

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



Т.В. Ивашкевич
«16» мая 2025 г.

ПРИЛОЖЕНИЕ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

Современные технологии программирования

Год утверждения программы: 2024

Разработчик рабочей программы дисциплины:
Е.В. Пальчевский

для студентов, обучающихся по направлению подготовки
38.03.05 – Бизнес-информатика,
ОП «Цифровая трансформация управления бизнесом»
профили: «ИТ-менеджмент в бизнесе»

Составитель актуализации: Селезнева Е.В.

*Рекомендовано Ученым советом Омского филиала Финуниверситета
(протокол от «16» мая 2025 г. №23)*

*Одобрено кафедрой «Естественно-научных и гуманитарных дисциплин»
(протокол от «30» апреля 2025 г. №4)*

Омск 2025

Приложение к рабочей программе дисциплины разработал:
кандидат тех. наук, доцент Селезнева Е.В.

Современные технологии программирования. 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, 2025 – 29 с.

© Омский филиал Финансового университета при
Правительстве Российской Федерации, 2025

Содержание

5.2. Содержание практических и семинарских занятий	4
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	8
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю (согласно таблицам 6, 7, 8).....	10
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	14
7.1. Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы	14
7.2. Показатели и критерии оценивания компетенций.....	15
7.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, владений	20
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	26
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	27
10. Методические указания для обучающихся по освоению дисциплины	28
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	29
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	29

5.2. 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	Обзор платформы и языка программирования Java	Комплект разработчика (Java development kit), среда выполнения Java (Java Runtime Environment). Интегрированная среда разработки (Integrated development Environment) IntelliJ Idea. Установка. Создание нового проекта, системы сборки проектов Maven. Структура проекта. Интерфейс IDE IntelliJ Idea. Создание исполняемого файла. Использование инструментов командной строки. Прimitивные и ссылочные типы данных, литералы. Операции. Преобразование типов. Переменные: объявление, инициализация. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
2	Основные языковые конструкции Java	Консольный ввод-вывод данных, команды ввода-вывода. Переменные и основные типы переменных. Объявление и инициализация переменных. Присвоение данных между разными типами переменных. Инициализация переменных. Специальные операторы. Операторы «вычисление остатка» и «вычисление частного». Логический тип (булевы значения). Ветвление (конструкция «if-else»). Класс Math. Циклы for и while. Массивы и методы. Строковые и символьные переменные. Классы String, StringBuilder, StringBuffer. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

3	Базовые принципы объектно-ориентированного программирования	Объекты и классы. Состояние и поведение. Построение классов в соответствии с принципами инкапсуляции, полиморфизма и наследование. Перегрузка конструкторов, методов. Области видимости, модификаторы доступа. Применение ключевого слова static. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
4.	Классы, методы и объекты Java	Объявление классов. Метод main. Отношения между классами. Имена классов, импорт. Инициализаторы. Преобразование ссылочных типов. Ключевые слова this и super. Обобщенные методы и конструкторы. Применение ключевого слова final к переменным экземпляра класса. <i>Источники: 8.1; 8.2; 8.3</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
5	Интерфейсы. Исключения. Отладка программ	Объявление, реализация и применение интерфейсов. Наследование интерфейсов. Методы интерфейсов: по умолчанию, статические. Ошибки при работе программы. Исключения (Exceptions). Конструкция try-catch-finally. Использование оператора throw. Отладка программ в среде программирования IDE IntelliJ IDEA. Запуск отладчика, точки останова. <i>Источники: 8.1; 8.2; 8.3,8.4</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
6	Коллекции	Интерфейсы Collection, Set, List, Map, Iterator. Каркас коллекций. Представления и оболочки. Унаследованные коллекции. Абстрактные классы, используемые при работе с коллекциями. Интерфейс Comparator. Применение ArrayList. TreeMap. Применение очередей. Алгоритмы: сортировка, двоичный поиск. <i>Источники: 8.1; 8.2; 8.3,8.4.</i>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
7.	Потоковая организация системы ввода-вывода. Многопоточность	Консольный ввод, запись консольного вывода. Классы-реализации потоков данных. Классы ByteArrayInputStream и ByteArrayOutputStream. Классы FileInputStream и FileOutputStream. StringBufferInputStream. Классы FilterInputStreem и FilterOutputStream. BufferedInputStream и	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

		BufferedOutputStream. DataInputStream и DataOutputStream. Класс Thread и интерфейс Runnable. Жизненный цикл потока. Фоновые задачи. Управление потоками и приоритеты потоков. Синхронизация. Приостановка, возобновление и остановка потока. <i>Источники: 8.1; 8.2; 8.3</i>	
8.	Работа с базами данных в Java. Основы объектно-реляционного отображения.	Установление соединения с системами управления баз данных MySQL и PostgreSQL. Драйверы JDBC. Установление соединения, классы DriverManager и DataSource, интерфейс Connection. Классы Statement, PreparedStatement. Запросы к базе данных. Добавление, удаление, изменение информации в базе данных. Обработка результатов выполнения запроса к базе данных, интерфейс ResultSet. Закрывание соединения. Драйверы JPA. Hibernate framework. Классы Entity, встраиваемые классы, наследование маппингов, fetch стратегии. Жизненный цикл Entity, EntityManager. <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 запаковка 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. Операции над основными числовыми типами данных.
2. Операции над булевыми переменными.
3. Динамическая типизация в Python.
4. Преобразование типов в Python.
5. Создание строк в Python.
6. Организация и пример цикла while в Python.
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.

Контрольная работа 2

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 функции.
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.
21. Наследование.
22. Абстрактные методы класса.
23. Статические методы класса.
24. Свойства класса.
25. Исключения. Обработка исключений.
26. Пользовательские исключения.

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 'Еeny, тeenу, mineу, мое'
3. Напишите программу на Python для печати следующей строки в определенном формате (см. Вывод). Пример с троки: «Мерцай, мерцай, маленькая звезда, Как я удивляюсь, кто ты! Вверх над миром так высоко, Как алмаз в небе. Мерцай, мерцай, маленькая звезда, Как я удивляюсь, кто ты » Вывод: Мерцание, мерцание, маленькая звезда, Как я жажду узнать, кто ты! Подняться над миром так высоко, Как алмаз в небе. Мерцание, мерцание,

маленькая звезда, Как я жажду узнать, кто ты.

4. Напишите программу на Python, чтобы получить версию Python, которую вы используете.
5. Напишите программу на Python для отображения текущей даты и времени. Пример вывода: Текущая дата и время: 2014-07-05 14:34:14
6. Напишите программу на Python, которая будет принимать имя пользователя от пользователя, и распечатайте его расширение. Пример имени файла: abc.java Вывод: Java.
7. Напишите программу на Python для отображения первого и последнего цветов из следующего списка. `color_list = [" Красный", " Зеленый", " Белый", " Черный"]`
8. Напишите программу на Python, которая принимает целое число (n) и вычисляет значение $n + nn + nnn$. Примерное значение n равно 5. Ожидаемый результат: 615

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.
5. Напишите программу на Python, которая принимает радиус круга а от пользователя и вычисляет площадь. Пример вывода: `г = 1,1 Площадь= 3.8013271108436504`
6. Напишите программу на Python, которая принимает имя и фамилию пользователя и печатает их в обратном порядке с пробелом между ними.
7. Напишите программу на Python, которая принимает от пользователя последовательность чисел, разделенных запятыми, и генерирует список и

кортеж с этими числами. Примерные данные: 3, 5, 7, 23 Выход: Список : ['3', '5', '7', '23'] Кортеж : («3», «5», «7», «23»)

8.Напишите программу на Python для печати календаря с указанным месяцем и годом. Примечание. Используйте модуль «календарь».

Примеры задач для индивидуального решения

Задача 1

Есть список **a** = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89].

Выведите все элементы, которые меньше 5.

Задача 2

Даны списки:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89];

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

Нужно вернуть список, который состоит из элементов, общих для этих двух списков.

Задача 3

Отсортируйте словарь по значению в порядке возрастания и убывания.

Задача 4

Напишите программу для слияния нескольких словарей в один.

Задача 5

Найдите три ключа с самыми высокими значениями в словаре **my_dict** = {'a':500, 'b':5874, 'c': 560, 'd':400, 'e':5874, 'f': 20}.

Задача 6

Напишите код, который переводит целое число в строку, при том что его можно применить в любой системе счисления.

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 пакеты прикладных программ.	Знает: профессиональные пакеты прикладных программ. Умеет: использовать профессиональные пакеты прикладных программ.	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 прикладных программ.	программ.		
<i>3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.</i>				
Знать: функционал прикладного программного обеспечения. Уметь: осуществлять обоснованный выбор программных средств для решения профессиональных задач.	Глубоко знает: функционал прикладного программного обеспечения. Уверенно умеет: осуществлять обоснованный выбор программных средств для решения профессиональных задач.	Знает: функционал прикладного программного обеспечения. Умеет: осуществлять обоснованный выбор программных средств для решения профессиональных задач.	Частично знает: функционал прикладного программного обеспечения. Частично умеет: осуществлять обоснованный выбор программных средств для решения профессиональных задач.	Не знает: функционал прикладного программного обеспечения. Не умеет: осуществлять 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				
<i>1. Четко описывает состав и структуру требуемых данных информации, грамотно реализует процессы их сбора, обработки и интерпретации</i>				

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.
<i>2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability</i>				
Знать современные приемы обоснования сущности происходящего, технологии анализа, обобщения и систематизации информации. Уметь применить на практике современные приёмы выявления закономерности происходящего, использовать современные технологии анализа, обобщения и систематизации информации.	Глубоко знает: современные приемы обоснования сущности происходящего, технологии анализа, обобщения и систематизации информации. Уверенно умеет: применить на практике современные приёмы выявления закономерности происходящего, использовать современные технологии анализа, обобщения и систематизации информации.	Знает: современные приемы обоснования сущности происходящего, технологии анализа, обобщения и систематизации информации. Умеет: применить на практике современные приёмы выявления закономерности происходящего, использовать современные технологии анализа, обобщения и систематизации информации.	Частично знает: современные приемы обоснования сущности происходящего, технологии анализа, обобщения и систематизации информации. Частично умеет: применить на практике современные приёмы выявления закономерности происходящего, использовать современные технологии анализа, обобщения и систематизации информации.	Не знает: современные приемы обоснования сущности происходящего, технологии анализа, обобщения и систематизации информации. Не умеет: применить на практике современные приёмы выявления закономерности происходящего, использовать 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 способы идентификации общих свойств элементов, образующих группы. оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп. Уметь формулировать признак классификации, выделяет соответствующие ему группы однородных «объектов».	Глубоко знает: эффективные способы идентификации общих свойств элементов, образующих группы. оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп. Уверенно умеет: формулировать признак классификации, выделяет соответствующие ему группы однородных «объектов».	Знает: эффективные способы идентификации общих свойств элементов, образующих группы. оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп. Умеет: формулировать признак классификации, выделяет соответствующие ему группы однородных «объектов».	Частично знает: эффективные способы идентификации общих свойств элементов, образующих группы. оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп. Частично умеет: формулировать признак классификации, выделяет соответствующие ему группы однородных «объектов».	Не знает: эффективные способы идентификации общих свойств элементов, образующих группы. оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп. Не умеет: формулировать признак классификации, выделяет соответствующие ему группы однородных «объектов».
<i>4. Грамотно, логично, аргументированно формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности</i>				
Знать способы грамотно, логично, аргументированно формировать собственные суждения и оценки. Уметь четко отличать факты от мнений, интерпретаций, оценок и т.д.	Глубоко знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки. Уверенно умеет: отличать факты от мнений, интерпретаций, оценок	Знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки. Умеет: четко отличать факты от мнений, интерпретаций, оценок	Частично знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки. Частично умеет: четко отличать факты от мнений, интерпретаций, оценок	Не знает: способы грамотно, логично, аргументированно формировать собственные суждения и оценки. 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 Jupyter Notebook 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 кодбинарного поиска данных. Напишите программу для слияния нескольких словарей в один. Напишите программу на Python, чтобы получить новую строку из заданной строки, где «ls» было добавлено вперед. Если заданная строка уже начинается с «ls», 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 вариабельности.	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/вариабельность 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 собственные суждения и оценки.	Вопросы к экзамену 1. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работа с array в Python.

	<i>Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников 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. <i>Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.</i>	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

4. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.
5. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.
6. Организация ветвлений в Python. Инструкция if...else. Множественный выбор.
7. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.
8. Циклы в Python, работа и устройство цикла while. Организация бесконечного цикла, аварийное завершение цикла.
9. Списки в Python. Обращение к элементам списка и создание срезов. Обход списка и поиск элементов в списке.
10. Ключевые операции, приводящие к изменению списка и порождающие измененные списки, копирование списков.
11. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python. Операции с представлениями словарей.
12. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.
13. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.
14. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.
15. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.
16. Функции стандартной библиотеки для работы с контейнерами.
17. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции.
18. Функция. Получение информации о функции. Способы передачи параметров при вызове функции.
19. Передача переменного количества параметров (именованных и не именованных) в функции Python.
20. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.
21. Синтаксис и семантика обработки исключительных ситуаций в Python. Создание пользовательских исключений и инструкция assert.

22. Базовые операции для работы с файлами в Python. Использование инструкции with ... as на примере работы с файлами.
23. CSV файлы. Методы и функции работы с CSV файлами.
24. Бинарные файлы. Методы и функции работы с бинарными файлами.
25. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля.
26. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

Примерные вопросы для подготовки к экзамену

3. Концепция класса и объекта. Принципы и механизмы ООП.
4. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python.
5. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.
6. Методы классов и статические переменные и методы в Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.
7. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка элементов функционального программирования в Python.
8. 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.
9. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.
10. Вложенные функции и замыкания, специфика реализации в Python.
11. Функции высшего порядка и декораторы в Python.
12. Концепция map/filter/reduce. Реализация map/filter/reduce в Python и пример их использования.
13. 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.
14. Функции генераторы и выражения генераторы: создание и применение в Python.
15. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работа с array в Python.

16. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.
17. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.
18. Связанные списки: однонаправленные и двунаправленные – принцип реализации. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве.
19. Деревья, создание деревьев с помощью структуры список, узлов и ссылок. Операции с деревьями. Обход дерева
20. Алгоритм обменной сортировки, сложность сортировки и возможности по ее улучшению.
21. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.
22. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве. Сложность поиска в отсортированном и не отсортированном массиве.
23. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.
24. Алгоритм быстрой сортировки, сложность сортировки и возможности по ее улучшению.
25. Алгоритм сортировки слиянием, сложность сортировки.

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. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В.М. Шелудько ; Южный федеральный университет. - Ростов-на Дону; Таганрог : Издательство Южного федерального университета, 2017. - 107 с. – ЭБС ZNANIUM.com. - URL: <https://znanium.com/catalog/product/1021664>– Текст: электронный.

Дополнительная литература:

3. Васильев, А. Н. Python на примерах: практический курс по программированию / А. Н. Васильев. — 2-е изд. — СПб. : Наука и Техника, 2017. — 432 с. — ISBN 978-5-94387-741-4. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL:

<http://www.iprbookshop.ru/73043.html>— Режим доступа: для авторизир. пользователей.

4. Маккинли, Уэс Python и анализ данных / Уэс Маккинли ; перевод А. Слинкина. — 2-е изд. — Саратов : Профобразование, 2019. — 482 с. — ISBN 978-5-4488-0046-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/88752.html>— Режим доступа: для авторизир. пользователей

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_rukovodstvami/
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/>.
5. Scikit-learn Machine Learning in Python [Электронный ресурс]: сайт. – Режим доступа: <http://scikit-learn.org>.
6. Официальный сайт продукта <https://www.python.org/>.
7. The Python Tutorial // <https://docs.python.org/3/tutorial/index.html>.
8. The Python Standard Library // <https://docs.python.org/3/library/index.html>.
9. SciPy // <http://docs.scipy.org/doc/scipy/reference/>
10. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>.
11. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>.
12. Электронно-библиотечная система BOOK.RU <http://www.book.ru>.
13. Электронно-библиотечная система «Университетская библиотека ОН-ЛАЙН» <http://biblioclub.ru/>.
14. Электронно-библиотечная система Znanium <http://www.znaniium.com>.
15. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>.

10. 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 исполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

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

Комплект лицензионного программного обеспечения:

1. 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.

12. 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.