления.

Тема 11. Функциональное программирование в Python

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.

Декораторы в Python: использование и создание собственных декораторов.

Тема 12. Структуры данных: массивы, стеки, очереди, списки.

Введение в анализ сложности алгоритмов.

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.

Тема 13. Алгоритмы поиска и сортировки

Поиск в списках/массивах, бинарный поиск. Сортировка и ее использование в прикладных задачах.

Простые методы сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками).

Эффективные методы сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. Сравнение различных сортировок.

Тема 14. Структуры данных: деревья

Деревья, бинарные деревья. Использование бинарных деревьев в прикладных задачах: представление выражения (предложения) в виде дерева. Обход бинарных деревьев. Двоичное дерево поиска.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Введение в программирование на языке Python	18	6	2	4	12
2	Управляющие конструкции, списки и кортежи	18	6	2	4	12
3	Словари, множества и выражения-генераторы	18	6	2	4	12
4	Функции	18	6	2	4	12
5	Обработка исключительных ситуаций	20	6	2	4	14
6	Работа с файлами	24	10	4	6	14
7	Модули и пакеты	24	10	4	6	14
8	Продвинутое коллекции	22	8	2	6	14
9	Объектно-ориентированное программирование в Python	22	8	2	6	14
10	Введение в функциональное программирование	20	6	2	4	14
11	Функциональное программирование в Python	20	6	2	4	14
12	Структуры данных: массивы, стеки, очереди, списки.	22	8	2	6	14
13	Алгоритмы поиска и сортировки	22	8	2	6	14
14	Структуры данных: деревья	20	6	2	4	14
	Итого	288	100	32	68	188

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в программирование на языке Python	Среда программирования на языке Python. Установка дистрибутива Anaconda. Программирование в командном и интерактивном режимах. Интерактивная оболочка iPython notebook. Строки и числа. Математические операции с числами. Ввод-вывод в языке Python.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Управляющие конструкции, списки и кортежи	Ветвление, циклы for и while. Создание и обработка списков. Методы работы со списками. Индексация и срезы. Многомерные списки. Кортежи. Особенности использования изменяемых и неизменяемых коллекций языка Python.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Словари, множества и выражения-генераторы	Создание словаря. Операции над словарями. Методы для работы со словарями. Объекты представления словаря ключи, значения, пары (ключ, значение). Создание множества. Методы работы с множествами. Выражения-генераторы.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Позиционные и именованные аргументы. Лямбда-функции.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Обработка исключительных ситуаций	Исключения. Инструкция try...except...else...finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Работа с файлами	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.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов. Обработка значения переменной __name__.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с

		преподавателем.
Продвинутые коллекции	Замороженные множества frozenset. Модуль collections: упорядоченный словарь OrderedDict, цепочка словарей ChainMap, счетчик Counter, словарь по умолчанию defaultdict, двунаправленная очередь deque, именованный кортеж namedtuple.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Объектно-ориентированное программирование в Python	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super(). Использование специальных методов для расширенной функциональности пользовательских классов. Методы классов, статические переменные и методы в Python.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Введение в функциональное программирование	Элементы функционального программирования в Python: функции как объекты первого класса. Глобальные и локальные переменные. Вложенные функции и замыкания, декораторы.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Функциональное программирование в Python	Реализация функций map, filter, reduce в Python. Итераторы в Python, итерируемый тип данных. Модуль itertools.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
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.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
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.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.
Структуры данных: деревья	Реализация на Python деревьев, бинарных деревьев. Использование бинарных деревьев в прикладных задачах: представление выражения (предложения) в виде дерева.	Индивидуальное выполнение заданий на компьютере, разбор результатов выполнения заданий с преподавателем.

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 языке Python	Среда программирования. Поиск сведений в документации языка Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, операции над списками.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Словари, множества и выражения-генераторы	Генераторы для словарей и множеств: практическое использование.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Функции	Возвращение нескольких значений из функции. Распаковка и упаковка параметров функции.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Обработка исключительных ситуаций	Иерархия классов исключений. Инструкции raise и assert.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Работа с файлами	Сохранение объектов в файл с помощью модулей pickle и shelve. Модуль csv.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Модули и пакеты	Установка и обновление модулей с помощью менеджера пакетов pip.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Продвинутое	Именованный кортеж namedtuple, словарь	Индивидуальное

коллекции	defaultdict.	выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Объектно-ориентированное программирование в Python	Проверка принадлежности к классу. Атрибуты <code>__class__</code> и <code>__dict__</code> объектов языка Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Введение в функциональное программирование	Реализация декораторов с параметрами в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Структуры данных: массивы, стеки, очереди, списки.	Реализация различных вариантов связанных списков на языке Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.
Структуры данных: деревья	Реализация обхода бинарных деревьев.	Индивидуальное выполнение заданий с использованием Jupyter Notebook и Python IDLE.

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

3 семестр

Примерные вопросы контрольной работы

1. Операции над основными числовыми типами данных.
2. Операции над булевыми переменными.
3. Динамическая типизация в Python.
4. Преобразование типов в Python.
5. Создание строк в Python.
6. Организация и пример цикла *while* в Python. Бесконечный цикл *while*.
7. Организация и пример цикла *for* в Python, применение *range* и *enumerate* в цикле *for*.
8. Операции над словарями в Python.
9. Выражения-генераторы для списков в Python.
10. Организация условий в языке Python, тернарный условный оператор.
11. Необязательные параметры функций в Python.
12. Заpackовка и распаковка параметров функции в Python.
13. Аннотация и документирование функций.
14. Создание и вызов функции.
15. Передача аргументов функции.
16. Создание объектов в Python.
17. Выполнение интроспекции в Python.
18. Компилятор и интерпретатор. Достоинства и недостатки.
19. Краткая характеристика основных классов языков программирования.

20. Встроенные числовые типы языка Python.
21. Списки. Создание, основные операции.
22. Основные методы списка.
23. Кортежи. Создание, основные методы и операции.
24. Словари. Создание, основные операции.
25. Методы для работы со словарями.
26. Множества. Создание, основные методы и операции.
27. Переменные. Правила именования переменных.
28. Динамическая типизация.
29. Операторы сравнения и логические операторы.
30. Модули. Инструкции *import* и *from*.
31. Обработка исключительных ситуаций в Python.
32. Создание пользовательских исключений и инструкция *assert*.

Примерные задания контрольной работы

1. Имеются два списка целых чисел, возможно разной длины, упорядоченные по возрастанию. Получите новый список, содержащий все элементы исходных списков, в котором элементы также упорядочены в порядке возрастания, не используя метод сортировки списка.
2. Какую функцию надо применить в сочетании с функцией *map*, чтобы список слов ["Я", "студент", "Финансового", "университета"] преобразовать в список чисел [1, 7, 11, 12]? Напишите программу, реализующую эту задачу.
3. С помощью функций *filter* и *lambda* в списке [4, 8, 15, 16, 23, 42] оставьте только элементы со значениями выше среднего.
4. Найдите все числа ряда Фибоначчи меньше заданного натурального числа *n*, используя рекурсивную функцию.
5. Имеется список городов вида [{"город", "год основания", "население"}, {"Гомель", 1142, 504, }, {"Бобруйск", 1387, 210}, {"Лида", 1323, 103}]. Выполните

задания: а) создайте новый список, в котором первый вложенный список остаётся на своём месте, а оставшиеся вложенные списки отсортированы по населению; б) создайте словарь, в котором ключами являются элементы первого вложенного списка, а значениями элементы остальных вложенных списков; в) распечатайте полученный в результате выполнения пункта б) словарь в виде таблицы, в которой названиями столбцов являются ключи, а значениями – значения словаря.

6. Участники олимпиады решали 3 задачи. Известны фамилии тех, кто решил первую, вторую и третью задачи (для каждой задачи отдельный список). Найдите и распечатайте фамилии тех, кто а) решил хотя бы одну задачу (любую); б) решил все задачи; в) решил ровно одну задачу (любую); г) решил ровно две задачи (любые); д) решил не больше двух задач (любых).

7. Имеются данные в формате JSON о множестве рецептов блюд национальной кухни. Каждый рецепт содержит множество ингредиентов. Найдите ингредиенты, входящие в состав а) одного блюда; б) одной кухни; в) максимального числа блюд.

8. Задать два двумерных массива *ar1* и *ar2* размерности (4, 7), состоящих из случайных целых чисел в пределах от 0 до 10. Построить двумерный массив размерности (4, 7), каждый элемент которого представляет собой максимум (минимум, среднее) из двух значений, находящихся на аналогичной позиции в массивах *ar1*, *ar2*.

9. Сгенерировать двумерный массив *arr* размерности (4, 7), состоящий из случайных действительных чисел, равномерно распределенных в диапазоне от 0 до 20. Нормализовать значения массива так, что после нормализации максимальный элемент массива будет равен 1.0, минимальный 0.0. Значения округлить до четырех десятичных знаков.

10. Имеются данные в формате CSV о финансово-экономической деятельности группы предприятий и содержащие сведения о типе выпускаемой продукции, наименовании продукции, цене, себестоимости, сегменте рынка, прибыли, дате поставки, размере скидки, стране. Найдите все сделки определенного производителя, поставленные в определенную страну при условии, что размер скидки ниже средней в соответствующем сегменте рынка.

11. Имеются данные в формате *pickle* о финансово-экономической деятельности группы предприятий: тип выпускаемой продукции, наименование продукции,

цена, себестоимость, сегмент рынка, прибыль, дата поставки, размер скидки, страна. Напишите программу, результатом которой является сводная таблица, в которой расположите наименования производителей по строкам, а сегменты рынка по столбцам, поместите в область значений среднюю прибыль каждого производителя в каждом сегменте. Рассчитайте общее среднее значение прибыли по каждому производителю и по каждому сегменту.

12. В квадратной матрице, элементами которой являются случайные целые числа, размера $n \times n$ найти среднее арифметическое положительных элементов главной диагонали и среднее арифметическое отрицательных элементов побочной диагонали. Матрицу вывести на экран.

4 семестр

Примерные вопросы контрольной работы

1. Базовые принципы объектно-ориентированного программирования.
2. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
3. Конструктор и деструктор.
4. Наследование
5. Статические методы класса.
6. Свойства класса.
7. Управление доступом к атрибутам класса в Python.
8. Алгоритм обменной сортировки.
9. Алгоритм сортировки выбором.
10. Алгоритм сортировки вставками.
11. Алгоритм быстрого поиска в отсортированном массиве.
12. Алгоритм сортировки Шелла.
13. Алгоритм быстрой сортировки.
14. Алгоритм сортировки слиянием.

15. Структуры данных: деревья.
16. Различные реализации рекурсивного обхода двоичных деревьев.
17. Функции-генераторы.
18. Лямбда-функции.
19. Итераторы, функция `next`, обработка исключения *StopIteration*.
20. Использование функций *map/filter/reduce*.

Примерные задания контрольной работы

1. Опишите класс *Car*, заданный маркой, моделью, годом выпуска и пробегом. Включите в описание класса методы: вывода информации о машине на экран, проверки, нужно ли произвести техническое обслуживание (пробег больше 10 000 км с последнего ТО), и свойство, позволяющее установить тип топлива (бензин, дизель и т. п.)
2. Создать иерархию классов для фруктов, продающихся в магазине. Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее 2-х атрибутов и 2-х 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.
3. Создать класс *Стек*. Создать экземпляр класса. Развернуть каждый второй элемент стека в обратном порядке.
4. Реализовать декоратор с именем *not_zero*, который генерирует исключительную ситуацию если декорируемая функция вернула значение ноль.
5. Написать функцию, которая принимает на вход список городов и сортирует его по населению с помощью алгоритма сортировки вставками. Функция должна возвращать отсортированный список.
6. Дана строка-предложение. Зашифровать ее, поместив вначале все символы, расположенные на четных позициях строки, а затем, в обратном порядке, все символы, расположенные на нечетных позициях (например, строка «Программа» превратится в «ргамамроП»).

7. Дана матрица размера $m \times n$. Удалить столбец, содержащий максимальный элемент матрицы.
8. Дана матрица размера $m \times n$. Поменять местами столбец с номером 1 и последний из столбцов, содержащих только положительные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.
9. Дан список слов. При помощи механизма *map/filter/reduce* рассчитать остаток от деления на 7 для каждой длины слова из списка и получить произведение тех остатков, величина которых больше 4.
10. Создать рекурсивную функцию *binary_to_decimal*, которая переводит число из двоичной системы счисления в десятичную.
11. Напишите функцию *reserve*, которая, получив на вход строку только из строчных (a..z) или прописных (A..Z) букв, возвращает строку согласно примерам: *reserve("abcd")* -> "A-Bb-Ccc-Dddd"; *reserve("RqaEzty")* -> "R-Qq-Aaa-Eeee-Zzzzz-Tttttt-Yyyyyyy"; *reserve("cwAt")* -> "C-Ww-Aaa-Tttt"
12. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа, рассчитанному с конца исходной строки. Пример: 'abcd' -> ['aaaa', 'bbb', 'cc', 'd'].

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

ПКН-3 Способность осуществлять сбор, обработку и статистический анализ данных, применять математические методы для решения стандартных профессиональных финансово-экономических задач, интерпретировать полученные результаты

1) Проводит сбор, обработку и статистический анализ данных для решения финансово-экономических задач

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

Знать: методы сбора, обработки и статистического анализа данных, применяемые при решении финансово-экономических задач

Уметь: выбирать и применять методы сбора, обработки и статистического анализа данных при решении финансово-экономических задач

Типовые контрольные задания

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

2) Формулирует математические постановки финансово-экономических задач, переходит от экономических постановок задач к математическим моделям

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

Знать: математические понятия, используемые при построении математических моделей экономических задач

Уметь: формулировать математические постановки финансово-экономических задач и переходить от экономических постановок задач к математическим моделям

Типовые контрольные задания

Создайте класс Order (Заказ), у которого есть свойства code (код_товара), price (цена), count (количество) и методы __init__ и __str__. Создайте два класса-потомка: Opt (Опт) и Retail (Розница). В этих классах создайте методы __init__, __str__ и summa (сумма_заказа), позволяющие узнать стоимость заказа.

3) Системно подходит к выбору математических методов и информационных технологий для решения конкретных финансово-экономических задач в профессиональной области

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

Знать: особенности математических методов и информационных технологий, используемых для решения финансово-экономических задач в профессиональной области

Уметь: обоснованно выбирать математические методы и информационные технологии для решения конкретных финансово-экономических задач в профессиональной области

Типовые контрольные задания

Опишите класс Employee, заданный фамилией, именем, должностью и зарплатой. Включите в описание класса методы: вывода информации о сотруднике на экран, проверки, является ли зарплата высокой (больше 100 000 рублей), и свойство, позволяющее установить стаж работы. Создайте магические методы сравнения сотрудников по стажу.

4) Анализирует результаты исследования математических моделей финансово-экономических задач и делает на их основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений

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

Знать: экономический смысл результатов, получаемых при математическом моделировании финансово-экономических задач, и методы их интерпретации для формулировки рекомендаций по принятию финансово-экономических решений

Уметь: анализировать результаты исследования математических моделей финансово-экономических задач и делать на их основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений

Типовые контрольные задания

Создайте класс Bank (Банк) с атрибутами название и список счетов. Каждый счет представлен классом Account (Счет) с атрибутами номер и баланс. Напишите методы для добавления счета в банк, удаления счета из банка и вычисления общего баланса всех счетов в банке. Используйте магический метод __len__ для определения количества счетов в банке.

УК-10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать 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: статистические методы сбора и обработки данных, методики анализа и интерпретации данных

Уметь: структурировать и ранжировать данные, организовать процесс сбора и обработки данных, представлять данные в графической и табличной формах

Типовые контрольные задания

Выполните обработку текстового документа, 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) Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности

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

Знать: методы работы с массивами данных, метрики эффективности, математические основы анализа данных

Уметь: определять связи и зависимости между элементами информации, использовать современные информационные технологии аналитики

Типовые контрольные задания

Сгенерировать двумерный массив агт размерности (4, 7), состоящий из случайных действительных чисел, равномерно распределенных в диапазоне от 0 до 20. Нормализовать значения массива так, что после нормализации максимальный элемент массива будет равен 1.0, минимальный 0.0. Значения округлить до двух десятичных знаков.

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

Файл `archives.pickle` содержит данные портала открытых данных Правительства Москвы о расположенных в Москве архивах. Источник: <https://data.mos.ru/opendata/1930?isDynamic=false> . С помощью модуля `pickle` загрузите эти данные в программу на Python и найдите распределение числа архивов по районам Москвы.

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 оценивания

Уметь: формировать суждения и оценки

Типовые контрольные задания

Напишите функцию Python, чтобы проверить, является ли число «совершенным». Совершенное число – это целое положительное число, равное сумме его собственных положительных делителей, то есть сумме его положительных делителей, исключая само число.

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: основы системного описания, позволяющую аргументированно и логично представлять свою точку зрения

Уметь: аргументировать свою точку зрения в рассуждениях, составлять системные описания объектов, процессов, явлений

Типовые контрольные задания

Задать два двумерных массива *ar1* и *ar2* размерности (*n*, *m*), состоящих из случайных целых чисел в пределах от 0 до 10. Построить двумерный массив *ar3* размерности (*n*, *m*), каждый элемент которого представляет собой максимум (минимум, среднее) из двух значений, находящихся на аналогичной позиции в массивах *ar1*, *ar2*.

УК-11 Способность к постановке целей и задач исследований, выбору оптимальных путей и методов их достижения

1) Аргументированно переходит от первоначальной субъективной формулировки проблемы к целостному структурированному описанию проблемной ситуации

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

Знать: основные формулировки проблемы, технологию структурированного описания проблемной ситуации

Уметь: аргументированно переходить от первоначальной субъективной формулировки проблемы к целостному структурированному описанию проблемной ситуации

Типовые контрольные задания

Создать класс *ПрограммноеОбеспечение* с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на текущую дату). Создайте дочерние классы *Свободное* (название, производитель), *УсловноБесплатное* (название, производитель, дата установки, срок бесплатного использования), *Коммерческое* (название, производитель, цена, дата установки, срок использования) со своими методами вывода информации на экран и определения возможности использования на текущую дату.

2) Обосновывает системную формулировку цели и постановку задачи управления

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

Знать: основные системные формулировки целей и постановки задач управления

Уметь: описывать и обосновывать системную формулировку проблемы

Типовые контрольные задания

Создать класс *Стек*, который будет хранить только элементы типа *str* и отсортированные по алфавиту. При добавлении элемента, если он не является строкой, то он не должен добавляться. При получении элементов из стека они должны быть отсортированы по алфавиту.

3) Взвешенно и системно подходит к анализу ситуации, формулировке критериев и условий выбора

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

Знать: основные методы анализа ситуации, технологию формирования и применения критериев оценки эффективности выбора

Уметь: анализировать динамику, состав и структуру информационных ресурсов, рассчитывать показатели эффективности выбора

Типовые контрольные задания

Дан массив целых чисел размерности n , заданных случайным образом из интервала от -20 до 20. Если сумма отрицательных элементов по модулю превышает сумму положительных, то отсортировать массив по возрастанию, иначе – по убыванию.

4) Критически переосмысливает свой выбор, сопоставляя с альтернативными подходами. Оценивает последствия принимаемых решений, учитывая неочевидные цепочки «последствия последствий» («причины причин») и контурные связи

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

Знать: основные методы оценки риска при решении задач, направленных на выработку решений

Уметь: применять методы обработки и анализа данных и оценивать последствия принимаемых решений

Типовые контрольные задания

Реализовать функцию, которая находит минимальный элемент в однонаправленном связном списке.

5) Корректно использует процедуры целеполагания, декомпозиции и агрегирования, анализа и синтеза при решении практических задач управления и подготовке аналитических отчетов

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

Знать: процедуры целеполагания, декомпозиции и агрегирования, анализа и синтеза

Уметь: применять решение практических задач управления и формировать аналитические отчеты

Типовые контрольные задания

Создать функцию, получающую на вход список строк, чисел и объектов логического типа и возвращающую список, состоящий только из чисел и объектов логического типа.

6) Логично, последовательно и убедительно излагает в отчете цели, задачи, теорию и методологию исследования, результаты и выводы

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

Знать: принципы написания научных текстов, структуру и правила оформления текстов в соответствии с нормами ГОСТ

Уметь: оформлять отчеты об исследовании, излагать результаты исследования

Типовые контрольные задания

Создать декоратор *deco(dlina, fill)* с параметрами *dlina* и *fill*. Декоратор превращает результат декорируемой функции в словарь, состоящий из *dlina* элементов в формате {порядковый номер элемента: элемент}. Если исходная функция возвращает меньше заданного количества элементов, то оставшиеся места заполняются значениями {порядковый номер элемента: *fill*}, в случае, если количество возвращаемых элементов больше *dlina*, то хвост последовательности отбрасывается.

ПКП-4 Способность применять математический аппарат при разработке вычислительных алгоритмов

1) Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов

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

Знать: основные математические понятия, необходимые для разработки вычислительных алгоритмов

Уметь: применять математический аппарат для разработки вычислительных алгоритмов

Типовые контрольные задания

Напишите рекурсивную функцию для реализации алгоритма поиска наибольшего общего делителя двух чисел.

2) Разрабатывает вычислительные алгоритмы для решения прикладных задач

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

Знать: методы разработки вычислительных алгоритмов для решения прикладных задач

Уметь: разрабатывать вычислительные алгоритмы для решения прикладных задач

Типовые контрольные задания

Напишите функцию, выполняющую рекурсивный обход каталогов файловой системы.

ПКП-5 Способность применять математическое моделирование для решения прикладных задач

1) Демонстрирует знания в области теории и методологии математического моделирования и системного анализа процессов

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

Знать: основные методы математического моделирования и системного анализа процессов для решения прикладных задач

Уметь: решать прикладные задачи с помощью методов математического моделирования

Типовые контрольные задания

Имеется массив (длиной не менее 3, но может быть и очень большим), содержащий целые числа. Массив либо полностью состоит из нечётных чисел, либо полностью из чётных чисел, за исключением одного числа p . Напишите функцию, которая принимает массив в качестве аргумента и возвращает этот «выброс» p .

2) Строит математические модели, анализирует и интерпретирует результаты моделирования для решения задач прикладного характера, в том числе в части проектирования программного обеспечения

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

Знать: математические модели и методы проектирования программного обеспечения

Уметь: применять математические модели и интерпретировать результаты моделирования для решения задач прикладного характера, в том числе в части проектирования

Типовые контрольные задания

Число Армстронга – это положительное число, представляющее собой сумму своих цифр, каждая из которых возведена в степень, соответствующую количеству цифр в данной системе счисления. Написать функцию `armstrong`, которая возвращает логическое значение `True` или `False` в зависимости от того, является ли заданное число числом Армстронга. Пример: `armstrong(153)=>True`, `armstrong(9474)=>True`.

Примеры практико-ориентированных заданий

Третий семестр

1. На основе данных столбца csv-файла создать очередь с помощью `collections.deque`. Передвинуть элементы очереди по кругу на 3 шага вправо, распечатайте результат.
2. Написать функцию, которая возвращает факториал числа при условии, что аргумент является целым и положительным числом. Если это не так, создать и обработать исключение.
3. Написать функцию сложения двух положительных чисел. Вызвать и обработать исключение `AssertionError` при вводе пользователем отрицательных чисел.
4. Написать функцию, которая принимает имя файла. Если файла нет, то возвращает текст "404. File (filename) not found", иначе возвращает содержимое файла.
5. Написать функцию вычисления $2/s$. На вход функции поступает переменная `s`. Перехватить все исключительные ситуации.
6. Написать функцию вычисления корней квадратного уравнения. На вход функции поступают коэффициенты `a`, `b`, `c`. С помощью команды `try-except` обработать случай отрицательного дискриминанта. Функция возвращает кортеж корней квадратного уравнения, если дискриминант неотрицательный. Если дискриминант отрицательный, функция возвращает `None` и распечатывает фразу «Отрицательный дискриминант».
7. Написать функцию подсчета суммы главной диагонали. Если матрица не является квадратной, должна вызываться ошибка.
8. Описать функцию `ring_s(r1, r2)`, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами `r1` и `r2`.
9. Описать функцию `triangle_p(b, h)`, находящую периметр равнобедренного треугольника по его основанию `b` и высоте `h`, проведенной к основанию. С помощью этой функции найти периметр треугольника, для которого даны основание и высота.

10. Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими) и другими знаками препинания. Вывести строку, содержащую эти же слова, разделенные одним пробелом и расположенные в обратном порядке и без знаков препинания: дефисов, запятых точек и других.
11. Дана строка, содержащая полный путь к файлу. Извлечь из этой строки только расширение файла без имени и точки.
12. Получите строку из заданной, в которой все буквы заменены на следующую букву в алфавите (например, 'a' → 'b', 'z' → 'a').
13. Вводятся координаты точки (x, y). Определить попадает ли точка в область, заданную условиями: $0 < y < 5\sin(x)$, $0 < x < \pi$.
14. Подсчитайте сумму двузначных чисел, сумма цифр которых не превышает 10.
15. Готовясь к соревнованиям, лыжник в первый день пробежал 10 км, затем каждый день увеличивал расстояние на 10%. Сколько километров пробежал он за неделю тренировок? На какой день он пробежал больше 15 км?
16. Описать функцию *compress_str*, выполняющую сжатие строки *string* по следующему правилу: каждая подстрока строки *string*, состоящая из более четырёх одинаковых символов *w*, заменяется текстом вида "*w{}*", где *k* — количество символов "*w*" (предполагается, что строка *string* не содержит фигурных скобок «{» и «}»). Например, для строки *string* = 'bbbccccce' функция вернет строку *bbbc{5}e*.

Четвертый семестр

1. Создать рекурсивную функцию *count_vowels*, находящую количество гласных букв в строке. С помощью этой функции найти количество гласных букв в заданной строке.
2. С помощью сортировки выбором (Selection Sort) отсортировать массив строк по убыванию. Результат вывести на экран.
3. С помощью сортировки вставками (Insertion Sort) отсортировать массив строк по длине. Результат вывести на экран.
4. С помощью сортировки Шелла отсортировать массив строк по числу гласных. Результат вывести на экран.
5. Написать функцию, создающую структуру бинарного дерева из математической формулы.
6. Напишите программу для создания бинарного дерева поиска, используя массив элементов, где элементы отсортированы по возрастанию.

7. Напишите функцию, которая позволяет в непустом двоичном дереве поиска (BST) найти значение, ближайшее к заданному.
8. Напишите функцию *kth_smallest(root, k)* для поиска k-го наименьшего элемента в заданном бинарном дереве поиска.
9. Дан список. Найти произведение всех элементов списка, расположенных справа от максимального элемента, не включая максимальный элемент.
10. Описать функцию *shift_right(a, b, c)*, выполняющую правый циклический сдвиг: значение a переходит в b, значение b – в c, значение c – в a. С помощью этой процедуры выполнить правый циклический сдвиг для очереди из трех чисел.
11. Открыть на чтение csv-файл с данными о цене акций организации и прочесть его с помощью модуля csv. Поместить полученные данные в очередь *collections.deque* с параметром максимальной длины равным трем. Распечатать очередь и интерпретировать результат.
12. Дана строка, содержащая полное имя файла, то есть имя диска, список каталогов (путь), собственно имя и расширение. Выделить из этой строки имя файла без расширения.
13. Реализовать функцию *repl*, которая принимает на вход строку и набор заранее неизвестных параметров. Результатом функции является строка, в которой слова совпадающие с именами параметров заменены на значения параметров. Пример: строка: 'replace my val abc', параметры *my='s1'*, *abc='fff'* -> Результат: 'replace s1 val fff'.
14. Задана строка, в которой через запятую перечислены слова. Создать словарь, в котором ключами будут слова из строки, а значениями - текст "номер <номер-слова-в-строке> в строке". Например, 'ten,one,five,two,three,four' преобразовать в {'three': 'номер 5 в строке', 'one': 'номер 2 в строке', 'ten': 'номер 1 в строке', 'two': 'номер 4 в строке', 'five': 'номер 3 в строке', 'four': 'номер 6 в строке'}

Примерные вопросы для подготовки к зачету, зачету

Третий семестр

1. Философия Python.
2. Форматированная строка. Примеры реализации.

3. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python.
4. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.
5. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python.
6. Изменяемые и неизменяемые коллекции языка Python.
7. Особенности копирования изменяемых объектов Python.
8. Операции со словарями, учитывающие возможное отсутствие ключа.
9. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств.
10. Выполнение основных операций с парой множеств в Python.
11. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.
12. Использование условий в выражениях-генераторах.
13. Логические ошибки, синтаксические ошибки, исключения.
14. Пользовательские исключения.
15. Структурирование программного кода с помощью модулей и пакетов.
16. Работа с файлами *pickle* и *shelve*.

Четвертый семестр

1. Вложенные функции, специфика реализации в Python.
2. Функции высшего порядка и декораторы в Python.
3. Итераторы в Python.
4. Встроенные функции для работы с итераторами и возможности модуля *itertools*.
5. Специфика массивов, как структур данных.
6. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.

7. Реализация основных операций в очереди на базе массива и связанного списка.
8. Связанные списки: однонаправленные и двунаправленные – принцип реализации.
9. Реализация двоичных деревьев в виде связанных объектов.
10. Замыкания в Python.
11. Особенности работы с замороженными множествами в Python.
12. Упорядоченный словарь *OrderedDict* из модуля *collections*. Назначение, примеры использования.
13. Цепочка словарей *ChainMap* из модуля *collections*. Назначение, примеры использования.
14. Счетчик *Counter* из модуля *collections*. Назначение, примеры использования.
15. Словарь по умолчанию *defaultdict* из модуля *collections*. Назначение, примеры использования.
16. Именованный кортеж *namedtuple* из модуля *collections*. Назначение, примеры использования.

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

Нормативные правовые акты:

Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных»

Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации»

Федеральный закон от 06.04.2011 № 63-ФЗ «Об электронной подписи»

Основная литература:

1. 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. [БИК ID: RU2fURAIT2f556864]

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) . [БИК ID: RU\infra-m\znanium\bibl\2166199]

3. 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. [БИК ID: RU\FA\books\137296]

Дополнительная литература:

1. Василекина, О. М. Учебно-методическое пособие по дисциплине «Алгоритмизация и программирование»: Структурное и процедурное

программирование на языке Python направление подготовки 09.03.03 Прикладная информатика профиль «Прикладная информатика в экономике» [Электронный ресурс] / Василекина О. М. Великие Луки : Великолукская ГСХА, 2024 104 с. Книга из коллекции Великолукская ГСХА - Информатика <https://e.lanbook.com/book/426992>. [БИК ID: RU-LAN-BOOK-426992]

2. Канева, О. Н. Введение в программирование на языке Python [Электронный ресурс] : учеб. пособие / Канева О. Н.,Финк Т. Ю. Омск : ОмГТУ, 2024 149 с. Книга из коллекции ОмГТУ - Информатика <https://e.lanbook.com/book/504279> ISBN 978 5 8149 3864 0. [БИК ID: RU-LAN-BOOK-504279]

3. Букунов, С. В. 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. [БИК ID: RU-LAN-BOOK-292856]

4. 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. [БИК ID: RU2fURAIT2f556864]

5. Бурдыко, Т. Г. Web-программирование на языке Python [Электронный ресурс] : учебно-методическое пособие / Бурдыко Т. Г.,Гавриленко А. В. Сургут : СурГУ, 2025 56 с. Книга из коллекции СурГУ - Информатика <https://e.lanbook.com/book/494693>. [БИК ID: RU-LAN-BOOK-494693]

6. Янцев, В. В. Web-программирование на Python [Электронный ресурс] : учебное пособие для вузов / Янцев В. В. 3-е изд., перераб. Санкт-Петербург : Лань, 2024 180 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/392993> ISBN 978-5-507-48364-8. [БИК ID: RU-LAN-BOOK-392993]

7. Чернышев, Станислав Андреевич Основы программирования на Python : учебник для вузов / С. А. Чернышев. 2-е изд., пер. и доп Электрон. дан. Москва : Юрайт, 2025 349 с (Высшее образование) URL: <https://urait.ru/bcode/567821> (дата

обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/567821> ISBN 978-5-534-17139-6 : 1729.00. [БИК ID: RU2fURAIT2f567821]

8. Богун, В. В. Программирование задач по финансовой математике на языке программирования Python : учебное пособие / В. В. Богун ; Финансовый университет при Правительстве Российской Федерации ; Департамент математики Москва : Прометей, 2025 172 с. : ил., табл. Библиогр. в кн Режим доступа: электронная библиотечная система «Университетская библиотека ONLINE», требуется авторизация <https://biblioclub.ru/index.php?page=book&id=721368> ISBN 978-5-00172-780-4. [БИК ID: BIBLIOCLUB\0000721368]

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. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>
4. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
5. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
6. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
7. •Электронно-библиотечная система Znanium <http://www.znanium.ru/>
8. Data Science Essentials / Microsoft. – <https://www.edx.org/course/data-science-essentials-microsoft-dat203-1x-3>
9. Электронный учебный курс <https://campus.fa.ru/course/view.php?id=12139>
10. Курс Онлайн-академии Финуниверситета <https://online.fa.ru/courses/course-v1:fa+itinresearch+2023/about>.

10. Методические указания для обучающихся по освоению дисциплины

На семинарских занятиях студенты с помощью преподавателя осваивают учебный материал и выполняют индивидуальные задания. Преподаватель распределяет задания между студентами. Преподаватель может пояснить задание в части требований к выполнению или стоящей за заданием математической концепцией. Все задания подразумевают творческий подход к их выполнению. После выполнения задания преподаватель может попросить студента провести публичную демонстрацию выполненного задания, включая устные комментарии написанного хода и алгоритмических решений, примененных студентом при решении задачи. Самостоятельная работа выполняется студентами в классах или медиатеке университета. При этом используется программное обеспечение Jupyter Notebook в составе пакета программ Anaconda, печатные учебные пособия и электронные образовательные ресурсы.

При подготовке к контрольным работам 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").

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. Jupyter Notebook (Windows 10)
3. Текстовый редактор
4. Anaconda

5. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Язык программирования Python 3. <https://pythonworld.ru/>
4. Размещение ЭУК: <https://www.campus.fa.ru/>

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)
4. Библиотека, читальный зал