

Общество с ограниченной ответственностью «1Т»
(ООО «1Т»)

УТВЕРЖДАЮ
Генеральный директор

В.В. Кармаза

(подпись)

«20» мая 2025 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ
ПРОГРАММА – ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«Мастерство Python»**

Москва 2025

СОДЕРЖАНИЕ

1. ОБЩИЕ ХАРАКТЕРИСТИКА ПРОГРАММЫ	3
1.1. Общие положения	3
1.2. Цель освоения	5
1.3. Планируемые результаты обучения	5
1.4. Учебно-тематический план	7
1.5. Календарный учебный график	8
1.6. Рабочая программа	9
1.7. Организационно-педагогические условия	23
1.8. Формы аттестации	25
2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ	26
2.1. Текущий контроль	26
2.2. Итоговая аттестация	26

1. ОБЩИЕ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Общие положения

1.1.1. Нормативные правовые основания разработки программы

Нормативные правовые основания для разработки дополнительной профессиональной программы – программы повышения квалификации «Мастерство Python» (далее – программа) составляют:

– Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

– Приказ Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам» (зарегистрирован в Министерстве юстиции Российской Федерации 20 августа 2013 г., регистрационный № 29444).

– Письмо Минобрнауки России от 22 апреля 2015 года № ВК-1030/06 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями-разъяснениями по разработке дополнительных профессиональных программ на основе профессиональных стандартов»).

Программа разработана на основе профессиональных стандартов «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель) (приказ Министерства труда и социальной защиты Российской Федерации от 18 октября 2013 года N 544н), «Педагог профессионального обучения, среднего профессионального образования» (приказ Министерства труда и социальной защиты Российской Федерации от 21 марта 2025 года № 136н (вступит в силу с 01.09.2025 г)), «Педагог дополнительного образования детей и взрослых» (приказ Министерства труда и социальной защиты Российской Федерации от 22 сентября 2021 года N 652н).

Программа разработана на основе требований федерального

государственного образовательного стандарта среднего профессионального образования 44.02.03 Педагогика дополнительного образования (приказ Министерства образования и науки Российской Федерации от 13 августа 2014 г. № 998) к результатам освоения образовательных программ.

Программа разработана на основе требований федеральных государственных образовательных стандартов высшего образования 44.03.01 Педагогическое образование (приказ Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 121), 44.03.05 Педагогическое образование (с двумя профилями подготовки) (приказ Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 125) к результатам освоения образовательных программ.

Требования к слушателям:

а) категория слушателей: педагоги общеобразовательных, профессиональных и организаций дополнительного образования.

б) требования к уровню профессионального образования: лица, имеющие среднее профессиональное и (или) высшее образование, а также право заниматься педагогической деятельностью по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ по современным языкам программирования, технологиям искусственного интеллекта и робототехнике.

Особенности адаптации образовательной программы для лиц с ограниченными возможностями здоровья

Разработка адаптированной образовательной программы для лиц с ОВЗ и/или инвалидностью или обновление существующей образовательной программы определяются индивидуальной программой реабилитации инвалида (при наличии), рекомендациями заключения ПМПК (при наличии) и осуществляются по заявлению слушателя (законного представителя).

1.1.2. Форма обучения: заочная с применением дистанционных образовательных технологий.

1.1.3. Трудоемкость освоения: 72 академических часа.

1.1.4. Период освоения: 24 календарных дня.

1.1.5. Форма документа, выдаваемого по результатам освоения программы: лицам, успешно освоившим дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации установленного образца.

1.2. Цель освоения

Целью освоения программы являются получение новой компетенции, необходимой для реализации образовательной деятельности в области программирования на Python, повышение профессионального уровня в рамках имеющейся квалификации в области профессиональной деятельности.

1.3. Планируемые результаты обучения

Таблица 1 – Планируемые результаты обучения

Для слушателей из системы дополнительного образования

Код и наименование компетенции	Показатели освоения компетенции		
	Знания	Умения	Практический опыт (при наличии)
ОК 9. Осуществлять профессиональную деятельность в условиях обновления ее целей, содержания, смены технологий.	Приемы и средства организации деятельности обучающихся при освоении Python. История и особенности языка программирования Python. Области применения Python.	Осуществлять деятельность, связанную с преподаванием Python.	Навыки определения возможностей применения Python в системе дополнительного образования детей.
ПК 1.3. Демонстрировать владение деятельностью, соответствующей избранной области	Знание языка Python: Команды вывода данных. Базовые операции. Типы данных.	Решать практико-ориентированные и прикладные математические	Навыки работы в Python с числами, арифметическими операциями, типами данных, простыми преобразованиями.

Код и наименование компетенции	Показатели освоения компетенции		
	Знания	Умения	Практический опыт (при наличии)
дополнительного образования.	Базовый синтаксис и принципы работы условных операторов. Функции, их структура и назначение. Синтаксис циклов for и while. Объектно-ориентированное программирование.	задачи с помощью Python. Понимать базовые конструкции языка и уметь писать код.	Навыки создания списков, обработки, добавления и удаления элементов списков. Навыки работы со словарями. Навыки решения задач на использование условных операторов. Навыки работы с функциями. Навыки решения задач, связанных с объектно-ориентированным программированием.

Таблица 2 – Планируемые результаты обучения

Для слушателей из системы общего и среднего профессионального образования

Код и наименование компетенции	Показатели освоения компетенции		
	Знания	Умения	Практический опыт (при наличии)
ОПК-2. Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)	Приемы и средства организации деятельности обучающихся при освоении Python. История и особенности языка программирования Python. Области применения Python. Знание языка Python: Команды вывода данных. Базовые операции. Типы данных. Базовый синтаксис и принципы работы условных операторов.	Осуществлять деятельность, связанную с преподаванием Python. Решать практико-ориентированные и прикладные математические задачи с помощью Python. Понимать базовые конструкции языка и уметь писать код.	Навыки определения возможностей применения Python в системе основного и дополнительного образования. Навыки работы с числами, арифметическими операциями, типами данных, простыми преобразованиями. Навыки создания списков, обработки, добавления и удаления элементов списков. Навыки работы со словарями. Навыки решения задач на использование условных операторов.

Код и наименование компетенции	Показатели освоения компетенции		
	Знания	Умения	Практический опыт (при наличии)
	Функции, их структура и назначение. Синтаксис циклов for и while. Объектно-ориентированное программирование.		Навыки работы с функциями. Навыки решения задач, связанных с объектно-ориентированным программированием.

1.4. Учебно-тематический план

Таблица 2 – Учебно-тематический план

Наименование разделов (модулей), тем, видов аттестации	Трудоемкость, ак. час			СР	Форма аттестации
	Итого	Виды занятий, в т.ч.			
		Л	ПЗ, ЛР		
Раздел 1. Базовый блок	21	10	11	0	
Тема 1. Введение в курс	1	1	0	0	
Тема 2. Вывод данных и базовые операции	7	3	4	0	
Тема 3. Типы данных	13	6	7	0	
Раздел 2. Углубленный блок	50	26	24	0	
Тема 4. Условные операторы	12	6	6	0	
Тема 5. Встроенные функции Python	11	6	5	0	
Тема 6. Циклы for и while	10	5	5	0	
Тема 7. Функции	10	5	5	0	
Тема 8 Объектно-ориентированное программирование	7	4	3	0	
Итоговая аттестация	1	0	0	1	Тестирование
Всего ак. часов	72	36	35	1	

1.6. Рабочая программа

Макет рабочей программы представлен в таблице 4.

Таблица 4

Наименование тем	Виды учебных занятий, ак. час	Содержание	
Раздел 1. Базовый блок			
Тема 1. Введение в курс	Лекция	1	Знакомство с основами языка программирования Python, его историей, особенностями и философией. Обзор типов языков программирования, преимуществ Python. Широкое применение в разных областях: веб-разработка, анализ данных, машинное обучение и автоматизация. Обзор философии Python и стандартов написания кода по PEP 8 для создания чистого и читаемого кода.
Тема 2 Вывод данных и базовые операции			
Тема 2.1. Команда print ()	Лекция	0.5	Использование функции print() для вывода данных на экран. Работа с аргументами, синтаксисом строковых литералов и особенностями передачи нескольких аргументов. Правила написания комментариев в Python, их виды и назначение для документирования и пояснения кода.
	Практическое занятие	1	Решение практических задач на использование команды print()
Тема 2.2. Присваивание переменных в Python	Лекция	0,5	Переменные как имена для хранения данных в памяти и оператор присваивания "=". Правила именования переменных в стиле snake_case и использование множественного присваивания для упрощения кода. Рассмотрение примеров решения практических задач на создание и использование переменных с различными типами данных.
	Практическое занятие	1	Выполнение тестовых заданий с вопросами разного типа, в том числе на просмотр кода и выбор правильного варианта вывода.

Наименование тем	Виды учебных занятий, ак. час		Содержание
			Вопросы для закрепления навыка присваивания переменных в Python.
Тема 2.3. Разделители аргументов print()	Лекция	1	Необязательные параметры функции print(): sep и end. Использование специальных символов и строк в качестве разделителей для форматирования текста. Создание гибких вариантов вывода. Перенос строк, многострочные литералы. Рассмотрение примеров решения практических задач на применение параметров sep и end для настройки вывода данных.
	Практическое занятие	1	Решение практических задач на использование параметров функции print(). Тестовые задания на ревью кода и выбор правильного варианта результата работы кода.
Тема 2.4. Операции с числами	Лекция	1	Целочисленные данные как числовой тип, включающий положительные, отрицательные числа и ноль, и их отличие от строковых значений. Основные арифметические операторы (+, -, *, /, **) и порядок их выполнения, включая особенности ассоциативности и приоритетов. Использование сокращённых операторов и множественного присваивания для упрощения работы с целыми числами. Понятие f-строк, их использование. Рассмотрение примеров решения практических задач на выполнение арифметических операций и применение правил приоритета.
	Практическое занятие	1	Решение задач на использование арифметических операторов в работе с числовым и строковым типом данных. Решение практико-ориентированных и прикладных математических задач с помощью Python.
Тема 3. Типы данных			
Тема 3.1. Числовые типы данных	Лекция	1	Изучение целочисленного типа int для работы с целыми числами и типа float для чисел с плавающей точкой. Разбор примеров использования, арифметических операций и особенностей хранения

Наименование тем	Виды учебных занятий, ак. час		Содержание
			числовых данных в Python. Объяснение различий между типами и их применения в математических вычислениях.
	Практическое занятие	1	Решение типовых задач на применение базовых операторов и формул на отработку навыков работы с числами, арифметическими операциями, типами данных и простыми преобразованиями в программировании.
Тема 3.2. Строковые типы данных	Лекция	1	Изучение строкового типа "str" для работы с текстовыми данными, включая создание строк, использование кавычек и многострочные блоки. Разбор операций конкатенации (сложения строк) и умножения строк, а также преобразования чисел в строки с помощью "str()". Срезы строк для извлечения подстрок с указанием начального, конечного индексов и шага.
	Практическое занятие	1	Решение тренировочных задач на базовые операции со строками в Python для закрепления практических навыков программирования и понимание базовых конструкций языка.
Тема 3.3. Типы данных list и tuple	Лекция	1	Знакомство со списками, типами хранимых данных и операциями. Создание списков, свойства, обращение к отдельным элементам. Изучение ключевых операций со списками: индексация, срезы с шагом для выборки элементов и изменение содержимого списков. Освоение методов объединения списков и разложения элементов в отдельные переменные. Изучение особенностей кортежей (tuple). Создание кортежей, работа с элементами, основные отличия от списков. Разбор случаев применения кортежей в Python-программах. Основные операции. Рассмотрение примеров практических задач на применение этих операций для обработки данных.
	Практическое занятие	1	Решение тренировочных задач на базовые операции со списками и кортежами в Python для закрепления практических навыков программирования и понимание базовых конструкций языка.

Наименование тем	Виды учебных занятий, ак. час		Содержание
			Работа с созданием списков, срезами, добавление и удаление элементов списков.
Тема 3.4. Типы данных set и dict	Лекция	1	Изучение множеств (set) в Python: их основные свойства, создание и операции. Разбор особенностей работы с изменяемыми и неизменяемыми (frozenset) множествами, а также их практического применения для обработки уникальных данных. Практика работы с операциями над множествами: объединение (), пересечение (&), разность (-) и симметрическая разность (^). Изучение методов сравнения множеств и проверки вхождения элементов. Что такое словарь в Python. Основные понятия, создание и вывод на печать. Изучение структуры данных "словарь" (dict): работа с парами "ключ-значение". Освоение основных операций: получение значений по ключу, добавление и удаление элементов, изменение существующих значений.
	Практическое занятие	1	Решение тренировочных задач на базовые операции с множествами и словарями в Python для закрепления практических навыков программирования и понимание базовых конструкций языка. Практическая отработка операций над множествами. Задачи на создание, изменение и дополнение словарей в Python, а также на извлечение из них значения по ключу.
Тема 3.5. Логический тип данных	Лекция	0,5	Изучение булева типа данных (bool) в Python: значения True и False, их особенности и числовые аналоги. Освоение преобразований других типов данных к логическому типу с помощью функции bool() и сравнения списков на равенство.
	Практическое занятие	1	Решение прикладных задач на использование логических выражений.
Тема 3.6. Операторы Python	Лекция	1	Изучение основных логических операторов в Python: in, or, not. Практическое применение операторов для проверки сложных условий с несколькими критериями, комбинирования условий в

Наименование тем	Виды учебных занятий, ак. час		Содержание
			<p>управляющих конструкциях, инверсии логических значений. Примеры практических задач на применение логических операторов</p> <p>Изучение операторов <code>in</code> и <code>not in</code> для проверки вхождения элементов в последовательности (строки, списки и другие итерируемые объекты). Практическое применение операторов для поиска данных и валидации значений.</p>
	Практическое занятие	1	Решение тренировочных практико-ориентированных задач на базовые операции с логическими и операторами принадлежности.
Тема 3.7. Изменяемость типов данных	Лекция	0,5	<p>Углубленное знакомство с переменными и адресами в Python. Изучение различий между изменяемыми и неизменяемыми типами данных. Анализ поведения объектов, использование функции <code>id()</code> для определения изменяемости объекта. Практическое сравнение основных типов данных.</p>
	Практическое занятие	1	Выполнение практико-ориентированных заданий на закрепление знаний об изменяемости типов данных в Python.
Раздел 2. Углубленный блок			
Тема 4. Условные операторы			
Тема 4.1. Условный оператор	Лекция	1	<p>Изучение базового синтаксиса и принципов работы условных операторов в Python. Освоение конструкции <code>if-else</code> для управления потоком выполнения программы на основе условий. Синтаксис <code>if-else</code>. Применение операторов сравнения. Особенности работы с разными типами данных (числа, строки, коллекции).</p> <p>Практическое применение условных операторов для проверки значений переменных, определения четности чисел, сравнения списков и словарей. Обзор типичных ошибок.</p>
	Практическое занятие	1	Решение базовых задач на использование условных операторов (<code>if-else</code>), проверки условий и работы с пользовательским вводом.

Наименование тем	Виды учебных занятий, ак. час		Содержание
Тема 4.1. Операторы сравнения	Лекция	1	Изучение основных операторов сравнения (==, !=, <, >, <=, >=) и их применения для разных типов данных. Особенности сравнения чисел, строк, списков и других структур. Практическое использование операторов сравнения в условных выражениях.
	Практическое занятие	1	Решение базовых задач на использование операторов (сравнения, проверки условий и работы с пользовательским вводом).
Тема 4.3. if-elif-else	Лекция	1	Изучение промежуточного условия elif ("else if") для создания сложных ветвлений в программах. Особенности проверки нескольких условий в цепочке if-elif-else и практическое применение для разных типов данных. Изучение синтаксиса и логики работы. Рассмотрение практических задач с использованием конструкции elif с разными типами данных.
	Практическое занятие	1	Выполнение практико-ориентированных задач на определение категории по диапазону значений, на множественный выбор и классификацию входных данных с помощью конструкции "if-elif-else".
Тема 4.4. Цепочки сравнений	Лекция	1	Изучение специального синтаксиса Python для компактной записи сложных условий проверки диапазонов значений. Практическое применение цепочек сравнений для улучшения читаемости кода. Синтаксис цепочек сравнений, особенности работы, комбинирования с if-elif-else. Рассмотрение примеров решения практических задач с использованием цепочки сравнений.
	Практическое занятие	1	Выполнение практико-ориентированных задач на определение категории по диапазону значений и классификацию входных данных с помощью конструкции "if-elif-else". Закрепление навыка использования цепочки сравнений.
Тема 4.5. Логические операторы	Лекция	1	Изучение ключевых логических операторов Python (and, or, not) и их комбинаций для создания сложных условий. Практика использования цепочек сравнений для компактной проверки

Наименование тем	Виды учебных занятий, ак. час		Содержание
			диапазонов значений. Приоритет выполнения, возможности группировки условий с помощью скобок. Примеры применения сложных конструкций и цепочек сравнений с логическими операторами.
	Практическое занятие	1	Выполнение практико-ориентированных задач на проверку условий с использованием логических операторов ("and", "or", "not").
Тема 4.6. Try и except	Лекция	1	Изучение базового синтаксиса и принципов работы конструкции try-except в Python. Освоение механизма обработки исключений для предотвращения аварийного завершения программ. Синтаксис try-except. Применение для различных типов исключений (ZeroDivisionError, ValueError, IndexError). Особенности обработки ошибок при работе с разными типами данных. Практическое применение конструкции try-except для обработки ошибок ввода пользователя, математических операций и работы с коллекциями. Типичные ошибки при использовании блоков try-except и способы их предотвращения.
	Практическое занятие	1	Выполнение практико-ориентированных задачи на отработку навыков написания программ с использованием конструкция try-except.
Тема 5. Встроенные функции Python			
Тема 5.1. Что такое функция и как ее читать	Лекция	1	Понятие функции в Python, её структура и назначение. Использование встроенных функций print() и input() для вывода и ввода данных с аргументами. Основы работы с параметрами функций и примеры вызова с разными типами аргументов.
Тема 5.2. Встроенные функции	Лекция	1	Обзор популярных встроенных функций Python для работы с данными: print(), input(), type(), len(), sum(). Использование функций range() для создания числовых последовательностей, sorted() и reversed() для сортировки и реверса, а также max(), min(), abs() и round() для математических операций и округления. Практическое

Наименование тем	Виды учебных занятий, ак. час		Содержание
			применение встроенных функций для упрощения и оптимизации программного кода.
	Практическое занятие	1	Выполнение практико-ориентированных задач на обработку коллекций данных (списков, множеств), подсчёт количества элементов, вычисление суммы и среднего значения с использованием встроенных функций Python.
Тема 5.3. Методы строк	Лекция	1	Изучение методов строк. В том числе методов изменения регистра строк: <code>capitalize()</code> , <code>lower()</code> , <code>upper()</code> , <code>title()</code> , <code>swapcase()</code> . Знакомство с методами поиска и замены строк: <code>count()</code> , <code>replace()</code> , <code>strip()</code> , <code>lstrip()</code> , <code>rstrip()</code> , <code>startswith()</code> , <code>endswith()</code> , <code>find()</code> , <code>index()</code> . Обзор методов классификации символов, включая <code>isalpha()</code> , <code>isalnum()</code> , <code>isspace()</code> и другие. Примеры практического применения методов строк.
	Практическое занятие	1	Решение практико-ориентированных задач на обработку, преобразование, классификацию символов и анализ строк с помощью стандартных методов строковых объектов Python (<code>capitalize()</code> , <code>lower()</code> , <code>title()</code> , <code>count()</code> и др.
Тема 5.4. Методы списков	Лекция	1	Знакомство с методами списков. Методы изменения элементов списка: <code>append()</code> , <code>extend()</code> . Операция удаления элементов с помощью оператора <code>del</code> по индексу и срезам. Особенности применения метода <code>pop()</code> . Использование оператора <code>*</code> для распаковки списка при передаче элементов в функцию. Рассмотрение примеров практических задач с использованием методов списков.
	Практическое занятие	1	Решение практико-ориентированных задач на обработку, изменение элементов списка, удаление элементов. Отработка навыков использования методов <code>append()</code> , <code>extend()</code> , <code>del()</code> , <code>pop()</code> и др. для решения базовых задач.
Тема 5.5. Методы множеств	Лекция	0,5	Изучение методов добавления и удаления элементов во множество, операции объединения, пересечения и разности множеств, а также

Наименование тем	Виды учебных занятий, ак. час		Содержание
			сравнение множеств между собой. Освоение операций по проверке принадлежности элемента множеству. Рассмотрение практических примеров операций над множествами.
	Практическое занятие	0,5	Решение практико-ориентированных задач на базовые операции с множествами: использование методов add(), update(), remove(), discard(), pop(), clear(). Отработка практических навыков применения операций над множествами: union(), intersection(), difference()
Тема 5.6. Методы словарей	Лекция	0,5	Изучение основных методов словарей в Python: добавление элементов с помощью update(), возврат значений по ключу через get(), удаление элементов методами pop(), popitem() и clear(). Рассмотрение методов получения ключей, значений и пар ключ-значение (keys(), values(), items()) и использование setdefault() для получения или создания элементов словаря. Практическое применение методов словарей.
	Практическое занятие	0,5	Решение практико-ориентированных задач на работу со словарями: добавление, удаление, очистка, извлечение и обновление данных с использованием стандартных методов словарей Python (update(), pop(), clear(), values(), setdefault()) и др
Тема 5.7. Формат данных json	Лекция	1	Изучение структуры и синтаксиса формата данных JSON, его преимуществ и основных понятий объектов и массивов. Рассмотрение связи JSON со словарями и списками Python, а также освоение процессов сериализации и десериализации данных. Практические примеры работы с JSON в Python.
	Практическое занятие	1	Выполнение задач на сериализацию и десериализацию данных: преобразование между строками в формате JSON и объектами Python (словарями) с использованием функций json.loads() и json.dumps().
Тема 6. Циклы for и while			

Наименование тем	Виды учебных занятий, ак. час		Содержание
Тема 6.1. Цикл for	Лекция	2	Изучение синтаксиса цикла for, принципов итерации по различным типам данных и назначения переменной цикла. Освоение синтаксиса цикла for и практическое применение при работе со списками, кортежами, множествами, строками, словарями и диапазонами. Особенности создания циклов for с разными типами итерируемых данных: list, tuple, set, string, dict, range. Работа с последовательностями и индексация элементов. Рассмотрение практических примеров операций.
	Практическое занятие	2	Решение типовых базовых задач на использование цикла for с разными типами коллекций в Python.
Тема 6.2. Освоение List Comprehension в Python	Лекция	1	Введение в синтаксис и принципы работы List / Dictionary / Set Comprehension для создания списков, словарей и множеств на основе итерируемых объектов. Освоение генерации объектов с использованием выражений, циклов и условных операторов в одной строке кода. Освоение синтаксиса List Comprehension с условием if для фильтрации и преобразования элементов коллекций. Разбор практических заданий на создание компактных и эффективных коллекций с помощью comprehensions.
	Практическое занятие	1	Решение практико-ориентированных задач на создание, преобразование и фильтрацию списков с помощью конструкции list comprehension для компактной и эффективной обработки данных.
Тема 6.3. Цикл While	Лекция	1	Изучение конструкции цикла while, принципов его работы и синтаксиса. Освоение приёмов считывания данных до стоп-значения, создание и предотвращение бесконечных циклов. Обзор преимуществ и недостатков использования циклов for и while, факторы выбора того или другого цикла для решения практической задачи. Рассмотрение примеров практических заданий на применение цикла while для решения различных задач.

Наименование тем	Виды учебных занятий, ак. час		Содержание
	Практическое занятие	1	Решение типовых задач на использование цикла while.
Тема 6.4. Управляющие конструкции	Лекция	1	Управляющие конструкции continue и break для управления потоком выполнения циклов for и while. Освоение синтаксиса и принципов работы этих операторов для пропуска итераций и досрочного выхода из циклов. Рассмотрение практических примеров использования continue и break для гибкого контроля выполнения программных циклов.
	Практическое занятие	1	Решение типовых задач на использование операторов break и continue для управления выполнением циклов: досрочное завершение цикла при выполнении условия (break), пропуск отдельных итераций (continue), фильтрация и выборка элементов в процессе обхода коллекций или диапазонов.
Тема 7. Функции			
Тема 7.1. Функции без параметров	Лекция	1	Создание и использование функций без параметров в Python, их синтаксис и преимущества для модульности и повторного использования кода. Освоение применения ключевого слова pass для создания заглушек функций и планирования структуры программы. Разбор практических задач на написание простых функций без аргументов и использование pass в процессе разработки.
	Практическое занятие	1	Решение типовых задач на создание, вызов и применение пользовательских функций для автоматизации действий, обработки входных данных и вывода результатов. Закрепление базовых навыков работы с функциями: определение функций, передача и получение данных, использование встроенных операторов и циклов внутри функций.
Тема 7.2. Локальные и глобальные переменные	Лекция	1	Различия между локальными и глобальными переменными в Python, их область видимости и жизненный цикл. Освоение

Наименование тем	Виды учебных занятий, ак. час		Содержание
			<p>принципов использования локальных переменных внутри функций и глобальных переменных для доступа из любой части программы. Рассмотрение особенностей применения ключевого слова <code>global</code> для изменения глобальных переменных внутри функций и правил организации чистого и модульного кода. Изучение примеров использования переменных разных типов при решении задач.</p>
	Практическое занятие	1	<p>Выполнение заданий на закрепление знаний о типах переменных, их различии и способах использования.</p>
Тема 7.3. Функции с параметрами	Лекция	1	<p>Изучение функций с параметрами, принципов передачи данных в функцию и использования значений по умолчанию для параметров. Освоение синтаксиса и структуры функций с обязательными и необязательными аргументами. Рассмотрение практических заданий на написание и вызов функций с параметрами для решения различных задач.</p>
	Практическое занятие	1	<p>Решение практико-ориентированных задач на разработку пользовательских функций для проверки условий, сравнения значений, обработки строк и списков, а также для поиска, подсчёта и фильтрации данных с использованием базовых конструкций Python.</p>
Тема 7.4. Функции с возвратом значения	Лекция	1	<p>Функции с возвратом значения с помощью ключевого слова <code>return</code>, принципы формирования результата и передачи его вызывающему коду. Освоение синтаксиса <code>return</code> для завершения работы функции и возврата значения любого типа. Разборы практических заданий на написание функций, возвращающих вычисленные результаты для дальнейшего использования в программе.</p>
	Практическое занятие	1	<p>Решение практико-ориентированных задач на разработку пользовательских функций для преобразования и обработки строк, списков и словарей с использованием базовых алгоритмов Python.</p>

Наименование тем	Виды учебных занятий, ак. час		Содержание
Тема 7.5. Анонимная функция	Лекция	1	Изучение лямбда-функций — компактных анонимных функций для выполнения простых выражений. Освоение синтаксиса lambda, передачи аргументов и применения лямбда-функций в качестве однострочных выражений для кратковременных задач. Рассмотрение практических заданий на создание и использование лямбда-функций для вычислений и передачи их в другие функции
	Практическое занятие	1	Решение практико-ориентированных задач на определение и использование лямбда-функций для выполнения простых операций над числами и строками: вычисление произведения, определение длины строки, проверка чётности числа, определение палиндрома.
Тема 8. Объектно-ориентированное программирование			
Тема 8.1. Принципы ООП	Лекция	2	Различия между процедурным и объектно-ориентированным программированием, принципы построения программ с использованием функций и объектов. Освоение базовых понятий ООП: классы, объекты, атрибуты (поля) и методы. Изучение основных принципов объектно-ориентированного программирования: абстракции, инкапсуляции, наследования и полиморфизма. Освоение концепций выделения существенных характеристик объектов, сокрытия внутренней реализации, создания новых классов на основе существующих и использования единых интерфейсов для разных типов объектов. Рассмотрение практических примеров применения каждого принципа для построения гибких и расширяемых программ.
Тема 8.2. Классы	Лекция	2	Изучение основных понятий ООП: класс как шаблон для создания объектов, определяющий их атрибуты (характеристики) и методы (действия); объект (экземпляр класса) как конкретная сущность с индивидуальными значениями атрибутов и общим набором методов. Создание классов и объектов. Инициализация объектов. Обзор статических и динамических атрибутов класса. Методы

Наименование тем	Виды учебных занятий, ак. час		Содержание
			<p>класса: методы объекта класса, статические методы, магические методы. Создание методов класса. Освоение структуры классов и объектов на примере реальных предметов. Геттеры, сеттеры и декораторы. Рассмотрение практических примеров их реализации.</p>
	Практическое занятие	3	<p>Решение практико-ориентированных задач на определение пользовательских классов, создание экземпляров (объектов), работу с атрибутами и методами, а также различие между обычными, статическими и специальными методами класса. Задачи направлены на освоение основ объектно-ориентированного программирования в Python: определение классов с помощью ключевого слова class, использование конструктора init для инициализации атрибутов, реализация методов и создание объектов.</p>
Итоговая аттестация	Самостоятельная работа	1	Тестирование

1.7. Организационно-педагогические условия

Реализация программы осуществляется в соответствии с требованиями законодательства Российской Федерации в области образования, нормативными правовыми актами, регламентирующими данное направление деятельности.

1.7.1. Требования к квалификации педагогических кадров

Реализация программы обеспечивается педагогическими работниками, а также лицами, привлекаемыми к реализации программы на иных условиях, имеющими высшее или среднее профессиональное образование в области программирования на Python, опыт преподавания по тематике модуля.

1.7.2. Требования к материально-техническому обеспечению

Образовательная организация обеспечивает функционирование информационно-образовательной среды, включающей в себя электронные информационные ресурсы, электронные образовательные ресурсы, совокупность информационных технологий, телекоммуникационных технологий, соответствующих технологических средств и обеспечивающей освоение слушателями образовательных программ полностью или частично независимо от места нахождения слушателей: каналы связи, компьютерное оборудование, периферийное оборудование, программное обеспечение.

Код и наименование компетенции	Материально-техническое обеспечение, необходимое для освоения ПК
ОК 9. Осуществлять профессиональную деятельность в условиях обновления ее целей, содержания, смены технологий.	Платформа, на которой проводится курс https://sprint.it.ru Telegram-канал Группа для взаимодействия и поддержки слушателей в Telegram Персональный компьютер с процессором x86, x64 Минимальная диагональ монитора – не менее 39,6 см.
ПК 1.3. Демонстрировать владение деятельностью, соответствующей избранной области	ПРОЦЕССОР: 2 GHz DUAL CORE (Core 2 Duo 2.4 GHz or Athlon X2 2.7 GHz) Веб-камера. Микрофон, колонка (акустическая система) или наушники. Клавиатура. Мышь.

Код и наименование компетенции	Материально-техническое обеспечение, необходимое для освоения ПК
дополнительного образования. ОПК-2. Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)	Скорость интернет-соединения: минимальные требования к пропускной способности канала связи - от 128 кбит/сек исходящего потока и от 128 кбит/сек входящего потока. При этом для комфортной работы необходимо хотя бы 1 Мбит/с. ОС: Windows 7 (SP1+), Windows 10 and Windows 11. Интерпретатор python https://www.python.org/downloads/ Среда разработки Pycharm https://www.jetbrains.com/pycharm/download/#section=windows Библиотека PyQt https://pypi.org/search/?q=Pyqt Библиотека

1.7.3. Требования к информационному и учебно-методическому обеспечению

Для реализации программы используются учебно-методическая документация, учебная литература и иные издания, информационные ресурсы.

Таблица 4 – Учебно-методическая документация, нормативные правовые акты, нормативная техническая документация, иная документация, учебная литература и иные издания, информационные ресурсы¹

1. Основная литература
1.1. Бизли Д., Джонс Б. К. Python. Книга рецептов / пер. с англ. Б. В. Уварова. – М.: ДМК Пресс, 2019. – 648 с.
1.2. Доусон М. Програмируем на Python, 3 е издание – Пер. с англ. – СПб.: Питер, 2014. – 650 с.
1.3. Зингаро Д. Python без проблем: решаем реальные задачи и пишем полезный код. Издательство: Питер, 2023. – 336 с.
1.4. Лутц М. Изучаем Python, 3 е издание – Пер. с англ. – СПб.: Символ Плюс, 2009. – 848 с.,
1.5. Чернышев, С. А. Основы программирования на Python: учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва: Издательство Юрайт, 2023. — 286 с.
2. Дополнительная литература
2.1. Николенко С. И., Кадурич А. А., Архангельская Е. О. Глубокое обучение: погружение в мир нейронных сетей. Издательство: Питер, 2022. – 480 с.
2.2 Персиваль Г. Python. Разработка на основе тестирования / Г. Персиваль. – Москва: Издательство ДМК Пресс, 2018. – 622с.

¹ Оформление раздела должно соответствовать требованиям ГОСТ Р 7.0.100-2018. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

2.3. Рашид, Тарик. Создаем нейронную сеть.: Пер. с англ. — СПб.: ООО «Диалектика», 2019. — 272 с.
3. Интернет-ресурсы
3.1. https://www.tutorialspoint.com/python/ (сайт на английском языке для изучения Python);
3.2. https://pythonworld.ru/samouchitel-python (сайт по изучению основ программирования на Python);
3.3. https://pypi.org/search/?q=Pyqt (сайт с кратким описанием библиотек).
4. Электронно-библиотечная система
4.1 https://www.python.org/downloads/ (интерпретатор Python);
4.2. https://www.jetbrains.com/pycharm/download/#section=windows (среда разработки Pycharm);
4.3. https://pypi.org/search/?q=Pyqt (библиотека Pyqt).

1.7.4. Общие требования к организации учебного процесса

Общие требования к организации учебного процесса определяются локальными нормативными актами образовательной организации.

1.7.5. Сетевая форма обучения

Сетевая форма обучения не предусмотрена.

1.8. Формы аттестации

Оценка качества освоения программы осуществляется в форме текущего контроля успеваемости и итоговой аттестации слушателей.

1.8.1 Текущий контроль успеваемости

В соответствии с учебно-тематическим планом и рабочей программой.

1.8.2. Итоговая аттестация

Освоение программы завершается итоговой аттестацией. Итоговая аттестация проводится на основе принципов объективности и независимости оценки качества подготовки слушателей. Итоговая аттестация является обязательной для слушателей.

Итоговая аттестация проводится в форме тестирования.

К итоговой аттестации допускаются слушатели, не имеющие академической задолженности и в полном объеме выполнившие учебно-тематический план программы. Порядок прохождения итоговой аттестации определяется локальными нормативными образовательной организации.

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Оценочные материалы обеспечивают проверку достижения планируемых результатов обучения по программе и используются в процедуре итоговой аттестации.

2.1. Текущий контроль

Форма оценочного материала: практическое задание

Оценивание: оценка индивидуальных образовательных достижений по результатам текущего контроля производится в форме «зачет – незачет».

Критерии оценивания:

- «зачтено» - слушателем выполнено задание в полном объеме;
- «не зачтено» - в задании имеются ошибки, которые требуют доработки; после устранения слушателем замечаний ставится «зачтено»;
- «отсутствует» - слушатель не выполнил, либо не отправил задание на проверку.

2.2. Итоговая аттестация

Форма оценочного материала: тест

Тест состоит из 30 вопросов.

Критерии оценивания: за правильный ответ начисляется 1 балл, за неправильный – 0 баллов.

Оценивание: оценка индивидуальных образовательных достижений по результатам итоговой аттестации производится в форме «зачет – незачет».

Процедура оценивания. Тест считается пройденным в случае, если слушателем набрано не менее 60 процентов правильных ответов от общего количества вопросов. В случае, если слушателем набрано менее 60% правильных ответов от общего количества вопросов, тест считается не пройденным.

Примеры вопросов теста:

1. Как создать переменную «x» со значением 5?

А) `x == 5`

Б) `x = 5`

В) `x: = 5`

Г) `let x = 5`

Правильный ответ: Б.

2. Что выведет `print(3 + 4 * 2)`?

А) 14

Б) 11

В) 10

Г) Ошибку

Правильный ответ: Б.

3. Что сделает параметр `sep` при выполнении кода: `print("A", "B", sep="...")`?

А) `'sep'` заменит пробел на `'...'`

Б) `'sep'` добавит `'...'` в конец строки

В) `'sep'` не влияет на вывод

Г) `'sep'` заменяет запятую на `'...'`

Правильный ответ: А.

4. Какие из этих операций выполняются с числами? (Выберите несколько правильных ответов)

А) `'+'`

Б) `'*'`

В) `len()`

Г) `upper()`

Правильные ответы: А, Б.

5. Какие команды выводят «Hello World»? (Выберите несколько правильных ответов)

А) `print("Hello", "World")`

Б) `print ("Hello" + "World»)`

В) `print ("Hello World")`

Г) `print(Hello World)`

Правильные ответы: А, В.

6. Какой тип данных у 3.14?

А) `int`

Б) `float`

В) `str`

Г) `bool`

Правильный ответ: Б.

7. Как создать список [1, 2, 3]?

А) `(1, 2, 3)`

Б) `{1, 2, 3}`

В) `[1, 2, 3]`

Г) `"1, 2, 3"`

Правильный ответ: В.

8. Чем отличается tuple от list?

А) tuple изменяемый, list — нет

Б) tuple неизменяемый, list — изменяемый

В) tuple хранит только числа

Г) list нельзя индексировать

Правильный ответ: Б.

9. Как создать множество {1, 2, 3}? (Выберите несколько правильных ответов).

А) `set([1, 2, 3])`

Б) `{1, 2, 3}`

В) `(1, 2, 3)`

Г) `[1, 2, 3]`

Правильные ответы: А, Б.

10. Какой тип данных у {"name": "Alice"}?

А) `list`

Б) tuple

В) dict

Г) set

Правильный ответ: В.

11. Как записать условие "если x больше 10"?

А) if x > 10:

Б) if x < 10:

В) if x == 10:

Г) if x != 10:

Правильный ответ: А.

12. Какой оператор проверяет "не равно"?

А) ==

Б) !=

В) >=

Г) <=

Правильный ответ: Б.

13. Что выведет код:

```
python
```

```
x = 5
```

```
print ("Yes" if x > 3 else "No")
```

А) Yes

Б) No

В) True

Г) Ошибку

Правильный ответ: А.

14. Какой оператор проверяет, что x НЕ равен 5?

А) x != 5

Б) x <> 5

В) not x == 5

Г) Все варианты верны

Правильный ответ: Г.

15. Что делает `len("abc")`?

- А) Возвращает 3
- Б) Возвращает "abc"
- В) Преобразует в число
- Г) Ничего

Правильный ответ: А.

16. Как преобразовать строку "123" в число?

- А) `str(123)`
- Б) `int("123")`
- В) `float("123")`
- Г) `list("123")`

Правильный ответ: Б.

17. Какие функции преобразуют типы данных? (Выберите несколько правильных ответов).

- А) `int()`
- Б) `str()`
- В) `list()`
- Г) `print()`

Правильные ответы: А, Б, В.

18. Какой метод строки разбивает ее на список по разделителю?

- А). `split()`
- Б). `join()`
- В). `strip()`
- Г). `cut()`

Правильный ответ: А.

19. Как перебрать список [1, 2, 3] в цикле `for`?

- А) `for i in [1, 2, 3]:`
- Б) `for i = 1 to 3:`
- В) `while [1, 2, 3]:``

Г) `for (i=0; i<3; i++)`

Правильный ответ: А.

20. Что делает код: `lst =list(range(5))`?

А) Создает число 5

Б) Создает список [0, 1, 2, 3, 4]

В) Возвращает 5

Г) Ничего

Правильный ответ: Б.

21. Что мы получим в результате выполнения кода:

```
for i in range(2):  
    print(i, end=" ")
```

А) 0 1

Б) 1 2

В) 0 1 2

Г) 2

Правильный ответ: А.

22. Как остановить бесконечный цикл `while True`?

А) `stop`

Б) `break`

В) `exit`

Г) `return`

Правильный ответ: Б.

23. Как объявить функцию без параметров?

А) `def func():`

Б) `function func():`

В) `func():`

Г) `def func`

Правильный ответ: А.

24. Что выведет следующий код:

```
def func(a, b=2):
```

```
    return a * b
print(func(3))
```

- А) 6
- Б) 5
- В) Ошибку
- Г) 3

Правильный ответ: А.

25. Как передать в функцию аргументы по ключу?

- А) `func(a=1, b=2)`
- Б) `func(1, 2)`
- В) `func(*args)`
- Г) `func(a, b)`

Правильный ответ: А.

26. Какие утверждения о функциях верны? (Выберите несколько правильных ответов).

- А) Функции могут возвращать значения
- Б) Функции не могут принимать аргументы
- В) Локальные переменные видны вне функции
- Г) `return` завершает функцию

Правильные ответы: А, Г.

27. Как создать класс Car?

- А) `class Car:`
- Б) `def Car:`
- В) `Car class:`
- Г) `new Car:`

Правильный ответ: А.

28. Что такое `self` в методах класса?

- А) Имя класса
- Б) Ссылка на экземпляр класса
- В) Глобальная переменная

Г) Ключевое слово Python

Правильный ответ: Б.

29. Какие принципы ООП существуют? (Выберите несколько правильных ответов).

А) Наследование

Б) Инкапсуляция

В) Полиморфизм

Г) Рекурсия

Правильные ответы: А, Б, В.

30. Что такое наследование?

А) Копирование методов одного класса в другой

Б) Передача свойств родительского класса дочернему

В) Скрытие данных

Г) Полиморфизм

Правильный ответ: Б.