

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №32

РАССМОТРЕНА
на заседании методического
совета

от « 25 » 04 20 24 г.
Протокол № 2

УТВЕРЖДАЮ

Директор МБОУ СОШ № 32
Л.Н.Прогонюк



« 24 » 04 20 24 г.
Приказ № 432-13-639/4

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ**

**«Программирование на языке
Python»**

Возраст учащихся:	13-16 лет
Количество часов в год:	38
Срок реализации программы:	1 год
Автор-составитель программы:	Никитина Татьяна Евгеньевна , педагог дополнительного образования

г. Сургут
2024 год

АННОТАЦИЯ

Дополнительная общеобразовательная (общеразвивающая) программа «Программирование на языке Python», технической направленности является модифицированной и составлена на основе программы Левшиной А.В. с изменением и учетом особенности возраста и уровня подготовки детей.

Программа рассчитана на учащихся 13-16 лет, срок реализации 1 год, объем программы 38 часов.

В процессе обучения, учащиеся познакомятся с современным языком программирования Python, научатся алгоритмически решать простейшие задачи с использованием среды программирования, а также получат возможность применить полученные знания на практике через решение разнообразных задач и создание приложения с графическим интерфейсом.

Формы проведения занятий: обсуждение, дискуссии, публичная защита собственных программ, тестирование, самостоятельные задания, проекты, устный опрос.

В процессе реализации программы «Программирование на языке Python» учащиеся более полно раскроют свой творческий потенциал и реализуют собственные идеи в изучаемой области знаний, создадут предпосылки по применению освоенных навыков программирования в других учебных курсах, а также поспособствуют возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

ПАСПОРТ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ
ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

Наименование образовательной организации МБОУ СОШ № 32

Название программы	Программирование на языке Python
Направленность программы	техническая
Уровень программы	стартовый
Ф.И.О. педагога, реализующего дополнительную общеобразовательную программу	Никитина Татьяна Евгеньевна
Год разработки	2024
Где, когда и кем утверждена дополнительная общеобразовательная программа	Утверждена директором МБОУ СОШ №32 Прогонюк Л.Н. 27.04.2024
Информация и наличии рецензии	Внешняя рецензия отсутствует. Рассмотрена на заседании методического совета
Цель	Формирование навыков программирования на языке Python
Задачи	Обучающие: 1. Сформировать представление о роли информационной технологий в современном обществе; 2. Познакомить с возможностями компьютерных технологий в отношении обработки и представления графической и текстовой информации посредством написания программ; 3. Сформировать навыки работы с современным свободным программным обеспечением (СПО); 4. Изучить различные парадигмы языка программирования Python, инструменты для создания графического интерфейса пользователя. Развивающие: 1. Развивать алгоритмические способности учащихся и развивать у школьников критическое мышление, логическое мышление, алгоритмическую культуру;

	2. Сформировать приемы умственной деятельности и реализация математических способностей учащихся в ходе составления программ на языке программирования. Воспитательные: 1. Прививать интерес к информатике. 2. Сформировать у учащихся интерес к профессиям, требующим навыков алгоритмизации и программирования; 3. Сориентировать школьников на достижение образовательных результатов для успешного продвижения на рынке труда.
Ожидаемые результаты освоения программы	Учащиеся должны знать/ понимать: - сущность алгоритма, его основных свойств, иллюстрировать их на конкретных примерах алгоритмов; - основные типы данных и операторы (процедуры) языка программирования Python; - назначение процедур и функций, их различие; - принципы работы с текстовыми файлами; - способы задания элементов массивов; - методы сортировки массивов и поиска элементов в массиве; - принципы работы со строками, записями, множествами. должны уметь: - разрабатывать и записывать на языке Python типовые алгоритмы; - использовать текстовые файлы; - сортировать одномерные массивы и искать элементы заданного свойства; - разрабатывать алгоритмы на обработку строк, записей, множеств.
Срок реализации программы	1 год
Количество часов в неделю/ год	1 час/38 часов
Возраст обучающихся	13-16 лет
Формы занятий	Лекция, семинары, круглый стол, мозговая атака и др.

Методическое обеспечение	Образовательные технологии: 1. интерактивные лекции в видеоформате; 2. тренинги по решению заданий – выполнение тренировочных заданий, позволяющее приобрести опыт решения задач. Форма организации и форма проведения занятия: фронтальная, индивидуально-фронтальная, комбинированное занятие. Методы и приемы организации учебно-воспитательного процесса: Словесные: объяснение, беседа, дискуссия. Наглядные: демонстрационные материалы, видеофрагменты, показ педагогом образца выполнения задания. Информационно-коммуникационные: электронные и информационные ресурсы аудио- и видеоинформацией. Практические: практические задания, упражнения. Методы проблемного обучения: поиск (самостоятельный поиск ответа на поставленные вопросы). Методы стимулирования и мотивации деятельности и поведения: одобрение, похвала, игровые эмоциональные ситуации, использование примера. Изучение программирования с использованием интерактивных методов позволяет активизировать познавательную активность учащихся, развивать способности к самостоятельному обучению, вырабатывают навыки работы в коллективе, развивают и формируют коммуникативные навыки, а самое важное повышают учебную мотивацию. В проведении занятий по программированию выбор метода осуществляется в зависимости от дидактической задачи. К примеру, задача обобщения ранее изученного материала происходит с использованием метода «Мозгового штурма», «групповой дискуссии», «Деловой игры», «Разбора ситуаций», «Работы в малых группах», «Работы в парах», «Метода проектов».
---------------------------------	---

Условия реализации программы (оборудование, инвентарь, специальные помещения, ИКТ и др.)	Материально-техническое обеспечение: 1) компьютерный класс с 13 персональными компьютерами для обучающихся; 2) локальная сеть с доступом в интернет; 3) проектор и демонстрационный экран; 4) доска школьная; 5) на каждом ученическом компьютере должна быть установлена среда программирования Python 3.12.

Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа составлена на основе программы «Основы программирования», разработанной Левшиной А.В. в 2023 году.

Направленность дополнительной общеобразовательной программы - техническая. Уровень освоения- стартовый.

Нормативной базой для реализации дополнительной общеобразовательной программы являются:

- Федеральный Закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями);
- Распоряжение Правительства Российской Федерации от 31.03.2022 № 678-р «Об утверждении Концепция развития дополнительного образования детей до 2030 года»;
- Приказ Министерства просвещения РФ от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

А также другими Федеральными законами, иными нормативными правовыми актами РФ, законами и иными нормативными правовыми актами субъекта РФ (Ханты-Мансийского автономного округа – Югры), содержащими нормы, регулирующие отношения в сфере дополнительного образования детей, нормативными и уставными документами МБОУ СОШ № 32.

Реализация дополнительной общеобразовательной (общеразвивающей) программы осуществляется за пределами Федеральных государственных образовательных стандартов и не предусматривает подготовку обучающихся к прохождению государственной итоговой аттестации по образовательным программам

Актуальность программы. Язык программирования Python на сегодняшний день является наиболее популярной высокоуровневой средой международного олимпиадного программирования. Python – мощный и простой для изучения язык программирования. В нем представлены проработанные высокоуровневые структуры данных и простой, но эффективный подход к объектно-ориентированному программированию. Данная программа ориентирована на обучающихся 7-10 классов, интересующихся информатикой, желающих расширить знания в этой области.

Кроме того актуальность заключается в применении методик преподавания с использованием компьютерных технологий. Обучающимся предлагается изучить материал через решение задач, требующих от них гибкости ума и аналитического мышления.

Новизна данного курса заключается в общедоступности и понятности, так как все материалы представлены в готовом видео-формате и не требуют высокого уровня профессиональной предметной подготовки преподавателя, присутствующего на занятии.

Направленность: техническая

Уровень изучения: стартовый.

Отличительными особенностями программы является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования. В рамках предлагаемого курса 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.

Практическая направленность программы на создание внешних образовательных продуктов – алгоритмов, простейших приложений с графическим интерфейсом на языке Python – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики, стимулируют развитие индивидуально-личностных качеств школьников. Активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Адресат программы: программа способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся 7-10 классов.

Количество обучающихся в группе: не более 15 человек.

Срок освоения программы: 1 год.

Объем программы: 38 часа.

Режим занятий: 1 раза в неделю по 1 академическому часу.

Форма(ы) обучения: очная. В период приостановки учебно-воспитательного процесса в очной форме возможно проведение занятий с применением дистанционных технологий. Дистанционные занятия предполагают общение с учителем по видеосвязи, учащиеся выполняют общефизические упражнения и (или) задания индивидуально, изучают теоретический материал онлайн на сайте или платформе

Особенности организации образовательного процесса: концентрированное изучение курса позволяет учащимся более полно выявить свои способности в изучаемой области знаний, создать предпосылки по применению освоенных умений в других учебных курсах, подготовить себя к осознанному выбору профессий, предусматривающих программирование.

Вид образовательной деятельности – техническое творчество.

Аттестация проводится по результатам итогового тестирования. С целью контроля и оценки результатов подготовки и учета индивидуальных образовательных достижений, обучающихся применяются: поурочный контроль знаний, итоговая диагностика.

Занятия включают лекционную (теоретическую) и практическую часть. Основной тип занятий – практикум. Большинство заданий программы выполняется с помощью персонального компьютера и необходимых

программных средств. Важной составляющей каждого из практических занятий является самостоятельная работа учащихся. Теоретическая и практическая части курса изучаются параллельно. Основа курса – личностная, практическая и продуктивная направленность занятий. Методы проведения занятий связаны не только с работой за компьютером. Кроме этого, проводятся обсуждения, дискуссии, публичная защита собственных программ, выполняется проверка знаний в виде тестирования, самостоятельных заданий, проектов, устного опроса, осуществляется участие в олимпиадах и конференциях и т.д.

Цель и задачи программы:

Цель программы — формирование навыков программирования на языке Python

Образовательные:

1. Сформировать представление о роли информационных технологий в современном обществе;
2. Познакомить с возможностями компьютерных технологий в отношении обработки и представления графической и текстовой информации посредством написания программ;
3. Сформировать навыки работы с современным свободным программным обеспечением (СПО);
4. Изучить различные парадигмы языка программирования Python, инструменты для создания графического интерфейса пользователя;
5. Научить алгоритмически решать простейшие задачи с использованием среды программирования, а также дать возможность применить полученные знания на практике через решение разнообразных задач и создание приложения с графическим интерфейсом.

Развивающие:

1. Развивать алгоритмические способности учащихся и развивать у школьников критическое мышление, логическое мышление, алгоритмическую культуру;
2. Сформировать приемы умственной деятельности и реализация математических способностей учащихся в ходе составления программ на языке программирования.

Воспитательные:

1. Прививать интерес к информатике.
2. Сформировать у учащихся интерес к профессиям, требующим навыков алгоритмизации и программирования;
3. Сориентировать школьников на достижение образовательных результатов для успешного продвижения на рынке труда.

Содержание программы

Учебный план состоит из 8 блоков (модулей), которые изучаются последовательно. В конце каждого блока выполняется небольшая творческая работа, которая выполняется как индивидуально, так и в группе. Последний модуль – это творческая работа, подводящая итог обучению.

Учебный план

№ п/п	Раздел, тема	Количество часов			форма контроля
		Теорети- -ческая	Практи- -ческая	Всего часов	
1.	Среда разработки, типы данных, переменные, функции ввода и вывода, простейшие арифметические функции.	2 часа	4 часа	6 часов	устный опрос, практическая работа, тестирование
2.	Условные конструкции и циклы.	2 часа	3 часа	5 часов	устный опрос, практическая работа, тестирование
3.	Строки, списки, словари, множества.	2 часа	4 часа	6 часов	устный опрос, практическая работа, тестирование
4.	Объявление функций, возвратные функции, работа с функциями.	2 часа	4 часа	6 часов	устный опрос, практическая работа, тестирование
5.	Модули, графический модуль turtle.	1 час	3 часа	4 часа	устный опрос, практическая работа, тестирование
6.	Основы ООП.	1 часа	3 часа	4 часа	устный опрос, практическая работа, тестирование.
7.	Графический модуль tkinter. Создание приложения с графическим интерфейсом.	2 час	4 часа	6 часов	устный опрос, практическая работа, тестирование
8.	Итоговое тестирование.		1 час	1 час	итоговое тестирование
	ИТОГО: 38 часов	12 часов	26 часов	38 часов	

Содержание учебного плана

Раздел 1. Среда разработки, типы данных, переменные, функции ввода и вывода, простейшие арифметические функции. (6 часов)

1.1 Среда разработки VSCode. Переменные. (1 час)

Теория: Знакомство со средой Visual Studio Code, с понятием переменной, простейшими типами данных и операциями над ними.

Практика: Решение тренировочных задач с переменными и простыми типами данных.

1.2 Вывод данных. Функция `print()`. (1 час)

Теория: Возможности и особенности работы функции `print()` и понятие функции в целом.

Практика: Решение тренировочных задач с применением функции `print()`.

1.3 Типы данных. Функции `int()` и `str()` и `float()`. (1 час)

Теория: Подробнее о типах данных, преобразование типов данных. Зачем нужны, для чего используются и как работают функции `int()` `str()` и `float()`.

Практика: Решение тренировочных задач с использованием функций `int()` `str()` и `float()`.

1.4 Ввод данных на компьютере. Функция `input()`. (1 час)

Теория: Особенности ввода данных с клавиатуры.

Практика: Работа функции `input()`, аргумент функции `input()`, ввод строк и ввод чисел, различия, приёмы.

1.5 Операции с числами в Python. (1 час)

Теория: Целочисленная арифметика в Python, практика, применение.

Практика: Определение последней, первой цифры числа с помощью целочисленного деления.

1.6 Работа с функциями и переменными. Решение задач. (1 час)

Теория: Использование функций на практике.

Практика: Написание программ и решение задач с использованием функций `print`, `input`, `str`, `int`, `float` и работы с переменными.

Раздел 2. Условные конструкции и циклы. (5 часов)

2.1 Условные конструкции. Оператор `if-else`. (1 час)

Теория: Знакомство с логическим типом данных, условные выражения, условный оператор `if-else`.

Практика: Решение тренировочных задач с условным оператором `if-else`.

2.2 Вложенные условные конструкции. Оператор `elif`. (1 час)

Теория: Знакомство с вложенными условными инструкциями. Зачем нужен `elif`, синтаксис, применение.

Практика: Решение тренировочных задач с условным оператором `elif`.

2.3 Циклы в программировании. Цикл `while`. (1 час)

Теория: Понятие цикла в программировании, его применение, принцип работы синтаксиса цикла `while`. Практика: Манипуляции с итерациями цикла `while`, применение операторов `break` и `continue`.

2.4 Цикл for. Функция range(). (2 часа)

Теория: Принцип работы и синтаксис цикла for. Отличие от цикла while, функция range и её аргументы.

Практика: Решение тренировочных задач с применением цикла while, функции range и её аргументов.

Раздел 3. Строки, списки, словари и множества. (6 часов)

3.1 Строки. Операции над строками. Обращение по индексу. (1 час)

Теория: Углубление в строки, методы строк. Строка как массив и обращение по индексу.

Практика: Решение тренировочных задач со строковыми данными.

3.2 Строки. Операции над строками. Срезы. Методы. (1 час)

Теория: Понятие подстроки строки, срезы строк, использование метода replace().

Практика: Решение тренировочных задач с использованием метода replace().

3.3 Списки. Операции со списками. Обращение по индексу. (1 час)

Теория: Что такое список, для чего он нужен, его отличие от обычной записи в переменную.

Практика: Обращение по индексу, добавление и удаление из списка.

3.4 Списки. Работа с элементами списка. Функции и методы. (1 час)

Теория: Использование функций min(), max().

Практика: Решение тренировочных задач со списками, нахождение наибольшего\наименьшего элемента с и без использования функций min(), max().

3.5 Словари. Операции со словарями. Ключи и значения. Методы словарей. (1 час)

Теория: Структура данных словарь, принцип работы, операции со словарём.

Практика: Решение тренировочных задач, используя метод словарей.

3.6 Множества. Операции с множествами. Методы множеств. (1 час)

Теория: Структура данных множество, принцип работы, операции со множествами.

Практика: Решение тренировочных задач, с применением операций со множествами.

Раздел 4. Объявление функций, возвратные функции, работа с функциями. (6 часов)

4.1 Объявление и использование функций в Python. Аргументы функций. (2 часа)

Теория: Создание собственных функций в Python, вызов собственных функций, выбор аргументов при создании функции.

Практика: Решение тренировочных задач, с использованием собственных функций.

4.2 Объявление и использование функций в Python. Типы функций. Оператор return. (2 часа)

Теория: Понятие возвращаемого значения, оператор return. Практика: Решение тренировочных задач с оператором return.

4.3 Работа с функциями. Простейший калькулятор. (2 часа)

Теория: Повторение функций.

Практика: Написание простейшего калькулятора с использованием функций.

Раздел 5. Модули, графический модуль turtle. (4 часа)

5.1 Библиотеки в программировании. Подключение библиотек. Модуль random(). (1 час)

Теория: Как подключать библиотеки в проект на Python,

Практика: Использование собственных программ как модулей, модуль random() и его функции.

5.2 Графический модуль turtle. Функции forward(), left(), exitonclick(). (1 час)

Теория: Подключение и работа с графическим модулем turtle.

Практика: Использование turtle для визуализации работы простейших программы.

5.3 Графический модуль turtle. Функции color(), shape(), begin_fill(), end_fill(). (1 час)

Теория: Функции изменения формы, цвета и заливки.

Практика: Продвинутое рисование с помощью turtle.

5.4 Графический модуль turtle. Практика. Проект “Картина”. (1 час)

Теория: Графический модуль turtle.

Практика: Проект “Картина”, основанный на применении собственных функций и функций встроенных в модуль turtle.

Раздел 6. Основы объектно-ориентированного программирования. (4 часа)

6.1 Введение в ООП. Понятие класса, объекта, методов на примере Turtle(). (1 час)

Теория: Знакомство с ООП. Классы, объекты, методы, атрибуты на примере модуля turtle.

Практика: Решение практических задач с применением модуля turtle.

6.2 Введение в ООП. Практика. Проект “Гонки”. (1 час)

Теория: Способ создания исполнителей turtle как экземпляров класса Turtle.

Практика: Проект “Гонки” – гонки черепашек с использованием функций модуля random.

6.3 Объявление и использование классов в Python. (1 час)

Теория: Синтаксис и создание классов в Python. Понятие конструктора класса, атрибутов, методов.

Практика: Создание классов в Python.

6.4 Объявление и использование классов в Python. (1 час)

Практика: Создание простой текстовой игры с применением Классов.

Раздел 7. Графический модуль tkinter. Создание приложения с графическим интерфейсом (6 часов)

7.1 Графический модуль tkinter. Понятие графического интерфейса. Виджеты. (1 час)

Теория: Что такое графический интерфейс. Виджеты label, button, entry, создание окна приложения.

Практика: Решение тренировочных задач с применением виджетов. 7.1

Графический модуль tkinter. Виды позиционирования виджетов. (1 час)

Теория: Абсолютное и относительное позиционирование виджетов.

Практика: Решение тренировочных задач с применением методов place(), pack(), grid().

7.2 Графический модуль tkinter. Практика. Интерфейс для проекта на модуле turtle. (2 часа)

Теория: Управление исполнителем turtle на основе виджетов tkinter. Практика: Решение тренировочных задач.

7.3 Графический модуль tkinter. Практика. Приложение с графическим интерфейсом. (2 часа)

Практика: Разработка оконного приложения с графическим интерфейсом, основанном на модуле tkinter.

8. Итоговая аттестация (1 час)

8.1 Прохождение тестирования по всем темам курса.

Планируемые результаты:

Учащиеся должны знать/ понимать:

- сущность алгоритма, его основных свойств, иллюстрировать их на конкретных примерах алгоритмов;
- основные типы данных и операторы (процедуры) языка программирования Python;
- назначение процедур и функций, их различие;
- принципы работы с текстовыми файлами;
- способы задания элементов массивов;
- методы сортировки массивов и поиска элементов в массиве;
- принципы работы со строками, записями, множествами.

должны уметь:

- разрабатывать и записывать на языке Python типовые алгоритмы;
- использовать текстовые файлы;
- сортировать одномерные массивы и искать элементы заданного свойства;
- разрабатывать алгоритмы на обработку строк, записей, множеств.

Календарный учебный график

Количество учебных недель: 38

Количество учебных дней: 38

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1	сентябрь			Учебное занятие	1 ч.	Среда разработки VSCode. Переменные.	Кабинет информатики	устный опрос
2	сентябрь			Учебное занятие	1 ч.	Вывод данных. Функция print.	Кабинет информатики	практическая работа
3	сентябрь			Учебное занятие	1 ч.	Типы данных. Функции int() и str() и float().	Кабинет информатики	практическая работа
4	сентябрь			Учебное занятие	1 ч.	Ввод данных на компьютере. Функция input().	Кабинет информатики	практическая работа
5	октябрь			Учебное занятие	1 ч.	Операции с числами в Python.	Кабинет информатики	практическая работа
6	октябрь			Учебное занятие	1 ч.	Работа с функциями и переменными. Решение задач.	Кабинет информатики	практическая работа
7	октябрь			Учебное занятие	1 ч.	Условные конструкции. Оператор if-else.	Кабинет информатики	устный опрос
8	октябрь			Учебное занятие	1 ч.	Вложенные условные конструкции. Оператор elif.	Кабинет информатики	устный опрос, практическая работа
9	октябрь			Учебное занятие	1 ч.	Циклы в программировании. Цикл while.	Кабинет информатики	практическая работа
10	ноябрь			Учебное занятие	1 ч.	Цикл for.	Кабинет информатики	практическая работа
11	ноябрь			Учебное занятие	1 ч.	Функция range().	Кабинет информатики	практическая работа
12	ноябрь			Учебное занятие	1 ч.	Строки. Операции над строками. Обращение по индексу.	Кабинет информатики	устный опрос
13	ноябрь			Учебное занятие	1 ч.	Строки. Операции над строками. Срезы. Методы.	Кабинет информатики	практическая работа

14	декабрь			Учебное занятие	1 ч.	Списки. Операции со списками. Обращение по индексу.	Кабинет информатики	практическая работа
15	декабрь			Учебное занятие	1 ч.	Списки. Работа с элементами списка. Функции и методы.	Кабинет информатики	практическая работа
16	декабрь			Учебное занятие	1 ч.	Словари. Операции со словарями. Ключи и значения. Методы словарей.	Кабинет информатики	практическая работа
17	декабрь			Учебное занятие	1 ч.	Множества. Операции с множествами. Методы множеств.	Кабинет информатики	практическая работа
18	январь			Учебное занятие	1 ч.	Объявление и использование функций в Python. Аргументы функций.	Кабинет информатики	устный опрос
19	январь			Учебное занятие	1 ч.	Объявление и использование функций в Python.	Кабинет информатики	устный опрос, прак. работа
	январь			Учебное занятие	1 ч.	Типы функций. Оператор return.	Кабинет информатики	практик. работа
	январь			Учебное занятие	1 ч.	Работа с функциями. Простейший калькулятор.	Кабинет информатики	практическая работа
	февраль			Учебное занятие	1 ч.	Библиотеки в программировании.	Кабинет информатики	практическая работа
	февраль			Учебное занятие	1 ч.	Подключение библиотек.	Кабинет информатики	практическая работа
	февраль			Учебное занятие	1 ч.	Модуль random().	Кабинет информатики	устный опрос, прак.р.
	февраль			Учебное занятие	1 ч.	Графический модуль turtle. Функции forward(), left(), exitonclick().	Кабинет информатики	практическая работа
	март			Учебное занятие	1 ч.	Графический модуль turtle. Функции color(), shape(), begin_fill(), end_fill().	Кабинет информатики	практическая работа
	март			Учебное занятие	1 ч.	Графический модуль turtle. Практика. Проект “Картина”.	Кабинет информатики	практическая работа
	март			Учебное занятие	1 ч.	Введение в ООП. Понятие класса, объекта, методов на примере Turtle().	Кабинет информатики	устный опрос

	март			Учебное занятие	1 ч.	Введение в ООП. Практика. Проект “Тонки”.	Кабинет информатики	практическая работа
	апрель			Учебное занятие	1 ч.	Объявление и использование классов в Python.	Кабинет информатики	практическая работа
	апрель			Учебное занятие	1 ч.	Объявление и использование классов в Python. Практика.	Кабинет информатики	практическая работа
	апрель			Учебное занятие	1 ч.	Графический модуль tkinter. Понятие графического интерфейса. Виджеты.	Кабинет информатики	устный опрос
	апрель			Учебное занятие	1 ч.	Виды позиционирования виджетов.	Кабинет информатики	устный опрос, практическая работа
	апрель			Учебное занятие	1 ч.	Практика. Интерфейс для проекта на модуле turtle.	Кабинет информатики	практическая работа
	май			Учебное занятие	1 ч.	Интерфейс для проекта на модуле turtle.	Кабинет информатики	практическая работа
	май			Учебное занятие	1 ч.	Приложение с графическим интерфейсом.	Кабинет информатики	практическая работа
	май			Учебное занятие	1 ч.	Приложение с графическим интерфейсом.	Кабинет информатики	практическая работа
	май			Учебное занятие	1 ч.	Итоговое тестирование	Кабинет информатики	тестирование

Условия реализации программы

Для эффективной реализации настоящей программы необходимы определённые условия:

- наличие помещения для учебных занятий, рассчитанного на 15 рабочих мест и отвечающего правилам СанПин;
- наличие ученических столов и стульев, соответствующих возрастным особенностям обучающихся;
- шкафы стеллажи для оборудования, а также разрабатываемых и готовых прототипов проекта;
- наличие необходимого оборудования согласно списку;
- наличие учебно-методической базы: научная и справочная литература, наглядный материал, раздаточный материал, методическая литература.

Методическое обеспечение программы:

В ходе реализации образовательной программы используются следующие образовательные технологии:

- интерактивные лекции в видеоформате;
 - тренинги по решению заданий – выполнение тренировочных заданий, позволяющее приобрести опыт решения задач.
- Форма организации и форма проведения занятия:* фронтальная, индивидуально-фронтальная, комбинированное занятие.

Методы и приемы организации учебно-воспитательного процесса:

- *Словесные:* объяснение, беседа, дискуссия.
- *Наглядные:* демонстрационные материалы, видеофрагменты, показ педагогом образца выполнения задания.
- *Информационно-коммуникационные:* электронные и информационные ресурсы аудио- и видеоинформацией.
- *Практические:* практические задания, упражнения.

Методы проблемного обучения: поиск (самостоятельный поиск ответа на поставленные вопросы).

Методы стимулирования и мотивации деятельности и поведения: одобрение, похвала, игровые эмоциональные ситуации, использование примера.

Изучение программирования с использованием интерактивных методов позволяет активизировать познавательную активность учащихся, развивать способности к самостоятельному обучению, вырабатывают навыки работы в коллективе, развивают и формируют коммуникативные навыки, а самое важное повышают учебную мотивацию. В проведении занятий по программированию выбор метода осуществляется в зависимости от дидактической задачи. К примеру, задача обобщения ранее изученного материала происходит с использованием метода «Мозгового штурма», «групповой дискуссии», «Деловой игры», «Разбора ситуаций», «Работы в малых группах», «Работы в парах», «Метода проектов».

Материально-техническое обеспечение программы:

- 1) компьютерный класс с 15 персональными компьютерами для обучающихся;
- 2) локальная сеть с доступом в интернет;
- 3) проектор и демонстрационный экран;
- 4) доска школьная;

На каждом ученическом компьютере должна быть установлена среда программирования Python 3.12.

Кадровые: наличие квалифицированного педагога дополнительного образования;

Для обучения с применением дистанционных образовательных технологий используются технические средства, а также информационно-телекоммуникационные сети, обеспечивающие передачу по линиям связи указанной информации (образовательные онлайн-платформы, цифровые образовательные ресурсы, размещенные на образовательных сайтах, видеоконференции, вебинары, онлайн -общение, E-mail, облачные сервисы и т.д.).

Воспитательный компонент:

Для успешной реализации дополнительной общеразвивающей программы должны быть обеспечены следующие психолого-педагогические условия:

- уважение взрослых к человеческому достоинству детей, формирование и поддержка их положительной самооценки, уверенности в собственных возможностях и способностях;
- использование в образовательной деятельности форм и методов работы с детьми, соответствующих их возрастным и индивидуальным особенностям (недопустимость как искусственного ускорения, так и искусственного замедления развития детей);
- построение образовательной деятельности на основе взаимодействия взрослых с детьми, ориентированного на интересы и возможности каждого ребенка и учитывающего социальную ситуацию его развития;
- поддержка взрослыми положительного, доброжелательного отношения детей друг к другу и взаимодействия детей друг с другом в разных видах деятельности;
- поддержка инициативы и самостоятельности детей в специфических для них видах деятельности;
- возможность выбора детьми материалов, видов активности, участников совместной деятельности и общения;
- поддержка родителей (законных представителей) в воспитании детей, охране и укреплении их здоровья, вовлечение семей непосредственно в образовательную деятельность

Формы промежуточной аттестации и итогового контроля

Критерии оценивания уровня учебных достижений:

1. Содержание и объем материала определяется программой дополнительного образования.
2. Основными формами проверки ЗУН учащихся по программе « **Программирование на языке Python**» являются устный опрос, самостоятельная работа, тестирование, практическая работа на компьютерах и зачеты.
3. Учащимся выдаются разноуровневые задачи по программированию.
4. Решение задач по программированию считается безупречным, если правильно выбран способ решения, само решение сопровождается необходимыми объяснениями, верно выполнен алгоритм решения, решение записано последовательно, аккуратно и синтаксически верно по правилам какого-либо языка или системы программирования.
5. Практическая работа на компьютере считается безупречной, если учащийся самостоятельно или с незначительной помощью учителя выполнил все этапы решения задачи на компьютере, и был получен верный ответ или иное требуемое представление задания.
6. Оценка учащегося при прохождении курса «Программирование на языке Python» по системе «зачет/незачет».

**Для практических работ учащихся по алгоритмизации и программированию
(учащимся выдаются разноуровневые задания)**

Уровни	Критерии
(высокий уровень)	- работа выполнена полностью; - в графическом изображении алгоритма (блок-схеме), в теоретических выкладках решения нет пробелов и ошибок; - в тексте программы нет синтаксических ошибок (возможны одна-две различные неточности, описки, не являющиеся следствием незнания или непонимания учебного материала).
(достаточный уровень)	- работа выполнена полностью, но обоснования шагов решения недостаточны (если умение обосновывать рассуждения не являлось специальным объектом проверки); - допущена одна ошибка или два-три недочета в блок-схеме или тексте программы.
(средний уровень)	- допущены более одной ошибки или двух-трех недочетов в блок-схеме или программе, но учащийся владеет обязательными умениями по проверяемой теме.
(начальный уровень)	- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями по данной теме в полной мере.
(критичный уровень)	- работа показала полное отсутствие у учащегося обязательных знаний и умений по проверяемой теме.

Практическая работа на компьютере

Уровни	Критерии
(высокий уровень)	- учащийся самостоятельно выполнил все этапы решения задач на компьютере; - работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы.
(достаточный уровень)	- правильно выполнена большая часть работы (свыше 85%), допущено не более трех ошибок; - работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.
(средний уровень)	- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на компьютере, требуемыми для решения поставленной задачи.
(начальный уровень)	- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на компьютере или значительная часть работы выполнена не самостоятельно.
(критичный уровень)	- работа показала полное отсутствие у учащихся обязательных знаний и навыков практической работы на компьютере по проверяемой теме.

Список литературы (для педагога)

1. Марк Лутц. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.
2. Мэтиз Эрик. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. — СПб.: Питер, 2020. — 512 с.: ил.— (Серия «Библиотека программиста»).
3. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2020.
4. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
5. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
6. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2001.
7. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. –М.: МПГУ, 2015.

Ссылки:

1. Задачи Python. Задачи с решениями для языка Python (pythoninfo.ru).
2. Сборник задач по изучению основ программирования с решениями на языке Python (ictdialog.ru).
3. Питонтьютор. [Электронный ресурс]. – URL: <https://pythontutor.ru>

Список литературы (для учащихся)

1. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
2. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
3. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.

Ссылки:

1. Сайт Полякова К.Ю. [Электронный ресурс]. – URL: <http://kpolyakov.spb.ru/school/probook/python.htm>
2. Программирование на языке Python. [Электронный ресурс]. – URL: <http://informatics.mccme.ru/course/view.php?id=156>

Критерии оценивания проектов обучающихся

Критерий «Постановка цели, планирование путей ее достижения»(максимум 3 балла)		Баллы
Цель не сформулирована		0
Цель сформулирована , но план ее достижения отсутствует		1
Цель сформулирована, обоснована , дан схематичный план ее достижения		2
Цель сформулирована, четко обоснована , дан подробный план ее достижения		3
Критерий «Глубина раскрытия темы проекта» (максимум 3 балла)		
Тема проекта не раскрыта		0
Тема проекта раскрыта фрагментарно		1
Тема проекта раскрыта, автор показал знание темы в рамках программы курса		2
Тема проекта раскрыта исчерпывающе, автор продемонстрировал глубокие знания , выходящие за рамки программы курса		3
Критерий «Разнообразие источников информации, целесообразность их использования» (максимум 3 балла)		
Использована не соответствующая теме и цели проекта информация		0
Большая часть представленной информации не относится к теме работы		1
Работа содержит незначительный объем подходящей информации из ограниченного числа однотипных источников		2
Работа содержит достаточно полную информацию из разнообразных источников		3
Критерий «Соответствие выбранных способов работы цели и содержанию проекта» (максимум 3 балла)		
Заявленные в проекте цели не достигнуты		0
Значительная часть используемых способов работы не соответствует теме и цели проекта		1
Использованные способы работы соответствуют теме и цели проекта, но являются недостаточными		2
Способы работы достаточны и использованы уместно и эффективно , цели проекта достигнуты		3
Критерий «Личная заинтересованность автора, творческий подход к работе» (максимум 3 балла)		
Работа шаблонная , показывающая формальное отношение автора		0
Автор проявил незначительный интерес к теме проекта, но не продемонстрировал самостоятельности в работе, не использовал возможности творческого подхода		1
Работа самостоятельная, демонстрирующая серьезную заинтересованность автора, предпринята попытка представить личный взгляд на тему проекта, применены элементы творчества		2
Работа отличается творческим подходом , собственным оригинальным отношением автора к идее проекта		3

Критерий «Качество проведения презентации» (максимум 3 балла)	
Презентация не проведена	0
Материал изложен с учетом регламента, однако автору не удалось заинтересовать аудиторию	1
Автору удалось вызвать интерес аудитории, но он вышел за рамки регламента	2
Автору удалось вызвать интерес аудитории и уложиться в регламент	3
Критерий «Качество проектного продукта» (максимум 3 балла)	
Проектный продукт отсутствует	0
Проектный продукт не соответствует требованиям качества (эстетика, удобство использования, соответствие заявленным целям)	1
Продукт не полностью соответствует требованиям качества	2
Продукт полностью соответствует требованиям качества (эстетичен, удобен в использовании, соответствует заявленным целям)	3

Система оценивания:

91%-100%- "5" или «высокий уровень»

70%-90% - "4" или «повышенный уровень»

50%-69% - "3" или

«средний уровень»Ниже

50% - "2" или «низкий

уровень»