
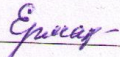





Муниципальное бюджетное общеобразовательное учреждение
Лицей «Держава»
города Обнинска

«Рассмотрено»
Руководитель методического
объединения учителей
математики
МБОУ «Лицей «ДЕРЖАВА» г.
Обнинск
 /Литвинчук С.Л. /
Протокол № 1
от «22» 08 2024г.

«Согласовано»
Заместитель директора по УВР
МБОУ «Лицей «ДЕРЖАВА»
г. Обнинск
 /Ермакова Н.В. /
«30» 08 2024г.

«Утверждаю»
Директор МБОУ «Лицей
«ДЕРЖАВА» г. Обнинска
 Копылова О.Н. /
Приказ № 001-08
от «30» 08 2024г.



ПРОГРАММА
курса «Изучение языка программирования «Python»
для 10-х классов.
(платные дополнительные образовательные услуги)
58 часов (2 часа в неделю)
на 2024-2025 учебный год

Составитель программы
Тякина Ольга Васильевна
учитель информатики и ИКТ

Обнинск 2024 г.

ПАСПОРТ ПРОГРАММЫ

Полное название программы	«Программирование на языке Python».
Автор программы, должность	Тякина Ольга Васильевна, учитель информатики
Адрес реализации программы <i>(адрес школы и телефон)</i>	город Обнинск, МБОУ «Лицей «Держава»
Вид программы	– по степени авторства – модифицированная; – по уровню освоения – общекультурная; – по уровню сложности – стартовый.
Направленность программы	техническая.
Вид деятельности	Информационная культура и информационные технологии.
Срок реализации	1 год, 63 часа.
Возраст учащихся	10 класс
Форма реализации программы	групповая.
Форма организации образовательной деятельности	курсы

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1.1. Пояснительная записка

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации до 2020 года и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей образовательной среды.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных учащихся. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

Актуальность программы состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Отличительные особенности программы:

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Целесообразность программы выражена в подборе интерактивных и практикоориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

Практическая значимость:

В рамках предлагаемого курса «Программирование на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизируют процесс индивидуально-личностного становления учащихся.

Адресат программы

Дополнительная общеразвивающая программа «Программирование на языке Python» рассчитана на обучающихся 9-х классов, проявляющих интерес к информационно-коммуникационным технологиям. Учащиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений законных представителей обучающихся. Группы формируются из школьников на добровольной внеконкурсной основе.

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Психолого-педагогические особенности учащихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

Особенности организации образовательного процесса

- Объем программы: 63 часа.
- Срок реализации программы: 1 год.
- Режим обучения: 1 раза в неделю по 90 минут.
- Форма обучения: очная.
- Особенности набора: свободный.
- Вид группы: среднешкольная.
- Состав группы: постоянный.

Методические принципы программы

- принцип развития: развитие индивидуальных способностей, общей культуры, навыков творческой продуктивной деятельности обучающихся;
- принцип демократизма: право каждого субъекта системы технического творчества на выбор своей траектории развития;
- принцип дифференциации и индивидуализации образования: выявление и развитие способностей обучающихся в области технического творчества, обеспечение их развития в соответствии с потенциалом, индивидуальными возможностями и интересами;
- принцип культуросообразности: ориентация на потребности общества и личности учащихся, единство человека и социокультурной среды, адаптация детей к современным условиям жизни общества;
- принцип единства учебного и воспитательного процесса;
- принцип систематичности и последовательности: вначале используется репродуктивный и интерактивный методы усвоения знаний с постепенным введением проблемного метода обучения, метода проекта.

Формы обучения и занятий

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их самостоятельно решению разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельности

ельностного подхода к обучению, в проблемном изложении материала педагогом, в переходе от продуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе – обучение учащихся навыкам самостоятельной творческой деятельности.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п.

Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

1.2. Цель и задачи программы

Цель: Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи

1. Образовательные:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

2. Воспитательные:

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;

3. Развивающие:

- развить познавательные процессы (внимание, восприятие, логическое мышление, память),
- развить креативность,
- развить способности к самореализации.

1.3. Содержание программы

Содержание учебного плана

Вводное занятие

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос.

Раздел 1. Знакомство с языком Python

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа. Установка программы Python Практическая работа. Режимы работы с Python Тест Знакомство с языком Python

Учащиеся должны знать/ понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами Тест.

Выражения и операции.

Учащиеся должны знать/ понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа. Логические выражения Практическая работа. "Условный оператор" Практическая работа Множественное ветвление Практическая работа. "Условные операторы"

Самостоятельная работа. Решение задач по теме "Условные операторы". "Составление программ с ветвлением".

Тест. "Условные операторы".

Учащиеся должны знать/ понимать:

- назначение условного оператора;

- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange Функция random. Примеры решения задач с циклом.

Практическая работа. "Числа Фибоначчи" Практическая работа. Решение задачи с циклом for.

Практическая работа . Реализация циклических алгоритмов Практическая работа. Случайные числа

Практическая работа. Решение задач с циклом. Самостоятельная работа "Составление программ с циклом"

Тест №4. Циклы

Творческая работа № 1. "Циклы"

Учащиеся должны знать/ понимать:

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа. Создание функций

Практическая работа Локальные переменные

Практическая работа Решение задач с использованием функций

Практическая работа Рекурсивные функции

Самостоятельная работа по теме "Функции" Тест №5. Функции

Учащиеся должны знать/ понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;

- способ передачи параметров.
- Учащиеся должны уметь:*
- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки- последовательности символов

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор `in`. Модуль `string`. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа Строки

Практическая работа Решение задач со строками.

Учащиеся должны знать/ понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Раздел 7. Сложные типы данных

Теория: Списки. Тип список (`list`). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция `range`. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (`dict`). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа Списки.

Практическая работа Решение задач со списками. Тест Списки

Учащиеся должны знать/ понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;

- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

1.4. Планируемые результаты освоения программы

Планируемые результаты курса

В рамках курса «Программирование на языке Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).
- Проверка достигаемых учащимися результатов производится в следующих формах:
- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка педагогом деятельности учащихся;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;
- Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1. Календарный учебный график

Работа по программе осуществляется с 01 октября по 28 мая. Каникулы не предусмотрены (см. приложение 1. «Рабочая программа»).

2.2. Условия реализации программы

Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;

- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

Программное обеспечение:

- Операционная система Windows 10 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JupyterLab/PyCharm;
- Foxit Reader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

–

2.3. Формы аттестации (контроля)

Входящий контроль осуществляется при комплектовании группы в начале учебного года. *Цель* – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточный контроль осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

Итоговый контроль осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

2.4. Методическое обеспечение

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2020 года.

Методы обучения:

1. **Словесный:** объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.
2. **Наглядный:** применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. **Практический:** индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. **Интерактивный:** создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Технологии:

1. **Технология проблемного диалога.** Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.
2. **Технология коллективного взаимообучения** («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.
3. **Игровая технология.** Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.
4. **Элементы здоровьесберегающих технологий** являются необходимым условием снижения утомляемости и перегрузки учащихся.
5. **Проектная технология** предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.
6. **Информационно-коммуникационные технологии** активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

Учебно-методическое обеспечение курса:

1. Марк Лутц – Изучаем Python и Программируем на Python
2. Форсье Дж., Биссекс П., Чан У., Django – веб-разработка приложений на Python
3. Вестра Э. — Разработка геоприложений на языке Python.
4. Андреас Мюллер, Сара Гвидо. Введение в машинное обучение с помощью Python.
5. Ян Эрик Содем Программирование компьютерного зрения на языке Python

Тематическое планирование по информатике на 2024/2025 уч.год

№ занятия	Наименование темы	Кол-во часов	Дата проведения
1	Множественное ветвление. Алгоритмы ветвления Реализация ветвления в языке Python.	1	02.10.24
2	Множественное ветвление. Реализация ветвления в языке Python.	1	02.10.24
3	Понятие цикла.Тело цикла.Условия выполнения тела цикла. Оператор цикла с условием.	1	09.10.24
4	Оператор цикла while. Бесконечные циклы.	1	09.10.24
5	Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.	1	16.10.24
6	Практическая работа . Реализация циклических алгоритмов	1	16.10.24
7	Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for.	1	24.10.24
8	Практическая работа. "Числа Фибоначчи" Практическая работа. Решение задачи с циклом for.	1	24.10.24
9	Вложенные циклы. Циклы в циклах.	1	06.11.24
10	Случайные числа. Функция randrange Функция random. Примеры решения задач с циклом.	1	06.11.24
11	Практическая работа. Решение задач с циклом. Самостоятельная работа "Составление программ с циклом"	1	13.11.24
12	Самостоятельная работа "Составление программ с циклом"	1	13.11.24
13	Самостоятельная работа "Составление программ с циклом"	1	20.11.24
14	Самостоятельная работа "Составление программ с циклом"	1	20.11.24
15	Самостоятельная работа "Составление программ с циклом"	1	27.11.24
16	Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения.	1	27.11.24
17	Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.	1	04.12.24
18	Практическая работа Решение задач с использованием функций	1	04.12.24
19	Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.	1	11.12.24
20	Практическая работа Рекурсивные функции	1	11.12.24

21	Практическая работа Рекурсивные функции	1	18.12.24
22	Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов.	1	18.12.24
23	Практическая работа Строки	1	25.12.24
24	Практическая работа Строки	1	25.12.24
25	Применение цикла для обхода строки.	1	15.01.25
26	Практическая работа Решение задач со строками.	1	15.01.25
27	Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string.	1	22.01.25
28	Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.	1	22.01.25
29	Практическая работа Решение задач со строками.	1	29.01.25
30	Практическая работа Решение задач со строками.	1	29.01.25
31	Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков.	1	05.02.25
32	Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков.	1	05.02.25
33	Практическая работа Списки.	1	12.02.25
34	Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.	1	12.02.25
35	Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.	1	19.02.25
36	Практическая работа Решение задач со списками.	1	19.02.25
37	Практическая работа Решение задач со списками.	1	26.02.25
38	Практическая работа Решение задач со списками.	1	26.02.25
39	Матрицы. Вложенные списки.	1	05.03.25
40	Практическая работа Решение задач со списками.	1	05.03.25
41	. Матрицы. Строки и списки. Генераторы списков в Python.	1	12.03.25
42	Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения	1	12.03.25
43	Практическая работа Кортежи.	1	19.03.25
44	Введение в словари. Тип словарь(dict). Словарные операции. Словарные методы.	1	19.03.25

45	Практическая работа Словарные операции.	1	02.04.25
46	Практическая работа Словарные методы.	1	02.04.25
47	Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.	1	09.04.25
48	Практическая работа Операции, допустимые над множествами	1	09.04.25
49	Практическая работа Операции, допустимые над множествами	1	16.04.25
50	Исключения в python. Конструкция try - except для обработки исключений	1	16.04.25
51	Иерархия встроенных в python исключений	1	23.04.25
52-53	Файлы. Работа с файлами.	2	23.04.25 30.04.25
54-55	With ... as - менеджеры контекста	2	30.04.25 07.05.25
56	Обработка ошибок python	1	07.05.25
57	Классы и объекты	1	14.05.25
58-59	Объектно-ориентированное программирование python	2	14.05.25 21.05.25
60-61	Инкапсуляция: скрытие информации	2	21.05.25
62	Наследование: поведение и характеристики	2	28.05.25
63	Итоговое тестирование	1	28.05.25