

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

Современные технологии программирования

Год утверждения программы: 2024

Разработчик рабочей программы дисциплины:

Е.В. Пальчевский

Направление подготовки: 38.03.05 – Бизнес-информатика,

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

Составитель актуализации: Сокур Е.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:
старший преподаватель *Е.А. Сокур*

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, 2026 – 22 с.

© Омский филиал Финансового университета при
Правительстве Российской Федерации, 2026

Содержание

5.2. Содержание практических и семинарских занятий	5
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.....	9
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	13
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	19
9. Перечень ресурсов информационно-телекоммуникационной сети.....	19
10. Методические указания для обучающихся по освоению дисциплины.....	20
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	21
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	22

5.2. Содержание практических и семинарских занятий

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. Примитивные 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. Потоквая 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 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>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

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
---	---	--

6.1. 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. Обзор платформы и языка программирования Java.	Установка IDE IntelliJ Idea. Переменные среды JDK. Регистрация github, создание проекта для выполнения семинарских занятий.	Обзор литературы и веб источников, прохождение онлайн-курсов
2. Основные языковые конструкции Java	Создание блоков кода. Стилевое оформление программного кода. Ключевые слова. Укороченные логические операции. Многоступенчатые конструкции «if – else – if». Оператор Switch. Цикл без тела. Вложенные циклы.	Обзор литературы и веб источников, прохождение онлайн-курсов
3. Базовые принципы объектно-ориентированного программирования	Поиск пакета и переменная среды CLASSPATH.	Обзор литературы и веб источников, прохождение онлайн-курсов
4. Классы, методы и объекты Java	Защищенные члены классов. Класс Object. Многоуровневая иерархия классов. Передача объектов методам. Инициализация полей по умолчанию. Вызов одного конструктора из другого.	Обзор литературы и веб источников, прохождение онлайн-курсов

5. Интерфейсы. Исключения. Отладка программ	Интерфейсы и абстрактные классы. Разрешение конфликтов в методах по умолчанию. Лямбда выражения. Генерация исключений в лямбда выражениях. Переопределение методов и исключения.	Обзор литературы и веб источников, прохождение онлайн-курсов
6. Коллекции	Связные списки. Хэш-множества. Очереди по приоритету. Древовидные множества. Взаимные преобразования списков и множеств. Стеки. Битовые множества.	Обзор литературы и веб источников, прохождение онлайн-курсов
7. Потокковая организация системы ввода вывода. Многопоточность.	Классы PipedInputStream и PipedOutputStream. SequenceInputStream. LineNumberInputStream. PushBackInputStream. PrintStream. Синхронизированные блоки. Параллельное программирование.	Обзор литературы и веб источников, прохождение онлайн-курсов
8. Работа с базами данных в Java. Основы объектно-реляционного отображения.	Исключения (SQL Exception). Потоки в JDBC. Основные этапы работы с базой данных при использовании JDBC. Уровни изоляции транзакций в JDBC. Хранимая процедура.	Обзор литературы и веб источников, прохождение онлайн-курсов

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

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

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. Функции-генераторы.
15. Лямбда-функции.
16. Модули. Инструкции import и from.
17. Базовые принципы объектно-ориентированного программирования. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
18. Конструктор и деструктор. Наследование.
19. Абстрактные методы класса.
20. Статические методы класса.
21. Свойства класса.
22. Исключения. Обработка исключений. 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 'Eeny, meeny, miney, 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: 2024-07-05 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 `a` 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
формированию 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**

**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.

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
УК – 4 Способность использовать прикладное программное обеспечение при решении профессиональных задач	1. <i>Использует основные методы средства получения, представления, хранения обработки данных.</i>	Знать: принципы ООП, основные структуры данных (HashSet, HashMap), работу с форматом данных (JSON, XML), основы JDBC, методы сериализации и алгоритмы обработки данных. Уметь: получать, представлять, хранить и обрабатывать данные с использованием коллекций, Stream API,	

		сериализации, а также работать с файлами и базами данных MySQL через JDBC.	
2. Демонстрирует Владение профессиональными Пакетами прикладных программ.		Знать: возможности и функционал профессиональных прикладных программ, их назначение и области применения, основы работы с офисными пакетами, инструментами для обработки данных, графики и программирования. Уметь: эффективно использовать инструменты для разработки ПО (например, IntelliJ IDEA, Eclipse), с целью решения профессиональных задач.	
3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.		Знать: классификацию и назначение прикладного программного обеспечения, основные функции и возможности офисных, инженерных, графических и специализированных программ. Уметь: анализировать задачу, определять требования к программному обеспечению и выбирать подходящее приложение в зависимости от поставленной цели.	
4. Использует прикладное программное обеспечение для решения		Знать: функциональные возможности прикладного программного	

	конкретных прикладных задач.	обеспечения, алгоритмы работы с ними и особенности применения в профессиональной деятельности. Уметь: применять специализированные программы для выполнения прикладных задач в зависимости от их специфики.	
УК – 10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	1. Четко описывает состав структуру	Знать: методы описания структуры данных, виды и форматы данных, способы их сбора, хранения и обработки, а также основы интерпретации полученной информации. Уметь: формулировать требования к данным, разрабатывать их структуру, организовывать процессы сбора (с использованием баз данных, API, файловых систем), обработки (через анализ, сортировку, фильтрацию) и представлять результаты в удобной и понятной форме (графики, отчеты, визуализации).	
	2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability.	Знать: основные принципы, объясняющие сущность информационных процессов, методы выявления закономерностей для анализа данных. Уметь: анализировать данные, выявлять причинно-следственные	

		связи, интерпретировать закономерности, оценивать природу и степень вариабельности с использованием статистических методов и инструментов обработки данных.	
	3. <i>Формулирует признак классификации, выделяет соответствующее «объектов», идентифицирует классификационные группы.</i>	Знать: принципы и методы классификации, критерии группировки однородных объектов, способы выявления общих свойств элементов групп и методы оценки полноты классификации. Уметь: формулировать признаки классификации, выделять группы однородных объектов, идентифицировать общие свойства элементов, оценивать полноту результатов классификации и демонстрировать прикладные навыки назначения классификационных групп.	
	4. <i>Грамотно, логично, аргументированно формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.</i>	Знать: принципы логического мышления и аргументации, методы критического анализа информации, различия между фактами, мнениями, интерпретациями и оценками, а также основы построения обоснованных суждений и выводов. Уметь: грамотно, логично и аргументированно формулировать собственные суждения и оценки;	

		отличать факты от мнений, интерпретаций и оценок в рассуждениях других участников; применять критическое мышление для анализа информации и аргументов; обоснованно оценивать точки зрения и предоставлять конструктивную обратную связь.	
	5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	Знать: принципы логического построения аргументов, методы системного описания и структурирования информации, правила эффективной коммуникации и убеждающей презентации, а также основы критического мышления и анализа. Уметь: аргументированно и логично представлять свою точку зрения посредством системного описания, использовать структурированные подходы для обоснования своих позиций, ясно и последовательно излагать мысли, подкрепляя их фактами и логическими связками.	

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

1. Концепция класса и объекта. Принципы и механизмы ООП.
2. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python.
3. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.
4. Методы классов и статические переменные и методы в Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.
5. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка элементов функционального программирования в Python.
6. Концепция «функции – граждане первого класса» в языке программирования, поддержка этой концепции в Python. Специфика лямбда-функций в Python их возможности и ограничения. Типичные сценарии использования лямбда-функций в Python.
7. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.
8. Вложенные функции и замыкания, специфика реализации в Python.
9. Функции высшего порядка и декораторы в Python.
10. Концепция map/filter/reduce. Реализация map/filter/reduce в Python и пример их использования.
11. Итераторы в Python: встроенные итераторы, создание собственных итераторов, типичные способы обхода итераторов и принцип их работы. Встроенные функции для работы с итераторами и возможности модуля itertools.
12. Функции генераторы и выражения генераторы: создание и применение в Python.
13. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работа с array в Python.
14. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.
15. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.
16. Связанные списки: однонаправленные и двунаправленные – принцип реализации. Сравнение скорости выполнения основных операций в

связанных списках и в динамическом массиве.

17. Деревья, создание деревьев с помощью структуры список, узлов и ссылок. Операции с деревьями. Обход дерева
18. Алгоритм обменной сортировки, сложность сортировки и возможности по ее улучшению.
19. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.
20. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве. Сложность поиска в отсортированном и не отсортированном массиве.
21. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.
22. Алгоритм быстрой сортировки, сложность сортировки и возможности по ее улучшению.
23. Алгоритм сортировки слиянием, сложность сортировки.

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, 2025. — 343 с. — (Высшее образование). - ISBN 978-5-16-020255-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2166199> (дата обращения: 29.04.2026). — Режим доступа: по подписке.

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, 2019.

— 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.znanium.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-х 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.

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.