МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Донской казачий государственный институт пищевых технологий и бизнеса (филиал) федерального государственного бюджетного образовательного учреждения «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)»

Донской казачий государственный институт пищевых технологий и бизнеса (филиал)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00AA7853CEFA35F0837BA866654F3503A6
Владелец: Павлятенко Геннадий Владимирович
Действителен: с 08.12.2025 до 03.03.2027

УТВЕРЖДАЮ
И.о. директора ДКГИПТиБ (филиала)
Г.В. Павлятенко
«10» апреля 2026 г.

Рабочая программа дисциплины (модуля)

Б1.О.03.03 МОДУЛЬ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ
Программирование

Кафедра:	Физика, математика и информационные технологии
Направление подготовки:	09.03.03 Прикладная информатика
Направленность (профиль):	Управление проектами в области информационных технологий
Квалификация выпускника:	Бакалавр
Форма обучения:	очно-заочная
Год набора:	2026
Общая трудоемкость:	144 часов/4 з.е.

Ростов-на-Дону, 2026 г.

Программу составил(и):
преподаватель Горбунов В.С.

Рабочая программа дисциплины (модуля)
"Программирование"
разработана на основании учебного плана, утвержденного ученым советом 26 марта 2026 г. протокол № 9 в соответствии с ФГОС ВО Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922),

Руководитель ОПОП
канд.экон.наук,доцент Левчук В.В._____ ___

Рабочая программа обсуждена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от 30 июня 2026 г. № 11
И.о.зав. кафедрой Левчук Валерия Владимировна_____

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Цели:

Цели освоения дисциплины "Программирование":
• обеспечить студентов базовыми знаниями в области разработки программных продуктов;
• заложить основы для последующих курсов, посвященных созданию современных информационных систем;
• познакомить студентов с прогрессивными парадигмами программирования и механизмами их реализации в программных продуктах;
• обучить студентов применению современных интегрированных инструментальных сред, предназначенных для разработки программ в интерактивном режиме;
• привить студентам навыки исследовательской работы, предполагающей самостоятельное изучение специфических инструментов и средств, необходимых для решения именно той конкретной проблемы, которая в качестве задачи поставлена перед ними.

1.2. Задачи:

-развитие и систематизация знаний, относящихся к математическим основам информатики, к принципам организации и функционирования программных и аппаратных средств вычислительной техники;
- расширение опыта создания, редактирования, хранения, передачи информационных объектов различного типа с помощью современных программных и коммуникационных средств с соблюдением соответствующих правовых и этических норм и требований информационной безопасности.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

Цикл (раздел) ОП: "Программирование"

Дисциплина относится к обязательной части ОПОП и обязательна для освоения.

Связь с предшествующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Пакеты прикладных программ для профессиональной деятельности	2	ОПК-2.1, ОПК-2.2, ОПК-2.3
2	Разработка программных приложений	2	ОПК-7.1, ОПК-7.2, ОПК-7.3
3	Основы алгоритмизации и программирования	1	ОПК-7.1, ОПК-7.2, ОПК-7.3
4	Основы информационных технологий	1	ОПК-2.1, ОПК-2.2, ОПК-2.3, УК-1.1, УК-1.2, УК-1.3
5	Ознакомительная практика	0	
6	Модуль Введение в информационные технологии	0	

Связь с последующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Ознакомительная практика	4	УК-7.1, УК-7.2, УК-7.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, УК-4.1, УК-4.2, УК-4.3
2	Операционные системы	5	ОПК-2.1, ОПК-2.2, ОПК-2.3
3	Интеллектуальные информационные системы	6	ОПК-2.1, ОПК-2.2, ОПК-2.3
4	Технологическая (проектно-технологическая) практика	6	УК-1.1, УК-1.2, УК-1.3, УК-2.1, УК-2.2, УК-2.3, УК-3.1, УК-3.2, УК-3.3, УК-3.4, УК-3.5, УК-3.6, УК-5.1, УК-5.2, УК-5.3, УК-8.1, УК-8.2, УК-8.3, УК-8.4, УК-8.5, УК-8.6, УК-9.1, УК-9.2, УК-9.3, УК-10.1, УК-10.2, УК-10.3, УК-10.4, УК-10.5, УК-10.6, ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-3.1, ОПК-3.2, ОПК-3.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-5.1, ОПК-5.2, ОПК-5.3, ОПК-6.1, ОПК-6.2, ОПК-6.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-8.1, ОПК-8.2, ОПК-8.3, ОПК-9.1, ОПК-9.2, ОПК-9.3, УК-4.4, УК-4.5, УК-4.6, УК-6.1, УК-6.2, УК-6.3
5	Базы данных	4	ОПК-5.1, ОПК-5.2, ОПК-5.3
6	Инженерная графика	0	
7	Интернет-технологии	0	
8	Управление информационными системами	7	ОПК-8.1, ОПК-8.2, ОПК-8.3

Распределение часов дисциплины

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		Итого	
Недель	16 3/6			
Вид занятий	УП	РП	УП	РП
Лекции	4	4	4	4
Лабораторные	24	24	24	24
В том числе электрон.	64	64	64	64
Итого ауд.	28	28	28	28
Контактная работа	28	28	28	28
Сам. работа	80	80	80	80
Часы на контроль	36	36	36	36
Итого	144	144	144	144

Вид промежуточной аттестации:

Экзамен 3 семестр

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины (модуля) направлен на формирование следующих компетенций и индикаторов их

ОПК-2:Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

ОПК-2.1: Знает процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); современные инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, используемые для решения задач профессиональной деятельности, и принципы их работы

ОПК-2.2: Умеет выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности; анализировать профессиональные задачи, выбирать и использовать подходящие информационные технологии

ОПК-2.3: Владеет навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности

ОПК-7:Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-7.1: Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий

ОПК-7.2: Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

ОПК-7.3: Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименования разделов, тем, их краткое содержание и результаты освоения /вид занятия/	Семестр	Часы	Интеракт.	Прак. подг.	Индикаторы достижения компетенции	Виды контроля
	Раздел 1.Программирование						
1.1	**Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.**	3	1	0	0	ОПК-2.1,ОПК-7.1	устный опрос, тестирование

	Краткое содержание: переменные и типы данных: как создавать переменные, какие типы данных поддерживает язык Python (целочисленные, вещественные, строки, булевы значения, списки, кортежи, множества, словари) и какие операции можно производить с переменными; операторы: арифметические, логические, операторы сравнения, операторы присваивания, операторы побитового сдвига, операторы доступа к элементам списка и словаря; условные операторы: if-else, elif, тернарный оператор, оператор assert. Рассматривается, какие действия можно выполнять в зависимости от значения переменных и каким образом использовать условные операторы; циклы: for, while, break, continue, else. Рассматривается, как работают циклы, какие действия можно выполнять внутри цикла, как использовать операторы break и continue для управления ходом выполнения цикла; функции: как определять функции, как передавать аргументы в функцию, как возвращать значения из функции, как использовать параметры функции по умолчанию и аргументы со звездочкой **Знать: как создавать переменные в Python, какие типы данных поддерживает язык; как определять тип данных переменной, как преобразовывать типы данных; какие операторы поддерживает Python; как использовать условные операторы; как создавать вложенные условия, как использовать логические операторы; как использовать циклы в Python; как создавать функции в Python, как передавать аргументы в функции /Лек/**						
1.2	**Тема 2. Структуры данных в Python: списки, кортежи, множества, словари.** **Краткое содержание: изучение различных структур данных, доступных в Python** **Знать: как создавать списки в Python; как создавать кортежи в Python; как создавать множества в Python; как использовать различные структуры данных в Python, включая списки, кортежи, множества и словари /Лек/**	3	1	0	0	ОПК-2.1,ОПК-7.1	Устный опрос, тестирование
1.3	**Тема 3. Объектно-ориентированное программирование в Python: классы, объекты, наследование,**	3	1	0	0	ОПК-2.1,ОПК-7.1	Устный опрос, тестирование

	полиморфизм. **Краткое содержание: основы объектно-ориентированного программирования: классы и объекты, наследование, полиморфизм** **Знать: как создавать классы в Python, как создавать объекты на основе класса, как использовать методы и атрибуты объектов; как использовать наследование в Python для создания новых классов на основе существующих, как переопределять методы и атрибуты в наследуемых классах; как использовать полиморфизм в Python, чтобы создавать код, который может работать с объектами разных классов /Лек/**						
1.4	**Тема 4. Работа с файлами и директориями в Python: чтение и запись файлов, создание и удаление директорий, работа с путями.** **Краткое содержание: представляет различных методов работы с файлами и директориями в Python: открытие файлов, закрытие файлов, работа с директориями, работа с путями** **Знать: как открыть файл в Python с помощью функции open(), как указать режимы открытия файлов, как читать и записывать данные в файл; как правильно закрыть файл в Python после его использования; как создавать и удалять директории в Python, как проверять существование директорий, как получать список файлов и поддиректорий в директории; как использовать модуль os.path в Python для работы с путями к файлам и директориям, как объединять и разделять пути, как получать информацию о путях /Лек/**	3	1	0	0	ОПК-2.1,ОПК-7.1	Устный опрос, тестирование
1.5	**Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.** **Краткое содержание: переменные и типы данных: как создавать переменные, какие типы данных поддерживает язык Python (целочисленные, вещественные, строки, булевы значения, списки, кортежи, множества, словари) и какие операции можно производить с переменными; операторы: арифметические, логические, операторы сравнения, операторы присваивания, операторы побитового сдвига,**	3	11	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	операторы доступа к элементам списка и словаря; условные операторы: if-else, elif, тернарный оператор, оператор assert. Рассматривается, какие действия можно выполнять в зависимости от значения переменных и каким образом использовать условные операторы; циклы: for, while, break, continue, else. Рассматривается, как работают циклы, какие действия можно выполнять внутри цикла, как использовать операторы break и continue для управления ходом выполнения цикла; функции: как определять функции, как передавать аргументы в функцию, как возвращать значения из функции, как использовать параметры функции по умолчанию и аргументы со звездочкой **Знать: как создавать переменные в Python, какие типы данных поддерживает язык; как определять тип данных переменной, как преобразовывать типы данных; какие операторы поддерживает Python; как использовать условные операторы; как создавать вложенные условия, как использовать логические операторы; как использовать циклы в Python; как создавать функции в Python, как передавать аргументы в функции** **Уметь: создавать переменные и работать с различными типами данных в Python, такими как числа, строки, списки и словари; создавать циклы, такие как циклы for и while, для повторения действий на протяжении определенного количества раз; создавать и использовать функции в Python, чтобы упростить свой код и повторно использовать его в различных местах.** **Владеть: навыками основных концепций языка Python, таких как переменные, типы данных, операторы, условные операторы, циклы и функции; навыками написания условных операторов, таких как if-else и операторов сравнения, для выполнения различных действий в зависимости от условий** **/Ср/**						
1.6	**Тема 2. Структуры данных в Python: списки, кортежи, множества, словари.** **Краткое содержание: изучение различных структур данных, доступных в Python**	3	10	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	Знать: как создавать списки в Python; как создавать кортежи в Python; как создавать множества в Python; как использовать различные структуры данных в Python, включая списки, кортежи, множества и словари **Уметь: создавать и использовать списки, кортежи, множества и словари; использовать операции над списками, такие как добавление, удаление, изменение элементов, срезы; создавать и использовать словари, а также выполнять операции добавления, удаления и изменения элементов в словаре; Применять списковые выражения (list comprehensions) и генераторы для создания списков и множеств на основе других списков или последовательностей** **Владеть: навыками анализа задач и выбора подходящей структуры данных для ее решения; навыками работы с различными методами и операциями над структурами данных; применения функций sorted() и reversed() для обработки списков и множеств; работы с элементами кортежей в качестве неизменяемых последовательностей. /Ср/**						
1.7	**Тема 3. Объектно-ориентированное программирование в Python: классы, объекты, наследование, полиморфизм.** **Краткое содержание: основы объектно-ориентированного программирования: классы и объекты, наследование, полиморфизм** **Знать: как создавать классы в Python, как создавать объекты на основе класса, как использовать методы и атрибуты объектов; как использовать наследование в Python для создания новых классов на основе существующих, как переопределять методы и атрибуты в наследуемых классах; как использовать полиморфизм в Python, чтобы создавать код, который может работать с объектами разных классов** **Уметь: создавать экземпляры классов и работать с их методами и атрибутами; использовать наследование для создания дочерних классов, которые наследуют методы и атрибуты родительского класса; применять полиморфизм для работы с объектами разных классов через единый интерфейс; проектировать и реализовывать объектно-ориентированные приложения в**	3	10	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	Python **Владеть: концепциями объектно-ориентированного программирования в Python /Ср/**						
1.8	**Тема 4. Работа с файлами и директориями в Python: чтение и запись файлов, создание и удаление директорий, работа с путями.** **Краткое содержание: представляет различных методов работы с файлами и директориями в Python: открытие файлов, закрытие файлов, работа с директориями, работа с путями** **Знать: как открыть файл в Python с помощью функции open(), как указать режимы открытия файлов, как читать и записывать данные в файл; как правильно закрыть файл в Python после его использования; как создавать и удалять директории в Python, как проверять существование директорий, как получать список файлов и поддиректорий в директории; как использовать модуль os.path в Python для работы с путями к файлам и директориям, как объединять и разделять пути, как получать информацию о путях** **Уметь: открывать файлы для чтения и записи с помощью функции open() и задавать различные режимы работы (чтение, запись, добавление); читать и записывать данные в файлы с помощью методов read(), readline(), write() и writelines(); работать с путями к файлам и директориям с помощью модуля os.path, создавать и удалять директории, проверять существование файлов и директорий** **Владеть: понимание различных режимов открытия файлов и их применения в практике; понимание принципов работы с путями и файловой системой операционной системы; навыками рекурсивного обхода директорий и поиска нужных файлов в них /Ср/**	3	10	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки
1.9	**Тема 5. Работа с базами данных в Python: подключение к базам данных, создание таблиц, выполнение SQL-запросов, обработка результатов запросов.** **Краткое содержание: различные методы работы с базами данных в Python: подключение к базам данных, создание таблиц, выполнение SQL-запросов, Обработка результатов запросов** **Знать: как работать с базами**	3	10	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	данных в Python, включая подключение к базам данных, создание таблиц, выполнение SQL-запросов и обработку результатов запросов **Уметь: подключаться к базе данных из Python с использованием стандартной библиотеки SQLite3 или других библиотек, таких как PostgreSQL или MySQL; использовать язык SQL для выполнения запросов к базе данных, включая выборку, вставку, обновление и удаление данных; обрабатывать результаты SQL-запросов в Python, например, преобразовывать их в словари или списки объектов; применять правильные методы для обработки ошибок при работе с базами данных** **Владеть: основами языка Python, включая переменные, типы данных, условные операторы и циклы; работать со структурами данных, такими как списки, кортежи и словари; навыками объектно-ориентированного программирования в Python, включая создание классов и объектов /Ср/**						
1.10	**Тема 6. Работа с сетью в Python: создание и использование сокетов, обмен данными по протоколам TCP/IP и UDP.** **Краткое содержание: методы работы с сетевыми соединениями в Python: создание и использование сокетов, обмен данными по протоколам TCP/IP и UDP, Обработка ошибок и исключений** **Знать: как работать с сетевыми соединениями в Python, включая создание и использование сокетов, а также обмен данными по протоколам TCP/IP и UDP; как создавать сокеты в Python и как использовать их для установления сетевых соединений; как отправлять и принимать данные по протоколам TCP/IP и UDP, в том числе установление соединения, передачу данных, закрытие соединения; как обрабатывать ошибки и исключения, связанные с сетевыми соединениями, и как защитить свои приложения от атак сетевых злоумышленников** **Уметь: создавать сокеты и выполнять настройку параметров соединения; обрабатывать исключения и ошибки, возникающие в процессе работы с сетью; производить асинхронную обработку соединений с использованием asyncio**	3	10	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	Владеть: навыками программирования на языке Python и основными конструкциями языка; навыками системного администрирования и настройки сетевых устройств; навыками работы с асинхронными операциями и потоками /Ср/						
1.11	**Тема 7. Модули и пакеты в Python: создание и использование модулей, пакетов, импортирование модулей.** **Краткое содержание: организация и использование модулей и пакетов в Python: создание модулей; импортирование модулей; создание пакетов; импортирование пакетов** **Знать: как организовать свои программы в модули и пакеты; как создавать и использовать модули и пакеты; как импортировать функции и классы из модулей и пакетов в свои программы на Python** **Уметь: создавать свои модули и использовать их в других программах; использовать стандартные модули Python для решения задач, например, для работы с датами и временем, математическими функциями, операционной системой и другими аспектами; создавать пакеты для организации своего кода и повторного использования его в других программах** **Владеть: навыками импорта модулей и пакетов в свои программы и использовать их функционал; использования встроенных функций dir() для получения списка методов и атрибутов модуля или объекта /Ср/**	3	10	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки
1.12	**Тема 8. Работа с графикой в Python: использование библиотеки matplotlib для создания графиков и диаграмм.** **Краткое содержание: использовать библиотеку Matplotlib для создания графиков и диаграмм в Python: установка и импортирование библиотеки Matplotlib, создание графиков, настройка графиков, сохранение графиков** **Знать: как использовать библиотеку Matplotlib для создания графиков и диаграмм в Python, как настраивать различные аспекты графиков и сохранять графики в различных форматах** **Уметь: установить и импортировать библиотеку Matplotlib; создавать базовые графики и диаграммы с помощью методов plot(), scatter(), bar() и**	3	9	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы для самоподготовки

	других; создавать множественные графики на одной фигуре и изменять их расположение и размеры **Владеть: навыками настройки графики с помощью различных опций: цвета, метки, заголовки; навыками создания гистограммы и ящики с усами для визуализации распределения данных /Ср/**						
1.13	**Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.** **Краткое содержание: переменные и типы данных: как создавать переменные, какие типы данных поддерживает язык Python (целочисленные, вещественные, строки, булевы значения, списки, кортежи, множества, словари) и какие операции можно производить с переменными; операторы: арифметические, логические, операторы сравнения, операторы присваивания, операторы побитового сдвига, операторы доступа к элементам списка и словаря; условные операторы: if-else, elif, тернарный оператор, оператор assert. Рассматривается, какие действия можно выполнять в зависимости от значения переменных и каким образом использовать условные операторы; циклы: for, while, break, continue, else. Рассматривается, как работают циклы, какие действия можно выполнять внутри цикла, как использовать операторы break и continue для управления ходом выполнения цикла; функции: как определять функции, как передавать аргументы в функцию, как возвращать значения из функции, как использовать параметры функции по умолчанию и аргументы со звездочкой** **Уметь: создавать переменные и работать с различными типами данных в Python, такими как числа, строки, списки и словари; создавать циклы, такие как циклы for и while, для повторения действий на протяжении определенного количества раз; создавать и использовать функции в Python, чтобы упростить свой код и повторно использовать его в различных местах.** **Владеть: навыками основных концепций языка Python, таких как переменные, типы данных, операторы, условные операторы, циклы и функции; навыками**	3	2	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе

	написания условных операторов, таких как if-else и операторов сравнения, для выполнения различных действий в зависимости от условий /Лаб/						
1.14	**Тема 2. Структуры данных в Python: списки, кортежи, множества, словари.** **Краткое содержание: изучение различных структур данных, доступных в Python** **Уметь: создавать и использовать списки, кортежи, множества и словари; использовать операции над списками, такие как добавление, удаление, изменение элементов, срезы; создавать и использовать словари, а также выполнять операции добавления, удаления и изменения элементов в словаре; Применять списковые выражения (list comprehensions) и генераторы для создания списков и множеств на основе других списков или последовательностей** **Владеть: навыками анализа задач и выбора подходящей структуры данных для ее решения; навыками работы с различными методами и операциями над структурами данных; применения функций sorted() и reversed() для обработки списков и множеств; работы с элементами кортежей в качестве неизменяемых последовательностей. /Лаб/**	3	2	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.15	**Тема 3. Объектно-ориентированное программирование в Python: классы, объекты, наследование, полиморфизм.** **Краткое содержание: основы объектно-ориентированного программирования: классы и объекты, наследование, полиморфизм** **Уметь: создавать экземпляры классов и работать с их методами и атрибутами; использовать наследование для создания дочерних классов, которые наследуют методы и атрибуты родительского класса; применять полиморфизм для работы с объектами разных классов через единый интерфейс; проектировать и реализовывать объектно-ориентированные приложения в Python** **Владеть: концепциями объектно-ориентированного программирования в Python /Лаб/**	3	2	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.16	**Тема 4. Работа с файлами и**	3	2	0	0	ОПК-	Отчет по лабораторной

	директориями в Python: чтение и запись файлов, создание и удаление директорий, работа с путями. **Краткое содержание: представляет различных методов работы с файлами и директориями в Python: открытие файлов, закрытие файлов, работа с директориями, работа с путями** **Уметь: открывать файлы для чтения и записи с помощью функции open() и задавать различные режимы работы (чтение, запись, добавление); читать и записывать данные в файлы с помощью методов read(), readline(), write() и writelines(); работать с путями к файлам и директориям с помощью модуля os.path, создавать и удалять директории, проверять существование файлов и директорий** **Владеть: понимание различных режимов открытия файлов и их применения в практике; понимание принципов работы с путями и файловой системой операционной системы; навыками рекурсивного обхода директорий и поиска нужных файлов в них /Лаб/**					2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	работе
1.17	**Тема 5. Работа с базами данных в Python: подключение к базам данных, создание таблиц, выполнение SQL-запросов, обработка результатов запросов.** **Краткое содержание: различные методы работы с базами данных в Python: подключение к базам данных, создание таблиц, выполнение SQL-запросов, Обработка результатов запросов** **Уметь: подключаться к базе данных из Python с использованием стандартной библиотеки SQLite3 или других библиотек, таких как PostgreSQL или MySQL; использовать язык SQL для выполнения запросов к базе данных, включая выборку, вставку, обновление и удаление данных; обрабатывать результаты SQL-запросов в Python, например, преобразовывать их в словари или списки объектов; применять правильные методы для обработки ошибок при работе с базами данных** **Владеть: основами языка Python, включая переменные, типы данных, условные операторы и циклы; работать со структурами данных, такими как списки, кортежи и словари; навыками**	3	4	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе

	объектно-ориентированного программирования в Python, включая создание классов и объектов /Лаб/						
1.18	**Тема 6. Работа с сетью в Python: создание и использование сокетов, обмен данными по протоколам TCP/IP и UDP.** **Краткое содержание: методы работы с сетевыми соединениями в Python: создание и использование сокетов, обмен данными по протоколам TCP/IP и UDP, Обработка ошибок и исключений** **Уметь: создавать сокеты и выполнять настройку параметров соединения; обрабатывать исключения и ошибки, возникающие в процессе работы с сетью; производить асинхронную обработку соединений с использованием asyncio** **Владеть: навыками программирования на языке Python и основными конструкциями языка; навыками системного администрирования и настройки сетевых устройств; навыками работы с асинхронными операциями и потоками /Лаб/**	3	4	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.19	**Тема 7. Модули и пакеты в Python: создание и использование модулей, пакетов, импортирование модулей.** **Краткое содержание: организация и использование модулей и пакетов в Python: создание модулей; импортирование модулей; создание пакетов; импортирование пакетов** **Уметь: создавать свои модули и использовать их в других программах; использовать стандартные модули Python для решения задач, например, для работы с датами и временем, математическими функциями, операционной системой и другими аспектами; создавать пакеты для организации своего кода и повторного использования его в других программах** **Владеть: навыками импорта модулей и пакетов в свои программы и использовать их функционал; использования встроенных функций dir() для получения списка методов и атрибутов модуля или объекта /Лаб/**	3	4	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе
1.20	**Тема 8. Работа с графикой в Python: использование библиотеки matplotlib для создания графиков и диаграмм.** **Краткое содержание: использовать**	3	4	0	0	ОПК-2.2,ОПК-2.3,ОПК-7.2,ОПК-7.3	Отчет по лабораторной работе

	библиотеку Matplotlib для создания графиков и диаграмм в Python: установка и импортирование библиотеки Matplotlib, создание графиков, настройка графиков, сохранение графиков **Уметь: установить и импортировать библиотеку Matplotlib; создавать базовые графики и диаграммы с помощью методов plot(), scatter(), bar() и других; создавать множественные графики на одной фигуре и изменять их расположение и размеры** **Владеть: навыками настройки графики с помощью различных опций: цвета, метки, заголовки; навыками создания гистограммы и ящики с усами для визуализации распределения данных /Лаб/**						
1.21	**Экзамен** **Знать:** **процессы, методы поиска, сбора, хранения, обработки, представления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); современные инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, используемые для решения задач профессиональной деятельности, и принципы их работы;процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий** **Уметь:** **выбирать и использовать современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе**	3	36	0	0	ОПК-2.1,ОПК-2.2,ОПК-2.3,ОПК-7.1,ОПК-7.2,ОПК-7.3	Вопросы к экзамену, итоговое тестирование

	отечественного производства, для решения задач профессиональной деятельности; анализировать профессиональные задачи, выбирать и использовать подходящие информационные технологии; выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий **Владеть:** **навыками работы с данными с помощью информационных технологий; навыками применения современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ и программных средств, в том числе отечественного производства, для решения задач профессиональной деятельности; навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач /Экзамен/**						

Перечень применяемых активных и интерактивных образовательных технологий:

Информационные технологии

Личностно ориентированная технология, способ организации самостоятельной деятельности учащихся, направленный на решение задачи учебного проекта

Технология обучения в сотрудничестве

Технология обучения в сотрудничестве используется в образовательной практике для преодоления последствий индивидуального характера учебной деятельности субъектов и их стремлений исключительно к индивидуальным образовательным достижениям. Она позволяет обогатить опыт и приобрести через учебный труд те навыки совместимой деятельности, которые затем могут стать необходимыми в будущей профессиональной и социальной деятельности в течение жизни. Цель технологии состоит в формировании умений у субъектов образовательного процесса эффективно работать сообща во временных командах и группах и добиваться качественных образовательных результатов

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная (внеаудиторная) работа обучающегося (далее - СРС) – это планируемая учебная, практическая, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время (свободное от аудиторных учебных занятий) по заданию и при методическом руководстве научно-педагогического работника (далее – преподаватель) и (или) лиц, привлекаемых к реализации основных профессиональных образовательных программ на иных условиях, но без их непосредственного участия.
СРС по заданию и при методическом руководстве преподавателя и (или) лиц, привлекаемых к реализации основных профессиональных образовательных программ на иных условиях, реализуется во время групповых консультаций и (или) индивидуальной работы обучающихся с преподавателями Университета и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях (в том числе индивидуальных консультаций), а также во время текущего контроля выполнения заданий, отнесенных к самостоятельной работе обучающихся.
Целью СРС является овладение формирование компетенций через овладение знаниями, умениями и навыками профессиональной деятельности по направлению подготовки (специальности). Самостоятельная работа является обязательной для каждого обучающегося.
Формы самостоятельной работы обучающихся определяются преподавателями кафедр Университета при разработке рабочих программ дисциплин (модулей), рабочих программ практик, программ государственной итоговой (итоговой) аттестации, методических указаний по выполнению практических, лабораторных работ, написанию курсовых работ/проектов и ВКР в соответствии с их содержанием.
В Университете оборудованы специальные помещения для самостоятельной работы обучающихся. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к информационно -телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

6.1. Перечень компетенций с указанием уровней их сформированности

ОПК-2:Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

Недостаточный уровень:

Знание основных понятий информационных технологий и современных инструментальных сред не сформированы

Умение осуществлять базовые операции поиска, сбора, хранения, обработки и представления информации с помощью простых программ и приложений не сформировано

Владение способами применения необходимых информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности не сформировано

Пороговый уровень:

Знание основных понятий информационных технологий и современных инструментальных сред имеет фрагментарный характер

Умеет применять различные программно-технические платформы и инструментальные среды для решения задач профессиональной деятельности

владеет базовыми навыками работы с данными и применения современных информационно-коммуникационных технологий

Продвинутый уровень:

Глубоко знаком с процессами и методами поиска, сбора, хранения, обработки, представления и распространения информации

Умеет создавать собственные программы и приложения, используя различные языки программирования и инструменты

владеет продвинутыми навыками работы с данными

Высокий уровень:

Осознает принципы работы современных инструментальных сред и программно-технических платформ

Использует передовые методы и технологии для решения сложных задач

Владеет навыками настройки и оптимизации вычислительных систем для работы с большими объемами данных; владеет способами применения необходимых информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-7:Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

Недостаточный уровень:

Знание понятий языка программирования и баз данных и основных принципов работ языков программирования и баз данных не сформированы

Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий не сформировано

не владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

Пороговый уровень:

Знание понятий языка программирования и баз данных и основных принципов работ языков программирования и баз данных носит фрагментарный характер

фрагментарно умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

Обладает практическим опытом разработки алгоритмов и компьютерных программ для решения практических задач

Продвинутый уровень:

Знаком с понятиями языка программирования и баз данных, понимает основные принципы работы языков программирования и баз данных

не в полной мере умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

Обладает практическим опытом разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения

Высокий уровень:

Глубоко разбирается в принципах работы языков программирования и баз данных

Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

Обладает опытом разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения

6.2. Описание критериев оценивания успеваемости

НЕДОСТАТОЧНЫЙ УРОВЕНЬ	ПОРОГОВЫЙ УРОВЕНЬ	ПРОДВИНУТЫЙ УРОВЕНЬ	ВЫСОКИЙ УРОВЕНЬ
Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий билета;	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала; - способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития;	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и объяснять связь практики и

- отсутствие умения выполнять практические задания, предусмотренные программой дисциплины (модуля); - отсутствие готовности (способности) к дискуссии и низкая степень контактности.	литературой, рекомендованной программой дисциплины (модуля); - умение без грубых ошибок решать практические задания, которые следует выполнить.	- правильные и конкретные, без грубых ошибок ответы на поставленные вопросы; - умение решать практические задания, которые следует выполнить; - владение основной литературой, рекомендованной программой дисциплины (модуля); - наличие собственной обоснованной позиции по обсуждаемым вопросам; - незначительные оговорки и неточности в раскрытии отдельных положений вопросов билета, присутствует неуверенность в ответах на дополнительные вопросы.	теории; - логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора; - умение решать практические задания; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.
Оценка «незачет», «неудовлетворительно»	**Оценка «зачтено/удовлетворительно», «удовлетворительно»**	**Оценка «зачтено/хорошо», «хорошо»**	**Оценка «зачтено/отлично», «отлично»**

6.3. Оценочные средства текущего контроля

ВОПРОСЫ К УСТНОМУ ОПРОСУ

Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.

1. Какие основные типы данных поддерживает Python?
2. Что такое переменная и как ее объявить в Python?
3. Какие операторы условий поддерживает Python?
4. Какие операторы цикла поддерживает Python?
5. Что такое функция в Python?
6. Что такое аргументы функции в Python?
7. Как передать аргументы функции по ссылке и по значению?
8. Как определить тип переменной в Python?
9. Какие способы форматирования строк поддерживает Python?
10. Какие встроенные функции поддерживает Python для работы со строками?

Тема 2. Структуры данных в Python: списки, кортежи, множества, словари

1. Какие типы данных в Python являются изменяемыми, а какие нет?
2. Что такое список в Python и как его создать?
3. Что такое кортеж в Python и как его создать?
4. Что такое множество в Python и как его создать?
5. Что такое словарь в Python и как его создать?
6. Как добавить элемент в список, кортеж, множество и словарь?
7. Как удалить элемент из списка, кортежа, множества и словаря?
8. Как получить элемент из списка, кортежа, множества и словаря?
9. Как проверить наличие элемента в списке, кортеже, множестве и словаре?
10. Как отсортировать список и словарь в Python?

Тема 3. Объектно-ориентированное программирование в Python: классы, объекты, наследование, полиморфизм.

1. Что такое класс в Python?
2. Что такое объект в Python?
3. Что такое наследование классов и как оно работает в Python?
4. Как переопределить метод в классе-наследнике?
5. Что такое полиморфизм и как он реализуется в Python?
6. Как создать абстрактный класс в Python?
7. Как использовать модуль abc для создания абстрактных классов?
8. Что такое инкапсуляция и как она реализуется в Python?
9. Какие особенности имеет множественное наследование в Python?
10. Что такое композиция объектов и как ее реализовать в Python?

Тема 4. Работа с файлами и директориями в Python: чтение и запись файлов, создание и удаление директорий, работа с путями

1. Как открыть файл в режиме чтения в Python?
2. Как открыть файл в режиме записи в Python?
3. Какой метод используется для чтения содержимого файла в Python?
4. Какой метод используется для записи содержимого в файл в Python?
5. Как проверить, существует ли файл в Python?

6. Как создать новую директорию в Python?
7. Как удалить файл в Python?
8. Как удалить пустую директорию в Python?
9. Как переименовать файл в Python?
10. Как скопировать файл в Python?

ВОПРОСЫ К САМОПОДГОТОВКЕ
Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.
1. Как объявить переменную в Python?
2. Какой оператор используется для ввода данных с клавиатуры?
3. Какие типы данных поддерживает Python?
4. Как проверить тип переменной в Python?
5. Какие операторы используются для математических операций в Python?
6. Какие операторы используются для сравнения значений в Python?
7. Какие операторы используются для логических операций в Python?
8. Какие условные операторы есть в Python?
9. Какие циклы поддерживает Python?
10. Как определить функцию в Python и как ее вызвать?

Тема 2. Структуры данных в Python: списки, кортежи, множества, словари
1. Как объявить список в Python?
2. Как добавить элемент в список?
3. Как удалить элемент из списка?
4. Как объявить кортеж в Python?
5. Как объявить множество в Python?
6. Как добавить элемент в множество?
7. Как удалить элемент из множества?
8. Как объявить словарь в Python?
9. Как добавить элемент в словарь?
10. Как удалить элемент из словаря?

Тема 3. Объектно-ориентированное программирование в Python: классы, объекты, наследование, полиморфизм.
1. Что такое объектно-ориентированное программирование?
2. Что такое классы в Python?
3. Как объявить класс в Python?
4. Что такое объекты в Python?
5. Как создать объект класса в Python?
6. Что такое наследование в Python?
7. Как создать класс-наследник в Python?
8. Что такое полиморфизм в Python?
9. Как реализовать полиморфизм в Python?
10. Какие особенности объектно-ориентированного программирования в Python?

Тема 4. Работа с файлами и директориями в Python: чтение и запись файлов, создание и удаление директорий, работа с путями
1. Как открыть файл для чтения в Python?
2. Как открыть файл для записи в Python?
3. Как прочитать содержимое файла в переменную в Python?
4. Как записать данные в файл в Python?
5. Как проверить существование файла или директории в Python?
6. Как создать новую директорию в Python?
7. Как удалить файл или директорию в Python?
8. Как переименовать файл или директорию в Python?
9. Как скопировать файл в Python?
10. Как переместить файл в Python?

Тема 5. Работа с базами данных в Python: подключение к базам данных, создание таблиц, выполнение SQL-запросов, обработка результатов запросов
1. Как подключиться к базе данных с помощью Python?
2. Как создать таблицу в базе данных с помощью Python?
3. Как выполнить SQL-запрос в базе данных с помощью Python?
4. Как получить результаты SQL-запроса в Python?
5. Как вставить данные в таблицу базы данных с помощью Python?
6. Как обновить данные в таблице базы данных с помощью Python?
7. Как удалить данные из таблицы базы данных с помощью Python?
8. Как выполнить транзакцию в базе данных с помощью Python?
9. Как создать индекс в базе данных с помощью Python?
10. Как работать с несколькими базами данных одновременно с помощью Python?

Тема 6. Работа с сетью в Python: создание и использование сокетов, обмен данными по протоколам TCP/IP и UDP.
1. Как создать TCP-сокет в Python?

2. Как создать UDP-сокет в Python?
3. Как отправить данные через сокет в Python?
4. Как получить данные через сокет в Python?
5. Как преобразовать данные в формат байтов в Python?
6. Как преобразовать данные из формата байтов в Python?
7. Как создать сервер, который слушает порт в Python?
8. Как создать клиент, который подключается к серверу в Python?
9. Как обработать несколько соединений одновременно в Python?
10. Как использовать протоколы HTTP и HTTPS в Python?

Тема 7. Модули и пакеты в Python: создание и использование модулей, пакетов, импортирование модулей.
1. Что такое модуль в Python и зачем он нужен?
2. Как создать свой модуль в Python?
3. Как импортировать модуль в Python?
4. Как выполнить код только при запуске модуля, а не при импортировании?
5. Что такое пакет в Python и зачем он нужен?
6. Как создать свой пакет в Python?
7. Как импортировать модуль из пакета в Python?
8. Как импортировать все модули из пакета в Python?
9. Как узнать список всех модулей, установленных в Python?
10. Как узнать, где находится файл модуля в Python?

Тема 8. Работа с графикой в Python: использование библиотеки matplotlib для создания графиков и диаграмм.
1. Что такое библиотека matplotlib и зачем она нужна?
2. Как установить библиотеку matplotlib в Python?
3. Как создать график с помощью библиотеки matplotlib?
4. Как настроить оси и заголовок графика с помощью библиотеки matplotlib?
5. Как создать разные типы графиков с помощью библиотеки matplotlib?
6. Как создать диаграмму с помощью библиотеки matplotlib?
7. Как создать множественные графики на одном холсте с помощью библиотеки matplotlib?
8. Как сохранить график в файл с помощью библиотеки matplotlib?
9. Как настроить стиль графика с помощью библиотеки matplotlib?
10. Как использовать аннотации и легенды на графике с помощью библиотеки matplotlib?

ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ
Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.
- Напишите программу, которая просит пользователя ввести свое имя, а затем выводит приветственное сообщение на экран с использованием этого имени.
- Напишите программу, которая запрашивает у пользователя два числа и выводит на экран их сумму, разность, произведение и частное.
- Напишите программу, которая запрашивает у пользователя радиус круга и выводит на экран его площадь и длину окружности.
- Напишите программу, которая проверяет, является ли введенное пользователем число четным или нечетным, и выводит соответствующее сообщение на экран.
- Напишите программу, которая запрашивает у пользователя число и выводит на экран все его делители.
- Напишите программу, которая запрашивает у пользователя число и проверяет, является ли оно простым, и выводит соответствующее сообщение на экран.
- Напишите программу, которая выводит на экран таблицу умножения от 1 до 10.
- Напишите программу, которая запрашивает у пользователя строку и выводит на экран количество гласных букв в ней.
- Напишите программу, которая запрашивает у пользователя две строки и выводит на экран сообщение о том, являются ли они анаграммами (т.е. содержат ли они одни и те же буквы в разном порядке).
- Напишите программу, которая запрашивает у пользователя список чисел и выводит на экран сумму всех четных чисел в этом списке.

Тема 2. Структуры данных в Python: списки, кортежи, множества, словари.
- Напишите программу, которая запрашивает у пользователя список целых чисел, затем выводит на экран все элементы списка, которые являются нечетными числами.
- Напишите программу, которая запрашивает у пользователя список слов, затем выводит на экран все слова, которые начинаются на букву "a".
- Напишите программу, которая запрашивает у пользователя два списка целых чисел, затем выводит на экран список из тех элементов, которые встречаются в обоих списках.
- Напишите программу, которая запрашивает у пользователя список целых чисел, затем находит и выводит на экран наименьшее и наибольшее значение в списке.
- Напишите программу, которая запрашивает у пользователя список целых чисел, затем удаляет из списка все повторяющиеся элементы и выводит на экран новый список.
- Напишите программу, которая запрашивает у пользователя два списка целых чисел одинаковой длины, затем создает новый список, в котором каждый элемент равен сумме соответствующих элементов входных списков.
- Напишите программу, которая запрашивает у пользователя список строк, затем создает новый список, в котором каждый элемент является длиной соответствующей строки из входного списка.
- Напишите программу, которая запрашивает у пользователя список целых чисел, затем сортирует его в порядке возрастания и выводит на экран отсортированный список.

3. Напишите программу на Python, которая создает новый файл new_file.txt и записывает в него все четные числа от 1 до 10, каждое число на новой строке.
4. Напишите программу на Python, которая считывает содержимое файла input.txt и записывает его в обратном порядке в файл output.txt.
5. Напишите программу на Python, которая создает новую директорию new_dir, а затем создает в ней 10 новых файлов с названиями file1.txt, file2.txt, ..., file10.txt. Каждый файл должен содержать номер файла в качестве текста.

Тема 5. Работа с базами данных в Python: подключение к базам данных, создание таблиц, выполнение SQL-запросов, обработка результатов запросов.
1. Напишите программу на Python, которая устанавливает соединение с базой данных SQLite и создает в ней таблицу "students" с полями "id", "name", "surname" и "age".
2. Напишите SQL-запрос на выборку всех студентов из таблицы "students" и выведите результат выполнения запроса на экран.
3. Напишите программу на Python, которая запрашивает у пользователя информацию о новом студенте (имя, фамилия, возраст) и добавляет эту информацию в таблицу "students".
4. Напишите SQL-запрос на выборку студентов, возраст которых больше 18 лет, и выведите результат выполнения запроса на экран.
5. Напишите программу на Python, которая изменяет возраст студента с указанным идентификатором в таблице "students".

Тема 6. Работа с сетью в Python: создание и использование сокетов, обмен данными по протоколам TCP/IP и UDP.
1. Напишите сервер, который принимает сообщения от клиентов и отправляет им обратно эти сообщения в верхнем регистре.
2. Напишите клиентскую программу, которая отправляет сообщения на сервер из предыдущего задания и выводит на экран ответы от сервера.
3. Напишите сервер, который принимает файлы от клиентов и сохраняет их на диск.
4. Напишите клиентскую программу, которая отправляет файлы на сервер из предыдущего задания.
5. Напишите программу, которая сканирует порты на заданном хосте и выводит список открытых портов.

Тема 7. Модули и пакеты в Python: создание и использование модулей, пакетов, импортирование модулей.
1. Создайте модуль с именем my_module и в нем определите следующие функции:
-greet(name), которая принимает строковый аргумент name и возвращает приветствие вида "Hello, {name}!"
-sum_range(start, end), которая принимает два целочисленных аргумента start и end и возвращает сумму всех целых чисел в диапазоне от start до end, включительно.
2. Импортируйте модуль my_module в другом файле и вызовите его функции.
3. Создайте пакет с именем my_package, содержащий следующие модули:
- module1 с функцией multiply(a, b), которая принимает два числа и возвращает их произведение.
- module2 с функцией divide(a, b), которая принимает два числа и возвращает результат их деления.
- module3 с функцией power(a, b), которая принимает два числа и возвращает результат возведения a в степень b.
4. Импортируйте пакет my_package в другом файле и вызовите его функции.

Тема 8. Работа с графикой в Python: использование библиотеки matplotlib для создания графиков и диаграмм.
1. Создайте список чисел от 1 до 10. Отобразите его на графике с помощью функции plot из библиотеки matplotlib.
2. Создайте список чисел от 1 до 10 и список квадратов этих чисел. Отобразите их на одном графике с помощью функции plot из библиотеки matplotlib. Добавьте подписи осей и заголовок графика.
3. Создайте график функции y = sin(x), где x принимает значения от 0 до 2*pi. Отобразите график с помощью функции plot из библиотеки matplotlib. Добавьте подписи осей и заголовок графика.
4. Создайте диаграмму с помощью функции pie из библиотеки matplotlib. Диаграмма должна отображать соотношение количества мужчин и женщин в группе из 20 человек.
5. Создайте столбчатую диаграмму, отображающую количество проданных товаров за 4 квартала. Данные должны храниться в словаре, где ключ - номер квартала, значение - количество проданных товаров. Добавьте подписи осей и заголовок графика.

ТЕСТОВЫЕ ЗАДАНИЯ ТЕКУЩЕГО КОНТРОЛЯ
Тема 1. Основы языка Python: переменные, типы данных, операторы, условные операторы, циклы, функции.
1. Какой тип данных не поддерживается в Python?
a) Integer (целое число)
b) Float (число с плавающей точкой)
c) Decimal (десятичное число)
d) Complex (комплексное число)

2. Какой оператор используется для выполнения целочисленного деления в Python?
a) /
b) //
c) %
d) *

3. Какой оператор используется для объединения двух списков в Python?
a) &
b) |
c) ^
d) +

1. Что из перечисленного является изменяемой структурой данных в Python? a) Список b) Кортеж c) Множество d) Словарь

2. Какой метод используется для добавления элемента в конец списка? a) append() b) insert() c) add() d) extend()

3. Какой тип данных используется для создания неизменяемой структуры данных? a) Список b) Кортеж c) Множество d) Словарь

4. Какой метод используется для удаления элемента из множества? a) remove() b) pop() c) discard() d) clear()

5. Как получить список ключей из словаря? a) keys() b) values() c) items() d) get()

Тема 3. Объектно-ориентированное программирование в Python: классы, объекты, наследование, полиморфизм.
1. Что такое класс в Python? a) Это способ организации данных b) Это функция в Python c) Это шаблон для создания объектов
2. Что такое объект в Python? a) Это функция в Python b) Это экземпляр класса c) Это способ организации данных
3. Что такое наследование в Python? a) Это способ организации данных b) Это механизм, позволяющий создавать новый класс на основе существующего c) Это функция в Python
4. Что такое полиморфизм в Python? a) Это способ организации данных b) Это механизм, позволяющий использовать один и тот же интерфейс для разных классов - Правильный ответ c) Это функция в Python
5. Что такое инкапсуляция в Python? a) Это механизм, позволяющий создавать новый класс на основе существующего b) Это способ организации данных c) Это механизм, позволяющий скрыть детали реализации класса от пользователя

Тема 4. Работа с файлами и директориями в Python: чтение и запись файлов, создание и удаление директорий, работа с путями.
1. Какой метод в Python используется для создания нового файла? a) create() b) open() c) new() d) make()

2. Какой метод в Python используется для записи текста в файл? a) write() b) append() c) read() d) close()

3. Какой метод в Python используется для удаления файла? a) delete() b) remove() c) erase() d) destroy()

6.4. Оценочные средства промежуточной аттестации

ВОПРОСЫ К ЭКЗАМЕНУ
ОПК-2
Знать:
Вопросы на установление последовательности
1. Установите правильную последовательность шагов для создания и использования функции в Python:
a) Вызов функции с передачей аргументов
b) Определение функции с помощью ключевого слова def
c) Использование возвращаемого значения функции
d) Указание параметров функции
2. Установите правильную последовательность действий для работы со списком в Python:
a) Добавление элемента в список с помощью метода append()
b) Создание пустого списка
c) Удаление элемента из списка с помощью метода pop()
d) Изменение элемента списка по индексу
3. Установите правильную последовательность этапов работы с файлом в Python:
a) Закрытие файла с помощью метода close()
b) Открытие файла с помощью функции open()
c) Чтение данных из файла с помощью метода read()
d) Запись данных в файл с помощью метода write()
4. Установите правильную последовательность шагов для создания класса и работы с объектом в Python:
a) Создание экземпляра класса
b) Определение класса с помощью ключевого слова class
c) Вызов метода объекта
d) Определение метода внутри класса
5. Установите правильную последовательность действий для работы с базой данных в Python с использованием SQLite:
a) Выполнение SQL-запроса с помощью метода execute()
b) Подключение к базе данных с помощью sqlite3.connect()
c) Закрытие соединения с базой данных
d) Создание курсора с помощью метода cursor()
Вопросы на установление соответствия
1. Установите соответствие между описанием и типом данных в Python:
1. Последовательность символов
2. Упорядоченная изменяемая последовательность элементов
3. Целое число
4. Набор пар ключ-значение

a) int
b) str
c) list
d) dict
e) float
2. Установите соответствие между назначением и оператором в Python:
1. Логическое "и"
2. Оператор сравнения на равенство
3. Целочисленное деление
4. Присваивание с прибавлением
a) +=
b) ==
c) and
d) //
e) *=
3. Установите соответствие между свойством и структурой данных:
1. Неизменяемая последовательность
2. Уникальные элементы без порядка
3. Изменяемая последовательность
4. Хранит данные в виде ключ-значение
a) Список
b) Кортеж
c) Множество
d) Словарь
e) Строка
4. Установите соответствие между использованием и условным оператором:
1. Выполняется, если ни одно из условий выше не истинно
2. Проверяет первое условие
3. Проверяет дополнительное условие, если предыдущее ложно
4. Вызывает исключение, если условие ложно
a) if
b) elif
c) else
d) assert
e) try
5. Установите соответствие между функцией и методом работы с файлами:
1. Записывает данные в файл
2. Читает содержимое файла
3. Закрывает файл
4. Открывает файл для работы
a) read()
b) write()
c) close()
d) open()
e) delete()

Уметь:
1. Выберите правильный ответ и обоснуйте свой выбор. Какой тип данных в Python является изменяемым?
a) str
b) tuple
c) list
d) int
2. Выберите правильный ответ и обоснуйте свой выбор. Какой оператор используется для проверки равенства двух значений?
a) =
b) ==
c) !=
d) is
3. Выберите правильный ответ и обоснуйте свой выбор. Что выведет код: x = 5; if x > 0: print("Положительное") else: print ("Отрицательное")?
a) Положительное
b) Отрицательное
c) Ничего
d) Ошибка
4. Выберите правильный ответ и обоснуйте свой выбор. Какой цикл лучше использовать для перебора элементов списка?
a) while
b) for
c) do-while
d) foreach
5. Выберите правильный ответ и обоснуйте свой выбор. Какой метод добавляет элемент в конец списка?
a) append()

b) insert()
c) extend()
d) pop()
6. Выберите правильный ответ и обоснуйте свой выбор. Что делает оператор break в цикле?
a) Пропускает текущую итерацию
b) Завершает цикл полностью
c) Переходит к следующему условию
d) Ничего не делает
7. Выберите правильный ответ и обоснуйте свой выбор. Какой режим открытия файла позволяет только чтение?
a) 'r'
b) 'w'
c) 'a'
d) 'r+'
8. Выберите правильный ответ и обоснуйте свой выбор. Какой метод возвращает значение из словаря по ключу?
a) get()
b) pop()
c) remove()
d) delete()
9. Выберите правильный ответ и обоснуйте свой выбор. Что делает ключевое слово class в Python?
a) Определяет функцию
b) Создаёт объект
c) Определяет класс
d) Импортирует модуль
10. Выберите правильный ответ и обоснуйте свой выбор. Какой метод используется для выполнения SQL-запроса в Python?
a) execute()
b) connect()
c) fetch()
d) close()

Владеть:
1. Вычислите результат выполнения следующего кода: x = 10
y = 3
z = x % y + x // y
print(z)
2. Рассчитайте, сколько раз выполнится тело цикла: for i in range(2, 8, 2):
print(i)
3. Вычислите длину строки после выполнения кода: s = "Python"
s = s + " " + "3"
print(len(s))
4. Рассчитайте сумму элементов списка: numbers = [1, 2, 3, 4]
total = sum(numbers)
print(total)
5. Вычислите значение переменной x после выполнения кода: x = 5
if x > 0:
x = x * 2
else:
x = x - 1
print(x)
6. Рассчитайте количество элементов в множестве: s = {1, 2, 2, 3, 4, 4}
print(len(s))
7. Вычислите результат выполнения функции: def calc(a, b=2):
return a + b
result = calc(3)
print(result)
8. Рассчитайте количество итераций цикла while: x = 10
while x > 5:
x -= 2
9. Вычислите сумму ключей словаря: d = {1: "a", 2: "b", 3: "c"}
total = sum(d.keys())
print(total)
10. Рассчитайте размер файла после записи: with open("test.txt", "w") as f:
f.write("Hello")
size = len("Hello")
print(size)

ОПК-7
Знать:
Вопросы на установление последовательности

1. Установите правильную последовательность шагов для создания и использования словаря в Python:
a) Добавление новой пары ключ-значение
b) Создание пустого словаря
c) Получение значения по ключу
d) Удаление элемента с помощью метода pop()
2. Установите правильную последовательность действий для работы с сетевым соединением через сокеты в Python:
a) Приём данных с помощью метода recv()
b) Создание сокета с помощью socket.socket()
c) Установка соединения с помощью метода connect()
d) Отправка данных с помощью метода send()
3. Установите правильную последовательность шагов для создания и импорта модуля в Python:
a) Использование функции из модуля в основном коде
b) Создание файла модуля с функцией
c) Импортирование модуля с помощью import
d) Сохранение файла модуля с расширением .py
4. Установите правильную последовательность действий для построения графика с помощью Matplotlib:
a) Отображение графика с помощью plt.show()
b) Импортирование библиотеки Matplotlib
c) Создание данных для графика
d) Построение графика с помощью plt.plot()
5. Установите правильную последовательность шагов для использования цикла for с прерыванием в Python:
a) Выполнение действия внутри цикла
b) Определение условия для прерывания с помощью break
c) Создание цикла for с итерируемым объектом
d) Проверка условия внутри цикла

Вопросы на установление соответствия
1. Установите соответствие между описанием и концепцией ООП:
1. Экземпляр класса
2. Шаблон для создания объектов
3. Возможность использовать один интерфейс для разных типов
4. Передача свойств и методов от одного класса другому
a) Наследование
b) Полиморфизм
c) Класс
d) Объект
e) Модуль
2. Установите соответствие между назначением и методом работы с базой данных:
1. Выполняет SQL-запрос
2. Создаёт объект для работы с запросами
3. Устанавливает соединение с базой данных
4. Сохраняет изменения в базе данных
a) connect()
b) execute()
c) cursor()
d) commit()
e) fetch()
3. Установите соответствие между функцией и сетевым методом сокетов:
1. Принимает данные
2. Отправляет данные
3. Создаёт сокет
4. Устанавливает соединение с сервером
a) socket()
b) send()
c) recv()
d) connect()
e) close()
4. Установите соответствие между назначением и функцией Matplotlib:
1. Создаёт столбчатую диаграмму
2. Отображает график
3. Создаёт точечную диаграмму
4. Создаёт линейный график
a) plot()
b) show()
c) scatter()
d) bar()
e) line()
5. Установите соответствие между ролью в циклах и ключевым словом в Python:
1. Прерывает выполнение цикла

2. Пропускает текущую итерацию
3. Создаёт цикл с условием
4. Создаёт цикл по итерируемому объекту
a) break
b) continue
c) for
d) while
e) pass

Уметь:
1. Выберите правильный ответ и обоснуйте свой выбор. Что выведет код: print(len([1, 2, 3]))?
a) 1
b) 2
c) 3
d) 4
2. Выберите правильный ответ и обоснуйте свой выбор. Какой модуль используется для работы с путями к файлам?
a) os
b) sys
c) pathlib
d) file
3. Выберите правильный ответ и обоснуйте свой выбор. Какой протокол используется для надёжной передачи данных в сокетах?
a) UDP
b) TCP
c) HTTP
d) FTP
4. Выберите правильный ответ и обоснуйте свой выбор. Какой метод создаёт сокет в Python?
a) socket.socket()
b) socket.connect()
c) socket.send()
d) socket.create()
5. Выберите правильный ответ и обоснуйте свой выбор. Как импортировать модуль в Python?
a) import module
b) from module import *
c) Вариант a и b
d) module.import()
6. Выберите правильный ответ и обоснуйте свой выбор. Что делает функция sorted()?
a) Изменяет исходный список
b) Возвращает новый отсортированный список
c) Удаляет дубликаты
d) Переворачивает список
7. Выберите правильный ответ и обоснуйте свой выбор. Какой метод Matplotlib отображает график?
a) plot()
b) show()
c) scatter()
d) figure()
8. Выберите правильный ответ и обоснуйте свой выбор/ Что делает метод pop() для множества?
a) Добавляет элемент
b) Удаляет случайный элемент
c) Удаляет элемент по значению
d) Очищает множество
9. Выберите правильный ответ и обоснуйте свой выбор/ Какой оператор используется для наследования класса?
a) :
b) extends
c) inherits
d) super()
10. Выберите правильный ответ и обоснуйте свой выбор/ Что выведет код: def func(x=5): return x * 2; print(func())?
a) 5
b) 10
c) Ошибка
d) Ничего

Владеть:
1. Вычислите количество строк в списке после выполнения кода: lines = ["a", "b", "c"]
lines.append("d")
print(len(lines))
2. Рассчитайте значение переменной x после выполнения кода с классом: class MyClass:
def __init__(self):
self.x = 1

def increment(self):
self.x += 2

obj = MyClass()
obj.increment()
print(obj.x)
3. Вычислите результат выполнения кода с наследованием: class Parent:
def get_value(self):
return 5

class Child(Parent):
def get_value(self):
return super().get_value() + 1

obj = Child()
print(obj.get_value())
4. Рассчитайте количество символов в файле после чтения: with open("data.txt", "w") as f:
f.write("Python")
with open("data.txt", "r") as f:
content = f.read()
print(len(content))
5. Вычислите сумму чисел в таблице базы данных: import sqlite3
conn = sqlite3.connect(":memory:")
c = conn.cursor()
c.execute("CREATE TABLE nums (value INTEGER)")
c.execute("INSERT INTO nums VALUES (2), (3), (4)")
c.execute("SELECT SUM(value) FROM nums")
result = c.fetchone()[0]
print(result)
6. Рассчитайте длину строки, переданной через сокет: import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
data = "Test"
length = len(data)
print(length)
7. Вычислите результат выполнения кода с модулем: def add(a, b):
return a + b

# main.py
from module import add
result = add(3, 7)
print(result)
8. Рассчитайте количество точек на графике: import matplotlib.pyplot as plt
x = [1, 2, 3]
y = [4, 5, 6]
plt.scatter(x, y)
count = len(x)
print(count)
9. Вычислите значение переменной после спискового выражения: numbers = [1, 2, 3, 4]
squares = [x**2 for x in numbers]
total = sum(squares)
print(total)
10. Рассчитайте количество элементов в кортеже после преобразования: lst = [1, 2, 3]
tup = tuple(lst)
print(len(tup))

ИТОГОВОЕ ТЕСТИРОВАНИЕ
ОПК-2
1. Какое значение будет иметь переменная x после выполнения кода: x = 5 + 3 * 2? a) 11 b) 16 c) 13 d) 15
2. Какая функция используется для ввода данных с клавиатуры в Python? a) input() b) print() c) scan() d) read()
3. Какая структура данных в Python является изменяемой? a) Список (list) b) Кортеж (tuple) c) Множество (set) d) Словарь (dictionary)
4. Что такое наследование в объектно-ориентированном программировании? a) Процесс создания нового класса на основе уже существующего b) Процесс изменения существующего класса c) Процесс создания объекта на основе класса d) Процесс добавления новых методов в класс
5. Какой протокол используется для обмена данными между сервером и клиентом в Интернете? a) TCP b) UDP c) FTP d) HTTP
6. Что такое класс в Python? a) функция, которая принимает на вход объекты и возвращает другие объекты b) совокупность свойств и методов, описывающих объекты c) переменная, которая хранит значение
7. Что такое объект в Python? a) функция, которая принимает на вход объекты и возвращает другие объекты b) совокупность свойств и методов, описывающих объекты c) переменная, которая хранит значение

12. Какой модуль Python используется для работы с базами данных SQLite? a) sqlite3 b) mysql c) psycopg2 d) pyodbc
13. Как подключиться к базе данных SQLite в Python? a) import sqlite3, connection = sqlite3.connect("database.db") b) import MySQLdb, connection = MySQLdb.connect("database.db") c) import psycopg2, connection = psycopg2.connect ("database.db")
14. Как создать таблицу в базе данных SQLite с помощью Python? a) cursor.execute("CREATE TABLE table_name (column1 type1, column2 type2)") b) cursor.create_table("table_name", "column1 type1", "column2 type2") c) connection.create_table("table_name", "column1 type1", "column2 type2")
15. Какое ключевое слово используется для создания множества в Python? a) tuple b) set c) list d) dict

ОПК-7
1. Как открыть файл для чтения в Python? a) file = open("myfile.txt", "w") b) file = open("myfile.txt", "r") c) file = open ("myfile.txt", "a") d) file = open("myfile.txt")
2. Как подключиться к базе данных SQLite в Python? a) import mysql.connector b) import sqlite3 c) import psycopg2 d) import pyodbc
3. Что такое сокет в Python? a) Тип данных, предназначенный для работы с сетью b) Функция, позволяющая создать соединение с удаленным сервером c) Функция, позволяющая принимать входящие подключения
4. Как создать модуль в Python? a) Создать файл с расширением .mod и определить в нем функции и переменные b) Создать файл с расширением .py и определить в нем функции и переменные c) Создать файл с расширением .module и определить в нем функции и переменные
5. Как создать пакет в Python? a) Создать каталог и поместить в него файлы с расширением .mod b) Создать каталог и поместить в него файлы с расширением .py c) Создать каталог и поместить в него файлы с расширением .pkg
6. Что такое наследование в Python? a. Это процесс создания нового класса на основе существующего b. Это процесс объединения двух классов в один c. Это процесс создания экземпляра класса d. Это процесс изменения существующего класса

6.5. Примерная тематика курсовых работ (проектов)
Учебным планом не предусмотрено

6.6. Методические указания для обучающихся по освоению дисциплины (модуля)
Методические рекомендации по подготовке к лекционным занятиям

Просмотрите конспект сразу после занятий. Пометьте материал конспекта лекций, который вызывает затруднения для понимания. Попытайтесь найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.
Работа с рекомендованной литературой:
При работе с основной и дополнительной литературой целесообразно придерживаться такой последовательности. Сначала прочитать весь заданный текст в быстром темпе. Цель такого чтения заключается в том, чтобы создать общее представление об изучаемом материале, понять общий смысл прочитанного. Затем прочитать вторично, более медленно, чтобы в ходе чтения понять и запомнить смысл каждой фразы, каждого положения и вопроса в целом. Чтение приносит пользу и становится продуктивным, когда сопровождается записями. Это может быть составление плана прочитанного текста, тезисы или выписки, конспектирование и др. Выбор вида записи зависит от характера изучаемого материала и целей работы с ним. Если содержание материала несложное, легко усваиваемое, можно ограничиться составлением плана. Если материал содержит новую и трудно усваиваемую информацию, целесообразно его законспектировать. План – это схема прочитанного материала, перечень вопросов, отражающих структуру и последовательность материала. Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов: - план-конспект – это развернутый детализированный план, в котором по наиболее сложным вопросам даются подробные пояснения, - текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника, - свободный конспект – это четко и кратко изложенные основные положения в результате глубокого изучения материала, могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом, - тематический конспект – составляется на основе изучения ряда источников и дает ответ по изучаемому вопросу. В процессе изучения материала источника и составления конспекта нужно обязательно применять различные выделения, подзаголовки, создавая блочную структуру конспекта. Это делает конспект легко воспринимаемым и удобным для работы.

Методические рекомендации по подготовке к практическим занятиям

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала предмета путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение практического занятия предполагает, например:
- индивидуальные выступления студентов с сообщениями по какому-либо вопросу изучаемой темы;
- фронтальное обсуждение рассматриваемой проблемы, обобщения и выводы;
- решение задач и упражнений по образцу;
- решение вариантных задач и упражнений;
- решение ситуационных производственных (профессиональных) задач;
- проектирование и моделирование разных видов и компонентов профессиональной деятельности;
- выполнение контрольных работ;

- работу с тестами.
При подготовке к практическим занятиям студентам рекомендуется: внимательно ознакомиться с тематикой практического занятия; прочесть конспект лекции по теме, изучить рекомендованную литературу; составить краткий план ответа на каждый вопрос практического занятия; проверить свои знания, отвечая на вопросы для самопроверки; если встретятся незнакомые термины, обязательно обратиться к словарю и зафиксировать их в тетради. Все письменные задания выполнять в рабочей тетради. Практические занятия развивают у студентов навыки самостоятельной работы по решению конкретных задач.

Методические рекомендации по подготовке к лабораторным работам

Лабораторные работы представляют одну из форм освоения теоретического материала с одновременным формированием практических навыков в изучаемой дисциплине (модуле). Их назначение – углубление проработки теоретического материала, формирование практических навыков путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к лабораторным работам включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение лабораторной работы предполагает:
- изучение теоретического материала по теме лабораторной работы (по вопросам изучаемой темы);
- выполнение необходимых расчетов и экспериментов;
- оформление отчета с заполнением необходимых таблиц, построением графиков, подготовкой выводов по проделанным экспериментам и теоретическим расчетам;
- по каждой лабораторной работе проводится контроль: проверяется содержание отчета, проверяется усвоение теоретического материала.
Контроль усвоения теоретического материала является индивидуальным.

Методические указания по выполнению отчёта к лабораторным работам

Основным требованием по выполнению лабораторных и практических работ является полное исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения и профессиональной подготовки студентов.
Методические указания обеспечивают комплексный подход в учебной работе студентов, единство и преемственность требований к оформлению результатов работы на разных этапах обучения. С единых позиций приведены основные требования к структуре, оформлению и содержанию отчета по лабораторным и практическим работам.
Структура отчёта:
- цель работы;
- краткие теоретические сведения;
- ход выполнения работы;
- выводы.
Дополнительные элементы:
- приложения;
- библиографический список.
Требования к содержанию отчёта:
1. Титульный лист
В верхнем поле листа указывают полное наименование учебного заведения.
В среднем поле указывается вид работы, в данном случае лабораторная или практическая работа с указанием курса, по которому она выполнена, и ниже ее название. Название работы приводится без слова "тема" и в кавычки не заключается.
Далее ближе к правому краю титульного листа указывают фамилию, инициалы и группу учащегося, выполнившего работу, а также фамилию, инициалы преподавателя, принявшего работу.
В нижнем поле листа указывается место выполнения работы и год ее написания (без слова год).
2. Цель работы должна отражать тему работы, а также конкретные задачи, поставленные студенту на период выполнения работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от нескольких строк до 0,5 страницы.
3. Краткие теоретические сведения. В этом разделе излагается краткое теоретическое описание изучаемой в работе темы. Материал раздела не должен копировать содержание методического пособия или учебника по данной теме, а ограничивается изложением основных понятий, требующихся для дальнейшей обработки полученных результатов. Объем литературного обзора не должен превышать 1/3 части всего отчета.
4. Ход выполнения работы. В данном разделе подробно излагается методика выполнения работы, процесс получения данных и способ их обработки. Если используются стандартные пакеты компьютерных программ для обработки экспериментальных результатов, то необходимо обосновать возможность и целесообразность их применения, а также подробности обработки данных с их помощью.
5. Выводы по работе - кратко излагаются результаты работы, полученные в результате выполнения работы, а также краткий анализ полученных результатов.
Отчет по лабораторной работе оформляется на листе формата А4. Допускается оформление отчета по лабораторной работе в электронном виде средствами Microsoft Office. Текст работы должен быть напечатан через полтора интервала шрифтом Times New Roman, кегль – 12. Поля должны оставаться по всем четырем сторонам печатного листа: левое – не менее 30 мм, правое – не менее 10, нижнее – не менее 20 и верхнее – не 15 мм.
Для защиты лабораторной работы студент должен подготовить отчет, провести самостоятельную работу, иметь отметку о проверенном отчете.
Результаты определяются по пятибалльной системе оценок.

Методические рекомендации по выполнению реферата

Реферат – письменная работа объемом 8–10 страниц. Это краткое и точное изложение сущности какого-либо вопроса, темы. Тему реферата студент выбирает из предложенных преподавателем или может предложить свой вариант. В реферате нужны развернутые аргументы, рассуждения, сравнения. Содержание темы излагается объективно от имени автора. Функции реферата: информативная, поисковая, справочная, сигнальная, коммуникативная. Степень выполнения этих функций зависит от содержательных и формальных качеств реферата и для каких целей их использует. Требования к языку реферата: должен отличаться точностью, краткостью, ясностью и простотой.
Структура реферата:
1. Титульный лист
2. Оглавление (на отдельной странице). Указываются названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.
3. Введение. Аргументируется актуальность исследования, т.е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками, перечисляются положения, которые должны быть обоснованы. Обязательно формулируются цель и задачи реферата.
4. Основная часть. Подчиняется собственному плану, что отражается в разделении текста на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала. В случае если используется чья-либо неординарная мысль, идея, то обязательно нужно сделать ссылку на того автора, у кого взят данный материал.
5. Заключение. Последняя часть научного текста. В краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования.
6. Приложение. Может включать графики, таблицы, расчеты.
7. Библиография (список литературы). Указывается реально использованная для написания реферата литература. Названия книг располагаются по алфавиту с указанием их выходных данных. Общие требования к построению, содержанию и оформлению».
При проверке реферата оцениваются:
- знание фактического материала, усвоение общих представлений, понятий, идей;
- характеристика реализации цели и задач исследования;
- степень обоснованности аргументов и обобщений;
- качество и ценность полученных результатов;
- использование литературных источников;
- культура письменного изложения материала;
- культура оформления материалов работы.
Правила написания научных текстов (реферат):
Здесь приводятся рекомендации по консультированию студентов относительно данного вида самостоятельной работы. Во время консультаций руководителю следует предложить к обсуждению следующие вопросы.
• Какова истинная цель Вашего научного текста – это поможет Вам разумно распределить свои силы и время.
• Важно разобраться, кто будет «читателем» Вашей работы.
• Начинать писать серьезную работу следует не раньше, чем возникнет ощущение, что по работе с источниками появились идеи, которыми можно поделиться.
• Должна быть идея, а для этого нужно научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного).
• Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно, а также стремясь структурировать свой текст.
• Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Методические рекомендации по выполнению контрольных работ

Контрольная работа выполняется по вариантам. На бланке указывается факультет, курс, группа, ФИО студента. Вопросы строятся на основе тестовых и ситуативных заданий. В тестовых заданиях, выбирается правильный(ые) ответ(ы). При решении ситуативных заданий выбирается правильная последовательность действий в рассматриваемой ситуации. Проверка контрольной работы позволяет выявить и исправить допущенные студентами ошибки, указать, какие вопросы дисциплины (модуля) ими недостаточно усвоены и требуют доработки. Студент должен внимательно ознакомиться с письменными замечаниями преподавателя и приступить к их исправлению, для чего еще раз повторить соответствующий материал.

Методические рекомендации по подготовке к коллоквиуму

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:
- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.

Коллоквиум - это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы. Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму студенту отводится 3-4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.
Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2-3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

Методические рекомендации по устному опросу/самоподготовке

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, используя лист опорных сигналов,
воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств. В случае необходимости следует рекомендовать еще раз внимательно разобраться в материале. Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение решать задачи или пройти тестирование по пройденному материалу. Однако преподавателю следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Методические рекомендации по подготовке к семинарским занятиям

Одним из видов внеаудиторной самостоятельной работы является подготовка к семинарским занятиям. Семинар – форма учебно-практических занятий, при которой студенты обсуждают сообщения, доклады и рефераты, выполненные ими по результатам учебных или научных исследований под руководством преподавателя. Преподаватель в этом случае является координатором обсуждений темы семинара, подготовка к которому является обязательной. Поэтому тема семинара и основные источники обсуждения предъявляются до обсуждения для детального ознакомления, изучения. Цели обсуждений направлены на формирование навыков профессиональной полемики и закрепление обсуждаемого материала. Семинар – это такая форма организации обучения, при которой на этапе подготовки доминирует самостоятельная работа учащихся с учебной литературой и другими дидактическими средствами над серией вопросов, проблем и задач, а в процессе семинара идут активное обсуждение, дискуссии и выступления учащихся, где они под руководством преподавателя делают обобщающие выводы и заключения. Семинар предназначен для углубленного изучения дисциплины, овладения методологией научного познания, то главная цель семинарских занятий – обеспечить студентам возможность овладеть навыками и умениями использования теоретического знания применительно к особенностям изучаемой отрасли.

Методические рекомендации по подготовке к эссе

Одним из видов самостоятельной работы студентов является написание творческой работы по заданной либо согласованной с преподавателем теме. Творческая работа (эссе) представляет собой оригинальное произведение объемом 500-700 слов, посвященное какой-либо значимой классической либо современной проблеме в определенной теоретической и практической области. Творческая работа не является рефератом и не должна носить описательный характер, большое место в ней должно быть уделено аргументированному представлению своей точки зрения студентами, критической оценке рассматриваемого материала и проблематики, что должно способствовать раскрытию творческих и аналитических способностей. Цели написания эссе – научиться логически верно и аргументировано строить устную и письменную речь; работать над углублением и систематизацией своих философских знаний; овладеть способностью использовать основы знаний для формирования мировоззренческой позиции. Приступая к написанию эссе, изложите в одном предложении, что именно вы будете утверждать и доказывать (свой тезис).

Методические рекомендации по подготовке к докладу

Для подготовки доклада необходимо выбрать актуальную тему. Желательно, чтобы тема была интересна докладчику и вызывала желание качественно подготовить материалы. Подготовка доклада предполагает: определение цели доклада; подбор необходимого материала, определяющего содержание доклада; составление плана доклада, распределение собранного материала в необходимой логической последовательности.
Композиция доклада имеет вступление, основную часть и заключение.
Вступление должно содержать: название доклада; сообщение основной идеи; современную оценку предмета изложения; краткое перечисление рассматриваемых вопросов; интересную для слушателей форму изложения. Основная часть, в

которой необходимо раскрыть суть темы, обычно строится по принципу отчёта. Задача основной части: представить достаточно данных для того, чтобы слушатели заинтересовались темой.
Заключение – чёткое обобщение и краткие выводы по излагаемой теме.

Методические рекомендации по подготовке к собеседованию

Собеседование – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.
Цель собеседования: проверка усвоения знаний; умений применять знания; сформированности профессионально значимых личностных качеств.
Подготовка к собеседованию предполагает повторение пройденного материала и приобретение навыка свободного владения терминологией и фактическими данными по определенному разделу дисциплины (модуля).

Методические рекомендации по подготовке к тестированию/ итоговому тестированию

Тестирование/итоговое тестирование – это система стандартизированных заданий, определенного содержания и упорядоченных в рамках определенной стратегии предъявления, позволяющая качественно оценить структуру и эффективно измерить уровень знаний, умений и навыков по дисциплине (модулю). Тестирование позволяет оценить у студентов уровень знаний фактического материала, умения логически мыслить, способности к рефлексии и творческому подходу к решению поставленной задачи на каждом этапе их обучения.
При самостоятельной подготовке к тестированию студенту необходимо:
- четко выяснить все условия тестирования заранее (сколько тестов будет предложено, сколько времени отводится на тестирование, какова система оценки результатов);
- проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- проработать информационный материал по разделу дисциплине;
- конспективно выписать из соответствующих разделов учебной дисциплины основные понятия, ключевые идеи, теоретические концепции и подходы;
- повторить теоретический материал лекций, самостоятельно изученной учебной и научной литературы, семинарских занятий.
При выполнении тестовых заданий рекомендуется:
- внимательно полностью прочитать вопрос и предлагаемые варианты ответов, выбрать правильные из них (их может быть несколько);
- при затруднении при ответе на сложный для студента вопрос целесообразно переходить к другим вопросам теста. По завершению ответов на все последующие вопросы теста необходимо вернуться к трудному вопросу и дать на него продуманный ответ;
- по окончании тестирования следует проверить правильность данных ответов на вопросы теста, чтобы избежать механических ошибок.

Методические рекомендации по подготовке к экзамену

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной дисциплине (модулю). Экзаменационная сессия – это серия экзаменов, установленных учебным планом. Между экзаменами интервал 2-4 дня, в течение студент систематизирует уже имеющиеся знания. На консультации перед экзаменом студенты должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки
вопросы. Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для
систематизации знаний.

Методические рекомендации по подготовке к зачету/зачету с оценкой

В ходе подготовки к зачету/зачету с оценкой студент, в первую очередь, должен систематизировать знания, полученные в ходе изучения дисциплины. К зачету/зачету с оценкой необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине (модулю). В самом начале учебного курса познакомьтесь со следующей учебно-методической документацией:
- программой дисциплины (модуля);
- перечнем знаний, умений и навыков, которыми студент должен владеть;
- тематическими планами лекций, семинарских занятий;
- учебниками, учебными пособиями по дисциплине, а также электронными ресурсами;
- перечнем вопросов к зачету/зачету с оценкой.
После этого у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине (модулю). Систематическое выполнение учебной работы на лекциях и лабораторных занятиях позволит успешно освоить дисциплину (модуль) и создать хорошую базу для сдачи зачета/зачета с оценкой.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература	
7.1.1. Основная литература	
Л.1.1	Батасова В. С., Воробьева И. А., Голубева И. В., Гречкина П. В., Маран М. М., Чибизова Н. В., Маран М. М. Программирование в примерах и задачах [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2024. - 260 с. – Режим доступа: https://e.lanbook.com/book/362825
Л.1.2	Борзунов С. В., Кургалин С. Д. Языки программирования. Python: решение сложных задач [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2023. - 192 с. – Режим доступа: https://e.lanbook.com/book/319394
Л.1.3	Никитина Т. П., Королев Л. В. Программирование. Основы Python для инженеров [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2023. - 156 с. – Режим доступа: https://e.lanbook.com/book/302720
Л.1.4	Янцев В. В. Web-программирование на Python [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2023. - 180 с. – Режим доступа: https://e.lanbook.com/book/310289
Л.1.5	Волк В. К. Базы данных. Проектирование, программирование, управление и администрирование [Электронный ресурс]:учебник для вузов. - Санкт-Петербург: Лань, 2023. - 244 с. – Режим доступа: https://e.lanbook.com/book/346439
Л.1.6	Букунов С. В., Букунова О. В. Разработка приложений с графическим пользовательским интерфейсом на языке Python [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2023. - 88 с. – Режим доступа: https://e.lanbook.com/book/292856
7.1.2. Дополнительная литература	
Л.2.1	Никифоров С. Н. Прикладное программирование [Электронный ресурс]:учебное пособие для спо. - Санкт-Петербург: Лань, 2020. - 124 с. – Режим доступа: https://e.lanbook.com/book/147402
7.2. Лицензионное и свободно распространяемое программное обеспечение в том числе отечественного производства	
7.2.1	Microsoft Office 2013 Standard
7.2.2	Kaspersky Endpoint Security
7.2.3	Microsoft®WINHOME 10 Russian Academic OLP lLicense NoLevel Legalization GetGenuine
7.2.4	Microsoft Visual Studio Professional 2017
7.2.5	Notepad ++
7.2.6	Microsoft Visual Code
7.3. Перечень профессиональных баз данных, информационных справочных систем и ресурсов сети Интернет	
7.3.1	Электронно-библиотечная система "Лань". Режим доступа: https://e.lanbook.com/
7.3.2	Электронно-библиотечная система "Университетская библиотека онлайн". Режим доступа: https://biblioclub.ru/
7.3.3	Электронно-библиотечная система "BOOK.ru". Режим доступа: https://book.ru/
7.3.4	Единое окно доступа к образовательным ресурсам. Режим доступа: http://window.edu.ru/
7.3.5	ПЛАТФОРМА ОНЛАЙН-ОБРАЗОВАНИЯ «РАЗУМ». Режим доступа: https://razoom.mgutm.ru/
7.3.6	"Электронная библиотека учебников" . Режим доступа: http://studentam.net/

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1	Адрес: 344002, Ростовская область, г. Ростов-на-Дону, пер. Семашко, д. 55: аудитория 20-28 - Компьютерный класс : 15 персональных компьютеров, объединённых в локальную вычислительную сеть на базе выделенного сервера приложений и web-сервера, наглядными пособиями, Мультимедийная установка «Toshiba» с экраном 3,5 м. Лицензионные программы:. ПО ОС Windows XP professional, MS Office 2003. Microsoft Visual Studio 2010-2013. Opera, Mozilla Firefox, Adobe Acrobat Reader 3.01,. WinDjView, КОМПАС-3D LT, 1С Предприятие 8.1, Система для моделирования бизнеса и оценки бизнес-проектов «Project Expert 7 ─ Эксперт Системс» (сетевая версия). CorelDRAW Graphics Suite X7. GRAPHISOFT ArchiCAD 15, Autodesk DWG Viewer, AVS DVD Player 2.4, CorelDraw Graphics Suite X4, Google SketchUp6.7, QuickTime, QBacic, Pascal, Delphi, Kaspersky Endpoint Security. IQlib. Все компьютеры подключены к сети Internet
8.2	Адрес: 344002, Ростовская область, г. Ростов-на-Дону, пер. Семашко, д. 55: аудитория 20-29 - Компьютерный класс : Оснащен: 12 персональными компьютерами, объединёнными в локальную вычислительную сеть на базе выделенного сервера приложений и web-сервера, наглядными пособиями, Мультимедийным проектором: Nec VT48 ANSI, с экраном 3,5 м. Лицензионные программы: ПО ОС Windows XP professional, MS Office 2003. Microsoft Visual Studio 2010-2013. Google Chome, Opera, Mozilla Firefox, Adobe Acrobat Reader 3.01, QBacic, Pascal, Delphi. WinDjView, КОМПАС-3D LT, GRAPHISOFT ArchiCAD 15, Audit XP. Система для моделирования бизнеса и оценки бизнес-проектов «Project Expert 7» (сетевая версия). 1С Предприятие 8.1, Autodesk DWG Viewer, AVS DVD Player 2.4, CorelDraw Graphics Suite X4, Google SketchUp6.7, QuickTime, Kaspersky Endpoint Security. IQlib. Инженерный калькулятор. Консультант Плюс. Все компьютеры подключены к сети Internet.

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Организация образовательного процесса для лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе

оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Студенты с ограниченными возможностями здоровья, в отличие от остальных студентов, имеют свои специфические особенности восприятия, переработки материала. Подбор и разработка учебных материалов производится с учетом индивидуальных особенностей. Предусмотрена возможность обучения по индивидуальному графику, при составлении которого возможны различные варианты проведения занятий: в академической группе и индивидуально, на дому с использованием дистанционных образовательных технологий.

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2027 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2027 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2028 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2028 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2029 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2029 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2030 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры
Физика, математика и информационные технологии
Протокол от __ _____ 2030 г. № __
И.о.зав. кафедрой Левчук Валерия Владимировна _____