



## Содержание

1. Пояснительная записка.....	3
2. Учебно-тематический план .....	5
3. Содержание программы .....	5
4. Методическое обеспечение .....	7
5. Материально-техническое обеспечение.....	7
6. Контрольно-измерительный материал.....	7
7. Кадровое обеспечение.....	8
8. Список литературы.....	8

## Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» разработана согласно требованиям следующих нормативно-правовых документов:

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

- Концепция развития дополнительного образования детей до 2030 года (Распоряжение Правительства РФ от 31 марта 2022 г. № 678-р).

- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства Российской Федерации от 29 мая 2015 г. № 996-р).

- Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

- Приказ Министерства просвещения РФ от 3 сентября 2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей».

- Санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» Утверждены постановлением Главного государственного санитарного врача РФ от 28.09.2020 года N 28.

- Правила персонифицированного финансирования дополнительного образования детей в Ярославской области, утв. приказом департамента образования ЯО от 27.12.2019 №47-нп.

- Положение о персонифицированном дополнительном образовании детей в городе Ярославле, утв. постановлением мэрии города Ярославля 11.04.2019 года № 428.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» способствует техническому, интеллектуальному развитию обучающихся, формируя основы компьютерной, информационной грамотности.

Программа направлена на освоение основ программирования и развитие компетенций, обучающихся в области информационных технологий посредством изучения одного из наиболее востребованных современных языков - Python. Данный язык является универсальным инструментом, применимым во множестве сфер деятельности от веб-разработки и анализа данных до автоматизации процессов.

**Актуальность** данной программы обусловлена высокой востребованностью специалистов, владеющих языком Python, который широко применяется в различных областях - от разработки программного обеспечения и анализа данных до веб-разработки. Простота и эффективность языка позволяют быстро освоить основы программирования даже новичкам, открывая широкие возможности для профессиональной реализации. Обучение Python развивает ключевые компетенции, необходимые для успешной адаптации в цифровом пространстве, обеспечивая прочную базу для карьерного роста в динамично развивающейся индустрии информационных технологий.

**Направленность:** техническая.

**Вид программы:** модифицированная.

**Категория обучающихся:** 12-16 лет.

**Срок реализации:** 1 год.

**Количество часов:** 72 часа.

**Количество обучающихся в группе:** 8 человек (согласно количеству ноутбуков).

Набор обучающихся в объединение производится по их желанию без предварительного конкурсного отбора.

**Календарный учебный график:**

- продолжительность реализации программы: сентябрь - май;
- количество учебных недель: 36;
- режим занятий: 1 раз в неделю;
- продолжительность занятия: 2 академических часа.

**Форма обучения:** очная.

**Форма работы:** групповая.

### Цель и задачи программы

**Цель:** формировать умения и навыки обучающихся программированию на языке Python.

**Задачи:**

**Обучающие:**

- обучить основам синтаксиса и структуры языка Python;
- обучить основам конструкций программирования (условия, циклы, функции);
- обучить приемам написания простых программ и понимания процесса разработки

ПО;

- обучить технологии применения Python в реальных проектах.

**Развивающие:**

- развивать алгоритмическое мышление и умение логически мыслить;
- развивать творческие способности путем самостоятельного проектирования небольших программных продуктов;
- развивать коммуникативные навыки через взаимодействие в команде над совместными проектами;
- развивать внимание и аккуратность при написании кода.

**Воспитательные:**

- формировать уважение к интеллектуальному труду и ответственность за качество выполняемых работ;
- формировать дисциплинированность и организованность в процессе самостоятельной работы;
- формировать чувства гордости за личные достижения и стремление к самосовершенствованию;
- формировать положительное отношение к профессиям в сфере информационных технологий и понимать важность непрерывного образования.

### Ожидаемые результаты

**В конце обучения обучающиеся должны**

**знать:**

- основы синтаксиса и структуры языка Python;
- типичные конструкции и методы работы с переменными, функциями, условиями и циклами;
- принципы организации и выполнения программных модулей;
- возможности и инструменты среды разработки Python;
- особенности чтения и понимания кода;
- правила эффективного оформления и комментирования своего кода.

**уметь:**

- создавать простые программы на языке Python, решающие конкретные задачи;
- работать с различными типами данных (числа, строки, списки, словари), осуществлять ввод-вывод информации;
- использовать стандартные библиотеки и для расширения функционала собственных программ;

- проводить отладку и тестирование разработанного программного продукта;
- анализировать алгоритмы и выбирать оптимальные подходы к решению поставленных задач;
- сотрудничать в группе при выполнении коллективных проектов, распределять обязанности между участниками команды;
- представлять итоговую работу, пояснять ход её выполнения.

### Учебно-тематический план

№ п/п	Название темы	Количество часов		
		Теория	Практика	Всего
1.	Инструктаж по ТБ. Введение в программирование и компьютерные технологии.	1	1	2
2.	Первая программа на Python: «Hello, world!»	1	1	2
3	Работа с числами и математическими выражениями.	1	1	2
4.	Изучение переменных и типов данных.	1	1	2
5.	Строки.	2	2	4
6.	Списки.	2	2	4
7.	Кортежи.	2	2	4
8.	Словари.	2	2	4
9.	Условия и ветвления: делаем выбор в программе.	1	1	2
10.	Конструкция if.	2	2	4
11.	Команды if и elif.	2	2	4
12.	Циклы: повторяем действия многократно.	1	1	2
13.	Использование цикла for.	2	2	4
14.	Цикл while.	2	2	4
15.	Списки и последовательности: работа с наборами данных.	2	2	4
16.	Графика.	2	2	4
17.	Использование модуля черепашки.	2	2	4
18.	Рисуем фигуры.	2	2	4
19.	Разработка игр.	1	5	6
20.	Проектная деятельность.	0	4	4
21.	Итоговое занятие.	0	2	2
<b>Итого:</b>		<b>31</b>	<b>41</b>	<b>72</b>

### Содержание программы

**Тема 1. Инструктаж по ТБ. Введение в программирование и компьютерные технологии.**

**Теория:** Что такое программирование, языки программирования. Компьютерная архитектура и её связь с программированием.

**Практика:** Знакомство со средами разработки программ.

**Тема 2. Первая программа на Python: «Hello, world!»**

**Теория:** Освоение синтаксиса и создание первой программы на Python.

**Практика:** Написание и запуск программы «Hello, world!».

**Тема 3. Работа с числами и математическими выражениями.**

**Теория:** Использование чисел и арифметических операций в Python.

**Практика:** Простые вычисления и выражение математических операций.

**Тема 4. Изучение переменных и типов данных.**

**Теория:** Понять значение переменных и изучить основные типы данных.  
**Практика:** Присваивание значений переменным и выполнение простых операций.

#### **Тема 5. Строки.**

**Теория:** Работа со строками, их свойства и способы манипуляций.

**Практика:** Создание и обработка строк.

#### **Тема 6. Списки.**

**Теория:** Понятие списков и их назначение в Python.

**Практика:** Создание списков, их изменение и обработка.

#### **Тема 7. Кортежи.**

**Теория:** Узнать о кортеже и отличие от списка.

**Практика:** Использование кортежей в программах.

#### **Тема 8. Словари.**

**Теория:** Научиться использовать словари для хранения связанных данных.

**Практика:** Создание и применение словарей.

#### **Тема 9. Условия и ветвления: делаем выбор в программе.**

**Теория:** Логика ветвления и принятие решений в программах.

**Практика:** Применять конструкцию if для выполнения разных путей выполнения программы.

#### **Тема 10. Конструкция if.**

**Теория:** Изучаем основную структуру if.

**Практика:** Выполнить задания с простыми условиями.

#### **Тема 11. Команды if и elif.**

**Теория:** Рассмотреть дополнительные варианты выбора с помощью elif.

**Практика:** Упражнения с разными путями выполнения программы.

#### **Тема 12. Циклы: повторяем действия многократно.**

**Теория:** Изучить принцип цикличности в Python.

**Практика:** Применение циклов для автоматизации процессов.

#### **Тема 13. Использование цикла for.**

**Теория:** Применение цикла for для обхода последовательностей.

**Практика:** Реализовать задачи с использованием цикла for.

#### **Тема 14. Цикл while.**

**Теория:** Когда и зачем использовать цикл while.

**Практика:** Написать программу с использованием цикла while.

#### **Тема 15. Списки и последовательности: работа с наборами данных.**

**Теория:** Учимся обрабатывать наборы данных в Python.

**Практика:** Работать с множествами и списками данных.

#### **Тема 16. Графика.**

**Теория:** Основы графической работы в Python.

**Практика:** Вывод простейших графических примитивов.

#### **Тема 17. Использование модуля черепашки.**

**Теория:** Знакомство с библиотекой Turtle.

**Практика:** Рисование простых фигур с помощью черепашьей графики.

#### **Тема 18. Рисуем фигуры.**

**Теория:** Работа с координатами и перемещением черепахи.

**Практика:** Создание более сложных графических композиций.

#### **Тема 19. Разработка игр.**

**Теория:** Простейшие концепции игровых движков.

**Практика:** Создание маленькой интерактивной игры на Python.

#### **Тема 20. Проектная деятельность.**

**Практика:** Завершить собственный учебный проект, применяя всё изученное.

#### **Тема 21. Итоговое занятие.**

**Практика:** Защита проекта.

## Методическое обеспечение

### Формы работы:

*Индивидуальная* - практическая работа обучающихся с оказанием педагога помощи, обучающимся при возникновении затруднения, не уменьшая активности обучающихся и содействуя выработке навыков самостоятельной работы.

*Групповая* - обучающимся предоставляется возможность самостоятельно построить свою деятельность на основе принципа взаимозаменяемости, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности.

### Технологии обучения:

*Использование ИКТ* - привлечение ресурсов интернет.

*Личностно-ориентированный подход в обучении* - признание индивидуальности, ценности каждого обучающегося, его развития как индивида. Целью личностно-ориентированного обучения является развитие познавательных и творческих способностей обучающихся, максимальное раскрытие индивидуальности ребенка.

### Методы обучения:

*Словесный* - передача информации теоретической части занятия.

*Практический* - показ технологии исполнения работы.

*Наглядный* - демонстрация ранее выполненных тематических работ.

*Метод самоконтроля* - выполнение самостоятельной части практического занятия, сравнение своего результата с образцом правильно выполненной работы.

*Метод проблемного обучения* - метод, когда процесс решения задачи обучающимся, со своевременной и достаточной помощью педагога, приближается к творческому процессу.

*Эвристический* - выработка логического и алгоритмического мышления.

## Материально-техническое обеспечение

- учебный класс, оснащенный: учебной мебелью (столы и стулья) на 12 учебных мест и 1 место педагога;
- учебной техникой (персональные компьютеры) на 8 учебных мест;
- демонстрационной учебной техникой (экран и проектор);
- копировальной техникой (принтер, сканер).

## Контрольно-измерительный материал

Для определения степени усвоения программы дополнительного образования осуществляются текущий, промежуточный и итоговый контроль. Текущий контроль (ТК) осуществляется в течение всего учебного года, промежуточный контроль (ПК) – после прохождения раздела, итоговый контроль (ИК) – в конце учебного года.

Текущий контроль выявляет степень сформированности практических умений и навыков обучающихся. Текущий контроль может проводиться в форме наблюдения, индивидуальное собеседование, групповая беседа, опрос. Текущий контроль осуществляется без фиксации результатов.

*Промежуточный контроль* проводится с целью установления уровня (высокий, средний, ниже среднего) освоения отдельной части или всего объема дополнительной программы.

- высокий – программный материал усвоен учащимся полностью, учащийся имеет высокие достижения;

- средний – усвоение программы в полном объеме, при наличии несущественных ошибок;

- ниже среднего – усвоение программы в неполном объеме, допускает существенные ошибки в теоретических и практических заданиях.

Формы промежуточной аттестации учащихся: выполнение практического задания.

*Итоговый контроль* проводится в конце обучения. Во время итогового контроля определяется фактическое состояние уровня знаний, умений, навыков, степень освоения материала по каждому изученному разделу и всей программе курса в виде защиты проекта.

### **Кадровое обеспечение**

Педагог дополнительного образования, работающий по данной программе, имеет высшее образование и соответствует профилю программы.

### **Список литературы**

1. Доусон М. Програмируем на Python. - СПб.: Питер, 2014. - 416 с.
2. Программирование для детей / К. Вордерман, Дж. Вудкок, Ш. Макаманус и др.; пер. с англ. С. Ломакина. - М.: Манн, Иванов и Фербер, 2015. - 224 с.
3. Python для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина; [науч. ред. Д. Абрамова]. - М.: Манн, Иванов и Фербер, 2017 - 320 с.