

**Государственное бюджетное общеобразовательное учреждение
средняя общеобразовательная школа № 438 Приморского района Санкт-Петербурга**

ПРИНЯТА
Педагогическим советом
(протокол от 31.08.2026 № 1)

УТВЕРЖДЕНА
Приказ от 31.08.2026 № 226-О
Директор _____ И.И. Боякова

ПРИНЯТА
С учетом мнения совета родителей
(законных представителей)
несовершеннолетних обучающихся
(протокол от 31.08.2026 № 1)

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Программирование на Python»**

Срок освоения: одногодичная
Возраст обучающихся: 13-17 лет

Разработчик программы
Петельская Наталья Александровна,
педагог доп. образования
первой категории

Санкт-Петербург

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеразвивающая программа «Программирование на Python» разработана в соответствии с основными нормативными документами в области дополнительного образования.¹

Дополнительная общеразвивающая программа «Программирование на Python» относится к программам **технической направленности**.

Актуальность программы.

Программа соответствует государственной политике в области дополнительного образования и социальному заказу общества. При определении актуальности программы использовался анализ детского и родительского спроса.

В ходе реализации дополнительной общеразвивающей программы «Программирование на Python» создаются условия для выявления, поддержки и развития способностей и талантов обучающихся, что соответствует основным задачам Федерального проекта «Успех каждого ребенка», национального проекта «Цифровая экономика» (федерального проекта «Кадры для цифровой экономики») и национального проекта «Образование» (федерального проекта «Молодые профессионалы»).

Программа «Программирование на Python» закладывает начальные знания и навыки в области алгоритмизации и программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования креативного мышления и навыков проектной деятельности.

Адресат программы

Программа предназначена для воспитанников 13-17 лет, заинтересованных в знакомстве с современным техническим творчеством и азам программирования.

Для обучения принимаются все желающие, не имеющие медицинских противопоказаний

Уровень освоения программы – общекультурный.

Объем и срок освоения программы

Срок освоения: одногодичный

Объем программы: 72 часов.

Отличительные особенности /новизна

Особенностью данной программы является нацеленность на формирование у учащихся навыков поиска собственного решения поставленной практической задачи, представленную в виде адаптированного кейса из реальной жизни, составления алгоритма решения и его реализации с помощью средств программирования.

Цель программы:

Формирование базовых знаний и навыков в области программирования на языке Python через практическую деятельность.

Задачи

- **Обучающие**
 - Познакомить с базовыми понятиями основ программирования.

¹ Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

Распоряжение Правительства Российской Федерации от 31.03.2022 № 678-р «об утверждении Концепции развития дополнительного образования детей до 2030 г. и плана мероприятий по ее реализации»;

Приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

Распоряжение Комитета по образованию от 25.08.2022 № 1676-р «Об утверждении критериев оценки качества дополнительных общеразвивающих программ, реализуемых организациями, осуществляющими образовательную деятельность, и индивидуальными предпринимателями Санкт-Петербурга».

- Познакомить с особенностями синтаксиса языка программирования Python.
- Познакомить с принципами объектно-ориентированного программирования.
- Сформировать навык правильного оформления кода.
- Сформировать навык использования итеративного подхода при решении различных задач.
- Формировать навык решения задач и разработки проектов с помощью Python.
- Формировать навык разработки игр с помощью языка программирования Python.
- Формировать навык тестирования и оптимизации созданных проектов.
- **Развивающие:**
 - Формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление.
 - Формировать навык публичного выступления и презентации.
 - Формировать навык планирования деятельности и основ тайм-менеджмента.
 - Формировать и расширять словарный запас на английском языке.
 - Формировать функциональную грамотность.
 - Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.
- **Воспитательные:**
 - Воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект.
 - Формировать умение давать качественную обратную связь и реагировать на нее.
 - Формировать информационную культуру.

Планируемые результаты освоения программы

Предметные

Учащиеся будут *знать*:

- основы логического и алгоритмического мышления, пространственного воображения и математической речи, основы счёта, измерения, записи и выполнения алгоритмов и кодов на языке Python;

Учащиеся будут *уметь*:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- настраивать рабочую среду Python;

Метапредметные

Учащиеся будут *способны*:

- понимать причины успеха, неуспеха учебной деятельности и способности конструктивно действовать даже в ситуациях неуспеха;
- осознавать начальную форму познавательной и личностной рефлексии;
- решать проблемы творческого и поискового характера;
- формировать умения планировать, контролировать и оценивать учебные действия в соответствии с поставленной задачей и условиями её реализации;
- определять наиболее эффективные способы достижения результата.

Личностные

У учащихся будут развиты:

- внимательность и наблюдательность, творческое воображение и фантазия;
- волевые качества личности (усидчивость, трудолюбие, терпение, дисциплинированность, ответственность);
- принятие и освоение социальной роли обучающегося, развитие мотивов учебной деятельности и формирование личностного смысла учения;
- способность принимать и сохранять цели и задачи учебной деятельности, поиск средств её осуществления;
- мотивации к творческому труду, работе на результат, бережному отношению к материальным и духовным ценностям.

Организационно-педагогические условия реализации программы

Язык реализации программы: государственный язык РФ – русский.

Форма обучения – очная.

Особенности реализации общеразвивающей программы

Дополнительная общеразвивающая программа «Программирование на Python» может быть реализована в период учебного года, включая каникулярный период (с 1 сентября по 31 мая).

Условия набора и формирования групп

На обучение по дополнительной общеразвивающей программе принимаются все желающие дети без предварительного отбора, не имеющие медицинских противопоказаний. Наличие базовых знаний в данной области не требуется.

Условия формирования групп:

Группы формируются из учащихся в возрасте 13 – 17 лет. Допускается обучение в разновозрастных группах.

Количество учащихся в группе – 10-15 человек;

Формы организации и проведения занятий

Формы организации деятельности на занятии:

- *фронтальная:* работа педагога со всеми учащимися одновременно (беседа, показ, объяснение);
- *коллективная:* организация проблемно-поискового или творческого взаимодействия между всеми детьми одновременно (создание коллективной работы и т.п.);
- *групповая:* организация работы (совместные действия, общение, взаимопомощь) в малых группах, в т.ч. в парах, для выполнения определенных задач; задание выполняется таким образом, чтобы был виден вклад каждого учащегося (группы могут выполнять одинаковые или разные задания, состав группы может меняться в зависимости от цели деятельности);
- *индивидуальная:* организуется для работы с одаренными детьми, а также для коррекции пробелов в знаниях и отработки отдельных навыков.

Формы проведения занятий:

- *общеразвивающие:* беседа, творческая встреча.
- *обучающие:* традиционное учебное занятие, мастер-класс, игра, самостоятельные творческие проекты.
- *отчетные:* защита своего проекта, фотосессия

Материально-техническое обеспечение программы

Для успешной реализации программы необходимо:

1. Кабинет, оснащенный по всем требованиям безопасности и охраны труда.
2. On-line выход в Интернет (желательно выделенная линия).
3. Столы.

4. Стулья.
5. Компьютеры (моноблоки) – 15 шт.
6. Мультимедиа проектор.
7. Экран.
8. Интерактивная доска.
9. Дисковые накопители/флешки.

Программное обеспечение:

1. Операционная система Linux Astra;
2. Среда программирования Python
3. Текстовый редактор Блокнот, Текстовый процессор
4. Редактор презентаций.

Методические материалы

1. Мультимедийные презентации
2. Электронные книги и учебники
3. Библиотеки программ.

Для успешного проведения занятий необходимо создать электронную библиотеку, на котором находились бы все материалы курса: конспекты лекций, визуальные материалы для занятий, практические задания и работы учащихся, список рекомендуемой литературы, материалы для дополнительного чтения.

Кадровое обеспечение

Программа реализуется педагогом дополнительного образования первой категории технической направленности с высшим образованием без привлечения иных специалистов.

УЧЕБНЫЙ ПЛАН ПРОГРАММЫ

№ пп/п	Наименование раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	Раздел 1. Введение в Python	4	3	1	Тест по модулю
	1.1. Инструкции и структура программы.	2	2	0	Устный опрос
	1.2. Установка Python. Ввод и вывод данных.	2	1	1	Решение задач
2.	Раздел 2. Типы данных и операции	12	6	6	Тест по модулю
	2.1. Переменные. Операторы.	4	2	2	Тест по теме
	2.2 Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	8	4	4	Решение задач
3.	Раздел 3. Инструкции и синтаксис	14	7	7	Тест по модулю
	3.1. Условный оператор if, Альтернативный условный оператор elif, else.	4	2	2	Решение задач
	3.2. Циклы while. Цикл for. Обработка вложений. Функция random. Случайные числа.	10	5	5	Решение задач с использованием циклов
4.	Раздел 4. Функции и модули в программировании	18	9	9	Тест по модулю
	4.1. Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат.	6	3	3	Тест по теме
	4.2. Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.	12	6	6	Решение задач с использованием строк

№ пп/п	Наименование раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
5.	Раздел 5. Сложные типы данных	14	7	9	Тест по модулю
	5.1. Списки. Срезы списков. Решение задач со списками	8	3	5	Решение задач
	5.2. Матрицы	2	1	1	Решение задач
	5.3. Словари	2	1	1	Решение задач
	5.4. Множества в языке Python	2	1	1	Решение задач
6.	Раздел 6. Написание и отладка программ. Самостоятельная работа	12	2	10	Защита проекта
	Итого	72	17	55	

Рабочая программа

Задачи

- **Образовательные:**

- Познакомить с основами программирования на Python
- Научить базовым принципам алгоритмики
- Сформировать навыки создания основных кодов для решения задач

- **Развивающие:**

- Развивать логическое мышление
- Развивать творческие способности
- Сформировать алгоритмическое мышление

- **Воспитательные:**

- Воспитать интерес к техническим наукам
- Сформировать командное взаимодействие
- Развивать усидчивость и внимательность

Содержание программы

Содержание учебного плана

Раздел 1. Введение в Python. Инструкции и структура программы.

Теория (3 ч.). Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка. Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации.

Практика (1 ч.). Практическая работа: Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран.

Формы и методы. Объяснительно-иллюстративный, репродуктивный, диалогический, эвристический, технология индивидуальных консультаций, практическая деятельность.

Результаты обучения.

Учащиеся должны знать:

- ✓ инструктаж по ТБ;
- ✓ организацию рабочего места;
- ✓ правила безопасного использования Интернета;
- ✓ правила установки и настройки Python
- ✓ правила ввода и вывода данных в среде программирования Python

Учащиеся должны уметь:

- ✓ работать с источниками информации;
- ✓ работать с браузерами;
- ✓ вносить собственные дополнения и изменения в работе,
- ✓ соблюдать аккуратность и внимательность в работе.

Диагностика и способы отслеживания результата:

устный опрос по пройденному теоретическому материалу, экспресс тест.

Раздел 2. Типы данных и операции

Теория (2 ч.). Переменные. Операторы.

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.

Практика (2 ч.). Практическая работа: Решение задач на элементарные действия с числами. Создание программы простейший калькулятор. Типы данных. Базовые функции для работы с различными типами данных.

Теория (4 ч.). Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)

Практика (4 ч.) Практическая работа: Решение математических задач с использованием функции import math.

Формы и методы. Объяснительно-иллюстративный, репродуктивный, диалогический, эвристический, технология индивидуальных консультаций, практическая деятельность.

Результаты обучения.

Учащиеся должны знать:

- ✓ Понятие переменных и операторов.
- ✓ Понятие о типах данных.
- ✓ Как преобразовывать типы данных.
- ✓ Различать типы данных.
- ✓ Базовые функции (abs, round, int, math).

Учащиеся должны уметь:

- ✓ работать с переменными и операторами;
- ✓ работать с разными типами данных;
- ✓ вносить собственные дополнения и изменения в работе,
- ✓ соблюдать аккуратность и внимательность в работе.

Диагностика и способы отслеживания результата:

устный опрос по пройденному теоретическому материалу, экспресс тест.

Раздел 3. Инструкции и синтаксис

Теория (2 ч.). **Условные операторы if, elif, else.**

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.

Практика (2 ч.) Практическая работа: Практическое закрепление знаний по условным операторам. Создание программ.

Теория (5 ч.). **Циклы while, for. Обработка исключений. Случайные числа.**

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.

Практика (5 ч.) Практическая работа: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

Формы и методы. Объяснительно-иллюстративный, репродуктивный, диалогический, эвристический, технология индивидуальных консультаций, практическая деятельность.

Результаты обучения.

Учащиеся должны знать:

- ✓ Понятие логический тип данных
- ✓ Понятие сложные условные выражения (логические операции and, or, not).
- ✓ Понятие цикла.
- ✓ Различать типы циклов.

Учащиеся должны уметь:

- ✓ работать с условными операторами;
- ✓ работать с разными типами цикла;

- ✓ вносить собственные дополнения и изменения в работе,
- ✓ соблюдать аккуратность и внимательность в работе.

Диагностика и способы отслеживания результата:

устный опрос по пройденному теоретическому материалу, самостоятельная работа, экспресс тест.

Раздел 4. Функции и модули в программировании

Теория (3 ч.). Встроенные и пользовательские функции.

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.

Практика (3 ч.) Практическая работа: Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.

Теория (6 ч.). Строки - последовательности символов.

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.

Практика (6 ч.) Практическая работа: Написание программ. Работа со строками.

Формы и методы. Объяснительно-иллюстративный, репродуктивный, диалогический, эвристический, технология индивидуальных консультаций, практическая деятельность.

Результаты обучения.

Учащиеся должны знать:

- ✓ Понятие встроенных и пользовательских функций.
- ✓ Понятие о типах данных - строка.
- ✓ Как преобразовывать типы данных.
- ✓ Различать типы данных.
- ✓ Рекурсивные функции.

Учащиеся должны уметь:

- ✓ работать со строками;
- ✓ работать с параметрами и аргументами;
- ✓ вносить собственные дополнения и изменения в работе,
- ✓ соблюдать аккуратность и внимательность в работе.

Диагностика и способы отслеживания результата:

устный опрос по пройденному теоретическому материалу, экспресс тест.

Раздел 5. Сложные типы данных.

Теория (3 ч.). Списки. Срезы списков. 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.

Практика (5 ч.) Практическая работа: Написание программ. Работа со списками.

Теория (1 ч.). Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.

Практика (1 ч.) Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Теория (1 ч.). Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Практика (1 ч.) Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Теория (1 ч.). Словари. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Практика (1 ч.) Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Теория (1 ч.). Множества в языке Python. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практика (1 ч.) Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Формы и методы. Объяснительно-иллюстративный, репродуктивный, диалогический, эвристический, технология индивидуальных консультаций, практическая деятельность.

Результаты обучения.

Учащиеся должны знать:

- ✓ Понятие списка, матрицы, кортежа, словари и множества в языке Python.
- ✓ Как преобразовывать типы данных.
- ✓ Различать списка, матрицы, кортежа, словари и множества в языке Python.
- ✓ Базовые операторы по теме.

Учащиеся должны уметь:

- ✓ работать со списками, матрицами, кортежами, словарями и множествами в языке Python;
- ✓ работать с разными типами данных;
- ✓ вносить собственные дополнения и изменения в работе,
- ✓ соблюдать аккуратность и внимательность в работе.

Диагностика и способы отслеживания результата:

устный опрос по пройденному теоретическому материалу, самостоятельная работа, экспресс тест.

Раздел 6. Написание и отладка программ. Самостоятельная работа

Теория (2 ч.). Техника безопасности. Области применения и перспективы. Инструкции и структура программы.

Практика (10 ч.) Практическая работа: Стиль программирования. Отладка программ. Проектная работа по курсу «Программирование на языке Python».

Формы и методы. Объяснительно-иллюстративный, репродуктивный, диалогический, эвристический, технология индивидуальных консультаций, практическая деятельность.

Результаты обучения.

Учащиеся должны знать:

- ✓ Понятие правил кода.
- ✓ Понятие о типах данных.
- ✓ Как преобразовывать типы данных.
- ✓ Различать типы данных.
- ✓ Базовые функции, операторы.

Учащиеся должны уметь:

- ✓ работать с переменными, операторами, функциями;
- ✓ работать с разными циклами;
- ✓ обрабатывать простые и сложные типы данных;
- ✓ вносить собственные дополнения и изменения в работе;
- ✓ соблюдать аккуратность и внимательность в работе.

Диагностика и способы отслеживания результата:

устный опрос по пройденному теоретическому материалу, самостоятельная работа, защита проекта.

Планируемые результаты

У обучающихся должны быть сформированы кроме основ общекультурных, общеучебных компетенций, компетентностей по алгоритмике и логики на основе программирования на языке Python, навыки и умения технического и логического проектирования и моделирования.

По окончании курса обучения по программе у обучающихся будут сформированы основы общекультурных, общеучебных и предметных компетенций, которые обеспечат ему комфортное вхождение в образовательную и социальную среду выбора старшеклассниками профиля дальнейшего обучения, будущей профессии. Самореализация учащихся путем участия в конкурсах, соревнованиях разного уровня.

Личностные образовательные результаты:

- готовность к самоидентификации в окружающем мире на основе критического анализа информации, отражающей различные точки зрения на смысл и ценности жизни;
- умение создавать и поддерживать индивидуальную информационную среду, обеспечивать защиту значимой информации и личную информационную безопасность, развитие чувства личной ответственности за качество окружающей информационной среды;
- приобретение опыта использования информационных ресурсов общества и электронных средств связи в учебной и практической деятельности;
- умение осуществлять совместную информационную деятельность, в частности при выполнении учебных проектов;
- повышение своего образовательного уровня и уровня готовности к продолжению обучения с использованием ИКТ.

Метапредметные образовательные результаты:

- планирование деятельности: определение последовательности промежуточных целей с учётом конечного результата, составление плана и последовательности действий;
- прогнозирование результата деятельности и его характеристики;
- контроль в форме сличения результата действия с заданным эталоном;
- коррекция деятельности: внесение необходимых дополнений и корректив в план действий;
- умение выбирать источники информации, необходимые для решения задачи (средства массовой информации, электронные базы данных, информационно-телекоммуникационные системы, Интернет, словари, справочники, энциклопедии и др.);
- умение выбирать средства ИКТ для решения задач из разных сфер человеческой деятельности;

Предметные образовательные результаты:

- Способность и готовность применять необходимые для построения моделей знания, принципов действия и математического описания составных систем (информационных и средств вычислительной техники);
- Способность реализовывать модели средствами вычислительной техники;
- Владение навыками разработки макетов информационных систем;
- Владение основами разработки алгоритмов и составления программ;
- Умение проводить настройку и отладку программ.

Календарно-тематическое планирование

№ п\	Название тем	Всего часов	ип/форма занятия	емые результаты обучения	Виды и формы контроля
Раздел 1. Введение в Python					
1.	1.1. Инструкции и структура программы. Вводное занятие. Знакомство с группой. Инструктаж по ТБ. Цели и задачи курса.	2	Диагностическая , Открытие новых знаний.	Расширение знаний учащихся о информации и информатике, о ее значении в жизни человека, о ее связях с другими науками	Наблюдение, беседы
2.	Установка Python. Ввод и вывод данных.	2	Открытие новых знаний. Просмотр видеоурока про Установку Python. Написание первой программы	Расширение знаний учащихся о языке программирования Python	Наблюдение, создание и отладка программы
Раздел 2. Типы данных и операции					
3.	Переменные. Операторы.	4	Комбинированное тематическое занятие, практическое занятие	Расширение знаний учащихся о Переменных. Операторах..	Наблюдение, решение задач
4.	Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math)	8	Открытие новых знаний. практическое занятие	Расширение знаний учащихся о типах данных. Преобразование типов. Различия типов данных. Базовые функции (abs,round, int, math).	Наблюдение, решение задач, тестирование.
Раздел 3. Инструкции и синтаксис					
5.	Условный оператор if, Альтернативный условный оператор elif, else.	4	Открытие новых знаний.	Расширение знаний учащихся о циклах и операторах.	Наблюдение, решение задач, тестирование.

6.	Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.	10	Открытие новых знаний.	Расширение знаний учащихся о циклах и операторах.	Наблюдение, Решение задач с использованием циклов, тестирование.
Раздел 4. Функции и модули в программировании					
7.	Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат.	6	Открытие новых знаний. Комбинированное тематическое занятие.	Расширение знаний учащихся о функциях и модулях в программировании. Научиться применять на практике теоретический материал по теме.	Наблюдение, создание программы
8.	Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.	12	Комбинированное тематическое занятие.	Научиться применять на практике теоретический материал по теме.	Наблюдение, Решение задач с использованием строк
Раздел 5. Сложные типы данных					
9.	Списки. Срезы списков. Решение задач со списками	8	Комбинированное тематическое занятие	Расширение знаний учащихся о математике, логике, алгоритме. Научиться применять на практике теоретический материал по теме.	Наблюдение, практика, участие в создании проекта
10.	Матрицы	2	Комбинированное тематическое занятие	Расширение знаний учащихся о математике, логике, алгоритме. Научиться применять на практике теоретический материал по теме.	Наблюдение, практика, решение задач

11.	Словари	2	Комбинированное тематическое занятие	Развитие внимания, памяти, логики. Научиться применять на практике теоретический материал по теме.	Наблюдение, практика, решение задач
12.	Множества в языке Python	2	Комбинированное тематическое занятие	Развитие внимания, памяти, логики, аналитических способностей. Научиться применять на практике теоретический материал по теме.	Наблюдение, практика, решение задач
Раздел 6. Написание и отладка программ.					
13.	Написание и отладка программ. Самостоятельная работа	10	Практическое занятие	Научиться применять на практике теоретический материал по теме. Самостоятельная работа.	Наблюдение, решение задач, создание и защита проекта.
14.	Итоговое занятие	2	Итоговое занятие	Научиться применять на практике теоретический материал по теме. Защитить проект. Провести самоанализ, самооценку. Доработать модели.	Наблюдение, защита проекта.

Методическое обеспечение

Методы и приемы, используемые на занятиях.

Методическое обеспечение программы включает приёмы и методы организации образовательного процесса, дидактические материалы, техническое оснащение занятий.

Методы и приёмы организации образовательного процесса при реализации программы:

Словесные методы: объяснение, беседа, комментированное чтение, рассказ. Практические методы: работа с текстом, составление планов, работа над проектами, выполнение творческих заданий: составление кроссвордов, сочинение загадок, рассказов, выпуск бюллетеней, сборников или альбомов с творческими работами и проектами.

Игровые методы: фантазирование, театральная импровизация, живая наглядность.

Наглядные методы: показ видеоматериалов, посещение выставок, проведение экскурсий.

Виды дидактических материалов, используемые при реализации программы:

Для обеспечения наглядности и доступности изучаемого материала педагог использует наглядные пособия следующих видов:

- схематические или символические (оформленные стенды и планшеты, таблицы, схемы, рисунки, графики, плакаты, диаграммы, чертежи, шаблоны и т.п.);

- картинные (иллюстрации, слайды, фотоматериалы и др.);
- звуковые (аудиозаписи);
- смешанные (видеозаписи, учебные кинофильмы и т.д.);
- дидактические пособия (карточки, рабочие тетради, раздаточный материал, вопросы и задания для опроса, тесты, практические задания, упражнения и др.).
- компьютерные программы в электронном виде (компьютеры с программами, CD, флеш-носители);
- учебные пособия, журналы, книги, Интернет-ресурсы.

При реализации программы с целью повышения качества и эффективности процесса обучения применяются современные эффективные технологии обучения, ориентированные не на накопление знаний, а на организацию активной деятельности обучающихся:

- технологии проектной деятельности;
- компьютерные (информационные) технологии;
- технологии учебно-игровой деятельности (моделирование);
- технологии коммуникативно-диалоговой деятельности;
- модульные технологии;
- квест-технологии;
- технологии личностно-ориентированного обучения;
- кейс-технологии.

Информационные технологии используются в различных видах деятельности:

- при подготовке и проведении занятий;
- для создания авторских мультимедийных презентаций;
- в рамках индивидуальной и групповой проектной деятельности;
- для самостоятельной работы;
- для накопления демонстрационных материалов к занятиям (видеоматериалы, таблицы, презентации, карты);

Одним из основных методов является метод проектного обучения, так как он является неотъемлемой частью учебного процесса. Исходный лозунг основателей системы проектного обучения – «Все из жизни, все для жизни». Обучение строится на активной основе, через практическую деятельность ученика, ориентируясь на его личный интерес и практическую востребованность полученных знаний в дальнейшей жизни, обучающийся имеет возможность через проектную деятельность освоить получаемые знания. Проекты представляются в виде готовых программ, презентаций проектов, научных докладов, моделей, демонстрации видеофильма. Достоинствами проектной деятельности являются:

- Уметь работать в коллективе;
- Брать ответственность за выбор решения на себя;
- Разделять ответственность с другими;
- Предоставлять ребенку свободу выбора темы, методов работы;
- Понимание каждым обучающимся важности работы и др.

Стимулы: отбор работы на конкурс, награждение грамотой или ценным призом.

СПИСОК ЛИТЕРАТУРЫ

Список литературы, используемой педагогом Основная:

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2016. - 800 с.

2. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил.

3. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.

4. УМК Босова Л.Л., Босова А.Ю. и др. Python: графика и анимация <https://bosova.ru/metodist/authors/informatika/3/python.php>

Дополнительная:

1. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.

2. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2012. – 520 с.

3. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Петербург, 2011. - 295 с.

4. ООП на Python: концепции, принципы и примеры реализации [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/python-oop/>

Список рекомендуемой литературы для обучающихся Основная:

1. Свейгар Эл. Учим python, делая крутые игры М: Эксмо, 2018. – 416 с.

2. Семакин И.Г. Основы алгоритмизации и программирования. М.: Академия, 2016. - 304 с.

Дополнительная:

1. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.

2. Уроки по Python для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>

3. Алгоритмизация. Программирование Python 3 [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/7215/promo>

Оценочные материалы.

Программа дополнительного образования «Робототехника» предполагает входной и текущий контроль, промежуточную аттестацию и итоговое оценивание для этого используют различные оценочные материалы и методы, например:

Текущий контроль. Проводится в конце каждой темы в виде устного опроса, выставки работ, мини-соревнований, защиты проектной работы, выступления конкурсах проектных работ.

Самостоятельная работа. Помогает отследить результативность обучающихся и выявить разные уровни подготовки:

Минимальный (наличие темы проекта).

Средний (наличие презентации проекта, хорошее знание материала).

Высокий (наличие презентации проекта с развёрнутым раскрытием темы, прекрасное знание материала).

Опрос по итогам профильных занятий. Участники отвечают на вопросы, чтобы продемонстрировать свои знания и умения.

Анкетирование. Участники заполняют анкеты, чтобы оценить свои достижения и опыт участия в программе.

Наблюдение. Педагог наблюдает за участниками в ходе их жизни в коллективе и оценивают их активность и вовлечённость.

Результаты освоения программного материала определяют по трём уровням: высокий, средний, низкий. Используют 3-балльную систему оценки: 3 балла — высокий уровень, 2 балла — средний уровень, 0–1 балл — низкий уровень.

Приложение 1

Общие параметры критериев педагогической оценки по мониторингу освоения дополнительной общеразвивающей программы «Программирование на Python»

Оценка по 3-х-балльной шкале.

Входной контроль	Теоретические задания. Тестирование. Собеседование.		0-1	Теоретические знания отсутствуют. Обучающийся никогда не занимался данным видом деятельности.
			2	Обучающийся имеет минимальные представления по выбранному направлению «Программирование на Python».
			3	Обучающийся имеет широкие представления по выбранному направлению «Программирование на Python». На определенном уровне владеет данным видом деятельности.
	Практические навыки. Контрольные задания.		0-1	Полное отсутствие практических навыков.
			2	Навыки находятся в начальной стадии формирования.
			3	У обучающегося сформированные определенные навыки.
	Личностное развитие. Наблюдение. Собеседование.		0-1	Отсутствие заинтересованности.
			2	Проявление частичного интереса к выбранному направлению.
			3	Обучающемуся интересен творческий процесс и результат этого процесса.
Итоговый контроль	Теоретические задания. Тестирование.		0-1	Обучающийся не усвоил (или усвоил только на начальном этапе) теоретические знания по направлению программирования
			2	Обучающийся усвоил базовые теоретические знания.
			3	Обучающийся полностью усвоил теоретические знания в соответствии с программой данного объединения.
	Практические навыки. Проектные задания.		0-1	Обучающийся не усвоил (или усвоил частично) практические навыки на базовом уровне.
			2	Обучающийся усвоил практические навыки на базовом уровне.
			3	Обучающийся полностью усвоил практические навыки по образовательной программе.
	Личностное развитие. Наблюдение. Собеседование.		0-1	Обучающийся не заинтересован в продолжении обучения по данному виду творчества.
			2	Обучающийся заинтересован в получении итоговых результатов, но не уверен в продолжении обучения.
			3	Обучающийся заинтересован в продолжение обучения и в том, чтобы выйти на более высокий уровень, как в теоретических, так и в практических знаниях по данному виду творчества.
	Общеучебные умения и навыки	Учебно-интеллектуальные умения:	0-1	Обучающийся не усвоил (или усвоил частично) навыки на базовом уровне.
			2	Обучающийся усвоил навыки на базовом

				уровне.
			3	Обучающийся полностью усвоил навыки по образовательной программе.
		Учебно-коммуникативные умения:	0-1	Обучающийся не усвоил (или усвоил частично) навыки на базовом уровне.
			2	Обучающийся усвоил навыки на базовом уровне.
			3	Обучающийся полностью усвоил навыки по образовательной программе.
		Учебно-организационные умения и навыки:	0-1	Обучающийся не усвоил (или усвоил частично) навыки на базовом уровне.
			2	Обучающийся усвоил навыки на базовом уровне.
			3	Обучающийся полностью усвоил навыки по образовательной программе.
		Предметные достижения на базе кружка «Робототехника»	0-1	Обучающийся не усвоил (или усвоил частично) навыки на базовом уровне.
			2	Обучающийся усвоил навыки на базовом уровне.
			3	Обучающийся полностью усвоил навыки по образовательной программе.

50% - минимальный уровень усвоения

50%-80% -базовый уровень усвоения

80%-100% - максимальный уровень усвоения

Карточка диагностики (входная) ДОП «Программирование на Python»

№п/п	ФИ обучающегося	Количество баллов		
		Теоретические задания. Тестирование. Собеседование.	Практические навыки. Контрольные задания.	Личностное развитие. Наблюдение. Собеседование.
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

**Итоговая карточка учета результатов обучающихся по дополнительной общеразвивающей программе
«Программирование на Python».**

[illegible]

