

муниципальное бюджетное учреждение  
дополнительного образования города Ростова-на-Дону  
«**Центр детского технического творчества**»

«Принято»

На заседании  
методического совета МБУ ДО ЦДТТ

Протокол № 1 от 28.08.2024 г.

«Утверждаю»

Директор МБУ ДО ЦДТТ

Н.А. Пивень

Приказ № 218 от 28.08.2024 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
технической направленности  
«Язык программирования Python»

срок реализации 2 года  
возраст обучающихся 10 – 16 лет

разработчик  
**Муравьев Юрий Александрович**  
педагог дополнительного образования

г. Ростов-на-Дону  
2024

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность и направленность программы. В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтения универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Подобные универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктивное мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Характерной чертой развития общества на протяжении последних десятилетий является его все более расширяющаяся информатизация. Отражением и следствием этой тенденции явилась потребность в подготовке подрастающего поколения к вступлению в информационное общество, любая профессиональная деятельность в котором будет связана с информатикой и информационными технологиями. Изучение данного курса имеет важное значение для развития мышления подростков. В современной психологии отмечается значительное влияние изучения информатики и использования компьютеров в обучении на развитие у школьников теоретического, творческого мышления, а также формирование нового типа мышления, так называемого операционного мышления, направленного на выбор оптимальных решений; открывает новые возможности для овладения такими современными методами научного познания, как формализация, моделирование, компьютерный эксперимент и т.д. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, полезные и в более широком контексте.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у обучающихся среднего школьного возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективной работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

Программа составлена с учетом задач национального проекта «Цифровая экономика» (федерального проекта «Кадры для цифровой экономики») и национального проекта «Образование» (федерального проекта «Молодые профессионалы»): создание условий для формирования у учащихся основ цифровых компетенций, необходимых для дальнейшего участия в разработке и поддержке инфраструктур передачи, обработки и хранения данных, а также цифровых продуктов для граждан, бизнеса и власти; разработка, внедрение цифровых технологий и платформенных решений.

Python — это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание

пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. ...

Программа «Программирование на Python» представляет собой углубленный курс по программированию, дающий представление о понятиях структурного программирования (данных, операциях, переменных, ветвлениях в программе, циклах и функциях, массивах, файлах). Программа направлена на развитие логического и пространственного мышления обучающегося, способствует раскрытию творческого потенциала личности, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Наш выбор Python для преподавания призван помочь обучающимся сделать первые шаги по одному из современных и перспективных путей развития IT-индустрии..

*Дополнительная образовательная программа* направлена на получение изначальных навыков работы с компьютером, ознакомление с программным обеспечением в области обучающих, развивающих программ, самостоятельную постановку задачи, структурирование и преобразование информации в текстовую и мультимедийную форму, использование ее для решения учебных и жизненных задач, использование проектно-исследовательского метода.

*Актуальность* овладения основами проектирования обусловлена, во-первых, тем, что данная технология имеет широкую область применения на всех уровнях. Во-вторых, к совместной проектной деятельности учителя и ученика сегодня побуждает сам ученик, осознающий свои собственные образовательные потребности и стремящийся к их удовлетворению. Реальным эффективным инструментом совместной проектной деятельности являются информационные технологии.

Отличительные особенности программы и новизна. Отличительной особенностью курса является его направленность 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. Для этого, учащиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают ее с помощью формального языка. Полученные знания учащиеся применяют при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями.

### ***ЦЕЛИ И ЗАДАЧИ:***

- научить учащихся ориентироваться и продуктивно действовать на сайте [stepik.org](http://stepik.org) на языке программирования Python;
- сформировать у учащихся самостоятельность в сборе и обработке материала;
- привить исследовательские навыки;
- более полно организовать межпредметные связи;
- познакомить со способами научно-технического мышления и деятельности, направленными на самостоятельное творческое познание и исследование;
- реализовать коммуникативные, технические и эвристические способности учащихся в ходе проектирования;
- ученики видят примеры применения изучаемой темы на практике, получают готовый к применению программный продукт;
- отрабатываются навыки проектирования и видения перспективы;
- сформировать элементы информационной и телекоммуникационной компетенций по отношению к знаниям, умениям.

### ***МЕТОДЫ ОБУЧЕНИЯ***

Основная методическая установка — обучение школьников навыкам самостоятельной индивидуальной и групповой работы. В методике

преподавания курса информатики особое место занимает проектная технология. Обучение с использованием метода учебных исследовательских проектов наиболее эффективный способ достижения высоких результатов образования. Проектная деятельность формирует креативно-интеллектуальную активность, коммуникативные умения, вносит вклад в общеучебную подготовку школьников, а также связана с использованием информационных технологий в различных предметных областях. Большинство заданий выполняется с помощью персонального компьютера и необходимых программных средств. Кроме индивидуальной, применяется и групповая работа. В задачи учителя входит создание условий для реализации ведущей подростковой деятельности — авторского действия, выраженного в проектных формах работы. Отбор методов обучения обусловлен необходимостью формирования информационной и коммуникативной компетентностей старшеклассников. Решение данной задачи обеспечено наличием в программе работы следующих элементов данных компетенций:

- социально-практическая значимость компетенции (для чего необходимо уметь создавать проекты);
- личностная значимость компетенции (зачем ученику необходимо быть компетентным в области создания ЭОР);
- перечень реальных объектов действительности, относящихся к данным компетенциям;
- знания, умения и навыки, относящиеся к данным объектам;
- способы деятельности по отношению к данным объектам;
- минимально необходимый опыт деятельности ученика в сфере данной компетенции;
- индикаторы — учебные и контрольно-оценочные задания по определению уровня компетентности ученика.

### ***ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНЫХ ЗАНЯТИЙ***

Основной тип занятий — практическая работа. Большинство заданий выполняется с помощью персонального компьютера и необходимых программных средств. Доступ в Интернет обязателен. Единицей учебного процесса является блок уроков. Каждый такой блок охватывает изучение отдельной информационной технологии или ее части.

Каждая тема начинается с постановки задачи — характеристики образовательного продукта, который предстоит создать ученикам. Изучение нового материала носит сопровождающий характер. Ученики изучают его с целью создания запланированного продукта. Далее проводится тренинг по отработке умений выполнять технические задачи, соответствующие минимальному уровню планируемых результатов обучения. Тренинг переходит в комплексную творческую работу по созданию учениками определенного образовательного продукта. Регулярное повторение способствует закреплению изученного материала. Возвращение к ранее изученным темам и использование их при изучении новых тем способствуют устранению весьма распространенного недостатка — формализма в знаниях учащихся — и формируют научное мировоззрение учеников.

### ***ДИАГНОСТИКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ***

В данном курсе предусмотрены проверка результатов освоения программы в виде мероприятий промежуточной аттестации:

- защиты индивидуальной творческой работы (проекта) по каждому разделу программы;
- выставки, конкурсы проектов;

Программа кружка рассчитана на два года. Занятия проводятся по 4 часа в неделю. Общий объем 144 часа в год.

Возраст детей, участвующих в реализации данной дополнительной образовательной программы составляет 10 -16 лет.

***Учащиеся должны иметь представление:***

- о сайте автоматической проверки программ [stepik.org](http://stepik.org);
- типах данных используемых в Python;

- формальном описании алгоритмов;
- интегрированной среде Python;
- основных алгоритмических конструкциях;
- объектах и их свойствах;
- данных и способе их организации.

***Учащиеся должны знать:***

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования;
- основы создания приложений в Python;
- инструментальные средства Python.

***Учащиеся должны уметь:***

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

**Учебно-тематический план программы дополнительного образования**

***Тематический план курса «Язык программирования Python:***

***1 год обучения***

Тема	Количество учебных часов		
	всего	теория	практика
<b><i>Вводное занятие.</i></b>	2	2	
<b><i>Раздел 1. Ввод-вывод данных</i></b>	20	8	12
<b><i>Раздел 2. Условный оператор</i></b>	18	6	12
<b><i>Раздел 3. Типы данных</i></b>	14	6	8
<b><i>Раздел 4. Циклы</i></b>	22	8	14
<b><i>Раздел 5. Строковый тип данных</i></b>	12	4	8
<b><i>Раздел 6. Списки</i></b>	22	8	14



<b>Раздел 7. Функции</b>	<b>10</b>	<b>4</b>	<b>6</b>
<b>Раздел 8. Мини проекты</b>	<b>24</b>	<b>4</b>	<b>20</b>
<b>Итого:</b>	<b>144</b>	<b>50</b>	<b>94</b>

### ***Введение 2 ч.***

Вводное занятие. Правила техники безопасности. Организация рабочего места. Вступительное слово преподавателя о программе курса.

### ***1. Раздел 1. Ввод-вывод данных 20ч.***

***Теоретический материал:*** Введение. Знакомство с Python. Вывод данных, команда print(). Ввод данных, команда input(). Необязательные параметры команды print(): sep, end. Целочисленный тип данных. Преобразование строки к целому числу. Операции над целыми числами: степень, нахождения остатка, целочисленного деления. Обработка цифр числа.

***Практический материал:*** Решение задач программирования с командами print() и input().

### ***Раздел 2. Условный оператор 18 ч.***

***Теоретический материал:*** Условный оператор. Отступы. Операторы сравнения. Логические операторы: «умножение», «сложение» и «не». Вложенные условия. Каскадные условия.

***Практический материал:*** Решение задач программирования с помощью условного оператора if.

### ***Раздел 3. Типы данных 14 ч.***

***Теоретический материал:*** Целочисленный тип данных int. Числа с плавающей точкой float. Встроенные функции max(), min(), abs(). Строковый тип данных str. Функции len() и str(). Конкатенация строк. Умножение строки на число. Оператор in. Модуль math.

***Практический материал:*** Решение задач программирования методом перевода данных из одного типа в другой.

### ***5. Циклы 22 ч.***

***Теоретический материал:*** Цикл for. Переменная цикла. Функции range() с одним, двумя и тремя параметрами. Цикл while. Считывание данных до стоп

значения. Бесконечный цикл. Использование цикла while для обработки цифр числа. Оператор break. Оператор continue. Ключевое слово else в циклах. Ревью кода. Поиск ошибок. Вложенные циклы. Операторы break и continue во вложенных циклах.

**Практический материал:** Решение задач программирования с помощью циклов. Частые сценарии: подсчет количества, вычисление суммы и произведения, обмен значений переменных, сигнальные метки, определение максимума и минимума.

#### **Раздел 5. Строковый тип данных 12 ч.**

**Теоретический материал:** Индексация строк. Итерирование строк. Срезы строк. Изменение символов строки. Методы строк: конвертация регистра, поиск и замена, классификация символов.

**Практический материал:** Решение задач программирования при обработке символьной информации.

#### **Раздел 6. Списки 22 ч.**

**Теоретический материал:** Введение в списки. Создание списков. Пустые списки. Функция списков. Основы работы со списками. Индексация и срезы. Методы добавления и удаления элементов. Вывод элементов списка. Строковые методы. Списочные выражения. Сортировка списков.

**Практический материал:** Решение задач программирования созданием списков.

#### **Раздел 7. Функции 10 ч.**

**Теоретический материал:** Функции без параметров. Функции с параметрами. Локальные и глобальные переменные. Функции с возвратом значений.

**Практический материал:** Решение задач программирования с помощью функций.

#### **Раздел 8. Мини проекты 24 ч.**

**Теоретический материал:** Модуль random.

**Практический материал:** Создание проектов: «Числовая угадайка» , «Магический шар 8», «Шифр Цезаря», «Угадайка слов», «Генератор безопасных паролей» , «Калькулятор систем счисления».

**Учебно-тематический план программы дополнительного образования**

**Тематический план курса «Язык программирования Python**

**2 год обучения**

Тема	Количество учебных часов		
	всего	теория	практика
<b>Вводное занятие.</b>	2	1	1
<b>Раздел 1. Тип данных bool и NoneType</b>	10	2,5	7,5
<b>Раздел 2. Вложенные списки</b>	18	4,5	13,5
<b>Раздел 3. Кортежи</b>	16	4	12
<b>Раздел 4. Множества</b>	18	4,5	13,5
<b>Раздел 5. Словари</b>	16	4	12
<b>Раздел 6. Модуль random и string</b>	8	2	6
<b>Раздел 7. Модули decimal, fractions и complex</b>	8	2	6
<b>Раздел 8. Функции</b>	36	4	32
<b>Раздел 9. Работа с файлами</b>	12	3	9
<b>Итого:</b>	144	31,5	112,5

**Введение 2 ч.**

Вводное занятие. Правила техники безопасности. Организация рабочего места. Вступительное слово преподавателя о программе курса.

**1. Раздел 1. Тип данных bool и NoneType 10 ч.**

**Теоретический материал:** Повторяем основные конструкции языка Python.

Логический тип данных bool. Логические операторы Булевы значения как числа. Функции bool(), type(), isinstance(). Тип данных NoneType. Литерал None. Сравнение None с другими типами данных

**Практический материал:** Решение задач программирования типами данных bool и NoneType.

## **Раздел 2. Вложенные списки 18 ч.**

**Теоретический материал:** Повторяем списки. Вложенные списки. Объявление и индексация. Функции len(), max(), min(). Списочные методы. Создание вложенных списков. Считывание вложенных списков. Перебор элементов вложенных списков. Обработка вложенных списков. Вывод вложенных списков. Работа с матрицами. Квадратные и прямоугольные матрицы. Функции ljust() и rjust(). Главная и побочная диагонали. Сложение матриц. Умножение матрицы на число. Умножение матриц.

**Практический материал:** Решение задач программирования с помощью вложенных списков.

## **Раздел 3. Кортежи 16 ч.**

**Теоретический материал:** Тип данных tuple. Особенности работы с кортежами. Функция tuple(). Особенности кортежей. Методы кортежей. Вложенные кортежи. Перебор кортежей. Сравнение кортежей. Сортировка кортежей. Преобразование кортежа в список и строку. Упаковка кортежей. Распаковка кортежей. Присваивание кортежей. Упаковка кортежей.

**Практический материал:** Решение задач программирования с помощью типа данных кортеж.

## **5. Множества 18 ч.**

**Теоретический материал:** Множества в математике. Числовые множества. Операции над множествами. Диаграммы Эйлера-Вена. Создание множеств. Пустые множества. Встроенная функция set(). Вывод множеств. Встроенные функции len(), sum(), min(), max(). Оператор принадлежности in. Перебор множеств. Форматированный вывод множеств. Сравнение множеств. Метод добавления элемента add(). Методы удаления элементов remove(), discard(), pop(). Метод удаления всех элементов clear(). Методы union(), intersection(), difference(), symmetric\_difference(). Методы update(), intersection\_update(), difference\_update(), symmetric\_difference\_update(). Операторы &, |, -, ^.

Методы `issuperset()`, `issubset()`, `isdisjoint()`. Сравнение множеств (`<`, `>`, `<=`, `>=`).  
Генераторы множеств. Неизменяемое множество `frozenset`.

**Практический материал:** Решение задач программирования с помощью типа данных множеств.

### **Раздел 5. Словари 16 ч.**

**Теоретический материал:** Новый тип коллекции. Отличия словарей от списков. Создание словарей. Обращение по ключу. Встроенная функция `dict()`. Создание словарей на основе списков и кортежей. Пустой словарь. Вывод словаря. Особенности словарей. Встроенные функции `len()`, `sum()`, `min()`, `max()`. Оператор принадлежности `in`. Перебор словарей. Распаковка словаря. Форматированный вывод словарей. Сравнение словарей. Методы `keys()`, `values()` и `items()`. Добавление и изменение элементов в словаре. Удаление элементов из словаря. Методы `get()`, `update()`. Методы `pop()`, `popitem()`. Методы `clear()`, `copy()`. Метод `setdefault()`. Вложенные словари. Генераторы словарей.

**Практический материал:** Решение задач программирования с помощью словарей.

### **Раздел 6. Модуль `random` и `string` 8 ч.**

**Теоретический материал:** Случайные числа. Псевдослучайные числа. Модуль `random`. Метод `shuffle()`. Метод `choice()`. Метод `sample()`. Модуль `string`. Метод Монте-Карло.

**Практический материал:** Решение задач программирования созданием случайным образом.

### **Раздел 7. Модули `decimal`, `fractions` и `complex` 8 ч.**

**Теоретический материал:** Числа с плавающей точкой `float`. Модуль `decimal`. Тип данных `Decimal`. Модуль `fractions`. Тип данных `Fraction`. Вещественные числа. Мнимые числа. Комплексные числа. Тип данных `complex`. Модуль `cmath`. Модуль `turtle`.

**Практический материал:** Решение задач программирования с помощью модулей `decimal`, `fractions` и `complex`.

## **Раздел 8. Функции 36 ч.**

**Теоретический материал:** Позиционные аргументы. Необязательные аргументы. Именованные аргументы. Аргументы \*args. Аргументы \*\*kwargs. Keyword-only аргументы. Парадигмы программирования. Императивное программирование. Структурное программирование. Объектно-ориентированное программирование. Логическое программирование. Функциональное программирование. Функции как объекты. Функции в качестве аргументов других функций. Встроенные функции, принимающие функции в качестве аргументов. Функции в качестве возвращаемых значений других функций. Функции высшего порядка. Самописные функции map(), filter(), reduce(). Встроенные функции map(), filter(), reduce(). Модуль operator. Определение анонимных функций. Условный оператор в теле анонимной функции. Передача аргументов в анонимную функцию. Ограничения анонимных функций. Функции all(), any(), enumerate() и zip().

**Практический материал:** Решение задач программирования с помощью функций.

## **Раздел 8. Работа с файлами 12 ч.**

**Теоретический материал:** Понятие файла. Файловый ввод и вывод. Работа с файлами. Типы файлов. Методы доступа к файлам. Имена файлов. Кодировка файлов. Открытие и закрытие файлов. Указание кодировки. Чтение данных из файла. Позиция (курсор) в файле. Менеджеры контекста. Запись данных в файл. Файловые методы write() и writelines().

**Практический материал:** Решение задач программирования с помощью файлов.

### **Литература для преподавателя:**

1. Авторский электронный обучающий курс Шакурова З. З. «Информатика, информационные технологии. Программирование».
2. Балаян Г.В. Метод проектов на уроке истории //Школьные технологии.- 1997.- № 1.- С. 116-119.
3. Бурков В.Н., Новиков Д.А. Как управлять проектами. – М., 1997.
4. Гузеев В.В. «Метод проектов» как частный случай интегральной технологии обучения// Директор школы. – 1995. - № 6. – С. 39-47.
5. Новикова Т.А. Проектные технологии на уроках и во внеурочной деятельности //Народное образование.- 2000.- № 7.-С. 151-157.
6. Сиденко А.С. Метод проектов: история, теория и практика //Наука и школа. 2002. №8. – С.49-53, №9. – С. 3-6.
7. Чечель И.Д. Исследовательские проекты в практике школы// Управление исследовательской деятельностью педагога и учащегося в современной школе.- М.: Сентябрь, 1998. – С. 83-128.
8. Шамова Т.И., Малинин А.Н., Тюлю Г.М. Инновационные процессы в школе.
9. Ясвин В.А. Образовательная среда: от моделирования к проектированию.- М., 1997.
10. <http://www.intergu.ru> –Интернет государство учителей
11. <http://www.uchportal.ru/> учительский портал

### **Литература для обучающихся:**

1. Тони Гэддис«Начинаем программировать на Python (5-е изд) »–Санкт-Петербург, БВХ-Петербург, 2022
2. Кольцов Д.М. «Справочник PYTHON. Кратко, быстро, под рукой»– Санкт-Петербург, Наука и техника, 2021

3. Создание тестов с помощью управляющих элементов сайт методическое пособие <http://www.stepik.org/>

### **Дидактический материал:**

1. Разработки уроков по темам: «Программирование».
2. Лабораторные практикумы по темам «Типы данных», «Операции в Python» и др.
3. Обучающий сайт [stepik.org](http://stepik.org).
4. Среда разработки Python.
7. Среда разработки Блок-схема.
8. Карточки с заданиями.
9. Разработки электронных учебников по различной тематике.
10. Разработки электронных рабочих тетрадей.
11. Мини проекты.