

УТВЕРЖДАЮ

Заместитель директора УрСЭИ (филиал)
ОУП ВО "АТиСО" по УВР

 О.В. Зубкова

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
Программирование на языке Python

Направление подготовки

09.03.03 Прикладная информатика

(код и наименование направления подготовки)

Направленность (профиль)

Корпоративные информационные системы

Квалификация выпускника

Бакалавр

Кафедра: Информатики и математики

Разработчики программы:

Кондрашкин Владимир Валентинович

Программа обсуждена на заседании кафедры Информатики и математики

Заведующий кафедрой Информатики и математики

ОГЛАВЛЕНИЕ

1. ЦЕЛЬ И ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	3
2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	3
2.1. Планируемые результаты обучения по дисциплине	3
2.2. Результаты освоения образовательной программы	3
3. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	4
4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) И ВИДЫ УЧЕБНОЙ РАБОТЫ	4
5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	5
5.1. Содержание дисциплины (модуля)	5
5.2. Разделы / темы дисциплины, их трудоемкость и виды занятий	6
6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)	7
7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА	10
8. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ И ИНТЕРНЕТ - РЕСУРСОВ, НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	11
9. ТРЕБОВАНИЯ К МАТЕРИАЛЬНО-ТЕХНИЧЕСКОМУ И УЧЕБНО-МЕТОДИЧЕСКОМУ ОБЕСПЕЧЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)	12
10. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	13
11. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ	13

1. ЦЕЛЬ И ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Цель:

Сформировать у студентов представление о современном языке программирования Python, приобретение навыков программирования в скриптовых языках, приобретение навыков использования библиотек и модулей для ускоренной обработки данных, использование модульного программирования для решения практических задач.

Задачи:

- 1.изучить Python как мультипарадигменное языковое средство, достаточно полно отражающее современные концепции разработки ПО
- 2.привить практические навыки разработки программных приложений на языке программирования Python
- 3.усовершенствовать и углубить навыки объектно-ориентированного и функционального программирования
- 4.познакомиться с основами создания приложений для взаимодействия с базами данных

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

2.1 Планируемые результаты обучения по дисциплине

Изучение дисциплины Программирование на языке Python направлено на формирование следующих компетенций:

профессиональных компетенций и индикаторов их достижения:

Категория (группа) ПК	Код и наименование универсальной компетенции	Код и наименование индикатора достижений универсальной компетенции
Проектные	ПК-1 Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов.	ПК-1.1. Знает методы описания алгоритмов, основные абстрактные типы данных и их реализации; основные способы верстки webстраниц, современные языки разметки; современные системы управления контентом сайта (CMS); основы создания интернет-приложений на основе ASP.NET с использованием языка программирования C# (в среде Visual Studio); основные команды для платформы 1С, приемы программирования в 1С способы интеграции различных модулей.. ПК-1.2. Умеет разрабатывать алгоритмы для конкретных задач, учитывать различия в представлении информационных объектов современными браузерами; проводить работы по интеграции программных модулей с использованием языков программирования; разрабатывать программные модули в 1С для заданной организации, встраивать их в существующую систему и проверять работоспособность разработанных модулей и системы в целом тестировать написанную программу на наличие ошибок, интегрировать различные модули в одну информационную систему.. ПК-1.3. Владеет навыками использования языка разметки гипертекста с языками программирования; навыками работы с программными комплексами и проверкой работоспособности выпусков программного продукта; навыками работы в 1С Конфигураторе и программирования на платформе 1С; способностью проводить интеграцию программных модулей, навыками тестирования разработанной программы..

2.2. Результаты освоения образовательной программы:

В результате изучения дисциплины студент должен:

Знать:

- 1.основные понятия языка программирования Python
- 2.методы описания алгоритмов, основные абстрактные типы данных и их реализации
- 3.методы объектно-ориентированного программирования.
- 4.классы задач, формулируемых и решаемых на Python

Уметь:

- 1.формулировать требования к создаваемым программным комплексам.
- 2.разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования.
- 3.проводить работы по интеграции программных модулей с использованием современных языков программирования
- 4.разрабатывать программы на языке Python

Владеть (навыки и/или опыт деятельности):

- 1.навыками работы в современной программно-технической среде в различных операционных системах.
- 2.навыками разработки программных комплексов для решения прикладных задач, оценки сложности алгоритмов и программ, использования современных технологий программирования, тестирования и документирования программных комплексов.
- 3.навыками работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов.
- 4.навыками моделирования прикладных задач; численными методами; навыками программирования в современных средах.
- 5.навыками самостоятельного создания простых приложений

3. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к дисциплинам части, формируемой участниками образовательных отношений направления подготовки 09.03.03 Прикладная информатика профиль "Корпоративные информационные системы".

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) И ВИДЫ УЧЕБНОЙ РАБОТЫ

Объем дисциплины	Всего часов		
	Для очной формы обучения	Для очно-заочной формы обучения	Для заочной формы обучения
Общая трудоемкость дисциплины (зачетных единиц/часов)	4 / 144	0 / 0	4 / 144
Контактная работа – аудиторные занятия:	58		16
Лекции	20		6
Семинары, практические занятия	34		10
Контактная работа (зачет, экзамен, курсовая работа)	4		
Самостоятельная работа обучающегося	54		119
Промежуточная аттестация	32		9
Вид промежуточной аттестации	Экзамен		Экзамен

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание дисциплины (модуля)

Тема 1. Установка и настройка среды программирования Python.

Интерпретаторы и компиляторы. Загрузка среды программирования Python с сайта разработчика. Установка Python в ОС Windows. Знакомство с интерфейсом среды программирования Python. Запуск программ, написанных на Python через командную строку ОС. Стандартные модули Python. Загрузка модулей в среду программирования и выполнение функция модуля.

Тема 2. Типы данных и составные объекты Python.

Ввод и вывод на Python. Форматированный ввод и вывод. Типизация Python. Специальные типы. Числа: целые, с плавающей точкой и комплексное. Строки. Последовательности. Отображения. Арифметические выражения. Составные объекты. Списки. Дополнительные возможности при конструировании списков. Кортежи. Множества. Словари. Файлы. Вспомогательные типы. Стеки. Очереди. Работа с файлами на диске.

Тема 3. Управляющие конструкции ходом выполнения программ на Python.

Полное и неполное ветвление. Реализация многовариантного выбора через ветвления. Параметрический цикл с вариативной и без вариативной части. Цикл с предусловием. Команды прерывания цикла и ее продолжения со следующей итерации (break, continue). Выполнение команд внутри контекста (with). Обработка исключений в программах. Встроенные типы исключений и определение новых исключений.

Тема 4. Пользовательские функции и основы функционального программирования.

Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. Функции, как объекты и замыкания. Декораторы. Итераторы. Генераторы и сопрограммы. Генераторы списков. Выражения-генераторы. Основы декларативного программирования. Оператор Lambda. Атрибуты функций. Выполнение неопределенных функций, командами: eval, exec, compile. Разработка рекурсивной функции на базе процедурной ее формы. Последовательная, параллельная и псевдопараллельная рекурсии. Обработка списков рекурсивными функциями. Работа со списком, как со стеком и очередью.

Тема 5. Модульное программирование. Стандартные и нестандартные модули Python.

Создание и использование модуля. Поиск модулей и компилированные файлы. Стандартные модули: sys, os. Пакеты. Краткая характеристика нестандартных модулей Python.

Тема 6. Математические модули.

Модуль чисел с плавающей точкой Decimal. Модуль рациональных чисел Fractions. Модуль стандартных математических функций Math. Модуль абстрактных базовых классов Numbers. Модуль псевдослучайных чисел Random. Модуль для работы с комплексными числами CMath. Модуль для работы с массивами Array. Модуль сортировки списков Bisect.

Тема 7. Разработка модуля численных методов.

Численные методы решения алгебраических и трансцендентных уравнений. Численные методы решения систем алгебраических уравнений. Численные методы интерполирования. Численные методы интегрирования. Статистическая обработка данных методом наименьших квадратов.

Тема 8. Работа с текстом и строками.

Кодировки символов на Python. Операции над строками. Стандартные модули обработки строк String и Codecs. Модуль приблизительного сравнения двух строк DiffLib. Модуль для работы с кодировкой и регулярными выражениями Re. Модуль преобразования данных Struct. Модуль доступа к базе символов UnicodeData.

Тема 9. Доступ к средствам операционной системы.

Модуль FileCmp для сравнения файловых объектов и каталогов ОС. Модуль Time для работы с системным временем. Модуль Sched для планирования задач ОС. Модуль GetPass для работы с секретной информацией. Модуль GetOpt для обработки данных командной строки. Модуль TempFile для работы с временными файлами. Модули Glob и fnMatch для работы с шаблонами. Модуль Mmap для отображения файлов в память.

5.2 Разделы/темы дисциплины, их трудоемкость и виды занятий Общая трудоемкость изучения дисциплины по учебному плану: 3.Е. (часов) 4/144

Форма обучения: очная

Вид контроля: Экзамен

№	Наименование темы	Всего	Лекции	Мини-ар	ци/А	те	ль	на	ом	пе	те	а	(с	ы
1	Тема 1. Установка и настройка среды программирования Python.	11	2	3		6				ПК-1			Л.1-7	
2	Тема 2. Типы данных и составные объекты Python.	11	2	3		6				ПК-1			Л.1-7	
3	Тема 3. Управляющие конструкции ходом выполнения программ на Python.	12	2	4		6				ПК-1			Л.1-7	
4	Тема 4. Пользовательские функции и основы функционального программирования.	12	2	4		6				ПК-1			Л.1-7	
5	Тема 5. Модульное программирование. Стандартные и нестандартные модули Python.	12	2	4		6				ПК-1			Л.1-7	
6	Тема 6. Математические модули.	12	2	4		6				ПК-1			Л.1-7	
7	Тема 7. Разработка модуля численных методов.	12	2	4		6				ПК-1			Л.1-7	
8	Тема 8. Работа с текстом и строками.	13	3	4		6				ПК-1			Л.1-7	
9	Тема 9. Доступ к средствам операционной системы.	13	3	4		6				ПК-1			Л.1-7	
10	Консультации (контактная)	4			4									
11	Промежуточная аттестация (часов)	32			32									
	Всего часов	144	20	34	36	54								

Форма обучения: заочная

Вид контроля: Экзамен

№	Наименование темы	Вс	ег	о	Ле	кц	ии	м	ин	ар	ци	и /	А	те	ль	на	ом	пе	те	а	(с	сы
1	Тема 1. Установка и настройка среды программирования Python.	15			1			1						13				ПК-1				Л.1-7
2	Тема 2. Типы данных и составные объекты Python.	15			1			1						13				ПК-1				Л.1-7
3	Тема 3. Управляющие конструкции ходом выполнения программ на Python.	15			1			1						13				ПК-1				Л.1-7
4	Тема 4. Пользовательские функции и основы функционального программирования.	15			1			1						13				ПК-1				Л.1-7
5	Тема 5. Модульное программирование. Стандартные и нестандартные модули Python.	15			1			1						13				ПК-1				Л.1-7
6	Тема 6. Математические модули.	15			1			1						13				ПК-1				Л.1-7

7	Тема 7. Разработка модуля численных методов.	14		1		13	ПК-1	Л.1-7
8	Тема 8. Работа с текстом и строками.	15		1		14	ПК-1	Л.1-7
9	Тема 9. Доступ к средствам операционной системы.	16		2		14	ПК-1	Л.1-7
10	Консультации (контактная)							
11	Промежуточная аттестация (часов)	9			9			
	Всего часов	144	6	10	9	119		

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Самостоятельная работа включает следующие ее виды:

- изучение тем самостоятельной подготовки по учебно-тематическому плану;
- самоподготовка к практическим и другим видам занятий – расчетно-аналитическая работа по индивидуальному учебному заданию;
- самостоятельная работа слушателей при подготовке к итоговому тестированию.

Эти виды работ предполагают:

- самостоятельную работу слушателя в библиотеке;
- изучение материалов специализированных сайтов по темам дисциплины в сети Интернет;
- изучение электронных учебных материалов (электронных учебников, методических материалов, представленных на сайте УрСЭИ);

Задания для самостоятельной работы слушателей включают в себя:

- выполнение тестовых заданий, размещенных в программной оболочке Moodle – по темам, вынесенным на самостоятельное изучение;
- подготовка докладов по отдельным вопросам курса.

Указания и задания по самостоятельной работе выдаются на практических занятиях с учетом индивидуального уровня освоения материала слушателями.

Перечень вопросов для самостоятельной работы

1. Типы данных.
2. Переменные.
3. Числовые типы данных.
4. Операции над числовыми типами данных.
5. Строки.
6. Строки unicod.
7. Вывод данных.
8. Ввод данных.
9. Форматированный ввод/вывод.
10. Списки.
11. Выражения в списках.
12. Оператор del.
13. Использование списков, как стеков.
14. Использование списков, как очередей.
15. Операции сравнения для списков.
16. Диапазоны.
17. Кортежи. Отличие кортежей от словарей
18. Словари.
19. Оператор if. Особенности операторов сравнения.
20. Операторы цикла. Оператор for. Оператор while. Завершение цикла.
21. Продолжение цикла. Оператор pass.
22. Определение функции.
23. Пространство имен функции.
24. Передача параметров. Ключи.
25. Передача в функцию переменного числа аргументов.
26. Элементы функционального программирования.
27. Использование лямбда функций.
28. Функции работы со структурами данных.
29. Функция map(). Примеры применения
30. Функция filter(). Примеры применения
31. Функция reduce(). Примеры применения
32. Документирование функций.
33. Создание модулей.
34. Указание кодировки.
35. Поиск модулей.
36. Компиляция модулей на Python.
37. Стандартные модули Python.
38. Использование функции __dir__.
39. Структурирование модулей в пакеты.
40. Импорт модулей и их составляющих из пакета.
41. Ссылки в пакетах.
42. Пакеты и файловая система.
43. Класс File.
44. Открытие файла.
45. Методы класса для File ввода-вывода.
46. Взаимодействие с файловой системой.
47. Модуль path.
48. Объекты и файловый ввод-вывод.
49. Объявление класса
50. Управление атрибутами и методами класса
51. Объявление объектов
52. Множественное наследование
53. Заимствование свойств и методов у родительского класса

Тема 1. Установка и настройка среды программирования Python.

Вопросы для дискуссии

1. Загрузка среды программирования Python с сайта разработчика. 2. Установка Python в ОС Windows. 3. Знакомство с интерфейсом среды программирования Python. Запуск программ, написанных на Python через командную строку ОС. 4. Стандартные модули Python. Загрузка модулей в среду программирования и выполнение функция модуля.

Практические задания

Установить среду разработки на личный ПК.

Тема 2. Типы данных и составные объекты Python.

Вопросы для дискуссии

1. Ввод и вывод на Python. Форматированный ввод и вывод. 2. Типизация Python. Специальные типы. Числа: целые, с плавающей точкой и комплексное. 3. Строки. Последовательности. Отображения. 4. Арифметические выражения. 5. Составные объекты. Списки. Дополнительные возможности при конструировании списков. 6. Кортежи. Множества. Словари. Файлы. 7. Вспомогательные типы. Стеки. Очереди. 8. Работа с файлами на диске.

Практические задания

Написать программу преобразования строки 123 в число 123.

Тема 3. Управляющие конструкции ходом выполнения программ на Python.

Вопросы для дискуссии

1. Полное и неполное ветвление. Реализация многовариантного выбора через ветвления. 2. Параметрический цикл с вариативной и без вариативной части. 3. Цикл с предусловием. Команды прерывания цикла и ее продолжения со следующей итерации (break, continue). 4. Выполнение команд внутри контекста (with). Обработка исключений в программах. Встроенные типы исключений и определение новых исключений.

Практические задания

Есть список a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]. Выведите все элементы, которые меньше 5.

Тема 4. Пользовательские функции и основы функционального программирования.

Вопросы для дискуссии

1. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. 2. Функции, как объекты и замыкания. 3. Декораторы. Итераторы. 4. Генераторы и сопрограммы. Генераторы списков. Выражения-генераторы. 5. Основы декларативного программирования. Оператор Lambda. Атрибуты функций. Выполнение неопределенных функций, командами: eval, exec, compile. 6. Разработка рекурсивной функции на базе процедурной ее формы. Последовательная, параллельная и псевдопараллельная рекурсии. Обработка списков рекурсивными функциями. Работа со списком, как со стеком и очередью.¶

Практические задания

Отсортируйте словарь по значению в порядке возрастания и убывания.

Тема 5. Модульное программирование. Стандартные и нестандартные модули Python.

Вопросы для дискуссии

1. Создание и использование модуля. 2. Поиск модулей и компилированные файлы. 3. Стандартные модули: sys, os. 4. Краткая характеристика нестандартных модулей Python.

Практические задания

Даны списки: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]; b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Нужно вернуть список, который состоит из элементов, общих для этих двух списков.

Тема 6. Математические модули.

Вопросы для дискуссии

1. Модуль чисел с плавающей точкой Decimal. 2. Модуль рациональных чисел Fractions. 3. Модуль стандартных математических функций Math. 4. Модуль абстрактных базовых классов Numbers. 5. Модуль псевдослучайных чисел Random. 6. Модуль для работы с комплексными числами CMath. 7. Модуль для работы с массивами Array. 8. Модуль сортировки списков Bisect.

Практические задания

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

Тема 7. Разработка модуля численных методов.

Вопросы для дискуссии

1. Численные методы решения алгебраических и трансцендентных уравнений. 2. Численные методы решения систем алгебраических уравнений. 3. Численные методы интерполирования. 4. Численные методы интегрирования. 5. Статистическая обработка данных методом наименьших квадратов.

Практические задания

Напишите программу, которая считывает три числа и выводит их сумму. Каждое число записано в отдельной строке.

Тема 8. Работа с текстом и строками.

Вопросы для дискуссии

1. Кодировки символов на Python. 2. Операции над строками. Стандартные модули обработки строк String и Codecs. 3. Модуль приблизительного сравнения двух строк DiffLib. 4. Модуль для работы с кодировкой и регулярными выражениями Re. 5. Модуль преобразования данных Struct. 6. Модуль доступа к базе символов UnicodeData.

Практические задания

Напишите программу, преобразующую строку Иванов Иван Иванович с три отдельных элемента (имя, отчество, фамилия).

Тема 9. Доступ к средствам операционной системы.

Вопросы для дискуссии

1. Модуль FileCmp для сравнения файловых объектов и каталогов ОС. 2. Модуль Time для работы с системным временем. 3. Модуль Sched для планирования задач ОС. 4. Модуль GetPass для работы с секретной информацией. 5. Модуль GetOpt для обработки данных командной строки. 6. Модуль TempFile для работы с временными файлами. 7. Модули Glob и fnMatch для работы с шаблонами. 8. Модуль Mmap для отображения файлов в память.

Практические задания

Напишите программу, позволяющую выбрать файл на диске вашего ПК.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА, НЕОБХОДИМАЯ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Основная литература

1. Черпаков, И. В. Алгоритмизация и программирование на Python : учебник для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2026. — 159 с. — (Высшее образование). — ISBN 978-5-534-21910-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582412>.

2. Майтак, Р. В. Python, Django, Data Science = [Python, Django, Наука о данных] : учебное пособие : [16+] / Р. В. Майтак, П. А. Пылов, А. В. Протодяконов. – Москва ; Вологда : Инфра-Инженерия, 2025. – 516 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=725620>. – Библиогр. в кн. – ISBN 978-5-9729-2143-0. – Текст : электронный.
3. Калитвин, В. А. Введение в программирование на Python : учебное пособие : [16+] / В. А. Калитвин ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2023. – 85 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=714538>. – Библиогр. в кн. – ISBN 978-5-907655-86-7. – Текст : электронный.
4. Федоров, Д. Ю. Программирование на Python : учебник для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 166 с. — (Высшее образование). — ISBN 978-5-534-22181-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/600874>.

Дополнительная литература

5. Баяк, О. А. Практикум по анализу данных на языках Python и R : учебное пособие для студентов, обучающихся по направлениям 38.03.01 «Экономика», 38.03.05 «Бизнес-информатика» : [16+] / О. А. Баяк, М. Р. Исаева, М. О. Самсонкин ; Финансовый университет при Правительстве Российской Федерации. – Москва : Прометей, 2023. – 100 с. : ил., табл., схем., граф. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=700938>. – Библиогр. в кн. – ISBN 978-5-00172-356-1. – Текст : электронный.
6. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588667>.
7. Богун, В. В. Программирование задач по финансовой математике на языке программирования Python : учебное пособие : [16+] / В. В. Богун ; Финансовый университет при Правительстве Российской Федерации, Департамент математики. – Москва : Прометей, 2025. – 172 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=721368>. – Библиогр. в кн. – ISBN 978-5-00172-780-4. – Текст : электронный.

8. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ И ИНТЕРНЕТ - РЕСУРСОВ, НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Программное обеспечение

№	Наименование	Описание
1	MS Office 2007 - 2019	Пакеты прикладных программ, включающие текстовые процессоры, табличные процессоры, инструменты создания презентаций.
2	Microsoft SQL Server	Система управления реляционными базами данных компании Microsoft.
3	MySQL	Системы управления базами данных MySQL.
4	QT среда разработки	Интегрированная среда разработки программного обеспечения.

5	PyCharm	Интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django.
---	---------	--

Интернет-ресурсы, информационные и справочные системы

№	Наименование	Описание
1	IT-портал «Сервер Информационных Технологий»	http://citforum.ru/
2	MSDN – сеть разработчиков Microsoft	https://msdn.microsoft.com/ru-ru/
3	Интернет Университет Информационных Технологий - бесплатное дистанционное образование.	www.intuit.ru/
4	Информационный ресурс для программистов SQL.	http://www.sql.ru/
5	Обучение Python. Сайт-самоучитель для начинающих.	https://python-ucheba.ru/samouchitel-python
6	Самоучитель Python.	http://pythonworld.ru/samouchitel-python

9. ТРЕБОВАНИЯ К МАТЕРИАЛЬНО-ТЕХНИЧЕСКОМУ И УЧЕБНО-МЕТОДИЧЕСКОМУ ОБЕСПЕЧЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Помещения, в которых проводятся занятия, представляют собой учебные аудитории для проведения учебных занятий. Данные аудитории оснащенные оборудованием и техническими средствами обучения.

Компьютеры, информационно-телекоммуникационные сети, аппаратно-программные и аудиовизуальные средства, находящиеся в учебных аудиториях:

- персональный компьютер или ноутбук (для преподавателя) с выходом в сеть интернет;
- проектор;
- акустическая система
- экран для проектора;
- доска маркерная (ученическая доска);

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой и имеют возможность подключения к сети интернет и обеспечивают доступ к электронной информационно-образовательной среде академии.

10. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В процессе освоения дисциплины используются следующие образовательные технологии:

1. Стандартные методы обучения:

- лекции;
- практические занятия, на которых обсуждаются основные проблемы, освещенные в лекциях и сформулированные в домашних заданиях;
- письменные или устные домашние задания;
- самостоятельная работа студентов, в которую входит освоение теоретического материала, подготовка к практическим/лабораторным занятиям, выполнение указанных выше письменных/устных заданий, работа с литературой.

2. Методы обучения с применением интерактивных форм образовательных технологий:

- круглые столы;
- групповые дискуссии и проекты;
- обсуждение результатов работы студенческих исследовательских групп.

11. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Код комп.	Идентиф.	Тип задания	Уровень	Время (мин)	Задание	Ключ
П К-1	Знать	Задание закрытого типа на установление соответствия	Базовый	3	Установите соответствие между типом данных Python (А–Г) и его описанием (1–4). Типы: А) Список (list); Б) Кортеж (tuple); В) Словарь (dict); Г) Множество (set). Описания: 1) Неупорядоченная коллекция уникальных элементов; 2) Упорядоченная изменяемая коллекция элементов; 3) Упорядоченная неизменяемая коллекция элементов; 4) Коллекция пар «ключ-значение».	А-2, Б-3, В-4, Г-1

П К-1	Уметь	Задание закрытого типа на установление соответствия	Повышенный	4	Установите соответствие между фрагментом кода Python (А–Г) и результатом его выполнения (1–4). Код: А) <code>print("Hello"[-1]);</code> Б) <code>print([1,2,3].pop());</code> В) <code>print(3 * 'Ha');</code> Г) <code>print("Hello"[:4]).</code> Результаты: 1) 'Hell'; 2) 'o'; 3) 3; 4) 'HaHaHa'.	А-2, Б-3, В-4, Г-1
П К-1	Знать	Задание закрытого типа на установление последовательности	Базовый	3	Расположите этапы выполнения цикла for при переборе списка в правильном порядке: 1) Выполняется тело цикла для текущего элемента; 2) Цикл завершается, когда элементы закончились; 3) Переменная цикла принимает следующий элемент списка; 4) Переменная цикла инициализируется первым элементом списка.	4, 1, 3, 2
П К-1	Уметь	Задание закрытого типа на установление последовательности	Повышенный	4	Расположите действия для чтения всех строк из текстового файла и вывода их на экран в правильной последовательности: 1) Вызвать метод <code>readlines();</code> 2) Открыть файл с	2, 1, 3, 4

					помощью open(); 3) Вывести полученные строки в цикле; 4) Закроить файл (или использовать with).	
П К-1	Практический опыт	Задание закрытого типа на установление последовательности	Повышенный	4	Расположите шаги по отладке программы, вызывающей исключение ZeroDivisionError, в правильном порядке: 1) Исправить код, добавив проверку делителя; 2) Запустить программу и убедиться, что ошибка исчезла; 3) Обернуть потенциально опасный участок в блок try-except; 4) Определить строку, в которой происходит деление на ноль.	4, 3, 1, 2
П К-1	Практический опыт	Задание комбинированного типа с выбором одного верного ответа и обоснованием	Повышенный	5	Дан список nums = [5, 2, 8, 1, 9]. Какой результат будет получен после выполнения кода: nums.sort(); print(nums[-2])? Варианты: 1) 8; 2) 2; 3) 5; 4) 9.	Ответ: 1. Обоснование: sort() сортирует список по возрастанию: [1,2,5,8,9]; индекс -2 соответствует предпоследнему элементу – 8.
П К-1	Знать	Задание комбинированного типа с выбором одного верного ответа и обоснованием	Базовый	2	Что такое «генератор списка» (list comprehension) в Python? Варианты: 1) Функция, создающая список из другого списка; 2) Синтаксическая	Ответ: 2. Обоснование: Генератор списка – компактный способ создания списка, например [x**2 for x in

					конструкция для создания списка на основе цикла и выражения; 3) Встроенный модуль для работы со списками; 4) Метод списка, добавляющий элементы.	range(10)].
П К-1	Уметь	Задание комбинированного типа с выбором одного верного ответа и обоснованием	Повышенный	5	Какой код правильно открывает файл data.txt для чтения и выводит его содержимое? Варианты: 1) f = open('data.txt'); print(f); f.close(); 2) with open('data.txt') as f: print(f.read()); 3) open('data.txt', 'r').readlines(); 4) file = open('data.txt'); file.read(); file.close.	Ответ: 2. Обоснование: Вариант 2 использует менеджер контекста with, который автоматически закрывает файл, и метод read() читает всё содержимое.
П К-1	Практический опыт	Задание комбинированного типа с выбором нескольких верных ответов и развёрнутым обоснованием	Повышенный	6	Какие из следующих действий помогут обработать ошибку ввода пользователя (вместо числа введена строка) в программе, запрашивающей число? Выберите два: 1) Использовать if not input().isdigit(): до преобразования; 2) Обернуть преобразование в try-except и перехватить ValueError; 3)	Ответы: 1, 2. Обоснование: Оба варианта корректно проверяют ввод: isdigit() проверяет, что строка состоит из цифр; try-except перехватывает исключение при преобразовании.

					Игнорировать ввод и завершить программу; 4) Использовать assert для проверки типа.	
П К-1	Знать	Задание комбинированного типа с выбором нескольких верных ответов и развёрнутым обоснованием	Базовый	3	Выберите два правильных утверждения о словарях Python: 1) Ключи словаря могут быть любого хешируемого типа (строки, числа, кортежи); 2) Словари сохраняют порядок добавления элементов (начиная с Python 3.7); 3) Значения словаря не могут быть изменяемыми объектами; 4) Для удаления элемента по ключу используется метод popitem().	Ответы: 1, 2. Обоснование: Ключи должны быть неизменяемыми и хешируемыми; с Python 3.7+ словари сохраняют порядок вставки. Значения могут быть любыми, popitem() удаляет последний элемент, а не по ключу.
П К-1	Уметь	Задание комбинированного типа с выбором нескольких верных ответов и развёрнутым обоснованием	Повышенный	5	Выберите два фрагмента кода, которые корректно создают список квадратов чисел от 1 до 5: 1) squares = [] for i in range(1,6): squares.append(i**2); 2) squares = [x**2 for x in range(1,5)]; 3) squares =	Ответы: 1, 3. Обоснование: Вариант 1 – классический цикл; вариант 3 – использование map и lambda. Вариант 2 генерирует квадраты для 1..4; вариант 4 – для 0..4.

					list(map(lambda x: x**2, range(1,6))); 4) squares = [i*i for i in range(5)].	
П К-1	Уметь	Задание комбинированног о типа с выбором нескольких верных ответов и развёрнутым обоснованием	Высокий	7	Выберите два правильных способа передать функции произвольное количество аргументов: 1) Использовать параметр вида *args в определении функции; 2) Использовать параметр вида **kwargs для именованных аргументов; 3) Определить параметр со значением по умолчанию; 4) Использовать глобальную переменную.	Ответы: 1, 2. Обоснование: *args собирает позиционные аргументы в кортеж, **kwargs – именованные в словарь. Остальные варианты не подходят.
П К-1	Знать	Задание открытого типа	Базовый	3	Что такое «исключение» в Python и для чего используется блок try-except?	Исключение – это объект, сигнализирующ ий об ошибке во время выполнения программы. Блок try-ехсепт позволяет перехватить исключение и выполнить код обработки ошибки, предотвращая аварийное завершение

						программы.
П К-1	Уметь	Задание открытого типа	Высокий	8	Напишите код функции, которая принимает список чисел и возвращает новый список, содержащий только чётные числа.	<pre>def filter_even(nums) : return [x for x in nums if x % 2 == 0]</pre>
П К-1	Уметь	Задание закрытого типа на установление соответствия	Базовый	3	Установите соответствие между методом списка Python (А–Г) и его действием (1–4). Методы: А) append(); Б) extend(); В) pop(); Г) remove(). Действия: 1) Удаляет первый элемент с указанным значением; 2) Добавляет элемент в конец списка; 3) Удаляет и возвращает элемент по индексу (по умолчанию последний); 4) Добавляет все элементы итерируемого объекта в конец списка.	А-2, Б-4, В-3, Г-1