

Министерство образования и науки Самарской области

Государственное бюджетное образовательное учреждение  
дополнительного образования Самарской области  
«Самарский областной центр детско-юношеского технического творчества»

Принята на заседании  
Методического Совета  
Протокол № 2

от « 20 » июня 2023 г.

УТВЕРЖДАЮ

Директор ГБОУ ДО СО СОЦДИЮТТ

/А.Ю. Богатов/

2023 г.



Дополнительная общеобразовательная общеразвивающая программа  
технической направленности

**«Программирование на Python»**

Возраст детей: 12-18 лет

Срок обучения: 1 год

**Разработчик:**

Яковлев Дмитрий Николаевич  
педагог дополнительного образования  
первой квалификационной категории

Самара, 2023

## **Оглавление**

Пояснительная записка	3
Учебно-тематический план	6
Содержание	7
Методическое обеспечение	11
Список литературы	14
Приложение Календарно-тематическое планирование	15

## Пояснительная записка

Дополнительная общеобразовательная программа «Программирование на Python» является общеразвивающей программой *технической* направленности. Автор программы - Яковлев Дмитрий Николаевич.

**Актуальность программы.** В настоящее время информационные технологии охватывают все сферы жизни. Новые технологии вводятся в различные профессии и требуют от сотрудников знания как пользовательских, так и базовых навыков программирования. Изучение Python является необходимым для формирования базовых навыков программирования и работы с данными.

- Python – это универсальный язык программирования. Он пригоден для создания самых разных программ, от игр до веб-сайтов;
- Python – современный, простой и удобный язык;
- В Python есть библиотеки готовых решений для использования в разных областях. Это позволяет создавать сложные программы быстро;
- Python - очень популярный и востребованный на рынке язык программирования. Например, в известном рейтинге ТЮВЕ он занимает третье место, при этом доля его в последнее время растет. Python – это язык будущего.

**Педагогическая целесообразность** программы заключается в том, что учащиеся приобретут навыки, которые позволят им создавать работающие приложения и станут основой для дальнейшего изучения 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. Гибкая структура программы позволит для каждого учащегося сформировать собственный, наиболее для него подходящий путь вхождения в профессию.

**Отличительная особенность** программы состоит в том, что она в краткой форме знакомит учащихся со основами языка программирования Python и его применения при работе с базами данных. В каждом модуле программы присутствует как теоретическая, так и практическая часть, поэтому учащиеся будут видеть конкретны результат освоения предмета.

**Адресат программы** - дети от 12 до 18 лет. Наполняемость групп: 5 – 12 человек.

**Объем и срок освоения программы.** Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python рассчитана на 1 год обучения.

**Режим занятий:** 72 академических часа в год, 2 академических часа в неделю.

**Форма реализации:** очная. Имеется возможность временно проводить занятия дистанционно через платформы для видеосвязи.

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

**Цель программы:** познакомить учащихся с современными информационными технологиями посредством изучения языка программирования «Python».

**Задачи:**

Обучающие:

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

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 учащихся усидчивость, аккуратность, умение доводить начатое дело до конца;
- формировать коммуникативные навыки.

**Личностные, предметные и метапредметные результаты выполнения программы.**

**Личностные результаты.**

- Формирование навыка доводить дело до конца

- Формирование мировоззрения, необходимого в современном цифровом мире.

- Формирования позитивного отношения к коллегам, их мнению.

- Формирование ответственного отношения к обучению

#### **Предметные результаты.**

- Умение разбивать решение задачи на подзадачи, составлять алгоритм;

- Умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;

- Умение искать и обрабатывать ошибки в коде;

- Умение писать грамотный, красивый код;

- Умение анализировать как свой, так и чужой код;

- Умение импортировать модули в программу;

- Умение работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);

- Умение грамотно строить коммуникацию, исходя из целей и ситуации;

- Умение работать с вычислительной техникой.

- Знание основных классических алгоритмов и способы их реализации;

- Знание основных элементы программирования

- Умение создавать оконные приложения.

#### **Метапредметные результаты**

- Умение создавать модели для решения практических задач.

- Умение разрешать конфликты на основе выработки общей позиции

- Умение критически оценивать сроки реализации задуманного проекта.

- Умение вносить изменение в проект, корректировать изначальный план

- Умение ставить для себя новые задачи

## Учебно-тематический план

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
<b>1</b>	<b>Введение в программирование</b>	<b>16</b>	<b>7</b>	<b>9</b>
1.1	Установка Python. Среда разработки IDLE. Понятие консольной программы.	2	1	1
1.2	Ввод-вывод данных на языке Python. Переменные	2	1	1
1.3	Типы данных. Строки и числа.	2	1	1
1.4	Условный оператор.	2	1	1
1.5	Логический тип. Объединение условий.	2	1	1
1.6	Циклы. Использование цикла while	2	1	1
1.7	Цикл for	2	1	1
1.8	Решение задач на циклические алгоритмы	2	0	2
<b>2</b>	<b>Основы языка Python</b>	<b>32</b>	<b>14</b>	<b>18</b>
2.1	Строки и списки	2	1	1
2.2	Кортежи и словари	2	1	1
2.3	Понятие функции. Использование встроенных функций	2	1	1
2.4	Создание своих функций	2	1	1
2.5	Двумерные списки	2	1	1
2.6	Понятие модуля (библиотеки). Подключение и работа с функциями из модуля. Модуль math – собрание математических функций.	2	1	1
2.7	Модуль random. Работа со случайными числами. Модуль numpy.	2	1	1
2.8	Работа с файлами	2	1	1
2.9	Основы ООП. Классы и объекты.	2	1	1
2.10	Наследование	2	1	1
2.11	Создание оконных приложений. Модуль tkinter	2	1	1
2.12	Интерфейс. Кнопки. События.	2	1	1
2.13	Графика с модулем tkinter	4	1	3
2.14	«Черепашья графика». Модуль turtle.	4	1	3
<b>3</b>	<b>Основы проектной деятельности</b>	<b>18</b>	<b>3</b>	<b>15</b>
3.1	Системы контроля версий. Работа с git, понятие “ветка” и “коммит”.	2	0	2
3.2	Работа в команде. Знакомство с github.	2	0	2
3.3	Проектирование и разработка приложения «Персональный помощник». Консольный вариант.	2	1	1
3.4	«Персональный помощник». Оконное приложение.	4	0	4
3.5	Разработка игры на клетчатом поле с использованием модуля tkinter.	4	1	3
3.6	Разработка чат-бота для Телеграм.	4	1	3
<b>4</b>	<b>Аттестация</b>	<b>6</b>	<b>-</b>	<b>6</b>
	<b>Итого:</b>	<b>72</b>	<b>28</b>	<b>44</b>

## Содержание

### Раздел 1. Введение в язык программирования Python

#### ***Тема 1.1. Установка Python. Среда разработки IDLE. Понятие консольной программы.***

*Теория.* 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.* Сохранение и запуск python-программ в среде разработки IDLE.

#### ***Тема 1.2. Ввод-вывод данных на языке Python. Переменные***

*Теория.* Основы синтаксиса языка Python. Функция print(). Понятие «переменная». Правила именования переменных в языке Python. Оператор присваивания. Функция input().

*Практика.* Проект «Символьная графика». Создание определённого рисунка с помощью символов. Отработка функции print(). Проект «Попугай». В данном проекте запомнить введенную строку и несколько раз ее повторить.

#### ***Тема 1.3. Типы данных. Строки и числа***

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

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

#### ***Тема 1.4. Условный оператор***

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

*Практика.* Решение задач на отработку условного оператора и операторов сравнения.

#### ***Тема 1.5. Логический тип. Объединение условий.***

*Теория.* True и False, операции and, or, not. Порядок выполнения операций. Переменные без значения – None.

*Практика.* Проект «Калькулятор»: создание приложения по определенным условиям.

#### ***Тема 1.6. Использование цикла while***

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

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

### ***Тема 1.7. Использование цикла for***

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

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

### ***Тема 1.8. Решение задач на циклические алгоритмы***

*Теория.* Виды циклов и их конструкции.

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

## **Раздел 2. Основы языка Python**

### ***Тема 2.1. Строки и списки***

*Теория.* Понятие «строка». Создание строк. Переменные внутри строк. Операции со строками. Понятие «список». Создание списков. Добавление/удаление. Операции со списками.

*Практика.* Решение задач на отработку операций со строками и списками.

### ***Тема 2.2. Кортежи и словари***

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

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

***Тема 2.3. Понятие функции. Использование встроенных функций.*** *Теория.* Понятие «функция», «параметр функции», «значение функции». Функции: abs, bool, dir, eval, exec, float, int, len, max, min, range, sum.

*Практика.* Решение задач на использование математических функций.

### ***Тема 2.4. Создание своих функций***

*Теория.* Ключевые слова def и lambda. Строение функции: имя, аргумент, тело. Создание и вызов функции. Переменные и область видимости.

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

### ***Тема 2.5. Двумерные списки. Особенности копирования структур данных.***

*Теория.* Вложенные списки. Особенности присваивания изменяемых и неизменяемых объектов в Питоне. Методы сору и деерсору.

*Практика.* Работа с таблицами.

### **Тема 2.6. Применение модулей. Математические возможности Python.**

*Теория.* Понятие «модуль». Импортирование модуля в программу. Полезные модули: math, time и другие.

*Практика.* Решение задач на применение модулей.

### **Тема 2.7. Модуль random. Работа со случайными числами. Модуль numpy.**

*Теория.* Функции random, randint, choice и другие.

*Практика.* Создание игры «Угадайка».

### **Тема 2.8. Работа с файлами**

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

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

### **Тема 2.9. Основы ООП. Классы и объекты**

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

*Практика.* Решение задач на отработку умения создавать классы и объекты.

### **Тема 2.10. Наследование**

*Теория.* Понятие предка и потомка.

*Практика.* Создание структуры классов для описания животных.

### **Тема 2.11. Модуль tkinter**

*Теория.* Понятие оконной программы. Модуль tkinter и его возможности.

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

### **Тема 2.12. Интерфейс. Кнопки. События**

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

*Практика.* Проект «Игра кликер».

### **Тема 2.13. Графика с модулем tkinter**

*Теория.* Модуль tkinter и его возможности.

*Практика.* Проект «Анимация»: отрисовывание объекта на холсте и программирование анимации. Проект «Фантастический мир»: рисование фигур, параметры которых задаются пользователем при запуске программы.

### **Тема 2.14. Рисование с помощью «Черепашьей графики»**

*Теория.* Модуль turtle. Импортирование модуля. Создание холста. Перемещение черепашки.

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

### **Раздел 3. Основы проектной деятельности.**

#### ***Тема 3.1. Системы контроля версий.***

*Теория.* История появления систем контроля версий. Их необходимость в работе. понятие «ветка» и «коммит».

*Практика.* Установка и работа с git.

#### ***Тема 3.2. Работа в команде. Знакомство с github.***

*Теория.* Обзор сайтов для совместной разработки.

*Практика.* Регистрация на github.com. Создание репозитория. Перенос проекта с локального компьютера в сеть. Команда push. Совместная работа над одним проектом.

#### ***Тема 3.3. Приложение «Персональный помощник». Консольный вариант.***

*Теория.* Обсуждение структуры программы: необходимые классы, способы хранения тестов. Работа с json-файлами.

*Практика.* Разработка программы, позволяющей хранить заметки по разным темам, составлять расписание и т.д. В программе должны быть разные роли (администратор, пользователь), возможность добавлять и редактировать события.

#### ***Тема 3.4. Приложение «Персональный помощник». Оконный вариант.***

*Теория.* Проектирование программ, допускающих дальнейшее расширение и переработку.

*Практика.* Добавление к разработанной программе оконного интерфейса.

#### ***Тема 3.5. Игра «Лабиринт»***

*Теория.* Обсуждение сюжета игры: класс Лабиринт, класс Игрок и их роль. Возможные способы реализации игры: хранение уровней в коде программы или в текстовых файлах, динамическое создание уровней

*Практика.* Приложение-игра «Лабиринт»: Задача пользователя: дойти до конца лабиринта, 2D-игра, вид сверху. Реализация чтения и отображения уровней. Реакция на кнопки. Добавление соперников. Добавление и отображения уровня здоровья/оставшихся жизней. Экран победы и поражения.

#### ***Тема 3.6. Разработка чат-бота для Телеграм***

*Теория.* Модули в Питоне для создания ботов. Структура бота. Добавление сценариев диалогов и кнопок.

*Практика.* Написание аналога проекта «Персональный помощник» в виде бота.

#### **Раздел 4. Аттестация**

*Практика.* Разработка собственного приложения. Создание презентации и подготовка выступления. Защита проекта.

## **Методическое обеспечение**

### *Материально-техническое обеспечение программы*

- столы, стулья (по росту и количеству детей);
- демонстрационная магнитная доска;
- технические средства обучения (ТСО) (мультимедийное устройство);
- ноутбуки (10 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО;

### *Формы подведения итогов реализации программы*

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: промежуточную и итоговую аттестацию результатов обучения детей.

Промежуточная аттестация 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.

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

### *Методы и приемы работы*

**Особенности организации образовательного процесса:** очная форма. Возможен временный переход на дистанционное обучение.

#### **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.

### **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 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, может проходить в виде мини-выставок, просмотров творческих работ и презентаций.

### **Алгоритм учебного занятия:**

1. Мотивация обучающихся.
2. Актуализация имеющихся знаний.
3. Теоретический блок нового материала.
4. Закрепление материала.

5. Перерыв.
6. Теоретический блок нового материала.
7. Закрепление материала.
8. Рефлексия.

#### **Контрольно-измерительный блок**

Форма	Описание	Критерии оценки
Защита проекта	Учащимся необходимо подготовить проект на Python. Тема выбирается самостоятельно.	<ul style="list-style-type: none"> <li>• Структурная сложность</li> <li>• Качество кода</li> <li>• Оригинальность идеи</li> <li>• Графика</li> <li>• Работоспособность</li> </ul>

## Список литературы

1. М. Лутц. Изучаем Python. Вильямс. 2019.
2. Великович Л., Цветкова М. Программирование для начинающих. – М.: Бином, 2008.
3. Программирование для детей. Перевод с английского Станислава Ломакина, Москва, «Манн, Иванов и Фербер», 2015 г.

## Календарно-тематический план

№	Тема занятия	Продолжительность	Дата
1	Установка Python. Среда разработки IDLE. Понятие консольной программы.	2	
2	Ввод-вывод данных на языке Python. Переменные	2	
3	Типы данных. Строки и числа.	2	
4	Условный оператор.	2	
5	Логический тип. Объединение условий.	2	
6	Циклы. Использование цикла while	2	
7	Цикл for	2	
8	Решение задач на циклические алгоритмы	2	
9	Строки и списки	2	
10	Кортежи и словари	2	
11	Понятие функции. Использование встроенных функций	2	
12	Создание своих функций	2	
13	Двумерные списки	2	
14	Понятие модуля (библиотеки). Подключение и работа с функциями из модуля. Модуль math – собрание математических функций.	2	
15	Модуль random. Работа со случайными числами. Модуль numpy.	2	
16	Работа с файлами	2	
17	Основы ООП. Классы и объекты.	2	
18	Наследование	2	
19	Создание оконных приложений. Модуль tkinter	2	
20	Интерфейс. Кнопки. События.	2	
21	Графика с модулем tkinter	2	
22	Графика с модулем tkinter	2	
23	«Черепашья графика». Модуль turtle.	2	
24	«Черепашья графика». Модуль turtle.	2	
25	Системы контроля версий. Работа с git, понятие “ветка” и “коммит”.	2	
26	Работа в команде. Знакомство с github.	2	
27	Проектирование и разработка приложения «Персональный помощник». Консольный вариант.	2	
28	«Персональный помощник». Оконное приложение.	2	
29	«Персональный помощник». Оконное приложение.	2	
30	Разработка игры на клетчатом поле с использованием модуля tkinter.	2	
31	Разработка игры на клетчатом поле с использованием модуля tkinter.	2	
32	Разработка чат-бота для Телеграм.	2	
33	Разработка чат-бота для Телеграм.	2	
<b>34</b>	<b>Аттестация</b>	<b>2</b>	
<b>35</b>	<b>Аттестация</b>	<b>2</b>	
<b>36</b>	<b>Аттестация</b>	<b>2</b>	

	<b>Итого:</b>	<b>72</b>	
--	---------------	-----------	--