

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финуниверситет)**

Владикавказский филиал Финуниверситета

Кафедра «Корпоративные инфокоммуникационные системы»

УТВЕРЖДАЮ
Директор филиала
Т.А. Хубаев
«15» апреля 2026 г.



М.А. Ковалева

Алгоритмы и структуры данных в языке Python

Рабочая программа дисциплины
для студентов, обучающихся по направлению подготовки
09.03.04 Программная инженерия,
ОП «Технологии разработки программного обеспечения»

*Рекомендовано Ученым советом Владикавказского филиала
Финансового университета
(протокол от «15» апреля 2026 г. № 30)*

*Одобрено на заседании кафедры «Корпоративные инфокоммуникационные
системы»
(протокол от «10» апреля 2026 г. № 8)*

Владикавказ 2026

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	4
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	5
5.1. Содержание дисциплины	5
5.2. Учебно-тематический план	9
5.3. Содержание семинаров, практических занятий.....	12
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	14
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	14
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	17
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	22
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	32
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	33
10. Методические указания для обучающихся по освоению дисциплины.....	34
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	40
11.1 Комплект лицензионного программного обеспечения	40
11.2 Современные профессиональные базы данных, и информационные справочные системы	40
11.3 Сертифицированные программные и аппаратные средства защиты информации	40
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	40

1. Наименование дисциплины

Дисциплина «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
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	1. Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования.	Знать: объектно-ориентированный язык программирования Python на уровне знания синтаксиса и семантики, основ стандартной библиотеки. Уметь: определять на уровне знания синтаксис и семантику, стандартные библиотеки языка Python, необходимые для решения прикладных задач.
		2. Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ.	Знать: технологии прикладного программирования, инструментальные средства программирования. Уметь: разрабатывать программы решения задач с использованием прикладного программирования, включая среды высокоуровневого программирования.
		3. Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев..	Знать: особенности создания программного кода, основы проектирования различных видов интерфейса программной системы. Уметь: разрабатывать программный код, ориентироваться в существующем коде, применять знание общепринятых

			соглашений и политик в области оформления кода, разрабатывать текстовый, программный или графический интерфейс программной системы исходя из ее назначения.
ОПК-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	1. Демонстрирует знания основ теории информации и алгоритмов, основных элементарных алгоритмов и структуры данных.	Знать: основы теории информации и алгоритмов, основные элементарные алгоритмы и структуры данных, процессы сбора, обработки и интерпретации информации. Уметь: описывать состав и структуру требуемых данных и информации
		2. Применяет простые алгоритмы и структуры данных к решению поставленной задачи, проводит выбор наиболее оптимальных методов.	Знать: простые алгоритмы и структуры данных, оптимальные методы решения задач Уметь: применять простые алгоритмы и структуры данных к решению поставленной задачи, проводить выбор наиболее оптимальных методов
		3. Проводит подробный количественных анализ реализованной программной системы с точки зрения оптимальности применяемых алгоритмических решений.	Знать: основные методы количественного анализа программных систем Уметь: выполнять количественных анализ реализованной программной системы с точки зрения оптимальности применяемых алгоритмических решений.

3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» является дисциплиной Общепрофессионального цикла обязательной части учебного плана образовательной программы «Технологии разработки программного обеспечения» по направлению подготовки 09.03.04 Программная инженерия, профиль «Технологии разработки программного обеспечения».

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

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 1 (в часах)	Семестр 2 (в часах)
Общая трудоемкость дисциплины	8 /288	144	144
Контактная работа- Аудиторные занятия	100	50	50
<i>Лекции</i>	32	16	16
<i>Семинары, практические занятия</i>	68	34	34
Самостоятельная работа	188	94	94
Вид текущего контроля	Контрольная работа, контрольная работа	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Экзамен, экзамен	Экзамен	Экзамен

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

5.1. Содержание дисциплины

Тема 1. Введение в программирование на Python

Общая информация о языке Python. История языка программирования, его связь с другими языками программирования, распространенность Python и основные сферы его применения. Знакомство с первыми примерами кода на Python. Философия Python. Базовая информация о языке Python. Основные типы данных. Основные числовые типы данных и операции над ними. Математические операции над числовыми типами данных. Преобразование типов данных. Переменные и специфика их объявления. Статическая и динамическая типизация. Работа с переменными. Управление памятью и сборка мусора в Python. Именованые переменных. Работа со строками: создание строк, специальные символы. Индексирование строк, получение срезов строк. Основные функции для работы со строками. Вывод на экран (работа с функцией print) и форматирование строк. Различные подходы к

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

Тема 2. Управляющие конструкции, списки и кортежи

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

Циклы в Python: while, for. Специфика циклов в Python. Функции range и enumerate и их использование в циклах. Списки и кортежи в Python. Специфика списков и их отличие от массивов. Создание списка, оперирование вложенными списками, копирование списков, операции над списками: индексация и срезы; изменение списка; поиск, сортировка и обход; изменение списка. Кортежи в Python: синтаксис, специфика использования.

Тема 3. Словари, множества и выражения-генераторы

Словари Python. Словари: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python. Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Пример: задача приведения списка к "плоскому"

виду. Выражения-генераторы для множеств и словарей. Кейсы использования и производительность решений с использованием выражений-генераторов.

Тема 4. Функции

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 упаковка параметров функции. Аннотации и документирование функций. Глобальные и локальные переменные.

Тема 5. Работа с файлами и обработка исключительных ситуаций

Обработка исключений в Python: кейсы для использования. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert. Работа с файлами в Python. Концепция файла в современных ОС и языках программирования. Операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов. Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.

Тема 6. Модули и пакеты

Модули и пакеты в Python: подход к структурированию программного кода с помощью модулей и пакетов. Синтаксис импортирования в Python. Создание и работа с пакетами в Python. Повторная загрузка модулей. Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.

Тема 7. Продвинутое коллекции

Модули стандартной библиотеки Python, предоставляющие дополнительный функционал по работе с коллекциями объектов. Класс Frozen set. Модуль collections. Класс Counter, класс defaultdict, класс OrderedDict, класс namedtuple. Модуль enum.

Тема 8. Обзор современных языков программирования

История и эволюция языков программирования. Распространенность современных языков программирования. Классификация языков программирования. Парадигмы программирования. Специализация языков программирования. Место Python в современном ландшафте языков программирования. Характеристики языков программирования: способ передачи параметров в функцию; способ управления динамической памятью; виды типизации переменных.

Тема 9. Введение в объектно-ориентированное программирование

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

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

Тема 10. Объектно-ориентированное программирование в Python

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

Тема 11. Введение в функциональное программирование

Парадигмы и идиомы программирования, общая концепция функционального программирования. Функциональные языки программирования. Функции "граждане первого класса", функции высшего порядка, замыкания, функции без побочных эффектов, рекурсия, хвостовая рекурсия. Не-меняемые структуры данных. Идиомы, распространенные в функциональных языках программирования: итераторы, последовательности, ленивые вычисления, сопоставление с образцом, монады. Элементы функционального программирования в Python: функции – граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python. Декораторы в Python: использование и создание собственных декораторов.

Тема 12. 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.

Тема 13. 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 14. 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 15. Структуры данных: деревья

Деревья, бинарные деревья. Использование бинарных деревьев в прикладных задачах: представление выражения (предложения) в виде дерева. Обход бинарных деревьев. Двоичное дерево поиска. Двоичные кучи и очереди с приоритетом.

Тема 16. Хеш-таблицы

Абстрактный тип данных – ассоциативный массив. Таблица с прямой адресацией. Хеш-таблица, хеш-функция: метод деления; метод MAD. Полиномиальная хеш-функция. Функция hash в Python. Методы разрешения коллизий.

5.2. Учебно-тематический план

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самост оатель ная работа	
			Общая, в т.ч.:	Лекци и	Семина ры, практич еские		
1	Тема 1. Введение в программировани е на Python	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
2	Тема 2. Управляющие	18	6	2	4	12	Опрос, собеседование по домашним

	конструкции, списки и кортежи						заданиям самостоятельной работы, решение задач (программирование)
3	Тема 3. Словари, множества и выражения- генераторы	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
4	Тема 4. Функции	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
5	Тема 5. Работа с файлами и обработка исключительных ситуаций	20	8	2	6	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
6	Тема 6. Модули и пакеты	20	8	2	6	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
7	Тема 7. Продвинутое коллекции	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
8	Тема 8. Обзор современных языков программирования	14	4	2	2	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование), защита контрольной работы
9	Тема 9. Введение в объектно- ориентированное программирование	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
	Тема 10. Объектно- ориентированное программирование	20	8	2	6	12	Опрос, собеседование по домашним заданиям самостоятельной

	е в Python						работы, решение задач (программирование)
	Тема 11. Введение в функциональное программирование	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
	Тема 12. Функциональное программирование в Python	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
	Тема 13. Структуры данных: массивы, стеки, очереди, списки	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
	Тема 14. Алгоритмы поиска и сортировки	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
	Тема 15. Структуры данных: деревья	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование)
	Тема 16. Хеш-таблицы	18	6	2	4	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение задач (программирование), защита контрольной работы
	В целом по дисциплине	288	100	32	68	188	Согласно учебному плану: две контрольные работы
	Итого в %	100	34,7	32	68	65,3	

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятия
Тема 1. Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка I Python notebook: принципы работы и применения.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 2. Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
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.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 4. Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
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.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 6. Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 7. Продвинутое коллекции	Модули стандартной библиотеки Python, предоставляющие дополнительный функционал по работе с коллекциями объектов.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с

	Класс Frozen set. Модуль collections	последующим коллективным обсуждением их результатов
Тема 8. Обзор современных языков программирования	Парадигмы программирования. Специализация языков программирования. Место Python в современном ландшафте языков программирования. Характеристики языков программирования: способ передачи параметров в функцию; способ управления динамической памятью	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов, защита контрольной работы
Тема 9. Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super().	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 10. Объектно-ориентированное программирование в Python	Методы классов и статические переменные, и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 11. 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. Декораторы в Python: использование и создание собственных декораторов.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 12. Функциональное программирование в Python	Реализация функций map, filter, reduce в Python. Итераторы в Python, итерируемый тип данных. Модуль itertools.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 13. 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	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов

	связанных списков.	
Тема 14. 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.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 15. Структуры данных: деревья	Реализация на Python деревьев, бинарных деревьев. Использование бинарных деревьев в прикладных задачах: представление выражения (предложения) в виде дерева.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов
Тема 16. Хеш-таблицы	Работа с функцией hash в Python. Использование специализированных библиотек для работы с хэш-функциями.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение задач с последующим коллективным обсуждением их результатов, защита контрольной работы

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
Тема 1. Введение в программирование на Python	Среда программирования. Использование документации.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 2. Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение,

		выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 3. Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 4. Функции	Возвращение нескольких значений из функции. Распаковка и упаковка параметров функции	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 5. Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 6. Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 7. Продвинутое коллекции	Класс Counter, класс defaultdict, класс OrderedDict, класс namedtuple. Модуль enum.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter

		Notebook.
Тема 8. Обзор современных языков программирования	Виды типизации переменных в различных современных языках программирования.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook, выполнение контрольной работы
Тема 9. Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 10. Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 11. Введение в функциональное программирование	Реализация декораторов с параметрами в Python.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 12. Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.

Тема 13. Структуры данных: массивы, стеки, очереди, списки	Реализация различных вариантов связанных списков на Python.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 14. Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 15. Структуры данных: деревья	Реализация обхода бинарных деревьев.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook.
Тема 16. Хеш-таблицы	Использование специализированных библиотек для работы с хэш-функциями: проверка хэш-конфликтов для различных наборов данных.	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы, в том числе заданий с использованием Jupyter Notebook, выполнение контрольной работы

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

Примерный перечень тем для подготовки к опросу

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. Регулярное выражение
24. В чем разница между списком и кортежем?
25. Как выполняется интерполяция строк?
26. Что такое декоратор?
27. Определите класс car с двумя атрибутами: color и speed. Затем создайте экземпляр и верните speed
28. В чем разница между методами экземпляра, класса и статическими методами в Python?
29. Переменные в Python передаются по ссылке или по значению?
30. Как объединить списки в Python?
31. В чем разница между списками и массивами?

32. Как объединить два массива?
33. Назовите изменяемые и неизменяемые объекты
34. Какая разница между словарями и JSON?
35. В чем разница между pass, continue и break?
36. Преобразуйте следующий цикл for в генератор списков (list comprehension)
37. Как перевести строку в верхний/нижний регистр?
38. Приведите пример генератора словарей (dict comprehension)
39. Каковы преимущества использования Python?
40. Как писать комментарии на Python?
41. Как прокомментировать несколько строк в Python?

Примеры задач

Задача 1.

Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python

Выберите библиотеку Python для хэширования данных в целях быстрого поиска текстовых строк

Задача 2.

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

Задача 3.

Опишите закономерности/вариативность в синтаксических конструкциях Python для работы со списками и словарями

Задача 4.

Опишите подходы используемых в Python для группировки функций ориентированных на обработку однотипных объектов и решение близких задач

Задача 5.

Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.

Примеры домашних заданий самостоятельной работы

Задание 1.

Используя программу Jupyter Notebook составьте код на Python, определяющий количество точек на плоскости, находящихся на заданном расстоянии от начала координат. Используйте библиотеку math.

Задание 2.

Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python. Выберите библиотеку Python для разработки системы управления складом малого предприятия.

Реализуйте программу на Python, которая реализует систему управления складом малого предприятия.

Примерные задания контрольной работы (семестр 1)

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 два 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: $c_0, c_1, c_2 \dots c_n$.

Примерные задания контрольной работы (семестр 2)

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

На вход программа должна принимать 2 строки (используйте стандартную функцию `input()` Python). Каждая строка содержит 3 действительных числа через пробел: a_i, b_i, c_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.

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

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

Примерные вопросы для подготовки к экзамену (семестр 1)

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.
2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.
3. 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.

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.

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.

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.

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.

Примерные вопросы для подготовки к экзамену (семестр 2)

1. Концепция класса и объекта. Принципы и механизмы ООП.
2. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python.
3. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.
4. Методы классов и статические переменные и методы в Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.
5. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка элементов функционального программирования в Python.
6. Концепция «функции – граждане первого класса» в языке программирования, поддержка этой концепции в Python. Специфика лямбда-функций в Python их возможности и ограничения. Типичные сценарии использования лямбда-функций в Python.
7. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.
8. Вложенные функции и замыкания, специфика реализации в Python.
9. Функции высшего порядка и декораторы в Python.
10. Концепция map/filter/reduce. Реализация map/filter/reduce в Python и пример их использования.
11. 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.

12. Функции генераторы и выражения генераторы: создание и применение в Python.
13. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работа с array в Python.
14. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.
15. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.
16. Связанные списки: однонаправленные и двунаправленные – принцип реализации. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве.
17. Алгоритм обменной сортировки, сложность сортировки и возможности по ее улучшению.
18. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.
19. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве. Сложность поиска в отсортированном и не отсортированном массиве.
20. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.
21. Алгоритм быстрой сортировки, сложность сортировки и возможности по ее улучшению.
22. Алгоритм сортировки слиянием, сложность сортировки.
23. Реализация двоичных деревьев в виде связанных объектов. Различные реализации рекурсивного обхода двоичных деревьев.
24. Двоичное дерево поиска – принципы реализации и логика реализации основных операций.
25. Двоичная куча – принципы реализации и логика реализации основных операций.

26. Абстрактный тип данных - ассоциативный массив и принцип его реализации на основе хэш-таблиц и хэш-функций.

27. Общая схема построения хэш-функции и возможная роль в этой схеме хэш-функции multiply-add-and-divide. Принцип работы хэш-функции multiply-add-and-divide.

28. Полиномиальная хэш-функция – принцип работы, специфика эффективной реализации и специфика применения хэш-функции.

29. Различные методы разрешения коллизий в хэш-таблицах.

Примерные практико-ориентированные задания к экзамену (семестр 1)

Практико-ориентированное задание

Используя программу Jupyter Notebook составьте код на Python, определяющий количество точек на плоскости, находящихся на заданном расстоянии от начала координат. Используйте библиотеку math.

Практико-ориентированное задание

Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных.

Практико-ориентированное задание

Выберите библиотеку Python для создания и обработки деревьев данных.

Реализуйте программный код вычисления различных арифметических операций, выбор которых осуществляется с помощью разработанного меню.

Примерные практико-ориентированные задания к экзамену (семестр2)

Практико-ориентированное задание

Реализуйте программу на Python которой в качестве аргумента командной строки передается имя CSV-файла, в первом столбце находятся числа, которые необходимо отсортировать. Программа создает новый файл, в

котором первый столбец отсортирован.

Практико-ориентированное задание

Напишите программу на Python, обрабатывающую текстовую строку так, чтобы в нем все заглавные буквы преобразовать в строчные.

Практико-ориентированное задание

Напишите скрипт на Python обрабатывающий текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV.

Практико-ориентированное задание

Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные.

Практико-ориентированное задание

Напишите скрипт на Python обрабатывающий текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV.

Пример экзаменационного билета

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Кафедра: **Корпоративные инфокоммуникационные системы**
Дисциплина: **Алгоритмы и структуры данных в языке Python**
Филиал: **Владикавказский**; Форма обучения: **Очная**
Семестр: 2 Направление: **09.03.04 Программная инженерия**
Профиль: **Технологии разработки программного обеспечения**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

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. Практико-ориентированное задание. Задание выполнить в **Jupyter Notebook**. Напишите программу, которая принимает список чисел и возвращает список тех, которые больше среднего значения. **(20 баллов)**
3. Практико-ориентированное задание. Задание выполнить в **Jupyter Notebook** Создайте класс Person с атрибутами name, age. Реализуйте: **(20 баллов)**
 - Метод introduce, выводящий информацию о человеке;
 - Дочерний класс Employee, добавляющий атрибуты position, salary;
 - Метод promote(new_position: str, raise_amount: float) для изменения должности и повышения зарплаты.

Подготовил: _____ М.А Ковалева

На основе перечня теоретических вопросов и практико-ориентированных заданий, утвержденного на заседании кафедры «Корпоративные инфокоммуникационные системы» протокол № ____ от _____.2026 г.

Утверждаю:
Заведующий кафедрой _____ М.А Ковалева
Дата _____.2026г.

**Примеры оценочных средств для проверки индикаторов
достижения компетенций, формируемых дисциплиной**

Код и наименование компетенции	Наименование индикатора достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	1. Разрабатывает алгоритмы решения простых информационных задач и выражает их на языке программирования.	Знать: объектно-ориентированный язык программирования Python на уровне знания синтаксиса и семантики, основ стандартной библиотеки. Уметь: определять на уровне знания синтаксиса и семантику, стандартные библиотеки языка Python, необходимые для решения прикладных задач.	Вопросы: 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. Практико-ориентированное задание Напишите программу на Python, обрабатывающую текстовую строку так, чтобы в нем все заглавные буквы преобразовать в строчные. Напишите скрипт на Python обрабатывающий текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV.
	2. Анализирует алгоритмы в части производительности, оптимальности, вырабатывает рекомендации для оптимизации алгоритмов программ.	Знать: технологии прикладного программирования, инструментальные средства программирования. Уметь: разрабатывать программы решения задач с использованием прикладного программирования,	Вопросы: 1. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for. 2. 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.

		включая среды высокоуровневого программирования.	<p>Задача</p> <p>Используя программу Jupyter Notebook составьте код на Python, определяющий количество точек на плоскости, находящихся на заданном расстоянии от начала координат. Используйте библиотеку math.</p> <p>Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных.</p>
	3. Проводит ручное и автоматизированное тестирование программных продуктов по методам черного и белого ящика, составляет набор тестовых случаев..	<p>Знать:</p> <p>особенности создания программного кода, основы проектирования различных видов интерфейса программной системы.</p> <p>Уметь:</p> <p>разрабатывать программный код, ориентироваться в существующем коде, применять знание общепринятых соглашений и политик в области оформления кода, разрабатывать текстовый, программный или графический интерфейс программной системы исходя из ее назначения.</p>	<p>Вопросы:</p> <ol style="list-style-type: none"> 1. Реализация двоичных деревьев в виде связанных объектов. Различные реализации рекурсивного обхода двоичных деревьев. 2. Двоичное дерево поиска – принципы реализации и логика реализации основных операций. 3. Двоичная куча – принципы реализации и логика реализации основных операций. <p>Задача</p> <p>Выберите библиотеку Python для создания и обработки деревьев данных. Реализуйте программный код вычисления различных арифметических операций, выбор которых осуществляется с помощью разработанного меню.</p> <p>Реализуйте программу на Python которой в качестве аргумента командной строки передается имя CSV-файла, в первом столбце находятся числа, которые необходимо отсортировать. Программа создает новый файл, в котором первый столбец отсортирован.</p>
ОПК-7 Способен применять в практической деятельности основные концепции, принципы,	1. Демонстрирует знания основ теории информации и алгоритмов,	<p>Знать:</p> <p>основы теории информации и алгоритмов, основные элементарные алгоритмы и структуры данных, процессы сбора,</p>	<p>Вопросы:</p> <ol style="list-style-type: none"> 1. Синтаксис и семантика обработки исключительных ситуаций в Python. Создание пользовательских исключений и инструкция assert. 2. Базовые операции для работы с файлами в Python.

теории и факты, связанные с информатикой	основных элементарных алгоритмов и структуры данных.	обработки и интерпретации информации. Уметь: описывать состав и структуру требуемых данных и информации	Использование инструкции with ... as на примере работы с файлами. Задача Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные. Напишите скрипт на Python обрабатывающий текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV.
	2. Применяет простые алгоритмы и структуры данных к решению поставленной задачи, проводит выбор наиболее оптимальных методов.	Знать: простые алгоритмы и структуры данных, оптимальные методы решения задач Уметь: применять простые алгоритмы и структуры данных к решению поставленной задачи, проводить выбор наиболее оптимальных методов	Вопросы: 1. Концепция класса и объекта. Принципы и механизмы ООП. 2. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python. 3. Полиморфизм, типизация, проверка принадлежности объекта к классу в языке Python. 4. Методы классов и статические переменные и методы в Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями. Задача Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python. Выберите библиотеку Python для разработки системы управления складом малого предприятия. Реализуйте программу на Python, которая реализует систему управления складом малого предприятия.

	<p>3. Проводит подробный количественный анализ реализованной программной системы с точки зрения оптимальности и применяемых алгоритмических решений.</p>	<p>Знать: основные методы количественного анализа программных систем</p> <p>Уметь: выполнять количественный анализ реализованной программной системы с точки зрения оптимальности применяемых алгоритмических решений.</p>	<p>Вопросы:</p> <p>1. Абстрактный тип данных - ассоциативный массив и принцип его реализации на основе хэш-таблиц и хэш-функций.</p> <p>2. Общая схема построения хэш-функции и возможная роль в этой схеме хэш-функции multiply-add-and-divide. Принцип работы хэш-функции multiply-add-and-divide.</p> <p>3. Полиномиальная хэш-функция – принцип работы, специфика эффективной реализации и специфика применения хэш-функции. Различные методы разрешения коллизий в хэш-таблицах</p> <p>Задача</p> <p>Выполните обработку текстового файла, в котором зафиксированы факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы. На основе полученных данных составьте CSV файл, хранящий полученные результаты. Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы.</p>
--	--	--	--

8. 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 python: учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 187 с. — (Высшее образование). — ISBN 978-5-534-14638-7. — URL: <https://urait.ru/bcode/556864> — 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: учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821>— Режим доступа: Электронно-библиотечная система Юрайт. — Текст: электронный

9. 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. Электронная библиотека издательского дома «Гребенников» <https://grebennikon.ru>

6. Электронно-библиотечная система издательства «Лань»
<https://e.lanbook.com>

7. Python Documentation [Электронный ресурс]: сайт. – Режим доступа: <http://python.org/doc/>.

8. Python Standard Library [Электронный ресурс]: сайт. – Режим доступа: <https://docs.python.org/2/library/>.

10. Методические указания для обучающихся по освоению дисциплины

Методика освоения дисциплины предусматривает подготовку обучающихся к лекциям, семинарам и практическим занятиям, выполнение студентами самостоятельной внеаудиторной работы, в том числе – контрольной работы.

Рекомендации по подготовке к лекционным занятиям.

Для наиболее полного освоения дисциплины студентам необходимо:

- перед каждой лекцией просматривать рабочую программу дисциплины, ее основные вопросы и рекомендуемую литературу. Это позволит сэкономить время на записывание основных вопросов темы;
- перед очередной лекцией просматривать материалы предыдущих, чтобы освоение материала не оставляло пробелов.

Рекомендации по подготовке к семинарам, практическим занятиям.

Студентам следует:

- проработать теоретический материал к занятию по рекомендованным литературным источникам и лекциям;
- использовать при подготовке к занятию нормативно-правовые документы, научные публикации, информационный материал, рекомендуемый преподавателем;
- перед занятиями задать вопросы по невыясненным в ходе самостоятельной подготовки темам или отдельным положениям темы;
- в ходе занятия давать четкие и исчерпывающие ответы на вопросы;

- на занятии демонстрировать понимание обсуждаемых тем и вопросов.

Студентам, по различным причинам пропустившим занятия, необходимо перед очередным занятием отработать пропущенный материал, подготовив его самостоятельно.

Методические рекомендации по выполнению различных форм самостоятельной работы

Студентам при организации самостоятельной работы следует руководствоваться Приказом Финансового университета № 1040/о от 11.05.2021г. «Об утверждении методических рекомендаций по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете».

Самостоятельная работа содержит в себе различные виды и формы работ. Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По теме учебной дисциплины студентам предлагается перечень заданий для самостоятельной работы.

В ходе изучения дисциплины предусмотрены следующие формы самостоятельной работы:

- подготовка к опросу;
- решение практико-ориентированных задач;
- выполнение контрольной работы;
- подготовка к экзамену.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также должны соответствовать установленным требованиям по оформлению.

Студентам следует:

- руководствоваться графиком самостоятельной работы,

определенным РПД;

- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, разбирать на занятиях и консультациях неясные вопросы;
- прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные фрагменты для их обсуждения на консультации.

Методические рекомендации для обучающихся по выполнению контрольной работы

Контрольная работа является обязательной формой внеаудиторной самостоятельной работы студентов по дисциплине и может реализовываться как в письменном виде, так и с использованием информационных технологий и специализированных программных продуктов.

Цель выполнения контрольной работы, содержащей комплект заданий – овладение студентами навыками решения типовых расчетных задач, формирование учебно-исследовательских навыков, закрепление умений самостоятельно работать с различными источниками информации; проверка сформированности компетенций.

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

Контрольная работа по дисциплине выполняется по вариантам.

Содержание заданий контрольных работ охватывают основной материал соответствующих разделов (тем) дисциплин. Контрольные задания разрабатываются по многовариантной системе. Варианты контрольных работ равноценны по объему и сложности.

Контрольная работа выполняется студентом под руководством преподавателя кафедры «Корпоративные инфокоммуникационные системы», ведущим семинарские (практические) занятия.

Контрольная работа состоит из нескольких частей. Состав контрольной работы и очередность размещения отдельных частей:

- титульный лист;
- основная часть;
- список использованных источников;
- приложения (при наличии).

Титульный лист является первой страницей и заполняется по определенным правилам.

Основная часть выполняется согласно заданиям (вопросам) контрольных работ.

В список использованных источников включаются названия законодательных актов, нормативных документов, книг, статей, учебных пособий и т. п., которые, так или иначе, использовались студентом при выполнении работы.

В Приложения выносятся вспомогательные материалы, которые не содержат основную информацию, либо материалы, которые сложно разместить по тексту работы (большие схемы, таблицы, графические материалы, расчетные справочные данные, образцы первичных документов и т.п.). Непременным условием включения данных материалов в приложение является ссылка на них в тексте работы.

Требования к выполнению контрольной работы:

- четкость и последовательность изложения материала (решения) в соответствии с составленным планом;
- наличие обобщений и выводов, сделанных на основе изучения информационных источников по данной теме;
- предоставление в полном объеме решений имеющихся в задании практических задач;
- использование современных способов поиска, обработки и анализа информации;
- самостоятельность выполнения.

Требования к оформлению контрольной работы.

Контрольная работа выполняется на компьютере (гарнитура Times New Roman, шрифт 13 или 14) через 1-1,5 интервала с полями: верхнее, нижнее - 2; правое - 3; левое - 1,5. Отступ первой строки абзаца - 1,25. Нумерация страниц – внизу в центре.

Иллюстративный материал (схемы, диаграммы, рисунки, таблицы и др.) встраивается в текст работы или выносится в Приложения.

При написании допускаются только общепринятые сокращения (например, тыс. руб.).

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

Объем контрольной работы составляет не более 6 страниц, не включая таблиц, графиков и т.п. (при наличии).

Законченная контрольная работа, содержащая все требуемые элементы оформления, вставленная в папку (или файл) и скрепленная по левому краю, сдается на кафедру или непосредственно руководителю контрольной работы – преподавателю; ведущему семинарские (практические) занятия по дисциплине. Он осуществляет проверку контрольной работы, а также оказывает помощь при подготовке к ее защите.

Контрольная работа защищается в назначенные сроки. Защита работы проводится до начала сессии (в крайнем случае, до начала экзамена по соответствующему предмету). При защите студент кратко излагает основные положения работы, последовательность ее выполнения, свои предложения.

При защите работы студент должен свободно ориентироваться в изложенном материале работы; ответить на все замечания преподавателя; уметь отвечать на вопросы преподавателя по выполненной работе.

Оценка контрольных работ студентов проводится в процессе текущего контроля успеваемости студентов.

Критерии оценки контрольной работы

Оценка «отлично» (5 баллов) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания вопросов контрольной работы /и/или умение уверенно применять их на практике при решении конкретных задач.

Оценка «хорошо» (3-4 балла) выставляется студенту, если он твердо знает материал контрольной работы, грамотно и по существу излагает его /и/или умеет применять полученные знания на практике при решении конкретных задач, но допускает некоторые неточности.

Оценка «удовлетворительно» (1-2 балла) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, обнаружившему нарушения логической последовательности в изложении материала, но при этом владеющему основными вопросами, выносимыми на контрольную работу и необходимыми для дальнейшего обучения /и/или умеющему применять полученные знания по образцу в стандартной ситуации.

Оценки «неудовлетворительно» (0 баллов) заслуживает студент, который не знает большей части основного содержания выносимых на контрольную работу вопросов, тем дисциплины, допускает грубые ошибки в формулировках основных понятий /и/или не умеет использовать полученные знания при решении типовых практических задач.

При оценивании контрольной работы на «неудовлетворительно» она должна быть переделана (исправлена) в соответствии с полученными замечаниями, сдана на проверку заново и защищена не позднее срока окончания ее приёма и защиты.

Оценка результатов текущего контроля успеваемости и промежуточной аттестации обучающихся осуществляется в соответствии с Балльно-рейтинговой системой Финансового университета (Приказ Финансового университета № 2187/о от 01.10.2024 г. «Об утверждении Положения о проведении текущего контроля успеваемости и промежуточной аттестации

студентов, обучающихся по образовательным программам высшего образования в Финансовом университете»).

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

11.1 Комплект лицензионного программного обеспечения

- 1) Антивирусная защита Kaspersky Security для виртуальных и облачных сред;
- 2) Windows, Microsoft Office или Astra Linux, Libre Office.
- 3) Дистрибутив Python Anaconda (свободно распространяемое ПО).
- 4) Браузер.

11.2 Современные профессиональные базы данных, и информационные справочные системы

1. Информационно-правовая система «Гарант»: <https://www.garant.ru>
2. Большая Российская энциклопедия: <https://bigenc.ru/>
3. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>.

11.3 Сертифицированные программные и аппаратные средства защиты информации

Не используются

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

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 45

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол компьютерный – 1 шт.

Стол (студенческий) двухместный – 13 шт.

Стулья – 27 шт.

Доска меловая – 1 шт.

Технические средства обучения:

Компьютер в сборе – 1 шт.

Доска интерактивная – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения:

Аудитория № 47

Специализированная мебель:

Стол компьютерный – 20 шт.

Стол (двухместный) – 7 шт.

Стул – 34 шт.

Шкаф – 1 шт.

Технические средства обучения:

Компьютер в сборе – 20 шт.

Мультимедиа-проектор – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Помещения для самостоятельной работы обучающихся:

Кабинет № 55. Читальный зал:

Специализированная мебель:

Стол – 20 шт.

Стул – 40 шт.

Шкаф для книг – 4 шт.

Стеллаж книжный – 13 шт.

Стеллаж выставочный – 4 шт.

Технические средства обучения:

Компьютер в сборе – 6 шт.

Телевизор – 1 шт.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Финансового университета