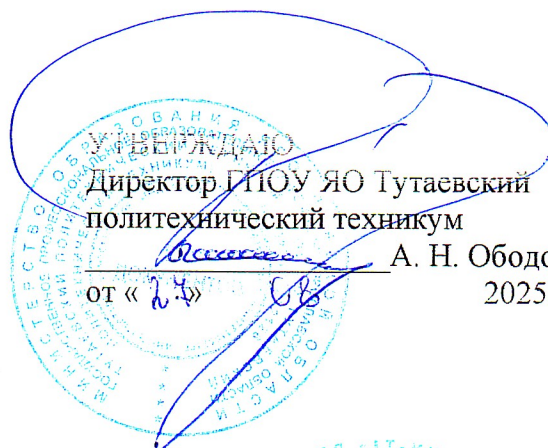


Государственное профессиональное образовательное учреждение
Ярославской области
Тутаевский политехнический техникум

РАССМОТРЕНО

на заседании методической комиссии

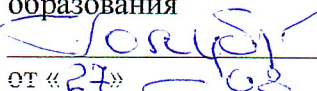
от «24» 08 2025г.

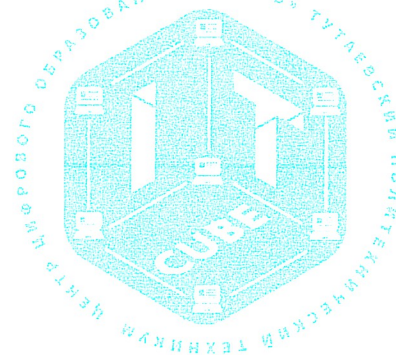


А. Н. Ободов
2025 г.

СОГЛАСОВАНО

Руководитель центра цифрового
образования

 Н.В.Голубкова
от «27» 08 2025 г.



Дополнительная общеобразовательная общеразвивающая программа
Технической направленности
«Программирование на языке Python»

Возраст детей: 12 - 18 лет

Срок реализации: 2 года

Наполняемость группы: 8-12 человек

Автор-составитель:

педагог дополнительного образования

Афоничева Александра Михайловна

Тутаев, 2025

Содержание

1.	Пояснительная записка	3
2.	Учебно-тематический план программы 1 года	9
3.	Учебно-тематический план 2 года обучения.....	10
4.	Содержание программы 1 года	11
5.	Содержание программы 2 года обучения.....	18
6.	Методическое обеспечение программы 1 года.....	23
7.	Методическое обеспечение 2 года обучения.....	25
8.	Воспитательная деятельность.....	27
9.	Список информационных источников.....	30
	Приложение.....	31

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа разработана на основе педагогического опыта по направлению «Основы программирования на языке Python.».

Программа разработана на основе:

- Федерального закона № 273 от 29.12.2012 «Об образовании в Российской Федерации»;
- Концепции развития дополнительного образования детей (Распоряжение Правительства РФ от 31 марта 2022 года № 678-р);
- Распоряжения Правительства РФ от 29.05.2015 № 996-р «Стратегия развития воспитания в РФ на период до 2025 года»;
- Приказа Министерства просвещения Российской Федерации от 27 июля 2022 года № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Целевой модели развития региональных систем дополнительного образования детей (Приказ Министерства Просвещения РФ от 3 сентября 2019 года № 467 «Об утверждении целевой модели развития региональных систем дополнительного образования»);
- Письма Министерства образования и науки РФ N 09-3242 от 18 ноября 2015 г. «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»;
- Постановления Главного государственного санитарного врача Российской Федерации 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»;
- Постановления Правительства № 527-п 17.07.2018 (в редакции постановления Правительства области от 15.04.2022 г. № 285-п) Концепция персонифицированного дополнительного образования детей в Ярославской области;
- Приказа департамента образования ЯО от 27.12.2019 №47-нп Правила персонифицированного финансирования ДОД.

Актуальность программы.

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, не только на пользовательском уровне, но и на уровне начинающего программиста. В школьном курсе информатики программирование нередко представлено лишь на элементарном уровне. Следствием этого является формальное восприятие

обучающимися основ современного программирования и неумение применять полученные знания на практике.

Педагогическая целесообразность программы заключается в том, что дети приобретут практические навыки, которые станут основой для дальнейшего изучения основ программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации обучающихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Осваивая данную программу, обучающиеся будут овладевать навыками, которые будут востребованы в ближайшие десятилетия в специальностях, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, получаемые в процессе обучения по программе.

Отличительная особенность программы состоит в том, что она позволяет привлечь детей среднего школьного возраста к углубленному изучению программирования при помощи языка Python, так как он обладает следующими достоинствами:

- Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разных программ, от текстовых процессоров до веб-браузеров;
- Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python совсем не сложно;
- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;
- Python используется как язык программирования крупными корпорациями, такими как Google.

Адресат программы. Данная образовательная программа разработана для работы с обучающимися от 12 до 18 лет. Программа не адаптирована для обучающихся с ОВЗ.

Объем и срок освоения программы. Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» рассчитана на 2 года обучения.

Режим занятий: 2 раза в неделю по 2 академических часа (по 45 минут) с 10-минутным перерывом, 144 академических часа в год.

В каникулярное время занятия проводятся в соответствии с календарным учебным графиком, допускается изменение форм занятий, проведение воспитательных мероприятий.

Занятия проводятся в кабинете, оборудованном согласно санитарно-эпидемиологическим требованиям. Форма обучения – очная, с использованием дистанционных технологий, ИКТ. Форма занятий - индивидуальная.

Наполняемость групп: 8 – 12 человек.

Цель программы: развитие алгоритмического и логического мышления посредством языка программирования «Python».

Задачи обучения:

Обучающие:

- сформировать у детей представление об основных элементах программирования;
- познакомить с синтаксисом языка программирования Python;
- сформировать у детей навыки работы в интегрированной среде разработки на языке Python;
- способствовать приобретению навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python у детей.

Развивающие:

- совершенствовать аналитические навыки;
- формировать навык алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование информации при решении задач;
- развивать умение планировать свои действия с учётом фактора времени;

Воспитательные:

- воспитывать в детях усидчивость, аккуратность, умение доводить начатое дело до конца;
- формировать коммуникативные навыки.

Условия реализации программы

Материально-техническое обеспечение программы:

- столы, стулья (по росту и количеству детей);
- демонстрационная магнитная доска, флипчарт;
- моноблочное интерактивное устройство;
- ноутбуки (12 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО.
- компьютерные мыши.

Кадровое обеспечение:

Программу реализует педагогический работник – педагог дополнительного образования.

Организация образовательного процесса

При организации обучения используется дифференцированный, индивидуальный подход. На занятиях используются следующие методы обучения:

- объяснительно-иллюстративный (беседы, объяснения, дискуссии);
- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);
- метод проблемного изложения;
- эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);
- исследовательский.

Образовательная программа содержит теоретическую и практическую подготовку, большее количество времени уделяется выработке практических навыков.

Формы организации образовательного процесса:

- фронтальная – подача материала всей учебной группе обучающихся;
- индивидуальная – самостоятельная работа обучающихся с оказанием педагогом помощи при возникновении затруднения;
- групповая – предоставление учащимся возможности самостоятельно построить свою деятельность, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности.

Формы занятий: комбинированные, практическая работа, соревнование, дискуссии, беседы, лекции.

Формы организации учебного занятия:

- *вводное занятие* – педагог знакомит обучающихся с техникой безопасности, особенностями организации деятельности и предлагаемым планом работы на текущий год;
- *ознакомительное занятие* – педагог знакомит обучающихся с новыми методами работы в зависимости от темы занятия;
- *тематическое занятие* – на котором детям предлагается работать над моделированием по определенной теме. Занятие содействует развитию творческого воображения обучающихся;
- *занятие-проект* – на таком занятии обучающиеся получают полную свободу в выборе направления работы, не ограниченного определенной тематикой. Обучающиеся, участвующие в работе по выполнению предложенного задания, рассказывают о выполненной работе, о ходе выполнения задания, о назначении выполненного проекта;

- *конкурсное игровое занятие* строится в виде соревнования для повышения активности обучающихся и их коммуникации между собой;
- *комбинированное занятие* проводится для решения нескольких учебных задач;
- *итоговое занятие* служит подведению итогов работы за учебный год.

Формы подведения итогов реализации программы

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: промежуточный и итоговый результат обучения детей.

Промежуточный контроль проводится в конце первого полугодия.

Итоговый контроль проводится в конце года, с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения.

Мониторинг обучающихся может проходить на итоговом занятии в форме презентации своего проекта.

Планируемые результаты

К концу года обучения, обучающиеся будут *знать*:

- основные классические алгоритмы и способы их реализации;
- синтаксис языка программирования Python;
- простые и сложные структуры данных, а также конструкции для работы с ними;
- некоторые модули (turtle, random, tkinter и др.);
- основные элементы программирования: ввод (вывод) информации в память (из памяти), данные, операции с данными, условное выполнение, циклы, подпрограммы.

будут *уметь*:

- разбивать решение задачи на подзадачи, то есть составлять алгоритм;
- объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- искать и обрабатывать ошибки в коде;
- писать грамотный, красивый код;
- анализировать как свой, так и чужой код;
- импортировать модули в программу;
- работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- грамотно строить коммуникацию, исходя из целей и ситуации;
- работать с вычислительной техникой.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

1 года обучения

Год обучения	Дата начало обучения по программе	Дата начало окончания по программе	Всего учебных недель	Количество учебных процессов	Режим занятий
1 год	1 сентября	31 мая	36	144	2 раза в неделю По 2 часа

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
1	Введение в язык программирования Python	32	14	18
2	Основы языка Python	64	38	26
3	Игры на Python	40	12	28
4	Аттестация	8	-	8
	Итого	144	64	80

3. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

2 год обучения

Год обучения	Дата начало обучения по программе	Дата начало окончания по программе	Всего учебных недель	Количество учебных процессов	Режим занятий
2 год	1 сентября	31 мая	36	144	2 раза в неделю По 2 часа

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
1	Библиотека PyQT. Создание десктопных приложений	32	14	18
2	Библиотека PyGame. Создание игр с помощью PyGame	64	38	26
3	Библиотека Flask. Основы создания web приложений	40	12	28
4	Аттестация	8	-	8
	Итого	144	64	80

4. СОДЕРЖАНИЕ ПРОГРАММЫ

1 года обучения

Раздел 1. Введение в язык программирования Python (32 часа)

Тема 1.1. Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ (4 часа)

Теория (2 часа)

Техника безопасности на занятии. Понятие «алгоритм», «исполнитель», «язык программирования», «программа», «интерпретатор». История языка программирования Python и его возможности. Виды окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей.

Практика (2 часа)

Сохранение и запуск python-программ в среде разработки IDLE.

Тема 1.2. Вывод данных на языке Python. Ввод и отладка программ в среде (4 часа)

Теория (2 часа)

Правила синтаксиса Python: правило начала, правило порядка, правило регистра. Понятие функции. Функция print().

Практика (2 часа)

Проект «Символьная графика». Создание определённого рисунка с помощью символов. Отработка функции print().

Тема 1.3. Переменные (4 часа)

Теория (2 часа)

Понятие «переменная». Правила именования переменных в языке Python. Оператор присваивания. Функция input().

Практика (2 часа)

Проект «Аватар». В данном проекте отрабатывается функция `input()`, с помощью которой становится возможным ввести свои данные в программу и отобразить их.

Тема 1.4. Выражения. Типы данных (4 часа)

Теория (2 часа)

Арифметические операции с помощью математических операторов `+`, `-`, `*`, `/`. Порядок выполнения операций. Понятие «выражение», «типы данных». Функции `int()` и `str()`.

Практика (2 часа)

Проект «Сумматор». При написании данной программы отрабатываются математические операторы и функции `int()` и `str()`.

Тема 1.5. Строки и списки (4 часа)

Теория (2 часа)

Понятие «строка». Создание строк. Переменные внутри строк. Операции со строками. Понятие «список». Создание списков. Добавление/удаление элементов в/из список/списка. Операции со списками.

Практика (2 часа)

Решение задач на отработку операций со строками и списками.

Тема 1.6. Кортежи и словари (6 часов)

Теория (2 часа)

Понятие «кортеж». Создание кортежа. Операции с кортежем. Понятие «словарь». Создание словаря.

Практика (4 часа)

Проект «Любимые вещи»: создание списка любимых развлечений и любимых лакомств.

Тема 1.7. Рисование с помощью «Черепашей графики» (6 часов)

Теория (2 часа)

Понятие «модуль». Модуль turtle. Импортирование модуля. Создание холста. Перемещение черепашки.

Практика (4 часа)

Проект «Я рисую»: рисование изображения, состоящего из линий с помощью команд модуля на холсте.

Раздел 2. Основы языка Python (64 часа)

Тема 2.1. Условный оператор (4 часа)

Теория (2 часа)

Понятие «условный оператор», «вложенные команды», «оператор сравнения». Конструкция if и её синтаксис. Операторы сравнения: <, >, >=, <=, !=, ==. Структура программы. Конструкция if-else. Команды if и elif.

Практика (2 часа)

Решение задач на отработку условного оператора и операторов сравнения.

Тема 2.2. Объединение условий (4 часа)

Теория (2 часа)

Логические операторы: and, or, not. Порядок выполнения операций. Переменные без значения – None.

Практика (2 часа)

Проект «Калькулятор»: создание приложения по определенным условиям.

Тема 2.3. Использование цикла for (4 часа)

Теория (2 часа)

Понятие «цикл», «цикл со счётчиком». Конструкция for и её синтаксис.

Практика (2 часа)

Проект «Таблица умножения»: создание приложения по определенным условиям.

Тема 2.4. Использование цикла *while* (4 часа)

Теория (2 часа)

Понятие «цикл с предусловием». Конструкция *while* и её синтаксис. Заикливание и выход из цикла с помощью команды *break*.

Практика (2 часа)

Проект «Бомба взорвалась!». Написание программы по определенным условиям.

Тема 2.5. Решение задач на циклические алгоритмы (4 часа)

Теория (2 часа)

Виды циклов и их конструкции.

Практика (2 часа)

Решение задач на применение циклов *for* и *while*.

Тема 2.6. Применение функций (4 часа)

Теория (2 часа)

Понятие «функция», «параметр функции», «значение функции». Строение функции: имя, аргумент, тело. Создание и вызов функции. Переменные и область видимости.

Практика (2 часа)

Решение задач на отработку понятия «функция», её строение и синтаксис.

Тема 2.7. Использование встроенных функций (4 часа)

Теория (2 часа)

Функции: *abs*, *bool*, *dir*, *eval*, *exec*, *float*, *int*, *len*, *max*, *min*, *range*, *sum*.

Практика (2 часа)

Решение задач на отработку функций.

Тема 2.8. Применение модулей (4 часа)

Теория (2 часа)

Понятие «модуль». Импортирование модуля в программу. Полезные модули: random, time, pickle.

Практика (2 часа)

Решение задач на применение модулей.

Тема 2.9. Классы и объекты (4 часа)

Теория (2 часа)

Понятие «объект». Концепция объектов. Понятие «класс». Использование классов в Python. Инициализация объектов.

Практика (2 часа)

Решение задач на отработку умения создавать классы и объекты.

Тема 2.10. События (6 часов)

Теория (2 часа)

Понятие «событие», «подписка на событие», «обработка события».

Практика (4 часа)

Проект «Поймай черепашку». Пишется приложение-игра, в котором создается холст и на нём прорисовывается черепашка, при нажатии на которую происходит ее перемещение в новое место на холсте.

Тема 2.11. Работа с файлами (6 часов)

Теория (2 часа)

Понятие «файл». Классификация файлов в зависимости от видов информации. Создание текстового файла. Открытие файла в Python. Запись в файл.

Практика (4 часа)

Проект «Блокнот»: создание простейшего текстового редактора в виде приложения.

Тема 2.12. Графика с модулем PIL (8 часов)

Теория (2 часа)

Рисование с использованием модуля PIL.

Практика (6 часов)

Проект «Фантастический мир»: рисование фигур, параметры которых задаются пользователем при запуске программы.

Тема 2.13. Графика с модулем tkinter (8 часов)

Теория (2 часа)

Модуль tkinter и его возможности.

Практика (6 часов)

Проект «Анимация»: отрисовывание объекта на холсте и программирование анимации.

Раздел 3. Игры на Python (40 часов)

Тема 3.1. Игра «Прыг-скок!» (12 часов)

Теория (4 часа)

Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка мяча. Программирование мяча: перемещение, отскоки от границ игрового холста, первоначальное направление движения мяча. Отрисовка ракетки. Программирование ракетки: управление, установления границ. Обработка события столкновения мяча и ракетки. Обработка события «проигрыш». Доработка игры: задержка перед началом игры; экран «Конец игры»; ускорение мяча; счет в игре.

Практика (8 часов)

Создание приложения-игра «Прыг-скок!»: неуправляемый пользователем объект «мяч» и управляемые пользователем объект «ракетка».

Тема 3.2. Игра «Крестики-нолики» (14 часов)

Теория (4 часа)

Обсуждение сюжета и правил игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Реализация хода соперника. Доработка игры: задержка перед началом игры; счет в игре и его отображение; создание лимита времени; экран «Конец игры».

Практика (10 часов)

Создание приложения-игра «Крестики-нолики»: реализовать выбор значка игрока, его хода и хода соперника. Задача пользователя: выиграть соперника в ходе игры.

Тема 3.3. Игра «Тетрис» (14 часов)

Теория (4 часа)

Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка падающих фигур. Реализация стирания линии, зачисление счета. Отрисовка экрана проигрыша.

Практика (10 часов)

Создание приложения-игра «Тетрис». Смысл игры: из падающих фигур составлять линии. При заполнении линии - она исчезает. За это игрок получает очки. Цель игры – набрать максимальное количество.

Раздел 4. Аттестация (8 часов)

Практика (8 часов)

Разработка собственного приложения-игры с сюжетом. Условия игры: обязательно должен присутствовать управляемый пользователем персонаж, должна быть прорисована и запрограммирована игровая обстановка.

5. СОДЕРЖАНИЕ ПРОГРАММЫ

2 год обучения

Раздел 1. Библиотека PyQT. Создание десктопных приложений

Тема 1.1. Классы. Повторение (4 часа)

Теория. Повторение объектов “классы”. Наследование, полиморфизм, инкапсуляция.

Практика. Написание классов для различных объектов.

Тема 1.2. Что такое QT и PyQT. Знакомство (4 часа)

Теория. Понятие десктопного приложения. Описание модуля PyQT. Взаимосвязь с классами.

Практика. Создание базового приложения, с помощью модуля PyQT.

Тема 1.3. QtDesigner, pyuic, два способа подключения uic-файла (4 часа)

Теория. Использование uic-файлов. Интерфейс QtDesigner.

Практика. Создание базового приложения, с помощью QtDesigner. Конвертация uic-файлов.

Тема 1.4. Обработка событий (4 часа)

Теория. Понятие события. Виды событий. Обработка событий

Практика. Обработка событий нажатия кнопок мыши и клавиатуры, кнопок приложения.

Тема 1.5. Работа с файлами (4 часа)

Теория. Описание используемых файлов в PyQT.

Практика. Применение файлов word, excel в десктопном приложении.

Тема 1.6. Диалоги, работа с изображениями (4 часа)

Теория. Понятие диалога. Виды диалогов. Использование изображений в PyQT.

Практика. Применение диалогов, вставка изображений в десктопное приложение.

Тема 1.7. Работа с базами данных. Основы SQL (8 часов)

Теория. Понятие базы данных. Основы языка SQL. Взаимосвязь с приложением.

Практика. Загрузка данных из базы данных в приложение. Использование языка SQL.

Раздел 2. Библиотека PyGame. Создание игр с помощью PyGame

Тема 2.1. Создание игрового окна. Игровой цикл. (6 часа)

Теория. Понятие игрового цикла. Описание модуля PyGame. Принцип работы приложения.

Практика. Создание игрового окна, с помощью модуля PyGame.

Тема 2.2. Рисование. Анимация объектов. FPS. Время (6 часа)

Теория. Виды рисуемых фигур в модуле PyGame. Способы задания анимации. Понятие FPS и времени в игре

Практика. Создание приложения с анимацией различных фигур.

Тема 2.3. Надписи. Рендер (4 часа)

Теория. Использование надписей в играх. Шрифты, цвет, размеры надписей. Рендер надписей.

Практика. Добавление различных надписей в игру.

Тема 2.4. Спрайты. Создание, рисование спрайтов. Взаимодействие объектов (8 часа)

Теория. Понятие спрайта. Способы создания, отрисовки спрайтов. Столкновение спрайтов.

Практика. Создание игры с использованием спрайтов.

Тема 2.5. Работа с файлами. Загрузка спрайтов. Отрисовка готовых спрайтов (4 часа)

Теория. Загрузка спрайтов из файлов.

Практика. Добавление готовых спрайтов в игру. Анимация спрайтов.

Тема 2.6. Игры на клетчатом поле. Наложение фонов (6 часа)

Теория. Виды игр на клетчатом поле. Примеры. Применение наложения фонов.

Практика. Использование наложения фонов в игре. Создание заготовки для игр на клетчатом поле.

Тема 2.7. Стартовое меню. Меню окончания. Запись рекордов (4 часа)

Теория. Способы добавления стартового меню и меню окончания игры. Способы записи рекордов игрока

Практика. Добавление стартового меню и меню окончания игры. Запись рекордов игрока.

Тема 2.8. Добавление музыки, звуков в проект (4 часа)

Теория. Способы добавления музыки и звуков в игру.

Практика. Добавление музыки и звуков в игру.

Тема 2.9. Игра “Змейка” (6 часа)

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка клетчатого поля. Отрисовка объектов. Хранение рекордов. Отрисовка экрана проигрыша.

Практика. Приложение-игра «Змейка». Смысл игры: змейка поедает фрукты и получает очки. При этом размер змейки увеличивается. Игра заканчивается, когда змейка врежется в край игрового поля, либо в саму себя.

Тема 2.10. Игра “Flappy bird” (6 часа)

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка поля. Отрисовка объектов. Хранение рекордов. Отрисовка экрана проигрыша.

Практика. Приложение-игра «Flappy bird ». Смысл игры: на игровом поле бесконечно летит птица и перелетает через препятствие. Игра заканчивается при ударе о препятствие, либо о край игрового поля.

Тема 2.11. Игра “Тетрис” (6 часа)

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка падающих фигур. Реализация стирания линии, зачисление счета. Отрисовка экрана проигрыша.

Практика. Приложение-игра «Тетрис». Смысл игры: из падающих фигур составлять линии. При заполнении линии - она исчезает. За это игрок получает очки. Цель игры – набрать максимальное количество.

Тема 2.12. Игра “Крестики-нолики”. Игра “Сапер” (8 часа)

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры.

Практика. Реализация классической игры “крестики-нолики”.

Тема 2.13. Создание лабиринтов (4 часа)

Теория. Способы создания лабиринтов. Поиск пути их решения.

Практика. Отрисовка лабиринтов различными способами.

Раздел 3. Библиотека Flask. Основы создания web приложений

Тема 3.1. Простейшее приложение на Flask. Понятие URL. Создание локального сервера. (4 часа)

Теория. Описание модуля Flask. Понятие сервер. Понятие URL.

Практика. Создание простейшего приложения на Flask.

Тема 3.2. Знакомство с html. Шаблоны. Модуль jinja2. (6 часов)

Теория. Основы языка html. Рендер html шаблонов. Использование шаблонов jinja2 в html файлах

Практика. Написание простейших html файлов. Применение шаблонов jinja2. Рендер шаблонов.

Тема 3.3. Создание сайта на Flask. Методы запросов. Взаимодействие с базами данных (8 часов)

Теория. Понятие методов запроса. Виды методов. Понятие тела запроса. Создание, подключение базы данных. Получение данных.

Практика. Создание приложения на Flask с использованием базы данных. Использование различных методов в приложении.

Тема 3.4. Модуль SQLAlchemy. Модуль marshmallow. Сериализация (6 часов)

Теория. Получение данных из базы данных, с помощью модуля SQLAlchemy. Понятие сериализации, с помощью модуля marshmallow.

Практика. Создание приложения на Flask с использованием SQLAlchemy. Вывод данных, с помощью модуля marshmallow.

Тема 3.4. Создание сайта с просмотром, добавлением постов (8 часов)

Теория. Обсуждение контекста сайта, его функционала и возможностей.

Практика. Создание сайта, с применением изученных навыков web разработки. Описание сайта: просмотр постов пользователя. Посты можно добавлять, удалять.

Раздел 4. Аттестация (8 часов)

Практика. Разработка собственного проекта на выбор: десктопное приложение, игра, сайт. Проект создается на основе изученных навыков.

6. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

1 года обучения

№ п/п	Наименование разделов, тем	Формы занятий	Формы контроля	Обеспечения
1	Вводное занятие	Беседа		Инструкция по ТБ. Игры на знакомство. ПО Python видеоролики.
2	Основы программирования на языке Python	Беседы Практические занятия	тестирование, контрольные задания, устный опрос, решение задач	ПО Python, видеоролики, тесты, викторина, упражнения
3	Игры на Python	Беседа, практические занятия, защита проектов	Индивидуальные и групповые проекты	ПО Python, тесты, проекты
4	Аттестация. Защита проекта.	Защита проекта	Защита проекта	ПО Python

В образовательном процессе используются следующие **методы**:

- объяснительно - иллюстративный;
- метод проблемного изложения (постановка проблемы и решение ее самостоятельно или группой);
- проектно - исследовательский;
- наглядный: демонстрация схем, таблиц, диаграмм; использование технических средств; просмотр видеороликов;
- практический: практические задания; анализ и решение проблемных ситуаций и т. д

Выбор методов обучения осуществляется исходя из анализа уровня готовности учащихся к освоению содержания раздела, степени сложности материала, типа учебного

занятия. На выбор методов обучения значительно влияет персональный состав группы, индивидуальные особенности, возможности и запросы детей.

Формы обучения:

фронтальная - предполагает работу педагога сразу со всеми учащимися в едином темпе и с общими задачами.

групповая - предполагает, что занятия проводятся с подгруппой. Для этого группа распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

индивидуальная - подразумевает взаимодействие преподавателя с одним обучающимся. Как правило, данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится фронтально, затем учащийся выполняет индивидуальные задания или общие задания в индивидуальном темпе.

7. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

2 года обучения

№ п/п	Наименование разделов, тем	Формы занятий	Формы контроля	Обеспечения
1	Вводное занятие Библиотека PyQt. Создание десктопных приложений	Беседа		Инструкция по ТБ. Игры на знакомство. ПО Python видеоролики.
2	Библиотека PyGame. Создание игр с помощью PyGame	Беседы Практические занятия	тестирование, контрольные задания, устный опрос, решение задач	ПО Python, видеоролики, тесты, викторина, упражнения
3	Библиотека Flask. Основы создания web приложений	Беседа, практические занятия, защита проектов	Индивидуальные и групповые проекты	ПО Python, тесты, проекты
4	Аттестация. Защита проекта.	Защита проекта	Защита проекта	ПО Python

В образовательном процессе используются следующие **методы**:

- объяснительное-иллюстративный;
- метод проблемного изложения (постановка проблемы и решение ее самостоятельно или группой);
- проектно-исследовательский;
- наглядный: демонстрация схем, таблиц, диаграмм; использование технических средств; просмотр видеороликов;
- практический: практические задания; анализ и решение проблемных ситуаций и т. д

Выбор методов обучения осуществляется исходя из анализа уровня готовности учащихся к освоению содержания раздела, степени сложности материала, типа учебного занятия. На выбор методов обучения значительно влияет персональный состав группы, индивидуальные особенности, возможности и запросы детей.

Формы обучения:

Фронтальная - предполагает работу педагога сразу со всеми учащимися в едином темпе и с общими задачами.

групповая - предполагает, что занятия проводятся с подгруппой. Для этого группа распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

индивидуальная - подразумевает взаимодействие преподавателя с одним обучающимся. Как правило, данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится фронтально, затем учащийся выполняет индивидуальные задания или общие задания в индивидуальном темпе.

8. ВОСПИТАТЕЛЬНАЯ ДЕЯТЕЛЬНОСТЬ

1. Цель

Развитие личности, самоопределение и социализация детей на основе социокультурных, духовно-нравственных ценностей и принятых в российском обществе правил и норм поведения в интересах человека, семьи, общества и государства, формирование чувства патриотизма, гражданственности, уважения к памяти защитников Отечества и подвигам Героев Отечества, закону и правопорядку, человеку труда и старшему поколению, взаимного уважения, бережного отношения к культурному наследию и традициям многонационального народа Российской Федерации, природе и окружающей среде (Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», ст. 2, п. 2).

Задач воспитания по программе являются:

- усвоение детьми знаний норм, духовно-нравственных ценностей, традиций танцевальной культуры; информирование детей, организация общения между ними на содержательной основе целевых ориентиров воспитания;
- приобретение детьми опыта поведения, общения, межличностных и социальных отношений в составе учебной группы, применение полученных знаний, организация детей, их ответственного поведения, создание, поддержка и развитие среды воспитания детей, условий безопасности, комфорта, и о общения, социализации, признания, самореализации, творчества при освоении программирование и метапредметного содержания программы.

Целевые ориентиры воспитания детей по программе:

- освоение детьми понятия о своей российской культурной принадлежности (идентичности);
- принятие и осознание ценностей языка, литературы, музыки, программирования, традиций, праздников;
- воспитание уважения к жизни, достоинству, свободе каждого человека, понимания ценности жизни, здоровья и безопасности (своей и других людей);
- воспитание уважение к труду, результатам труда, уважения к старшим;
- развитие творческого самовыражения в программировании, реализация традиционных и своих собственных представлений.

2. Формы и методы воспитания

Решение задач информирования детей, создания и поддержки воспитывающей среды общения и успешной деятельности, формирования межличностных отношений на основе

российских традиционных духовных ценностей осуществляется на каждом из учебных занятий.

В воспитательной деятельности с детьми по программе используются методы воспитания: метод убеждения (рассказ, разъяснение, внушение), метод положительного примера (педагога и других взрослых, детей); метод упражнений (приучения); методы одобрения и осуждения поведения детей, педагогического требования (с учётом преимущественного права на воспитание детей их родителей (законных представителей), и стимулирования, поощрения (индивидуального и публичного); метод переключения в деятельности; методы руководства и самовоспитания, развития самоконтроля и самооценки детей в воспитании; методы воспитания воздействием группы.

3. Условия воспитания, анализ результатов

Воспитательный процесс осуществляется на учебной базе реализации программы дополнительного образования детей в соответствии с нормами и правилами работы организации.

Анализ результатов воспитания проводится в процессе педагогического наблюдения за поведением детей, их общением, отношениями детей друг с другом, их отношением к педагогам, к выполнению своих заданий по программе. Косвенная оценка результатов воспитания, достижения целевых ориентиров воспитания по программе проводится путём опросов родителей в процессе реализации программы (отзывы родителей) и после её завершения (итоговые исследования результатов реализации программы за учебный период, учебный год).

Анализ результатов воспитания по программе не предусматривает определение персонифицированного уровня воспитанности, развития качеств личности конкретного ребёнка, обучающегося, а получение общего представления о воспитательных результатах реализации программы, продвижения в достижении определённых в программе целевых ориентиров воспитания, влияния реализации программы на обучающихся: что удалось достичь, а что является предметом воспитательной работы в будущем. Результаты, полученные в ходе оценочных процедур - опросов, используются только в виде агрегированных усреднённых и анонимных данных.

Календарный план воспитательной работы

№ п/п	Название события, мероприятия	Сроки	Форма проведения	Практический результат и информационный продукт, иллюстрирующий успешное достижение цели события
1	День пожилого человека	1 октября	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся
2	День учителя	5 октября	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся
3	Новый год	31 декабря	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся
4	День Защитника Отечества	23 февраля	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся
5	Международный женский день	8 марта	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся
6	День космонавтики	12 апреля	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся
7	День Победы	9 мая	Участие в дистанционном конкурсе	Дипломы и сертификаты обучающихся

9. СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Марк Лутц «Изучаем Python» (5-е издание, 2022) Классика, подробное руководство по Python 3.
2. Бизли Д., Джонс Б. «Python. Книга рецептов» (3-е издание, 2020) Практические примеры и решения задач на Python 3.
3. Луסיано Рамальо «Python. К вершинам мастерства» (2-е издание, 2022) Продвинутые техники и лучшие практики Python.
4. Михал Рысь «Python. Разработка на основе тестирования» (2021) TDD (Test-Driven Development) в Python.
5. Адитья Бхаргава «Грокаем алгоритмы» (2-е издание, 2023) Алгоритмы и структуры данных с примерами на Python.
6. Разбор сложных тем (асинхронность, ООП, алгоритмы). «Stepik - Python для начинающих»

*Сводная диагностическая таблица
по дополнительной общеобразовательной общеразвивающей
программе «Программирование Python» за учебный год 2025-2026 учебный год*

Наименование группы _____

№	Ф.И.О ребёнка	1. Теоретическая подготовка Обучающийся:	2. Практическая подготовка ребёнка	
		Теоретические знания (по основным разделам образовательной программы)	Практические умения и навыки, предусмотренные программой (по основным разделам образовательной программы)	Творческие навыки

Минимальный уровень - 3 балла.

Средний уровень - 4 балла.

Максимальный уровень – 5 баллов.