

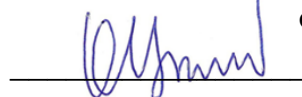
Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»

Омский филиал Финуниверситета

Кафедра «Естественно-научные и гуманитарные дисциплины»

УТВЕРЖДАЮ

Директор Омского филиала
Финуниверситета



О.Р. Онищенко
«23» июня 2026 г.

**ПРИЛОЖЕНИЕ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ
АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ В ЯЗЫКЕ PYTHON**

Год утверждения программы: 2023

Разработчики рабочей программы дисциплины:

Горохова Р.И., Макрушин С.В.

для студентов, обучающихся по направлению
подготовки

38.03.05 – Бизнес-информатика,

ОП «Цифровая трансформация управления бизнесом»
профили: «ИТ-менеджмент в бизнесе».

Составитель актуализации: Шамис В.А.

*Рекомендовано Ученым советом Омского филиала Финуниверситета
(протокол от «23» июня 2026 г. №34)*

*Одобрено кафедрой «Естественно-научные и гуманитарные дисциплины»
(протокол от «29» апреля 2026 г. №4)*

Омск 2026

Приложение к рабочей программе дисциплины разработал:
кандидат психологических наук, доцент Шамис В.А.

Алгоритмы и структуры данных в языке PYTHON. Приложение к рабочей программе дисциплины для студентов, обучающихся по направлению 38.03.05 – Бизнес-информатика, ОП «Цифровая трансформация управления бизнесом» профили: «ИТ-менеджмент в бизнесе» – Омск, 2026– 37 с.

© Омский филиал Финансового университета при
Правительстве Российской Федерации, 2026

Содержание

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	4
5.3.Содержание практических и семинарских занятий	4
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	6
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 (согласно таблицам 6, 7, 8)	9
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	13
7.1. Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы.....	13
7.2. Показатели и критерии оценивания компетенций	14
7.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, владений	19
7.4. Методические материалы, определяющие процедуру оценивания знаний, умений, владений	25
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	25
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	26
10. Методические указания для обучающихся по освоению дисциплины	26
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	28
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	28

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

Общая трудоёмкость дисциплины составляет 6 зачетных единиц.

Вид промежуточной аттестации – зачет, экзамен.

Вид учебной работы	Часы (очная)		
	Всего (в часах и з.е.)	Семестр 3 (в часах)	
Общая трудоёмкость дисциплины	144 (4 з.е.)	144 (4 з.е.)	
Контактная работа - аудиторные занятия	50	50	
Лекции			
Семинары	50	50	
Самостоятельная работа	94	94	
Вид текущего контроля	контрольные работы	контрольные работы	
Вид промежуточной аттестации	Зачет,	Зачет,	

5.3. Содержание практических и семинарских занятий

№ п/п	Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
1	Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
2	Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

3	Словари, множества и выражения-генераторы	Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python. Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
4.	Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
5	Работа с файлами и обработка исключительных ситуаций	Исключения. Инструкция try...except...else...finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert. Работа с файлами в Python, операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ...as и ее использование для файлов. <i>Источники: 8.1; 8.2; 8.3, 8.4</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
6	Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов. <i>Источники: 8.1; 8.2; 8.3, 8.4.</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
7.	Введение в объектно-ориентированное программирование	Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super(). <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

8.	Объектно-ориентированное программирование в Python	Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
9.	Введение в	Элементы функционального	Индивидуальное

	функциональное программирование	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. <i>Источники: 8.1; 8.2; 8.3, 8.4</i>	выполнение заданий, групповой разбор результатов выполнения заданий
10.	Функциональное программирование в Python	Реализация функций map, filter, reduce в Python. Итераторы в Python, итерируемый тип данных. Модуль itertools. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
11.	Структуры данных: массивы, стеки, очереди, списки	Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с динамическими массивами. Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

12.	Алгоритмы поиска и сортировки	Реализация на Python простых методов сортировки: обменные сортировки (с различными вариациями); сортировка выбором; сортировка (извлечение м); сортировка включением (вставками). Реализация на Python эффективных методов сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
-----	-------------------------------	---	---

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

Основная цель внеаудиторной самостоятельной работы бакалавра – закрепление и систематизация теоретических знаний, полученных в ходе лекционных занятий, формирование умений и навыков и овладение опытом творческой, исследовательской деятельности. Этот вид самостоятельной работы способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровней. Внеаудиторная самостоятельная работа является обязательной для каждого студента.

Внеаудиторная самостоятельная работа бакалавра в процессе освоения дисциплины включает в себя:

- для овладения знаниями: чтение текста (учебника, первоисточника, дополнительной литературы); составление плана текста; графическое изображение структуры текста; конспектирование текста; выписки из текста; работа со словарями и справочниками: ознакомление с нормативными документами; учебно-исследовательская работа; использование аудио- и видеозаписей, компьютерной техники и Интернета и др.;

- для закрепления и систематизации знаний: работа с конспектом лекции; работа над учебным материалом (учебника, первоисточника, дополнительной литературы, аудио- и видеозаписей); составление плана и тезисов ответа; составление таблиц для систематизации учебного материала; изучение нормативных материалов; ответы на контрольные вопросы; аналитическая обработка текста (аннотирование, рецензирование, реферирование и др.); подготовка тезисов сообщений к выступлению на семинаре, конференции; подготовка рефератов, докладов: составление библиографии, тематических кроссвордов и др.;

- для формирования умений: решение задач и упражнений по образцу; решение вариативных задач и упражнений; выполнение графиков, схем; выполнение расчетно-графических работ; решение ситуационных производственных (профессиональных) задач; подготовка к деловым играм; проектирование и моделирование разных видов и компонентов профессиональной деятельности; подготовка контрольных работ.

В качестве форм и методов контроля внеаудиторной самостоятельной работы использованы тестирование, контрольные работы.

Критериями оценки результатов внеаудиторной самостоятельной работы являются:

- уровень освоения учебного материала,
- умение использовать теоретические знания при выполнении практических задач,
- полнота общеучебных представлений, знаний и умений по изучаемой теме, к которой относится данная самостоятельная работа,
- обоснованность и четкость изложения ответа на поставленный по внеаудиторной самостоятельной работе вопрос,
- оформление отчетного материала в соответствии с известными или заданными преподавателем требованиями, предъявляемыми к подобного рода материалам.

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

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
1. Введение в программирование на Python	Среда программирования. Использование документации.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
2. Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
3. Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
4. Функции	Возвращение нескольких значений из функции. Распаковка и заpackовка параметров функции.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.

5. Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
6. Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.

7. Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
8. Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
9. Введение в функциональное программирование	Реализация декораторов с параметрами в Python.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.

10. Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
11. Структуры данных: массивы, стеки, очереди, списки	Реализация различных вариантов связанных списков на Python.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.
12. Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Использование компьютерной техники и Интернета. Работа с учебной литературой, работа со справочным материалом. Выполнение индивидуальных заданий.

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

Примерные вопросы к контрольным работам

Контрольная работа 1

1. Основные структуры данных Python: списки, кортежи, множества. Операции над ними.
2. Словари в Python: создание, доступ к элементам, методы словарей.
3. Генераторы списков, словарей и множеств. Примеры использования.
4. Итераторы и итерируемые объекты в Python. Протокол итерации.
5. Обработка исключений: блоки try, except, else, finally. Создание собственных исключений.
6. Работа с файлами в Python: открытие, чтение, запись, контекстный менеджер with.
7. Рекурсия в Python: принципы, примеры, ограничения.
8. Алгоритмы сортировки: пузырьковая сортировка, сортировка вставками. Реализация на Python.
9. Алгоритмы поиска: линейный и бинарный поиск. Реализация и оценка сложности.
10. Стек (stack): реализация, основные операции, примеры применения.
11. Очередь (queue): реализация, типы очередей, применение в задачах.
12. Связный список: односвязный и двусвязный список, реализация на Python.
13. Деревья: основные понятия, бинарное дерево, обход дерева.
14. Хеш-таблицы: принципы работы, коллизии, реализация в Python.
15. Алгоритмы на графах: представление графов, поиск в ширину (BFS).
16. Поиск в глубину (DFS) на графах. Реализация и применение.
17. Оценка сложности алгоритмов: нотация Big O, временная и пространственная сложность.

18. Жадные алгоритмы: принципы, примеры задач.
19. Динамическое программирование: основные идеи, примеры задач.
20. Модули и пакеты в Python: создание, импорт, организация кода.
21. Лямбда-функции и функциональное программирование в Python: map, filter, reduce.

В строке, содержащей перечень
товаров через запятую, оставить только те
слова, длина которых не менее 5 символов.

Ответ представить в виде строки. Пример:
строка 'Стол, Стул, Лампа, Ковёр, Ваза, Шкаф'
будет преобразована в:

'Стол, Лампа, Ковёр, Шкаф'.

1. Из списка списков, элементами которого являются данные о подразделениях, собрать строку, где каждый вложенный список преобразован в формат "Отдел: руководитель (бюджет)", а записи разделены точкой с запятой. Пример: [['IT', 'Иванов', '500000'], ['HR', 'Петрова', '300000']] будет преобразован в строку 'IT: Иванов (500000); HR: Петрова (300000)'.
2. Напишите программу на Python для заполнения шаблона делового письма. Дана строка: "Уважаемый {client}, ваш заказ №{order} на сумму {sum} руб. передан в доставку." и словарь с данными. Выведите итоговый текст с подставленными значениями. Пример вывода: Уважаемый Алексей, ваш заказ №1042 на сумму 12500 руб. передан в доставку.
3. Напишите программу на Python для получения количества логических ядер процессора и названия операционной системы, на которой запущен интерпретатор.
4. Напишите программу на Python, которая принимает две даты в формате "ГГГГ-ММ-ДД" и вычисляет количество полных дней между ними. Пример ввода: "2024-01-01", "2024-03-15". Ожидаемый вывод: 74 дня.
5. Напишите программу на Python, которая принимает строку с электронным адресом сотрудника и извлекает из неё логин (часть до @) и домен компании (часть после @). Пример: manager@techcorp.ru → Вывод: Логин: manager, Домен: techcorp.ru.
6. Напишите программу на Python для нахождения общих элементов двух списков идентификаторов клиентов и вывода результата в виде отсортированного списка. Пример: list1 = ["C01", "C02", "C05"], list2 = ["C02", "C03", "C05"]. Вывод: ['C02', 'C05'].
7. Напишите программу на Python, которая принимает трёхзначное целое число и меняет местами его первую и последнюю цифры. Пример: введено 321. Ожидаемый результат: 123.

Примеры задач для индивидуального решения

Задача 1

Дан список показателей эффективности процессов: `metrics = [12, 45, 23, 67, 8, 34, 51, 9, 100]`.

Выведите все элементы, значения которых меньше 50.

Задача 2

Даны списки идентификаторов клиентов из двух филиалов:

`branch_A = ['CL-101', 'CL-105', 'CL-112', 'CL-120']`

`branch_B = ['CL-105', 'CL-108', 'CL-112', 'CL-115', 'CL-120']`

Необходимо вернуть список, состоящий из идентификаторов, которые присутствуют в обоих филиалах одновременно.

Задача 3

Отсортируйте словарь, содержащий данные о квартальной выручке по направлениям, по значению в порядке возрастания и убывания.

`revenue = {'Консалтинг': 150000, 'Разработка': 280000, 'Поддержка': 120000, 'Обучение': 95000}`

Задача 4

Напишите программу для объединения трёх словарей, содержащих параметры конфигурации системы (базовые, пользовательские, сессионные), в один итоговый словарь. При совпадении ключей приоритет имеет значение из последнего переданного словаря.

Пример: `cfg1 = {'theme': 'light', 'lang': 'ru'}`, `cfg2 = {'lang': 'en', 'debug': False}`, `cfg3 = {'debug': True, 'version': 2}`

Задача 5

Найдите два ключа с наименьшими значениями в словаре нагрузки серверов:

`server_load = {'Node_A': 85, 'Node_B': 42, 'Node_C': 67, 'Node_D': 29, 'Node_E': 55}`

Выведите их в порядке возрастания нагрузки.

Задача 6

Напишите функцию, которая принимает целое неотрицательное число и основание системы счисления (от 2 до 36) и возвращает строковое представление этого числа в заданной системе. Для значений ≥ 10 используйте заглавные латинские буквы (A=10, B=11, ..., Z=35).

Пример: `to_base(255, 16) → 'FF'`, `to_base(10, 2) → '1010'`.

Распределение максимальных баллов по видам работы и формирование итоговой оценки

Требования к результатам освоения дисциплины	Оценка или зачет
<i>Текущая работа в семестре (текущий контроль)</i>	
Выполнение и защита текущих заданий на занятиях	10 баллов
Выполнение контрольной работы в семестре	10 баллов
Выполнение и защита домашней контрольной работы	20 баллов

<i>Промежуточный контроль</i>	
Знание основных положений дисциплины	20 баллов
Выполнение практического задания	40 баллов
Итого	100 баллов
<i>Итоговая оценка</i>	
Зачет /Экзамен ставится при условии набора	49 баллов и выше

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

7.1. Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы

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

7.2. Показатели и критерии оценивания компетенций

Зачет (экзамен) по дисциплине выставляется студенту при условии сформированности по каждой компетенции как минимум порогового уровня.

Результаты обучения по дисциплине	Шкала и критерии оценки результатов			
	Высокий уровень (отлично)	Продвинутый уровень (хорошо)	Пороговый уровень (удовлетворительно)	Пороговый уровень не достигнут (неудовлетворительно)
УК-4 Способность использовать прикладное программное обеспечение при решении профессиональных задач				
<i>1. Использует основные методы и средства получения, представления, хранения и обработки данных.</i>				
Знать: основные методы и средства получения, представления, хранения управленческой информации. Уметь: применять современные методы и средства получения, представления, хранения и обработки информации в финансовой сфере.	Глубоко знает: основные методы и средства получения, представления, хранения и обработки управленческой информации. Уверенно умеет: применять современные методы и средства получения, представления, хранения и обработки информации в финансовой сфере.	Знает: основные методы и средства получения, представления, хранения и обработки управленческой информации. Умеет: применять современные методы и средства получения, представления, хранения и обработки информации в финансовой сфере.	Частично знает: основные методы и средства получения, представления, хранения и обработки управленческой информации. Частично умеет: применять современные методы и средства получения, представления, хранения и обработки информации в финансовой сфере.	Не знает: основные методы и средства получения, представления, хранения и обработки управленческой информации. Не умеет: применять современные методы и средства получения, представления, хранения и обработки информации в финансовой сфере.
<i>2. Демонстрирует владение профессиональными пакетами прикладных программ.</i>				

Знать: профессиональные пакеты прикладных программ. Уметь: использовать профессиональные пакеты прикладных программ.	Глубоко знает: профессиональные пакеты прикладных программ. Уверенно умеет : использовать профессиональные	Знает: профессиональные пакеты прикладных программ. Умеет: использовать профессиональные пакеты прикладных	Частично знает: профессиональные пакеты прикладных программ. Частично умеет: использовать профессиональные пакеты прикладных программ.	Не знает: профессиональные пакеты прикладных программ. Не умеет: использовать профессиональные пакеты прикладных программ.
--	--	--	--	--

	пакеты прикладны х программ.	программ.		
<i>3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.</i>				
Знать: функциона л прикладного программного обеспечения. Уметь: осуществлят ь обоснованный выбор программных средств для решения профессиональных задач.	Глубоко знае т: функционал прикладного программного обеспечения. Уверенно умеет: осуществлять обоснованный выбор программных средств для решени я профессиональных задач.	Знает: функциона л прикладного программного обеспечения. Умеет: осуществлять обоснованный выбор программных средств для решени я профессиональных задач.	Частично знае т: функционал прикладного программного обеспечения. Частично умее т: осуществлять обоснованный выбо р программных средств для решения профессиональных задач.	Не знает: функционал прикладного программного обеспечения. Не умеет: осуществлят ь обоснованный выбор программных средств для решения профессиональных задач.
<i>4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.</i>				

Знать: виды и инструментарий прикладного программного обеспечения для решения конкретных прикладных задач. Уметь: использовать прикладное программное обеспечение для решения конкретных прикладных задач.	Глубоко знает: виды и инструментарий прикладного программного обеспечения для решения конкретных прикладных задач Уверенно умеет: использовать прикладное программное обеспечение для решения конкретных прикладных задач.	Знает: виды и инструментарий прикладного программного обеспечения для решения конкретных прикладных задач Умеет: использовать прикладное программное обеспечение для решения конкретных прикладных задач.	Частично знает: виды и инструментарий прикладного программного обеспечения для решения конкретных прикладных задач Частично умеет: использовать прикладное программное обеспечение для решения конкретных прикладных задач.	Не знает: виды и инструментарий прикладного программного обеспечения для решения конкретных прикладных задач Не умеет: использовать прикладное программное обеспечение для решения конкретных прикладных задач.
УК-10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач				
1. Четко описывает состав и структуру требуемых данных информации, грамотно реализует процессы их сбора, обработки и интерпретации				

Знать основные принципы описания состава и структуры данных и информации и методы её обработки. Уметь осуществлять процесс поиска, хранения и обработки информации, используя технические средства.	Глубоко знает : основные принципы описания состава и структуры данных и информации и методы её обработки. Уверенно умеет : осуществлять процесс поиска, хранения и обработки информации, используя технические средства.	Знает: основные принципы описания состава и структуры данных и информации и методы её обработки. Умеет: осуществлять процесс поиска, хранения и обработки информации, используя технические средства.	Частично знает : основные принципы описания состава и структуры данных и информации и методы её обработки. Частично умеет : осуществлять процесс поиска, хранения и обработки информации, используя технические средства.	Не знает: основные принципы описания состава и структуры данных и информации и методы её обработки. Не умеет: осуществлять процесс поиска, хранения и обработки информации, используя технические средства.
2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability				
Знать современные приемы обоснования сущности происходящего,	Глубоко знает : современные приемы обоснования	Знает: современные приемы обоснования сущности	Частично знает : современные приемы обоснования	Не знает: современные приемы обоснования сущности

технологии , обобщения и систематизации информации. Уметь применить на практике современные приёмы выявления закономерности происходящего, использовать современные технологии , обобщения и	анализа	и происходящего, технологии анализа , обобщения и систематизации информации.	сущност	происходящего, технологии анализа , обобщения и систематизации информации.	и происходящего, технологии анализа , обобщения и систематизации информации.	сущност	происходящего технологии анализа , обобщения и систематизации информации.
		Уверенно : применить на практике современные приём ы выявления закономерности происходящего, использовать	умеет	Умеет: применить на практике современные приёмы выявления закономерности происходящего, использовать современные	Частично : применить на практике современные приём ы выявления закономерности происходящего, использовать	умеет	применить на практике современны е приёмы выявлени я закономерности происходящего, использовать современны е технологии анализа , обобщения и систематизации информации.

систематизации информации.	современные технологии анализа, обобщения и систематизации информации.	технологии анализа , обобщения и систематизации информации.	современные технологии анализа, обобщения и систематизации информации.	
<i>3. Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных</i>				

<i>групп</i>					
Знать	эффективные	Глубоко	знает	Знает:	эффективные
способы	идентификации	:	эффективные	способы	идентификации
общих свойств элементов,	образующих группы	и	общих свойств элементов,	образующих группы	и
оценивает	полнот	оценивает	полнот	оценивает	полнот
у результатов	классификации, показывает	у результатов	классификации, показывает	у результатов	классификации, показывает
прикладное назначение	классификационных групп.	прикладное назначение	классификационных групп.	прикладное назначение	классификационных групп.
Уметь	формулировать	Уверенно	умеет	Частично	умеет
признак	классификации,	признак	классификации,	признак	классификации,
выделяет соответствующие	групп.	выделяет соответствующие	групп.	выделяет соответствующие	групп.
ему группы однородных		ему группы однородных		ему группы однородных	

«объектов».	формулировать призна к классификации, выделяет соответствующие ему	признак классификации, выделяет соответствующие ем у группы однородных	формулировать призна к классификации, выделяет соответствующие ем у группы однородны х	«объектов».
4. Грамотно, логично, аргументированно формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности				
Знать способы грамотно, логично, аргументированно формировать собственные суждения и оценки.	Глубоко знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки.	Знает: способ грамотно, логично, аргументированно формировать собственные суждения и оценки.	Частично знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки.	Не знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки.
Уметь четко отличать факты от мнений, интерпретаций, оценок и т.д.	Уверенно умеет : отличать факты от мнений, интерпретаций, оценок	Умеет: четко отличать факты от мнений, интерпретаций, оценок	Частично умеет: четк о отличать факты от мнений, интерпретаций, оценок	Не умеет: четко отличать факты от мнений, интерпретаций, оценок и т.д.

<i>5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания</i>				
Знать методические	Глубоко знает :	Знает: методические	Частично знает:	Не знает: методические

приемы, применяемые при системном анализе проблемы или объекта. Уметь аргументированно и логично представлять свою точку зрения	методические приемы, применяемые при системном анализе проблемы или объекта. Уверенно умеет аргументированно и логично представлять свою точку зрения	приемы, применяемые при системном анализе проблемы или объекта. Умеет аргументированно и логично представлять свою точку зрения	методические приемы, применяемые при системном анализе проблемы или объекта. Частично умеет аргументированно и логично представлять свою точку зрения	приемы, применяемые при системном анализе проблемы или объекта. Не умеет аргументированно и логично представлять свою точку зрения
---	---	--	---	--

7.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, владений.

Промежуточная аттестация проводится в форме зачета, экзамена.

Наименование компетенции	Наименование индикаторов достижения компетенции	Типовые контрольные задания
УК – 4 Способность использовать прикладное программное обеспечение при решении профессиональных задач	1. <i>Использует основные методы и средства получения, представления, хранения и обработки данных.</i>	Вопросы к зачету 1. Функция. Получение информации о функции. Способы передачи параметров при вызове функции. 2. Передача переменного количества параметров (именованных и не именованных) в функции Python. 3. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре. Практико-ориентированные задания 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. 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. <i>Демонстрирует владение профессиональными пакетами прикладных программ.</i>	Практико-ориентированные задания Установите пакет Anaconda запустите программу JupyterNotebook составьте тестовый код на Python. Отсортируйте словарь по значению в порядке возрастания и убывания. Составить программу калькулятор на языке программирования Python.

	Напишите программу на Python, чтобы найти расположение исходных
--	---

		кодов модуля Python. Название операционной системы: POSIX Название плат формы: Linux Релиз плат формы : 4.4.0-47-родовой
	3. <i>Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.</i>	Практико-ориентированные задания Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных. Напишите программу для слияния нескольких словарей в один. Напишите программу на Python, чтобы получить новую строку из заданной строки, где «Is» было добавлено вперед. Если заданная строка уже начинается с «Is», вернуть строку без изменений.
	4. <i>Использует прикладное программное обеспечение для решения конкретных прикладных задач.</i>	Вопросы к зачету 1. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python. Практико-ориентированные задания Создайте для класса «студент» функцию, которая считает средний балл по трем дисциплинам, оценки по которым вводятся пользователями. Реализуйте программу двумя способами: без использования конструктора __init__ и с ним. 2. Создайте функцию, которая считает и выводит в файл заработную плату сотрудника склада в зависимости от количества отработанных дней в месяце. Оклад является защищаемым атрибутом класса «сотрудник».
УК – 10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	1. <i>Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации.</i>	Вопросы к зачету 1. Что такое класс в Python? Каковы его основные характеристики? 2. Дайте определение атрибута класса и опишите его основные особенности. 3. Как осуществляется перегрузка специальных методов класса в Python? Практико-ориентированные задания Студент регистрируется на информационно-образовательном портале вуза. Составьте концептуальную модель главного цикла обработчика событий, который включает основные события, обработчик событий, вызываемые методы. Вы принимаете от пользователя последовательность чисел, разделённых

		запятой. Составьте список и кортеж с этими числами.
	2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability.	Вопросы к экзамену 1. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность. 2. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка. Практико-ориентированные задания Опишите закономерности/variability в синтаксических конструкциях Python для работы со списками и словарями. Определите количество элементов списка, значения которых совпадают с ключами заданного словаря и количество элементов списка, значения которых равны значениям в словаре.
	3. Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.	Вопросы к экзамену 1. Связанные списки: однонаправленные и двунаправленные – принцип реализации. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве. 2. Деревья, создание деревьев с помощью структуры список, узлов и ссылок. Операции с деревьями. Обход дерева. Практико-ориентированные задания Опишите подходы используемые в Python для группировки функций, ориентированных на обработку однотипных объектов и решение близких задач. Выполните классификацию с использованием бинарного дерева. Оцените количество операций и глубину дерева. Приведите примеры классов объектов, элементы которых могут иметь разную реализацию. Напишите программу, которая выводит чётные числа из заданного списка и останавливается, если встречается число 237.
	4. Грамотно, логично, аргументировано формирует	Вопросы к экзамену 1. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работа с array в Python.

	<i>е суждения собственны и оценки.</i>	
--	--	--

	<i>Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участника в деятельности.</i>	2. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность. Практико-ориентированные задания Сформируйте собственные суждения о применимости Python для решения задач машинного обучения. Нужно проверить, все ли числа в последовательности уникальны. Создайте класс «студент» (название можно написать на русском языке), который будет содержать неизменяемые атрибуты: ФИО, направление подготовки и год поступления. Курс является изменяемым атрибутом. Создайте два экземпляра класса и протестируйте программу.
	5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	Вопросы к экзамену 1.Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python. 2.Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python. 3.Методы классов и статические переменные и методы в Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями. Практико-ориентированные задания Пользуясь предыдущим примером, создайте окно с названием «Студенты», где разместите кнопки «Вывести список», «Ввести данные», «Создать запрос», «Выйти» тремя рассмотренными способами. Сравните результат и прокомментируйте преимущества и недостатки использования данных методов. Сформулируйте задачи по обработке массивов, которые вы сможете решить на основе языка Python Напишите программу, которая принимает имя файла и выводит его расширение. Если расширение у файла определить невозможно, выбросите исключение. Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.

Примерные вопросы для подготовки 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.
22. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля.

23. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

Примерные вопросы для подготовки к экзамену

7.4. Классы и объекты в Python. Конструктор `__init__`, магические методы `__str__` и `__repr__`. Различие между атрибутами класса и экземпляра.

- 1. Наследование и полиморфизм.** Множественное наследование, разрешение порядка методов (MRO), функция `super()`. Абстрактные базовые классы (abc модуль).
- 2. Инкапсуляция и управление доступом.** Публичные, защищённые и приватные атрибуты. Свойства (`@property`), геттеры и сеттеры.
- 3. Магические методы (dunder methods).** Реализация арифметических операций, сравнения, вызова объектов. Протоколы последовательностей и отображений.
- 4. Декораторы функций и классов.** Встроенные декораторы `@staticmethod`, `@classmethod`, `@property`. Создание декораторов с параметрами и сохранение метаданных (`functools.wraps`).
- 5. Генераторы и итераторы.** Протокол итерации (`__iter__`, `__next__`), ленивые вычисления. Генераторные выражения и `yield from`.
- 6. Менеджеры контекста.** Протокол `__enter__` и `__exit__`, модуль `contextlib`. Применение для управления транзакциями и соединениями с БД.
- 7. Обработка JSON данных.** Модуль `json`, сериализация и десериализация. Работа с вложенными структурами, кастомные `encoder/decoder`.
- 8. Работа с датой и временем.** Модули `datetime`, `time`, `calendar`. Часовые пояса (`pytz`, `zoneinfo`), форматирование и парсинг дат.
- 9. Регулярные выражения в Python.** Модуль `re`, шаблоны, группы, поиск и замена. Валидация email, телефонов, парсинг логов.
- 10. Сортировка и поиск.** Алгоритмы сортировки (быстрая, слиянием, кучей). Бинарный поиск, модуль `bisect`. Функции `sorted()` и метод `sort()`.
- 11. Анализ сложности алгоритмов.** Нотация Big O, временная и пространственная сложность. Оценка эффективности алгоритмов для бизнес-задач.
- 12. Структуры данных: стек, очередь, дек.** Реализация через `list`, `collections.deque`. Применение в задачах обработки транзакций и очередей сообщений.
- 13. Деревья и графы.** Представление графов (матрица смежности, списки смежности). Обход в ширину (BFS) и глубину (DFS).
- 14. Хеш-таблицы и хеш-функции.** Принцип работы словарей и множеств. Коллизии, методы разрешения, требования к хешируемости.
- 15. Рекурсия и динамическое программирование.** Примеры рекурсивных алгоритмов. Мемоизация (`functools.lru_cache`), табуляция.

16. **Многопоточность и параллелизм.** Модули threading, multiprocessing. GIL (Global Interpreter Lock), CPU-bound и I/O-bound задачи.
17. **Асинхронное программирование.** Модуль asyncio, корутины async/await. Event loop, конкурентное выполнение I/O операций.
18. **Работа с базами данных SQLite.** Модуль sqlite3, выполнение SQL-запросов, параметризация. Контекстные менеджеры для транзакций.
19. **ORM SQLAlchemy.** Модели, сессии, отношения между таблицами. Query API, ленивая и жадная загрузка данных.
20. **Тестирование кода.** Модули unittest, pytest. Фикстуры, параметризация тестов, покрытие кода (coverage).
21. **Логирование в приложениях.** Модуль logging, уровни логирования, handlers, formatters. Структурированное логирование для бизнес-аналитики.
22. **Работа с внешними API.** Модуль requests, GET/POST запросы, аутентификация, обработка JSON ответов. Работа с REST API бизнес-сервисов.
23. **Парсинг HTML и XML.** Библиотеки BeautifulSoup, lxml. Извлечение данных из веб-страниц, навигация по DOM-дереву.
24. **Упаковка и распространение кода.** Структура проекта, setup.py, pyproject.toml. Создание pip-пакетов, публикация в PyPI, управление зависимостями.

7.5. Методические материалы, определяющие процедуру оценивания знаний, умений, владений

Соответствующие приказы, распоряжения ректората, директора филиала о контроле знаний студентов.

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

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С. Р. Гуриков. — Москва: ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат).—ЭБС ZNANIUM.com.-URL: <https://znanium.com/catalog/product/1356003>— Текст: электронный.
2. Шкодина, Т. А. Алгоритмы и структуры данных в Python : лабораторный практикум : учебное пособие : [16+] / Т. А. Шкодина ; Ростовский государственный экономический университет (РИНХ), Кафедра информационных систем и прикладной информатики. — Ростов-на-Дону : Издательско-полиграфический комплекс РГЭУ (РИНХ), 2022. — 80 с. : ил., табл. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=704757> (дата обращения: 11.04.2026). — ISBN 978-5-7972-3037-3. — Текст : электронный.
3. Калитвин, В. А. Введение в программирование на Python : учебное пособие : [16+] / В. А. Калитвин ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. — Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2023. — 85 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=714538> (дата обращения: 11.04.2026). — Библиогр. в кн. —

ISBN 978-5-907655-86-7. – Текст : электронный.

Дополнительная литература:

4. Богун, В. В. Программирование задач по финансовой математике на языке программирования Python : учебное пособие : [16+] / В. В. Богун ; Финансовый университет при Правительстве Российской Федерации, Департамент математики. – Москва : Прометей, 2025. – 172 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=721368> (дата обращения: 11.04.2026). – Библиогр. в кн. – ISBN 978-5-00172-780-4. – Текст : электронный.
5. Тазиева, Р. Ф. Python. Обработка данных : учебно-методическое пособие : [16+] / Р. Ф. Тазиева, А. Н. Титов ; Казанский национальный исследовательский технологический университет. – Казань : Казанский национальный исследовательский технологический университет (КНИТУ), 2022. – 104 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=702252> (дата обращения: 11.04.2026). – Библиогр. в кн. – ISBN 978-5-7882-3171-6. – Текст : электронный.

9. Перечень ресурсов информационно-телекоммуникационной сети

«Интернет», необходимых для освоения дисциплины

1. Сайты для изучения Python: полезная подборка сайтов с руководствами
https://codernet.ru/articles/python/sajtyi_dlya_izucheniya_python_poleznaya_podbor_ka_sajtov_s_ukovodstvami/
2. Python Data Analysis Library [Электронный ресурс]: сайт. – Режим доступа: <http://pandas.pydata.org/>
3. Python Documentation [Электронный ресурс]: сайт. – Режим доступа: <http://python.org/doc/>.
- 4 Python Standard Library [Электронный ресурс]: сайт. – Режим доступа: <https://docs.python.org/2/library/>.
6. Scikit-learn Machine Learning in Python [Электронный ресурс]: сайт. – Режим доступа: <http://scikit-learn.org>.
7. Официальный сайт продукта <https://www.python.org/>.
8. The Python Tutorial // <https://docs.python.org/3/tutorial/index.html>.
9. The Python Standard Library // <https://docs.python.org/3/library/index.html>.
10. SciPy // <http://docs.scipy.org/doc/scipy/reference/>
11. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>.
12. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>.
13. Электронно-библиотечная система BOOK.RU <http://www.book.ru>.
14. Электронно-библиотечная система
«Университетская библиотека ОН- ЛАЙН» <http://biblioclub.ru/>.
15. Электронно-библиотечная система Znanium <http://www.znaniy.com>.
16. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>.

1. Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет.

Студентам необходимо ознакомиться:

- с содержанием рабочей программы дисциплины (далее - РПД), с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, методическими разработками по данной дисциплине, имеющимися на образовательном портале и сайте кафедры, с графиком консультаций преподавателей данной кафедры.

При подготовке к лекционным занятиям студентам необходимо:

- перед каждой лекцией просматривать рабочую программу дисциплины, что позволит сэкономить время на записывание темы лекции, ее основных вопросов, рекомендуемой литературы;
- на отдельные лекции приносить соответствующий материал на бумажных носителях, представленный лектором на портале или присланный на «электронный почтовый ящик группы» (таблицы, графики, схемы). Данный материал будет охарактеризован, прокомментирован, дополнен непосредственно на лекции;
- перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях. Не оставляйте «белых пятен» в освоении материала.

При подготовке к практическим занятиям студентам следует:

- приносить с собой рекомендованную преподавателем литературу к конкретному занятию;
- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин) и не имеющим выполненного практического задания, рекомендуется не позже чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, рассматриваемой на занятии. Студенты, не отчитавшиеся по каждой не проработанной ими на занятиях теме к началу зачетной сессии, упускают возможность получить положенные баллы за работу в соответствующем семестре.

Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По каждой теме учебной дисциплины студентам

предлагается перечень заданий для самостоятельной работы.

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

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

Комплект лицензионного программного обеспечения:

1. Astra Linux Special Edition

Современные профессиональные базы данных и информационные справочные системы:

1. Справочная правовая система «КонсультантПлюс» (<http://www.consultant.ru>).
2. Информационно-образовательный портал Финансового университета.
- <https://org.fa.ru>, <http://campus.fa.ru>

Сертифицированные программные и аппаратные средства защиты информации – не предусмотрено.

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

Освоение дисциплины предполагает использование учебной аудитории для проведения лекционных и семинарских занятий с необходимыми техническими средствами (компьютер, проектор, интерактивная доска, презентации в системе Power Point, банк данных видео по курсу), специально оборудованные аудитории и компьютерные классы с локальной сетью и выходом в Интернет.