

Комитет по образованию администрации
Всеволожского муниципального района
Ленинградской области
Муниципальное общеобразовательное бюджетное учреждение
«Средняя общеобразовательная школа
«Центр образования «Кудрово»

РАССМОТРЕНО
на Педагогическом совете МОБУ «СОШ
«ЦО «Кудрово»
Протокол №34 от 25.08.2025г.

УТВЕРЖДЕНО
Директор МОБУ «СОШ «ЦО
«Кудрово»
И.Ю.Соловьев
Приказ от 25.08.2025 г. № 581



Дополнительная общеобразовательная общеразвивающая программа
Язык программирования Python
Техническая направленность

Педагог дополнительного
образования
Корягин Сергей Андреевич
Уровень: базовый
Продолжительность освоения
программы - 2 года
Возраст учащихся - 13-17 лет

г. Кудрово
Ленинградская область
2025 год

Содержание

Пояснительная записка.....	3
Учебно-тематический план	14
Содержание программы... ..	16
Контрольно-оценочные средства.....	19
Условия реализации программы.....	21
Список литературы.....	22
Приложение	24

Пояснительная записка

Общеразвивающая программа дополнительного образования «Язык программирования Python» (далее - Программа) разработана в соответствии с:

- Конвенцией о правах ребенка
- Конституцией Российской Федерации
- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»
- Постановлением Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодёжи»
- Приказом Министерства образования и науки Российской Федерации от 31 мая 2021 г. № 286 «Об утверждении федерального государственного образовательного стандарта начального общего образования»
- Приказом Минобрнауки России от 17.05.2012 № 413 (ред. от 12.08.2022) «Об утверждении федерального государственного образовательного стандарта среднего общего образования» (Зарегистрировано в Минюсте России 07.06.2012 № 24480)
- Приказом Министерства образования и науки Российской Федерации от 31 мая 2021 г. № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования»
- Приказом Министерства просвещения Российской Федерации от 03 сентября 2019 г. № 467 «Об утверждении Целевой модели развития региональных систем развития дополнительного образования детей»

- Приказом Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- Распоряжением Правительства РФ от 31.03.2022 № 678-р «Об утверждении Концепции развития дополнительного образования детей и признании утратившим силу Распоряжения Правительства РФ от 04.09.2014 N 1726-р» (вместе с «Концепцией развития дополнительного образования детей до 2030 года»)
- Распоряжением Правительства Российской Федерации от 29 мая 2015 г. № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года».

Программа направлена на обучение учащихся школьного возраста (7-11 классы) основам программирования на языке Python с учетом современных требований к цифровым компетенциям. Программа сочетает теоретическое изучение фундаментальных концепций программирования с практическим освоением технологий разработки на Python.

В процессе освоения программы учащиеся получат комплексные знания о синтаксисе Python, принципах структурного и объектно-ориентированного программирования, работе с основными типами данных и структурами. Особое внимание уделяется развитию навыков решения практических задач с использованием стандартной библиотеки Python и популярных сторонних модулей.

Программа включает изучение ключевых аспектов применения Python в современных ИТ-сферах: веб-разработке, анализе данных, автоматизации процессов и создании скриптов. Учащиеся освоят принципы работы с файлами, базами данных, сетевыми протоколами, а также научатся разрабатывать полноценные приложения с графическим интерфейсом.

Программа разработана в соответствии с актуальными образовательными стандартами и ориентирована на подготовку специалистов, способных эффективно использовать Python для решения широкого круга задач. Выпускники программы приобретут навыки, необходимые для дальнейшего профессионального роста в области программирования и информационных технологий.

Программа предназначена для учащихся 7-11 классов МОБУ «СОШ «Центр образования «Кудрово», проявляющих интерес к углубленному изучению современных информационных технологий и программирования. Особый акцент сделан на формировании практических навыков, востребованных в профессиональной ИТ-среде.

Актуальность программы, педагогическая целесообразность

Современные образовательные учреждения играют ключевую роль в подготовке учащихся к профессиональной деятельности, особенно в столь динамичной сфере, как информационные технологии. В условиях стремительного развития цифровой экономики особую значимость приобретает формирование у школьников не только технических навыков программирования, но и soft skills, необходимых для успешной работы в ИТ-индустрии.

Программа изучения языка Python призвана создать условия для развития компетентных специалистов, способных применять современные технологии программирования для решения практических задач. Python, как один из наиболее востребованных языков, открывает перед учащимися широкие возможности в различных областях — от веб-разработки и анализа данных до искусственного интеллекта и автоматизации процессов.

Важной особенностью программы является сочетание глубокого изучения технических аспектов программирования с развитием навыков командной работы, проектного мышления и способности к постоянному профессиональному росту. Практико-ориентированный подход позволяет учащимся не только освоить синтаксис и парадигмы программирования, но и научиться применять полученные знания для создания реальных ИТ-решений.

В процессе обучения особое внимание уделяется формированию у школьников способности адаптироваться к быстро меняющимся технологическим реалиям, самостоятельно осваивать новые инструменты и frameworks, а также участвовать в разработке проектов, направленных на решение актуальных социальных и экономических задач.

Программа «Язык программирования Python» в МОБУ «СОШ «Центр образования «Кудрово» представляет собой комплексную систему подготовки, которая интегрирует фундаментальные знания в области программирования с практическими навыками их применения, что позволяет выпускникам становиться востребованными специалистами в современной цифровой среде.

Цель - развитие творческого потенциала личности обучающегося, формирование у него системы знаний, умений и навыков в области системного администрирования, необходимых для дальнейшего обучения в области ИТ.

Для достижения данной цели необходимо решить следующий **комплекс задач**.

Обучающеее

- Способствовать овладению базовым синтаксисом и конструкцией языка Python

- Обучить принципам объектно-ориентированного программирования
- Формировать навыки работы с основными структурами данных
- Научить понимать алгоритмы обработки и анализа информации
- Научить методам работы с файлами и внешними источниками данных
- Способствовать овладению основами тестирования и отладки программного кода
- Формировать профессиональные коммуникации в ИТ-среде

Развивающие

- Развивать алгоритмическое и логическое мышление
- Совершенствовать навыки анализа и оптимизации кода
- Формировать системный подход к решению задач
- Развивать креативность при проектировании программ
- Совершенствовать навыки самостоятельного обучения
- Развивать презентационные навыки при защите проектов
- Прививать культуру оформления документации и комментирования кода
-

Воспитательные

- Формировать ответственное отношение к написанию кода
- Развивать навыки командной работы над проектами
- Воспитывать настойчивость при решении сложных задач
- Формировать этические нормы использования программного обеспечения

Организационно-педагогические условия

Срок реализации – 2 года.

Возраст учащихся

Программа предназначена для работы с обучающимися 13 -17 лет.

Режим занятий

Занятия проводятся 1 раз в неделю по 2 часа, 72 часа в год, 144 за два года обучения. Продолжительность занятия 40 минут.

*Календарный учебный график
на 2025 – 2026 учебный год*

<i>Этапы образовательного процесса</i>	<i>1 год обучения</i>	<i>2 год обучения</i>

Этапы образовательного процесса	1 год обучения	2 год обучения
<i>Продолжительность занятия</i>	<i>7-8 лет: до 40 минут (физкультурно-спортивная и техническая направленности) 9-18 лет: до 80 минут</i>	<i>7-8 лет: до 40 минут (физкультурно-спортивная и техническая направленности) 9-18 лет: до 80 минут</i>
<i>Промежуточная аттестация</i>	<i>2-26 декабря</i>	<i>21-26 декабря</i>
<i>Аттестация по завершении реализации программы</i>	<i>23-31 мая</i>	<i>23-31 мая</i>
<i>Окончание учебного года</i>	<i>31 мая</i>	
<i>Зимние каникулы</i>	<i>31.12.2025 – 11.01.2026, 12 дней</i>	
<i>Летние каникулы</i>	<i>31.05.2026 – 31.08.2026</i>	
<i>Праздничные дни</i>	<i>04 ноября; 23 февраля ; 8 марта; 1 мая; 9 мая</i>	

Форма обучения Очная.

Форма организации образовательной деятельности учащихся

Групповая, индивидуальная, всем составом. Наполняемость групп до 35 человек.

Форма занятия

- Теоретические занятия (лекции, объяснение нового материала)
- Практические занятия (решение задач, написание кода, работа над проектами)
- Дискуссии и обсуждения (разбор различных подходов к решению задач)
- Викторины и конкурсы по программированию
- Встречи с ИТ-специалистами (гостевые лекции, мастер-классы)
- Хакатоны и соревнования по программированию

Результаты освоения программы:

- Выполненные практические работы и проекты
- Презентации и защиты разработанных программных продуктов
- Участие в школьных и городских ИТ-мероприятиях
- Создание портфолио программных разработок

Данная дополнительная общеразвивающая программа **построена на основе:**

- педагогической теории «Педагогика сотрудничества», авторами которой являются педагоги – С.Т. Шацкий, В.А. Сухомлинский, А.С. Макаренко, К.Д. Ушинский, Н.И. Пирогов, Л.Н. Толстой, Ж.Ж. Руссо, Я. Корчак;

- современных принципов воспитания;
- учета индивидуальных особенностей формирования классного коллектива;
- учета возрастных особенностей учащихся класса;
- учета особенностей контингента родителей учащихся класса.

Идеи программы

Идея индивидуальности – программа учитывает уникальные особенности каждого учащегося, предлагая дифференцированные задания и индивидуальные образовательные траектории. При изучении Python учащиеся развиваются собственный стиль программирования, сохраняя при этом понимание общих принципов разработки. Педагог помогает каждому ученику осознать свои сильные стороны и области для роста, формируя персональную стратегию освоения материала.

Идея развития и самоактуализации личности – обучение Python становится инструментом профессионального и личностного роста. Учащиеся последовательно переходят от простых скриптов к сложным проектам, раскрывая свой потенциал. Программа создает условия для осознанного стремления к мастерству, где каждый ученик может определить интересующие его направления (веб-разработка, анализ данных, автоматизация) и углубленно их изучать.

Идея самовыражения и творчества – в процессе работы с информационными системами ученик развивает свои творческие способности, учится нестандартно подходить к решению задач. Через творчество он находит свои пути самовыражения, будь то настройка сети, администрирование серверов или создание собственных проектов, что способствует дальнейшему раскрытию его личных возможностей.

Идея выбора и ответственности – программа "Язык программирования Python" формирует у учащихся осознанный подход к профессиональной деятельности через развитие навыков самостоятельного принятия решений. Учащиеся получают возможность выбирать направления для углубленного изучения в зависимости от своих интересов – будь то веб-разработка, анализ данных или создание игр. Такой подход учит их анализировать последствия своих решений и нести ответственность за качество создаваемого программного кода.

Идея сотрудничества и педагогической поддержки – обучение строится на принципах коллективного взаимодействия, где каждый участник вносит вклад в общий результат. Педагоги создают среду, в которой учащиеся помогают друг другу осваивать материал – через парное программирование, командные проекты и взаимное рецензирование кода. Родители и профессиональное сообщество вовлекаются в образовательный процесс через презентации проектов и встречи с IT-специалистами, что обеспечивает комплексную поддержку на всех этапах обучения.

Принципы реализации программы

Принцип личностно-ориентированного воспитания – программа ориентирована на развитие индивидуальности каждого обучающегося, предоставляя свободу в выборе средств и путей освоения программирования. В процессе обучения языку Python учащиеся получают возможность двигаться в комфортном для себя темпе, выбирать интересующие направления (веб-разработка, анализ данных, автоматизация и т. д.) и самостоятельно определять способы решения задач. Педагогическая поддержка направлена на развитие навыков самоанализа, рефлексии и стремления к постоянному профессиональному росту.

Принцип гуманистической направленности – программа строится на основе уважения, доверия и взаимопонимания между преподавателями и учащимися. Это особенно важно при обучении Python, где учащиеся часто работают в парах или группах над проектами. Поддержка и уважение в команде способствуют не только комфортной атмосфере, но и эффективному обмену знаниями и совместному преодолению трудностей при разработке программных решений.

Принцип природообразности – материал и методы преподавания подбираются с учетом возрастных, психологических и физиологических особенностей обучающихся. Обучение Python проходит поэтапно: от простого к сложному, обеспечивая логичность, доступность и последовательность подачи материала. Это позволяет каждому учащемуся усваивать знания в соответствии со своими способностями и личным темпом, создавая комфортные условия для освоения языка программирования.

Принцип социального взаимодействия – особое внимание уделяется формированию среды для активного общения, сотрудничества и обмена опытом между всеми участниками образовательного процесса. В рамках изучения Python реализуются групповые проекты, совместные исследования и хакатоны, в которых учащиеся учатся работать в команде, решать практические задачи, обсуждать идеи и находить компромиссы. Такое взаимодействие развивает навыки социальной адаптации и профессионального самоопределения.

Принцип компетентностного подхода – обучение Python строится с акцентом на формирование практических умений и навыков, необходимых для реальной профессиональной деятельности в сфере ИТ. В ходе обучения учащиеся осваивают базовые и продвинутые конструкции языка, учатся писать читаемый, эффективный код, решать прикладные задачи, разрабатывать собственные проекты. Компетентностный подход позволяет формировать такие важные качества, как критическое мышление, способность к самостоятельному поиску решений и адаптация к меняющимся условиям в профессиональной среде.

Компетентностный подход обеспечивает направленность образовательного процесса на подготовку учащихся к решению реальных задач, характерных для сферы программирования. Освоив Python,

учащиеся смогут уверенно работать с алгоритмами, структурами данных, библиотеками и фреймворками, что позволит им успешно справляться с профессиональными вызовами и строить карьеру в ИТ-индустрии.

Средства и методы реализации программы

1) Аналитико-диагностическая деятельность:

- наблюдение;
- тестирование;
- соцопросы;
- анкетирование;
- консультирование;
- анализ.

2) Информационно-организационная деятельность

- конференции;
- диспуты;
- выпуск плакатов, коллажей;
- консультации специалистов;
- ролевые игры;
- просмотр и обсуждение фильма;
- проектная деятельность;
- родительские лектории.

Планируемые результаты 1 года обучения

Освоение содержания дополнительной обще развивающей общеобразовательной программы обеспечивает достижение следующих результатов:

Предметные

Обучающийся научится:

- Контролировать версионность программного обеспечения и адаптироваться к изменениям в используемых библиотеках и средах разработки, таких как работа с основными типами данных, операциями над строками, списками и кортежами.
- Принимать обоснованные решения при проектировании архитектуры программного продукта, выборе алгоритмов и структур данных.
- Брать ответственность за работоспособность и устойчивость разработанных приложений, включая выполнение практических заданий и мини-проектов.
- Осознавать этические аспекты программирования, включая честность при использовании чужого кода, защиту персональных данных, соблюдение авторских прав и безопасность пользователей.

Метапредметные:

Обучающийся научится:

- Формулировать и планировать задачи в процессе программирования, от разработки алгоритма до его реализации на языке Python.
- Разбивать задачи на этапы (псевдокод, написание, отладка, тестирование), используя принципы логики и структурного подхода.
- Применять критерии качества программного кода: читаемость, эффективность, модульность, повторное использование и тестируемость.
- Работать в команде над проектами, используя системы контроля версий, такие как Git.
- Анализировать ошибки и находить причины сбоев в работе программ, проводить отладку, оптимизацию и улучшение кода.
- Овладевать основами цифровой безопасности, включая защиту данных, предотвращение уязвимостей и безопасную обработку пользовательского ввода.
- Эффективно коммуницировать с одноклассниками, преподавателями и потенциальными пользователями при обсуждении программных решений, исправлении ошибок и внедрении новых функций.

Личностные Обучающийся

научится:

- Проявлять устойчивую мотивацию к изучению более сложных аспектов программирования, осознавая значимость Python для решения задач в таких областях, как веб-разработка, анализ данных и работа с API.
- Самостоятельно определять цели своего развития в области программирования, планировать освоение новых технологий, таких как Flask, базы данных и библиотеки для работы с данными.
- Развивать навыки самообучения и стремиться к постоянному совершенствованию программных решений, включая работу с инструментами для тестирования и отладки.

Планируемые результаты 2 года обучения

Освоение содержания дополнительной обще развивающей общеобразовательной программы обеспечивает достижение следующих результатов:

Предметные

Обучающийся научится:

- Контролировать версионность программного обеспечения и адаптироваться к изменениям в используемых библиотеках и средах разработки, таких как Flask, NumPy, Pandas и библиотеки для работы с базами данных.
- Принимать обоснованные решения при проектировании архитектуры более сложных программных продуктов, выборе алгоритмов, структур данных и способах хранения информации.
- Брать ответственность за работоспособность и устойчивость разработанных приложений, включая работу с веб-технологиями, базами данных и API.

Метапредметные:

- Обучающийся научится:
- Формулировать и планировать задачи для создания более сложных программных решений, включая работу с API, создание веб-приложений и обработку данных.
- Разбивать задачи на этапы (псевдокод, написание, отладка, тестирование), применяя методы работы с внешними библиотеками, такими как Flask, NumPy, Pandas.
- Применять критерии качества программного кода для создания масштабируемых и надежных программных решений.
- Работать в команде, используя системы совместной разработки (например, Git), планировать задачи, управлять версиями и распределять роли.
- Анализировать ошибки и находить причины сбоев в работе программ, проводить отладку и оптимизацию кода.
- Овладевать более сложными принципами цифровой безопасности, включая защиту данных при взаимодействии с внешними сервисами и базами данных.

- Осознавать этические аспекты программирования, включая защиту данных при работе с внешними сервисами и обеспечение безопасности пользователей.

Личностные

Обучающийся научится:

- Проявлять устойчивую мотивацию к изучению более сложных аспектов программирования, осознавая значимость Python для решения задач в таких областях, как веб-разработка, анализ данных и работа с API.
- Самостоятельно определять цели своего развития в области программирования, планировать освоение новых технологий, таких как Flask, базы данных и библиотеки для работы с данными.
- Развивать навыки самообучения и стремиться к постоянному совершенствованию программных решений, включая работу с инструментами для тестирования и отладки.

Учебно – тематический план. Первый год обучения

№ раздела	Основные направления	Количество часов			Формы аттестации/контроля
		Теория	Практика	Всего	
1	Введение в программирование. История и области применения Python	2	4	6	Тестирование, анкетирование
2	Переменные, типы данных, ввод и вывод	2	4	6	Тестирование, практическая работа
3	Условные операторы и логика выполнения	2	4	6	Тестирование
4	Циклы и итерации	2	4	6	Практическая работа
5	Функции и работа с аргументами	2	4	6	Мини-проект
6	Строки и операции над ними	2	4	6	Тестирование
7	Списки, кортежи, множества	2	4	6	Тестирование
8	Словари и работа с коллекциями данных	2	4	6	Практическое задание
9	Работа с файлами (чтение/запись)	2	4	6	Проект
10	Исключения и обработка ошибок	2	4	6	Тестирование
11	Модули и импорт внешних библиотек	2	4	6	Практическое задание

12	Итоговый проект: создание простого приложения на Python	2	4	6	Защита проекта
	Итого	24	48	72	

Учебно – тематический план. Второй год обучения

№ раздела	Основные направления	Количество часов			Формы аттестации/контроля
		Теория	Практика	Всего	
1	Работа с файлами и каталогами, расширенное взаимодействие с ОС	2	4	6	Практическая работа
2	Структуры данных и алгоритмы	2	4	6	Тестирование
3	Объектно-ориентированное программирование (ООП)	2	4	6	Проект
4	Работа с библиотеками NumPy и Pandas	2	4	6	Практическое задание
5	Визуализация данных с помощью Matplotlib/Seaborn	2	4	6	Мини-проект
6	Работа с API, форматом JSON и HTTP-запросами	2	4	6	Тестирование
7	Веб-программирование с использованием Flask	2	4	6	Проект

8	Работа с базами данных и SQLite	2	4	6	Практическое задание
9	Тестирование и отладка кода	2	4	6	Тестирование
10	Работа с виртуальным и средами и системами контроля версий (Git)	2	4	6	Практическая работа
11	Основы автоматизации и и написание скриптов	2	4	6	Практическое задание
12	Итоговый проект: создание приложения или аналитической системы	2	4	6	Защита проекта
	Итого	24	48	72	

Содержание программы

Первый год обучения:

Раздел 1. Введение в программирование на Python – 6 ч

Теория: Введение в язык Python, области его применения, установка и настройка окружения разработки – 2 ч

Практика: Выполнение первых программ, работа с интерфейсом среды разработки – 4 ч

Раздел 2. Переменные и типы данных – 6 ч

Теория: Типы данных, переменные, правила именования, преобразование типов – 2 ч

Практика: Решение задач с использованием чисел, строк и логических выражений – 4 ч

Раздел 3. Условные операторы – 6 ч

Теория: if, elif, else, логические операторы, вложенные условия – 2 ч

Практика: Написание программ с использованием условий – 4 ч

Раздел 4. Циклы – 6 ч

Теория: Циклы for и while, управление циклом с помощью break и continue – 2 ч

Практика: Задачи на повторяющиеся действия, генерация числовых последовательностей – 4 ч

Раздел 5. Функции – 6 ч

Теория: Создание и вызов функций, параметры и возвращаемые значения – 2 ч

Практика: Разработка программ с использованием функций – 4 ч

Раздел 6. Строки – 6 ч

Теория: Методы строк, срезы, форматирование строк – 2 ч

Практика: Преобразование и анализ строк, работа с вводом пользователя – 4 ч

Раздел 7. Списки, кортежи, множества – 6 ч

Теория: Работа с коллекциями, методы списков, неизменяемые структуры – 2 ч

Практика: Решение задач с применением списков и кортежей – 4 ч

Раздел 8. Словари – 6 ч

Теория: Структура словаря, методы и операции – 2 ч

Практика: Использование словарей в прикладных задачах – 4 ч

Раздел 9. Работа с файлами – 6 ч

Теория: Чтение и запись файлов, работа с путями – 2 ч

Практика: Написание программ, сохраняющих и обрабатывающих информацию из файлов – 4 ч

Раздел 10. Исключения – 6 ч

Теория: Обработка ошибок, конструкции try/except, raise – 2 ч

Практика: Создание надежных программ с контролем ошибок – 4 ч

Раздел 11. Модули и библиотеки – 6 ч

Теория: Импорт модулей, обзор стандартных библиотек Python – 2 ч

Практика: Использование библиотек math, random и др. – 4 ч

Раздел 12. Итоговый проект – 6 ч

Теория: Планирование, структура и презентация проекта – 2 ч

Практика: Реализация проекта: программа на Python по выбору – 4 ч

Итого: Теория – 24 ч, Практика – 48 ч, Всего – 72 ч

Раздел 1. Объектно-ориентированное программирование – 6 ч

Теория: Классы, объекты, методы, наследование – 2 ч

Практика: Создание классов и работа с ними – 4 ч

Раздел 2. Алгоритмы и структуры данных – 6 ч

Теория: Основы алгоритмов, стеки, очереди, сортировки – 2 ч

Практика: Реализация алгоритмов на Python – 4 ч

Раздел 3. NumPy и Pandas – 6 ч

Теория: Работа с массивами и таблицами данных – 2 ч

Практика: Анализ и обработка данных с использованием библиотек – 4 ч

Раздел 4. Визуализация данных – 6 ч

Теория: Принципы визуализации, обзор инструментов – 2 ч

Практика: Создание графиков с использованием Matplotlib и Seaborn – 4 ч

Раздел 5. Веб-программирование с Flask – 6 ч

Теория: Основы веб-разработки, маршрутизация – 2 ч

Практика: Создание простого веб-приложения – 4 ч

Раздел 6. Работа с API – 6 ч

Теория: REST API, HTTP-запросы, JSON – 2 ч

Практика: Получение и обработка данных с внешних сервисов – 4 ч

Раздел 7. Работа с базами данных – 6 ч

Теория: Основы SQL, подключение баз данных – 2 ч

Практика: Работа с SQLite, SQLAlchemy – 4 ч

Раздел 8. Тестирование программ – 6 ч

Теория: Виды тестов, модульное тестирование – 2 ч

Практика: Использование unittest, написание тестов – 4 ч

Раздел 9. Git и окружения – 6 ч

Теория: Контроль версий, виртуальные среды – 2 ч

Практика: Работа с GitHub, настройка venv – 4 ч

Раздел 10. Скрипты и автоматизация – 6 ч

Теория: Принципы автоматизации задач – 2 ч

Практика: Создание скриптов для повседневных задач – 4 ч

Раздел 11. Практикум: проект на Python – 6 ч

Теория: Подготовка проекта, постановка задачи – 2 ч

Практика: Разработка приложения, аналитической системы и т.п. – 4 ч

Раздел 12. Защита итогового проекта – 6 ч

Теория: Подготовка презентации, оформление документации – 2 ч

Практика: Демонстрация проекта, ответы на вопросы – 4 ч

Итого: Теория – 24 ч, Практика – 48 ч, Всего – 72 ч

Контрольно-оценочные материалы

Контроль и оценка уровня образовательных результатов освоения дополнительной общеразвивающей программы осуществляется педагогом в процессе проведения занятий.

Основными видами контрольно-оценочных средств являются педагогическое наблюдение за деятельностью обучающихся, игры на взаимодействие и тесты.

При отслеживании диагностических результатов освоения программы используются различные методы: опрос, беседа, тест, творческие зачетные задания, викторины. При этом учитываются психолого-возрастные особенности детей.

Определение результативности реализации образовательной программы проводится при анализе результатов промежуточной и итоговой диагностики.

Промежуточная диагностика проводится через полгода обучения в форме собеседования и интеллектуальной игры (своя игра).

Итоговая диагностика по программе проводится в конце года в форме тестирования.

1. Что такое переменная в Python?

- а) Специальная команда для создания цикла
- б) Область памяти для хранения данных
- в) Импортированный модуль
- г) Комментарий в коде

2. Какой тип данных используется для хранения текста?

- а) int
- б) list
- в) str
- г) float

3. Что делает функция input()?

- а) Завершает выполнение программы
- б) Получает данные от пользователя через консоль
- в) Выводит данные в консоль
- г) Определяет переменную

4. Какой символ используется для начала комментария в Python?

- а) //
- б) <!--
- в) #

г) /

5. Какой результат будет у выражения `len([10, 20, 30])`?

- а) 10
- б) 30
- в) 3
- г) Ошибка

6. Что делает оператор `if` в Python?

- а) Повторяет блок кода
- б) Выполняет блок кода при выполнении условия
- в) Завершает программу
- г) Останавливает цикл

7. Какой тип цикла выполняется, пока условие истинно?

- а) `for`
- б) `return`
- в) `while`
- г) `loop`

8. Как вызвать функцию с именем `greet`?

- а) `greet[]`
- б) `greet()`
- в) `call greet`
- г) `run(greet)`

9. Что такое список (list) в Python?

- а) Комментарий
- б) Функция
- в) Условие
- г) Упорядоченная коллекция элементов

10. Что делает метод `.append()` у списка?

- а) Удаляет элемент
- б) Переводит список в строку
- в) Сортирует список
- г) Добавляет элемент в конец списка

Ключ к тесту:

1. б
2. в
3. б
4. в
5. в
6. б
7. в

8. б
9. г
- 10.г

Оценивание:

- 8–10 правильных ответов – «5» (отлично)
- 6–7 правильных ответов – «4» (хорошо)
- 4–5 правильных ответов – «3» (удовлетворительно)
- 0–3 правильных ответов – «2» (не усвоил материал)

Условия реализации программы

Материально-техническое оснащение

Кабинет программирования оборудован необходимыми средствами для комфорtnого и результативного проведения занятий по Python:

- 13 рабочих мест для обучающихся (столы и стулья);
- 12 персональных компьютеров с мониторами, клавиатурами и мышами;
- 1 интерактивная доска или мультимедийный проектор для демонстрации кода, видеоуроков и презентаций;
- Наличие программного обеспечения: интерпретатор Python, среда разработки (например, PyCharm, VS Code), установленные библиотеки и доступ к виртуальным окружениям.

Методические материалы

Методическое сопровождение программы включает в себя разнообразные дидактические ресурсы:

- обучающие презентации по основным темам Python;
- тематические карточки и инструкции к практическим заданиям;
- практические упражнения разного уровня сложности — от базовых до проектных задач;
- методические разработки, ориентированные на формирование алгоритмического мышления и навыков программирования.

Организация учебного процесса

Занятия по Python реализуются через различные формы подачи материала, соответствующие целям и задачам каждого этапа обучения:

- объяснение нового материала с демонстрацией кода;
- практическая работа за компьютером;
- интерактивные обсуждения, поиск и исправление ошибок в коде;
- самостоятельная работа учащихся по шаблонам и проектам;
- зачеты, мини-проекты и защита практических заданий.

Структура учебных занятий по курсу включает:

1. Вводное занятие (знакомство с Python и средой разработки);
2. Теоретические и демонстрационные занятия;
3. Уроки отработки и закрепления навыков;
4. Практико-ориентированные занятия на основе проектного подхода;
5. Контрольные и диагностические занятия;
6. Комбинированные формы (теория + практика в одном блоке).

Дидактические и технические материалы

Для успешной реализации программы используются следующие средства обучения:

- авторские задания, тесты, упражнения на закрепление материала;
- компьютерные программы и среды (Python, Jupyter Notebook, PyCharm и др.);
- учебные видеоматериалы, анимации алгоритмов, проекты;
- интернет-ресурсы и онлайн-документация;
- печатные и электронные материалы по Python;
- доступ к принтерам, сканерам, средствам визуализации (интерактивная доска/проектор);
- стабильный доступ в интернет.

Требования к учащимся

Для продуктивного освоения курса обучающимся необходимо:

- иметь базовые навыки работы на компьютере;
- понимать принципы логики, алгоритмизации и структуры данных на начальном уровне;
- быть готовыми к самостоятельному поиску информации, анализу ошибок и работе с документацией.

Обучение поддерживается методическим материалом, включающим краткие конспекты, задания с пошаговыми инструкциями, примеры кода, а также домашние задания с возможностью дальнейшего совершенствования.

Список литературы

Список литературы для педагогов

1. Лутц, М. Изучаем Python / М. Лутц. — СПб.: Питер, 2022. — 1216 с.
2. Свигарт, А. Автоматизация рутинных задач с помощью Python / А. Свигарт. — М.: ДМК Пресс, 2020. — 592 с.
3. Маттис, Дж., Уэстлерман, Т. Python для детей: самоучитель по

программированию / Дж. Маттис, Т. Уэстлерман. — СПб.: Питер, 2021. — 368 с.

4. Кеннеди, Дж. Пишем тесты на Python / Дж. Кеннеди. — СПб.: Питер, 2022. — 320 с.

5. МакКинни, У. Python и анализ данных / У. МакКинни. — М.: ДМК Пресс, 2021. — 544 с.

6. Шоу, З. А. Научитесь думать как программист с Python 3 / З. А. Шоу. — М.: Эксмо, 2020. — 320 с.

7. Вирт, Н. Алгоритмы + структуры данных = программы / Н. Вирт. — М.: Вильямс, 2006. — 288 с.

Список литературы для учеников и родителей

1. Шоу, З. А. Изучаем Python. Простое и увлекательное введение в программирование / З. А. Шоу. — М.: Эксмо, 2022. — 400 с.

2. Свигарт, А. Python для начинающих. Учимся программировать, играя / А. Свигарт. — М.: ДМК Пресс, 2021. — 384 с.

3. Бриггс, Дж. Python для детей. Самоучитель по программированию / Дж. Бриггс. — СПб.: Питер, 2022. — 336 с.

4. Северенс, Ч. Python для всех / Ч. Северенс. — [Электронный ресурс]. — Бесплатное издание, доступное онлайн: <https://www.py4e.com>

5. Бэрри, П. Программирование на Python. Подробный справочник / П. Бэрри. — СПб.: Символ-Плюс, 2020. — 704 с.

6. Лутц, М. Простой Python: современный стиль программирования / М. Лутц. — СПб.: Питер, 2021. — 448 с.

Приложение

Приложение 1

Календарно-тематическое планирование. 1 год обучения

Номер темы	Тема	Дата
1	Введение в программирование на Python: Введение в Python, области применения, установка и настройка окружения разработки	03.09.2025
1	Введение в программирование на Python: Выполнение первых программ, работа с интерфейсом среды разработки	10.09.2025
1	Введение в программирование на Python: Выполнение первых программ, работа с интерфейсом среды разработки	17.09.2025
2	Переменные и типы данных: Типы данных, переменные, правила именования, преобразование типов	24.09.2025
2	Переменные и типы данных: Решение задач с использованием чисел, строк и логических выражений	01.10.2025
2	Переменные и типы данных: Решение задач с использованием чисел, строк и логических выражений	08.10.2025
3	Условные операторы: if, elif, else, логические операторы, вложенные условия	15.10.2025
3	Условные операторы: Написание программ с использованием условий	22.10.2025
3	Условные операторы: Написание программ с использованием условий	29.10.2025
4	Циклы: Циклы for и while, управление циклом с помощью break и continue	05.11.2025
4	Циклы: Задачи на повторяющиеся действия, генерация числовых последовательностей	12.11.2025
4	Циклы: Задачи на повторяющиеся действия, генерация числовых последовательностей	19.11.2025
5	Функции: Создание и вызов функций, параметры и возвращаемые значения	26.11.2025
5	Функции: Разработка программ с использованием функций	03.12.2025
5	Функции: Разработка программ с использованием функций	10.12.2025
6	Строки: Методы строк, срезы, форматирование строк	17.12.2025
6	Строки: Преобразование и анализ строк, работа с вводом пользователя	24.12.2025
6	Строки: Преобразование и анализ строк, работа с вводом пользователя	31.12.2025
7	Списки, кортежи, множества: Работа с коллекциями, методы списков, неизменяемые структуры	07.01.2026
7	Списки, кортежи, множества: Решение задач с применением списков и кортежей	14.01.2026
7	Списки, кортежи, множества: Решение задач с применением списков и кортежей	21.01.2026
8	Словари: Структура словаря, методы и операции	28.01.2026

Номер темы	Тема	Дата
8	Словари: Использование словарей в прикладных задачах	04.02.2026
8	Словари: Использование словарей в прикладных задачах	11.02.2026
9	Работа с файлами: Чтение и запись файлов, работа с путями	18.02.2026
9	Работа с файлами: Написание программ, сохраняющих и обрабатывающих информацию из файлов	25.02.2026
9	Работа с файлами: Написание программ, сохраняющих и обрабатывающих информацию из файлов	04.03.2026
10	Исключения: Обработка ошибок, конструкции try/except, raise	11.03.2026
10	Исключения: Создание надежных программ с контролем ошибок	18.03.2026
10	Исключения: Создание надежных программ с контролем ошибок	25.03.2026
11	Модули и библиотеки: Импорт модулей, обзор стандартных библиотек Python	01.04.2026
11	Модули и библиотеки: Использование библиотек math, random и др.	08.04.2026
11	Модули и библиотеки: Использование библиотек math, random и др.	15.04.2026
12	Итоговый проект: Планирование, структура и презентация проекта	22.04.2026
12	Итоговый проект: Реализация проекта: программа на Python по выбору	29.04.2026
12	Итоговый проект: Реализация проекта: программа на Python по выбору	06.05.2026

Календарно-тематическое планирование. 2 год обучения

Номер темы	Тема	Дата
1	Объектно-ориентированное программирование: Классы, объекты, методы, наследование	03.09.2025
1	Объектно-ориентированное программирование: Создание классов и работа с ними	10.09.2025
1	Объектно-ориентированное программирование: Создание классов и работа с ними	17.09.2025
2	Алгоритмы и структуры данных: Основы алгоритмов, стеки, очереди, сортировки	24.09.2025
2	Алгоритмы и структуры данных: Реализация алгоритмов на Python	01.10.2025
2	Алгоритмы и структуры данных: Реализация алгоритмов на Python	08.10.2025
3	NumPy и Pandas: Работа с массивами и таблицами данных	15.10.2025
3	NumPy и Pandas: Анализ и обработка данных с использованием библиотек	22.10.2025
3	NumPy и Pandas: Анализ и обработка данных с использованием библиотек	29.10.2025
4	Визуализация данных: Принципы визуализации, обзор инструментов	05.11.2025
4	Визуализация данных: Создание графиков с использованием Matplotlib и Seaborn	12.11.2025
4	Визуализация данных: Создание графиков с использованием Matplotlib и Seaborn	19.11.2025
5	Веб-программирование с Flask: Основы веб-разработки, маршрутизация	26.11.2025
5	Веб-программирование с Flask: Создание простого веб-приложения	03.12.2025
5	Веб-программирование с Flask: Создание простого веб-приложения	10.12.2025
6	Работа с API: REST API, HTTP-запросы, JSON	17.12.2025
6	Работа с API: Получение и обработка данных с внешних сервисов	24.12.2025
6	Работа с API: Получение и обработка данных с внешних сервисов	31.12.2025
7	Работа с базами данных: Основы SQL, подключение баз данных	07.01.2026
7	Работа с базами данных: Работа с SQLite, SQLAlchemy	14.01.2026
7	Работа с базами данных: Работа с SQLite, SQLAlchemy	21.01.2026
8	Тестирование программ: Виды тестов, модульное тестирование	28.01.2026
8	Тестирование программ: Использование unittest, написание тестов	04.02.2026
8	Тестирование программ: Использование unittest, написание тестов	11.02.2026
9	Git и окружения: Контроль версий, виртуальные среды	18.02.2026
9	Git и окружения: Работа с GitHub, настройка venv	25.02.2026
9	Git и окружения: Работа с GitHub, настройка venv	04.03.2026

10	Скрипты и автоматизация: Принципы автоматизации задач	11.03.2026
10	Скрипты и автоматизация: Создание скриптов для повседневных задач	18.03.2026
10	Скрипты и автоматизация: Создание скриптов для повседневных задач	25.03.2026
11	Практикум: проект на Python: Подготовка проекта, постановка задачи	01.04.2026
11	Практикум: проект на Python: Разработка приложения, аналитической системы и т.п.	08.04.2026
11	Практикум: проект на Python: Разработка приложения, аналитической системы и т.п.	15.04.2026
12	Защита итогового проекта: Подготовка презентации, оформление документации	22.04.2026
12	Защита итогового проекта: Демонстрация проекта, ответы на вопросы	29.04.2026
12	Защита итогового проекта: Демонстрация проекта, ответы на вопросы	06.05.2026