Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 30.05.2026 14:54:43
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационных технологий»
_____ / Д.Г. Демидов /
« 26 » февраля 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование

Направление подготовки/специальность
09.03.03 Прикладная информатика

Профиль/специализация
Большие и открытые данные

Квалификация
бакалавр

Формы обучения
очная, заочная

Москва, 2026 г.

Разработчик(и):

Старший преподаватель  / О.В. Дедёхина /

Согласовано:
Заведующий кафедрой «Прикладная информатика»,
К.э.н, доцент  / С.В. Суворов /

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 6
3.3 Содержание дисциплины 6
3.4 Тематика семинарских/практических и лабораторных занятий 8
3.5 Тематика курсовых проектов (курсовых работ) 10
4 Учебно-методическое и информационное обеспечение 10
4.1 Нормативные документы и ГОСТы 10
4.2 Основная литература 11
4.3 Дополнительная литература 11
4.4 Электронные образовательные ресурсы 11
4.5 Лицензионное и свободно распространяемое программное обеспечение 11
4.6 Современные профессиональные базы данных и информационные справочные системы 11
5 Материально-техническое обеспечение 11
5.1 Требования к оборудованию и помещению для занятий 11
5.2 Требования к программному обеспечению 12
6 Методические рекомендации 12
6.1 Методические рекомендации для преподавателя по организации обучения 12
6.2 Методические указания для обучающихся по освоению дисциплины 12
7 Фонд оценочных средств 13
7.1 Методы контроля и оценивания результатов обучения 13
7.2 Шкала и критерии оценивания результатов обучения 13
7.3 Оценочные средства 16

1 Цели, задачи и планируемые результаты обучения по дисциплине

К **основным целям** освоения дисциплины относится:

- формирование у студентов знаний о современных принципах, методах и средствах программирования на примере программирования прикладных задач на языке Python;
- подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра по направлению, в том числе формирование умений по выявлению необходимых совершенствований и разработке нового программного обеспечения;
- закрепление получаемых в семестре знаний и навыков на практике;
- формирование взаимосвязей, получаемых в семестре знаний и навыков с изученными ранее и изучаемых параллельно с данной дисциплиной;
- подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра.

К **основным задачам** дисциплины относятся:

- освоение методологии, анализа и выбора принципов и методов программирования прикладных задач на языке программирования высокого уровня.
- изучение и освоение теоретического материала, как в процессе контактной, так и в ходе самостоятельной работы;
- выполнение предоставленных практических заданий различных форм, как в процессе контактной, так и в ходе самостоятельной работы;
- самостоятельная работа над тематикой дисциплины для формирования компетенций основной образовательной программы (далее, ООП).

Обучение по дисциплине «Программирование» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.	**Знать**: Способы инсталляции программного и аппаратного обеспечения для информационных и автоматизированных систем. **Уметь**: Инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем. **Владеть**: Инсталляции программного и аппаратного обеспечения для информационных и автоматизированных систем.
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	**Знать**: Этапы разработки, внедрения, адаптации и настройки, алгоритмы разработки программ для практического применения. **Уметь**: Разрабатывать алгоритмы прикладных программ; разрабатывать, устанавливать и настраивать прикладное программное обеспечение. **Владеть**: Навыками разработки алгоритмов и программ для практического применения, разработки, инсталляции и

	настройки прикладного программного обеспечения.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части (части, формируемой участниками образовательных отношений) блока Б1 «Дисциплины (модули)».

Дисциплина «Программирование» взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

В базовой части базового цикла (Б.1.1):

– Теоретические основы информатики;

– Вычислительные сети и системы;

– Математический анализ;

– Дискретная математика.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных(е) единиц(ы) (180 часов).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестр
			1
1	**Аудиторные занятия**	**48**	48
	В том числе:		
1.1	Лекции	16	16
1.2	Семинарские/практические занятия		
1.3	Лабораторные занятия	32	32
2	**Самостоятельная работа**	**132**	132
3	**Промежуточная аттестация**		
	Зачет/диф.зачет/экзамен		экзамен
	Итого:	**180**	180

3.1.2 Заочная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестр
			5
1	**Аудиторные занятия**	**20**	20
	В том числе:		
1.1	Лекции	8	8
1.2	Семинарские/практические занятия		
1.3	Лабораторные занятия	12	12
2	**Самостоятельная работа**	**160**	160
3	**Промежуточная аттестация**		
	Зачет/диф.зачет/экзамен		экзамен
	Итого:	**180**	180

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Раздел 1.						
1.1	Введение в программирование на Python	9	1				8
1.2	Операции. Оператор условия и циклы. Модули Math и Random	19	1		4		14
1.3	Строки. Строковые методы. Литералы строк	20	2		4		14
1.4	Словари, множества, кортежи	22	2		4		16
1.5	Списки	22	2		4		16
1.6	Функции	22	2		4		16
1.7	Работа с файлами и обработка исключительных ситуаций	22	2		4		16
1.8	ООП	22	2		4		16
1.9	Введение в функциональное программирование	22	2		4		16
Итого		**180**	**16**		**32**		**132**

3.2.2 Заочная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Раздел 1.						
1.1	Введение в программирование на Python	21	1		2		18
1.2	Операции. Оператор условия и циклы. Модули Math и Random	21	1		2		18
1.3	Строки. Строковые методы. Литералы строк	21	1		2		18
1.4	Словари, множества, кортежи	20	1		1		18

1.5	Списки	20	1		1		18
1.6	Функции	20	1		1		18
1.7	Работа с файлами и обработка исключительных ситуаций	20	1		1		18
1.8	ООП	19	1		1		17
1.9	Введение в функциональное программирование	18			1		17
Итого		**180**	**8**		**12**		**160**

3.3 Содержание дисциплины

Раздел 1.

Тема 1.1. Введение в программирование на Python

Общая информация о языке Python. История языка программирования, его связь с другими языками программирования, распространенность Python и основные сферы его применения. Знакомство с первыми примерами кода на Python. Философия Python.

Базовая информация о языке Python. Основные типы данных. Основные числовые типы данных и операции над ними. Математические операции над числовыми типами данных. Преобразование типов данных. Переменные и специфика их объявления. Статическая и динамическая типизация. Работа с переменными. Управление памятью и сборка мусора в Python. Именование переменных.

Тема 1.2. Операции. Оператор условия и циклы. Модули Math и Random

Управляющие конструкции в Python. Булев тип: объявление и операции. Операции сравнения в Python. Условные операторы в Python. Реализация задачи case в Python.

Циклы в Python: while, for. Специфика циклов в Python. Функции range и enumerate и их использование в циклах.

Тема 1.3. Строки. Строковые методы. Литералы строк

Работа со строками: создание строк, специальные символы. Индексирование строк, получение срезов строк. Основные функции для работы со строками. Вывод на экран (работа с функцией print) и форматирование строк. Различные подходы к форматированию строк, форматирование с помощью fстрок. Расширенное форматирование в Python.

Тема 1.4. Словари, множества и выражения-генераторы

Словари Python. Словари: семантика, синтаксис создания, операции над словарями, перебор элементов словаря.

Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python.

Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Пример: задача приведения списка к "плоскому" виду. Выражения-генераторы для множеств и словарей. Кейсы использования и производительность решений с использованием выражений-генераторов.

Тема 1.4 Списки

Списки и кортежи в Python. Специфика списков и их отличие от массивов. Создание списка, оперирование вложенными списками, копирование списков, операции над списками: индексация и срезы; изменение списка; поиск, сортировка и обход; изменение списка. Кортежи в Python: синтаксис, специфика использования.

Тема 1.5. Функции

Функции в Python: общая семантика. Создание функции и ее вызов. Расположение определений функций. Анонимные функции в Python. Необязательные параметры функций и сопоставление по ключам. Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции. Аннотации и документирование функций. Глобальные и локальные переменные.

Тема 1.6. Работа с файлами и обработка исключительных ситуаций

Обработка исключений в Python: кейсы для использования. Инструкция try ... except ... else ... finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert.

Работа с файлами в Python. Концепция файла в современных ОС и языках программирования. Операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов.

Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.

Тема 1.7. ООП в Python

Предпосылки и история появления ООП. Объекты и классы в ООП. Принципы и основные механизмы ООП. Логика работы абстракции, инкапсуляции, наследования и полиморфизма.

Python как объектно-ориентированный язык программирования. Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super(); проверка принадлежности к классу. Базовые типы в Python.

Тема 1.8. Введение в функциональное программирование

Парадигмы и идиомы программирования, общая концепция функционального программирования. Функциональные языки программирования.

Функции "граждане первого класса", функции высшего порядка, замыкания, функции без побочных эффектов, рекурсия, хвостовая рекурсия. Неизменяемые структуры данных. Идиомы, распространенные в функциональных языках программирования: итераторы, последовательности, ленивые вычисления, сопоставление с образцом, монады.

Элементы функционального программирования в Python: функции – граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python.

Декораторы в Python: использование и создание собственных декораторов.

Подход: map, filter, reduce. Реализация функций map, filter, reduce в Python.

Итераторы в Python, итерируемый тип данных. Модуль itertools.

Функции-генераторы и выражения-генераторы в Python.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

нет

3.4.2 Лабораторные занятия

Тема 1.2. Операции. Оператор условия и циклы. Модули Math и Random

Получение навыков работы с операторами преобразования и присваивания, условными операторами, операторами цикла, переменными, функциями, модулями Math и Random.

Задачи:

1. Изучение теоретического материала по конструкциям языка Python;
2. Создание в среде программирования на Python программ, операторами преобразования и присваивания, условными операторами, операторами цикла, переменными, функциями.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Дайте определение терминам инкремент/декремент. Уточните специфику применения и особенности в Python.
2. Назовите основной инструмент выбора в Python. .
3. Дайте характеристику оператору elif.
4. Дайте характеристику оператору continue.
5. Дайте характеристику оператору break..

6. Назовите операторы цикла. Дайте характеристику оператору if.
7. Назовите операторы цикла. Дайте характеристику оператору while.
8. Назовите типы данных и дайте характеристику типу None.
9. Дайте определение функции
10. Дайте определение модуля
11. Что такое базовые операции?

Тема 1.3. Строки. Строковые методы. Литералы строк

Цель: Получение навыков работы со строками в Python.

Задачи:

1. Изучение теоретического материала по работе с массивами в Python;
2. Создание в среде программирования на Python программ, работающих со строками.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Как проверить то, что каждое слово в строке начинается с заглавной буквы?
2. Как найти индекс первого вхождения подстроки в строку?
3. Как подсчитать то, сколько раз определённый символ встречается в строке?
4. Как получить срез строки?

Тема 1.4. Словари, множества и выражения-генераторы

Цель: Получение навыков работы с разными видами массивов в Python.

Задачи:

1. Изучение теоретического материала по работе с массивами в Python;
2. Создание в среде программирования на Python программ, работающих с разными видами массивов.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Перечислите возможные виды массивов в Python.
2. Перечислите способы создания словарей. Назовите метод, который возвращает значения в словаре.
3. Перечислите способы создания словарей. Назовите метод, который возвращает пары (ключ, значение).

Тема 1.4 Списки

Цель: Получение навыков работы со списками в Python.

Задачи:

1. Изучение теоретического материала по работе со списками в Python;
2. Создание в среде программирования на Python программ работающих со списками и множествами.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Списки. Создание, основные операции.
2. Основные методы списка.

Тема 1.5. Функции

Цель: Получение навыков работы с функциями в Python.

Задачи:

1. Изучение теоретического материала по работе с функциями в Python;
2. Создание в среде программирования на Python различных функций.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Назовите возможные виды функций в Python.
2. Каким образом возможно создать функцию с неизвестным количеством параметров?

Тема 1.6. Работа с файлами и обработка исключительных ситуаций

Цель: Получение навыков работы с файлами в Python.

Задачи:

1. Изучение теоретического материала по работе с файлами в Python;
2. Создание в среде программирования на Python программ, работающих с файлами.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Назовите способы открытия файлов в разных режимах в Python.
2. Как используют функцию print() для записи файлов?
3. Назовите методы чтения содержимого документов

Тема 1.7. ООП в Python

Цель: Получение навыков использования ооп в Python.

Задачи:

1. Изучение теоретического материала по ооп в Python;
2. Создание в среде программирования на Python программ с использованием классов.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Создание объектов в Python.
2. Управление доступом к атрибутам класса в Python.
3. Что такое наследование и полиморфизм;
4. Что такое функция super().

Тема 1.8. Введение в функциональное программирование

Цель: Получение навыков использования ооп в Python.

Задачи:

1. Изучение теоретического материала по ооп в Python;
2. Создание в среде программирования на Python программ с использованием классов.
3. Защита лабораторной работы.

Вопросы для подготовки к лабораторной работе:

1. Что такое «граждане первого класса»?.
2. Какие бывают рекурсии?
3. Что такое наследование и полиморфизм;
4. Что такое функция super().

3.5 Тематика курсовых проектов (курсовых работ)

Не предусмотрено

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный государственный образовательный стандарт высшего образования (уровень бакалавриата) по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 №922 «Об утверждении федерального государственного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.03 Прикладная информатика. https://fgos.ru/fgos/fgos-09-03-03-prikladnaya-informatika-922/

4.2 Основная литература

1. Рагимханова, Г. С. Программирование на Python : учебное пособие / Г. С. Рагимханова. — Махачкала : ДГПУ, 2022. — 126 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/330071 (дата обращения: 16.10.2023). — Режим доступа: для авториз. пользователей.

2. Тарланов, А. Т. Основы языка программирования Python : учебно-методическое пособие / А. Т. Тарланов, Ш. Г. Магомедов. — Москва : РТУ МИРЭА, 2019. — 107 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/171465 (дата обращения: 16.10.2023). — Режим доступа: для авториз. пользователей.

3. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. —Москва : ИНФРА-М, 2022. —343 с. —(Высшее образование: Бакалавриат). –ЭБС ZNANIUM.com. -URL: https://znanium.com/catalog/product/1356003 (дата обращения: 10.06.2022). –Текст : электронный

4.3 Дополнительная литература

1. Никитина, Т. П. Программирование. Основы Python / Т. П. Никитина, Л. В. Королев. — Санкт-Петербург : Лань, 2023. — 156 с. — ISBN 978-5-507-45283-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/302714 (дата обращения: 16.10.2023). — Режим доступа: для авториз. пользователей.

4.4 Электронные образовательные ресурсы

1.Программирование https://online.mospolytech.ru/course/view.php?id=8703

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Операционная система, Windows 11 (или ниже) - Microsoft Open License
2. Офисные приложения, Microsoft Office 2013(или ниже) - Microsoft Open License
3. Anaconda Python

4.6 Современные профессиональные базы данных и информационные справочные системы

1. Python Documentation [Электронный ресурс]: сайт. – Режим доступа: http://python.org/doc/
2. Python Standard Library [Электронный ресурс]: сайт. – Режим доступа: https://docs.python.org/2/library/
3. Официальный сайт продукта https://www.python.org/

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены

современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

5.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

Microsoft Windows.

Веб-браузер, Chrome.

ПО, предоставленное преподавателем.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, семинары и практики.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

самоконтроль и самооценка студента;

контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

уровень освоения студентом учебного материала;

умения студента использовать теоретические знания при выполнении практических задач;

сформированность компетенций;

оформление материала в соответствии с требованиями..

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

Лабораторные работы, экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.				
ОПК-5.1. Знать: Способы инсталляции программного и аппаратного обеспечения для информационных и автоматизированных систем. **ОПК-5.2. Уметь:** Инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем. **ОПК-5.3. Владеть**: Инсталляции программного и аппаратного обеспечения для информационных и автоматизированных систем.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.				
ОПК-7.1. Знать: Этапы разработки,	Обучающийся демонстрирует	Обучающийся демонстрирует	Обучающийся демонстрирует	Обучающийся демонстрирует

внедрения, адаптации и настройки, алгоритмы разработки программ для практического применения. **ОПК-7.2. Уметь:** Разрабатывать алгоритмы прикладных программ; разрабатывать, устанавливать и настраивать прикладное программное обеспечение. **ОПК-7.3. Владеть:** Навыками разработки алгоритмов и программ для практического применения, разработки, инсталляции и настройки прикладного программного обеспечения.	полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

Шкала оценивания результатов промежуточной аттестации и ел описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине, при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	**Описание**
Отлично	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации,** предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при переносе знаний и умений на новые, нестандартные задачи.
Хорошо	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации,** предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. В то же время при ответе допускает несущественные погрешности, задачи решает с недочетами, не влияющими на общий ход решения.
Удовлетворительно	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации,** предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. Но показывает неглубокие знания, при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами, в решении задач могут содержаться грубые ошибки. Для получения правильного ответа требуются уточняющие вопросы.
Неудовлетворительно	Не выполнены обязательные **условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины, ИЛИ студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями.

7.3 Оценочные средства

7.3.1 Текущий контроль

Защита лабораторных работ

7.3.2 Промежуточная аттестация

Примерные вопросы к экзамену

1. Перечислите основные элементы синтаксиса.

2. Дайте определение терминам инкремент/декремент. Уточните специфику применения и особенности в Python.

3. Назовите основной инструмент выбора в Python. Пример.

4. Дайте характеристику оператору elif. Пример.

5. Дайте характеристику оператору continue. Пример.

6. Дайте характеристику оператору break. Пример.

7. Назовите операторы цикла. Дайте характеристику оператору for. Пример.

8. Назовите операторы цикла. Дайте характеристику оператору while. Пример.

9. Назовите типы данных и дайте характеристику типу None.

10. Перечислите возможные виды массивов в Python. Дайте характеристику списку.

11. Перечислите возможные виды массивов в Python. Дайте характеристику кортежу.

12. Перечислите возможные виды массивов в Python. Дайте характеристику множеству.

13. Перечислите возможные виды массивов в Python. Дайте характеристику словарю.

14. Назовите исключение, от которого берут начало все остальные.

15. Что такое «менеджер контекста». В каких ситуациях применяется?

16. Дайте определение срезов. Алгоритм работы.

17. Назовите различие между инструкциями def и lambda. Использование return.

18. Перечислите способы создания словарей. Назовите метод, который возвращает значения в словаре.

19. Перечислите способы создания словарей. Назовите метод, который возвращает пары (ключ, значение).

20. Дайте определение термину Инкапсуляция. Пример.

21. Дайте определение термину Наследование. Пример.

22. Дайте определение термину Полиморфизм. Пример.

23. Назовите функцию ввода данных с клавиатуры.

24. При помощи какого метода возможно превратить введенный строку в список? Какой может быть разделитель?

25. Специфика лямбда-функций в Python их возможности и ограничения. Пример

26. Специфика создания объектов и присвоения в Python. Пример.

27. Базовые операции для работы с файлами в Python. Пример.

28. Использование инструкции with … as на примере работы с файлами. Пример.

29. Глобальные и локальные переменные в функциях на примере Python.

30. Кортежи в Python. Отличия кортежей от списков.

31. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков. Пример.

32. Итерирование по словарям, преобразование между словарями и списками в Python.

33. Модули. Инструкции import и from.

34. Назовите и дайте краткую характеристику основных классов языков программирования.

35. Анонимные функции. Типичное применение анонимных функций.

36. Функции и методы для работы со строками.

37. Назовите основные принципы ООП

38. Назовите особенности ООП в Python.

39. Чем являются строки в Python? Как это влияет на работу с текстовыми данными? Пример.

40. Назовите способы форматирования строк.