

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

Тульский филиал Финуниверситета

Кафедра «Математика и информатика»

УТВЕРЖДАЮ



Директор Тульского филиала

Г.В. Кузнецов

26 мая 2026 г.

Приложение к рабочей программе дисциплины

Алгоритмы и структуры данных в языке Python

Направление подготовки 38.03.05 – Бизнес-информатика,
образовательная программа «Цифровая трансформация управления
бизнесом»,

Профиль: ИТ-менеджмент в бизнесе

Фамилия И.О. научно-педагогического работника филиала, ответственного
за актуализацию РПД Р.А. Жуков

Год утверждения рабочей программы дисциплины – 2023

*Одобрено кафедрой «Математика и информатика»
Протокол от 05 мая 2026 г. № 11*

Тула 2026

СОДЕРЖАНИЕ ПРИЛОЖЕНИЯ

№ п/п	РАЗДЕЛЫ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ С ВНЕСЁННЫМИ ИЗМЕНЕНИЯМИ И ДОПЛНЕНИЯМИ	стр.
2.	Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения, соотнесенных с планируемыми результатами обучения по дисциплине	3
4.	Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	3
5.2	Учебно-тематический план	4
5.3.	Содержание семинаров, практических занятий.....	5
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	7
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	7
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю.	8
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	10
7.1	Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, знаний и умений.....	10
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.....	14
10.	Методические указания для обучающихся по освоению дисциплины.....	15
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем.....	17
11.1.	Комплект лицензионного программного обеспечения	17
11.2.	Современные профессиональные базы данных и информационные справочные системы	17
11.3.	Сертифицированные программные и аппаратные средства защиты информации	17
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	17

2. Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения, соотнесенных с планируемыми результатами обучения по дисциплине

В результате изучения дисциплины «Алгоритмы и структуры данных в языке Python» у студентов должны быть сформированы следующие компетенции:

Таблица 1

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКН-2	Способность анализировать и проектировать информационные потоки организации	1. Анализирует информационные потоки организации.	Знать: типы и структуры данных, используемые в языке Python. Уметь: выбирать структуры данных и алгоритмы, позволяющие решить поставленную задачу наиболее оптимальным способом.
		2. Создают модели «как есть» и «как должно быть» информационных потоков организации.	Знать: технологии обработки, анализа и интерпретации данных различной природы. Уметь: применять алгоритмы для поиска и выявления зависимостей в данных.

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

Таблица 2

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 1 (в часах)	Семестр 2 (в часах)
Общая трудоемкость дисциплины	7/252	108	144
<i>Контактная работа-Аудиторные занятия</i>	100	50	50
Лекции	32	16	16
Семинары, практические занятия	68	34	34
Самостоятельная работа	152	58	94
Вид текущего контроля	контрольные работы	контрольная работа	контрольная работа
Вид промежуточной аттестации	1-Зачет 2 -экзамен	Зачет	Экзамен

5.2 Учебно-тематический план

Таблица 3

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоят ельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Введение в программирование на Python	16	6	2	4	10	Обсуждение теоретических вопросов. Решение задач (программирование) по тематике практического занятия.
2	Управляющие конструкции, списки и кортежи	16	6	2	4	10	
3	Словари, множества и выражения-генераторы	18	6	2	4	12	
4	Функции	18	12	4	8	6	
5	Работа с файлами и обработка исключительных ситуаций	20	12	4	8	8	
6	Модули и пакеты	20	8	2	6	12	
	Итого в 1 семестре	108	50	16	34	58	Согласно учебному плану: контрольная работа
7	Введение в объектно- ориентированное программирование	20	6	2	4	14	Обсуждение теоретических вопросов. Решение задач (программирование) по тематике практического занятия.
8	Объектно- ориентированное программирование в Python	22	6	2	4	16	
9	Введение в функциональное программирование	20	6	2	4	14	
10	Функциональное программирование в Python	32	18	6	12	14	
11	Структуры данных: массивы, стеки, очереди, списки	24	6	2	4	18	
12	Алгоритмы поиска и сортировки	26	8	2	6	18	
	Итого во 2 семестре	144	50	16	34	94	Согласно учебному плану: контрольная работа
	В целом по дисциплине	252	100	32	68	152	Согласно учебному плану: контрольные работы
Итого в %:		100	40	13	27	60	

5.3. Содержание семинаров, практических занятий

Таблица 4

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8, 9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Словари, множества и выражения-генераторы	Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python. Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Работа с файлами и обработка исключительных ситуаций	Исключения. Инструкция try...except...else...finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert. Работа с файлами в Python, операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений

Модули и пакеты	Устройство модулей и пакетов, инструкции <code>import</code> и <code>from</code> . Создание собственных модулей и пакетов. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция <code>super()</code> . Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
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. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Функциональное программирование в Python	Реализация функций <code>map</code> , <code>filter</code> , <code>reduce</code> в Python. Итераторы в Python, итерируемый тип данных. Модуль <code>itertools</code> . Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
Структуры данных: массивы, стеки, очереди, списки	Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с динамическими массивами. Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков. Рекомендуемые источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений

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. Рекомендуются источники: 8.1; 8.2; 8.3	Выполнение практических заданий под руководством преподавателя и самостоятельно. Решение задач. Разбор решений
-------------------------------	---	--

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

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

Таблица 5

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в программирование на Python	Среда программирования. Использование документации.	Работа с литературой. Выполнение заданий для самостоятельной работы
Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Работа с литературой. Выполнение заданий для самостоятельной работы
Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование.	Работа с литературой. Выполнение заданий для самостоятельной работы
Функции	Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции.	Работа с литературой. Выполнение заданий для самостоятельной работы
Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Работа с литературой. Выполнение заданий для самостоятельной работы
Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Работа с литературой. Выполнение заданий для самостоятельной работы
Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Работа с литературой. Выполнение заданий для самостоятельной работы
Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Работа с литературой. Выполнение заданий для самостоятельной работы
Введение в	Реализация декораторов с	Работа с литературой.

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
функциональное программирование	параметрами в Python.	Выполнение заданий для самостоятельной работы
Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Работа с литературой. Выполнение заданий для самостоятельной работы
Структуры данных: массивы, стеки, очереди, списки	Реализация различных вариантов связанных списков на Python.	Работа с литературой. Выполнение заданий для самостоятельной работы
Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Работа с литературой. Выполнение заданий для самостоятельной работы

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

На практических занятиях используются методические материалы, выдаваемые преподавателем в виде файлов, а также материалы с сайтов, указанных в разделах 8 и 9 («Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины» и «Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины»).

Задания для самостоятельной работы содержатся в методических материалах, выдаваемых преподавателем на практических занятиях.

Примерные вопросы текущего контроля в 1 семестре

1. Операции над основными числовыми типами данных.
2. Операции над булевскими переменными.
3. Динамическая типизация в Python.
4. Преобразование типов в Python.
5. Создание строк в Python.
6. Организация и пример цикла while в Python.
7. Организация и пример цикла for в Python.
8. Операции над словарями в Python.
9. Выражения-генераторы для списков в Python.
10. Необязательные параметры функций в Python.
11. Заpackовка и распаковка параметров в Python.
12. Аннотации и документирование функций.
13. Создание объектов в Python.
14. Управление доступом к атрибутам класса в Python.
15. Выполнение интроспекции в Python.
16. Замыкания в Python.

17. Использование декораторов в Python.
18. Создание собственных декораторов в Python.
19. Компилятор и интерпретатор. Достоинства и недостатки.
20. Назовите и дайте краткую характеристику основных классов языков программирования.
21. Встроенные числовые типы языка Python.

Примеры заданий контрольных работ

Контрольная работа 1

1. В строке содержащей последовательность слов, разделенных запятыми удалить все нечетные слова. Ответ представить в виде строки. Пример: строка 'SIX,SEVEN,EIGHT,NINE,TEN' будет преобразована в: 'SIX,EIGHT,TEN'.
2. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку 'Eeny,meeny,miney,мое'

Примерные вопросы текущего контроля во 2 семестре

1. Списки. Создание, основные операции.
2. Основные методы списка.
3. Кортежи. Создание, основные методы и операции.
4. Словари. Создание, основные операции.
5. Методы для работы со словарями.
6. Множества. Создание, основные методы и операции.
7. Переменные. Правила именования переменных.
8. Динамическая типизация.
9. Операторы сравнения и логические операторы.
10. Инструкция if...else.
11. Инструкция цикла while.
12. Инструкция цикла for.
13. Создание и вызов функции.
14. Передача аргументов функцию.
15. Функции-генераторы.
16. Лямбда-функции.
17. Модули. Инструкции import и from.
18. Базовые принципы объектно-ориентированного программирования.
19. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
20. Конструктор и деструктор.

Примеры заданий контрольных работ

Контрольная работа 2

1. Создать иерархию классов для фруктов, продающихся в магазине.

Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее 2х атрибутов и 2х методов. Реализовать механизм автоматического подсчета количества всех созданных фруктов и автоматического присвоения каждому фрукту уникального идентификатора. Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать работу созданного механизма.

2. Реализовать декоратор с именем `not_none`, который генерирует исключительную ситуацию если декорируемая функция вернула значения `None`. (20 баллов)

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

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

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

Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их 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».

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

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые задания
ПКН-2 Способность анализировать и проектировать информацион	1. Анализирует информационные потоки организации.	Знать: типы и структуры данных, используемые в языке Python. Уметь: выбирать структуры данных и алгоритмы,	Задание 1. Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные. Задание 2. Напишите скрипт на Python обрабатывающий

ные потоки организации		позволяющие решить поставленную задачу наиболее оптимальным способом.	текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV. Задание 3. Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы.
	2. Создают модели «как есть» и «как должно быть» информационных потоков организации.	Знать: технологии обработки, анализа и интерпретации данных различной природы. Уметь: применять алгоритмы для поиска и выявления зависимостей в данных.	Задание 1. Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных. Задание 2. Выберите библиотеку Python для хэширования данных. Выполните программный код быстрого поиска текстовых строк. Задание 3. 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.

Перечень примерных вопросов к зачету

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.
2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.
3. Организация ветвлений в Python. Инструкция if...else. Множественный выбор.
4. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.

5. Циклы в Python, работа и устройство цикла while. Организация бесконечного цикла, аварийное завершение цикла.
6. Списки в Python. Обращение к элементам списка и создание срезов. Обход списка и поиск элементов в списке.
7. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков.
8. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python. Операции с представлениями словарей.
9. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.
10. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.
11. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.
12. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.
13. Функции стандартной библиотеки для работы с контейнерами.
14. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции.
15. Функция. Получение информации о функции. Способы передачи параметров при вызове функции.
16. Передача переменного количества параметров (именованных и не именованных) в функции Python.
17. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.
18. Синтаксис и семантика обработки исключительных ситуаций в Python. Создание пользовательских исключений и инструкция assert.
19. Базовые операции для работы с файлами в Python. Использование инструкции with ... as на примере работы с файлами.
20. CSV файлы. Методы и функции работы с CSV файлами.
21. Бинарные файлы. Методы и функции работы с бинарными файлами.

Перечень примерных вопросов к экзамену

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. Деревья, создание деревьев с помощью структуры список, узлов и ссылок. Операции с деревьями. Обход дерева.

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

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

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

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

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

23. Алгоритм сортировки слиянием, сложность сортировки.

Пример экзаменационного билета

Федеральное государственное образовательное бюджетное учреждение
высшего образования

«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Кафедра «Математика и информатика»

Дисциплина «Алгоритмы и структуры данных в языке Python»

Филиал Тульский

Семестр: 2 Направление: Бизнес-информатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Создание базового окна. Элементы графического интерфейса (виджеты) (10 баллов).
2. Модули. Инструкции `import` и `from`. Обзор стандартной библиотеки Python (10 баллов).
3. Напишите скрипт, реализующий последовательность действий по созданию таблицы прихода / расхода / списания Товара1 в течение месяца в файл `mov.csv`, организовать считывание из этого файла данных по строкам и столбцам в структуру `table` (40 баллов).

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

Основная литература

1. Жуков Р. А. Язык программирования Python: практикум [Электронный ресурс]: учебное пособие. М.: ИНФРА-М, 2021. 216 с. <https://znanium.com/catalog/product/1412168> (дата обращения: 12.01.2025).

Дополнительная литература

2. Гуриков С. Р. Основы алгоритмизации и программирования на Python [Электронный ресурс]: учебное пособие. М.: ФОРУМ; ИНФРА-М, 2020. 343 с. <https://znanium.com/catalog/product/1206074> (дата обращения: 12.01.2025).
3. Федоров Д. Ю. Программирование на языке высокого уровня Python [Электронный ресурс]: учебное пособие. 2-е изд., перераб. и доп. М.: Юрайт, 2021. 161 с. <https://urait.ru/bcode/472985> (дата обращения: 12.01.2025).
4. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования [Электронный ресурс]: учебное пособие. М.: Юрайт, 2021. 235 с. <https://urait.ru/bcode/469759> (дата обращения: 12.01.2025).

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

1. Сайт Департамента анализа данных и машинного обучения Финансового университета.
2. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
3. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
4. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
5. Электронно-библиотечная система Znanium <http://www.znanium.com>
6. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.urait.ru/>
7. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>
8. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
9. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
10. Национальная электронная библиотека <http://нэб.рф/>

10. Методические указания для обучающихся по освоению дисциплины

Основная цель освоения содержания дисциплины получить теоретические знания в ходе лекционных занятий и семинаров, закрепить их в процессе систематической самостоятельной работы и сформировать знания, умения и владения в соответствии с компетенциями, содержащимися в рабочей программе по дисциплине.

Самостоятельная работа студента в процессе освоения дисциплины включает в себя:

- подготовку к семинарским занятиям;
- работу с электронными учебными ресурсами;
- изучение основной и дополнительной литературы по дисциплине;
- выполнение контрольных работ;
- индивидуальные или групповые консультации по наиболее сложным вопросам;
- подготовка к экзамену.

В рамках изучения предусмотрено выполнение контрольных работ.

Студенты, не выполнившие работу или не прошедшие собеседование по ней, к экзамену по дисциплине не допускаются.

Организация выполнения контрольной работы включает несколько этапов. Прежде всего, студент должен изучить рекомендуемую учебную литературу и внимательно ознакомиться с методическими указаниями по выполнению контрольной работы.

Выполненная и надлежащим образом оформленная работа сдается на кафедру «Математика и информатика» в сроки, установленные учебным графиком. Если студент не допущен к защите, то работа должна быть доработана согласно замечаниям руководителя.

Критерии оценивания контрольной работы:

1) четкость структуры работы, представленной в содержании (рубрики дают однозначное понимание раскрываемой тематики и оформлены стилями заголовков соответствующего уровня);

2) наличие всех разделов, предложенных в методических указаниях, таблиц, иллюстраций (схем, рисунков и т.п.), ссылок на источники заимствованного материала;

3) наличие списка использованной литературы, оформленного в соответствии с правилами библиографического описания;

4) соответствие оформления расчетно-аналитической работы общим требованиям.

Контрольная работа оформляется в тетради или на компьютере с использованием текстового редактора:

- гарнитура шрифта – Times New Roman;
- размер шрифта – 14;
- межстрочный интервал – полуторный;
- размеры полей: левого – 3 см, верхнего – 1,5, правого – 1, нижнего – 1,5 см;
- ориентация – книжная;
- форматирование основного текста и ссылок – по ширине;
- цвет шрифта – черный;
- абзацный отступ – 1,5 см.
- нумерация страниц – справа, внизу.

Объем контрольной работы не должен превышать 25 страниц машинописного текста формата А4.

Каждую структурную часть контрольной работы нужно начинать с нового листа. Точка в конце заголовка структурной части работы не ставится.

Необходимо стремиться к ясности, краткости и самостоятельности изложения материала.

Цитаты и заимствованный материал должны сопровождаться ссылками на источники, описание которых необходимо привести в списке использованной литературы (например: [2, с. 15]).

Сокращение слов и использование аббревиатур, за исключением общепринятых, в контрольной работе не допускаются. Общепринятые аббревиатуры необходимо расшифровать в тексте контрольной работы при их первом употреблении.

Цифровой материал целесообразно представлять в виде таблиц. Для этого в правом верхнем углу помещают надпись «Таблица» и указывают ее порядковый номер. Таблицу снабжают тематическим заголовком, который располагают посередине страницы.

Приводимые в работе иллюстрации (схемы, диаграммы, графики, технические рисунки) должны быть выполнены четко, аккуратно и разборчиво. Рисунок должен иметь номер и подрисуночную подпись (например: Рис. 4. График функции $f(x)$).

На все таблицы и рисунки необходимо сделать ссылки в тексте работы.

На последней странице студент должен поставить подпись и дату сдачи контрольной работы на рецензирование.

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

лицензионное отечественное:

1. Антивирусная защита Kaspersky Endpoint Security.
2. Astra Linux.
3. LibreOffice.

лицензионное импортное:

ПО для ТСО лиц с ограниченными возможностями здоровья
свободно распространяемое:
Архиватор KDE (Ark).

11.2. Современные профессиональные базы данных и информационные справочные системы

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Информационно-образовательный портал Финансового университета:
<http://org.fa.ru>.

11.3. Сертифицированные программные и аппаратные средства защиты информации

Не используются.

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

Образовательный процесс по учебной дисциплине осуществляется в учебных аудиториях для проведения учебных занятий. При освоении дисциплины используются технические средства мультимедийной техники аудиторий.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения в сети «Интернет» и обеспечением доступа в электронно-образовательную среду Финансового университета.