

**Муниципальное общеобразовательное учреждение  
Борисоглебская средняя общеобразовательная школа №1**

Согласовано  
Методический совет  
от 30.08.2022 г. Протокол № 1

Принята на заседании педагогического  
совета МОУ Борисоглебская СОШ №1  
Протокол №1 от 30.08.2022 г.



Утверждаю  
Директор МОУ БОСШ №1  
Огурцов А.К  
Приказ № 01-07/136  
от 01.09.2022 г.

**Техническая направленность**

Дополнительная  
общеобразовательная общеразвивающая программа  
**«IT-технологии и программирование»**

**Нормативный срок изучения предмета:** 1 год

**Возраст детей:** 12-15 лет

Составитель:  
Учитель информатики  
Ахапкин Н.А.  
Ф.И.О.

## Содержание

<b>1. Пояснительная записка.....</b>	<b>3-6</b>
<b>2. Учебно-тематический план .....</b>	<b>7</b>
<b>3. Содержание программы.....</b>	<b>8</b>
<b>4. Методическое обеспечение программы.....</b>	<b>9-10</b>
<b>5. Список литературы.....</b>	<b>11</b>
<b>Приложение 1.....</b>	<b>12-13</b>
<b>Приложение 2.....</b>	<b>14-19</b>

## 1. Пояснительная записка.

Язык программирования Python на сегодняшний день является наиболее популярной высокоуровневой средой международного олимпиадного программирования. Python – мощный и простой для изучения язык программирования. В нем представлены проработанные высокоуровневые структуры данных и простой, но эффективный подход к объектно-ориентированному программированию.

Недостаточные знания различных языков программирования затрудняют решения многих задач по проектированию и прогнозированию.

Python - современный и прогрессивный язык программирования, который используют миллионы инженеров по всему миру. Этот язык по праву считается одним из самых лаконичных, что помогает разработчикам сконцентрировать все усилия на достижении результата и при этом получать удовольствие от самого процесса. Python используют такие IT-гиганты, как Google, Dropbox, Yandex, Facebook, Mail.ru и многие другие.

Дополнительная общеобразовательная общеразвивающая программа «Мой Python» (далее Программа) имеет **техническую направленность**, предметом изучения является язык программирования.

### ***Новизна и актуальность***

Программа направлена на расширение знаний учащихся в направлении изучения языков программирования, в частности, языка программирования Python.

Программа является синтезом основных тем, которые используются при изучении языков программирования, расширяющих и дополняющих знания умения и навыки учащихся 13-16 лет, при работе с программным кодом, решении задач. Программа предназначена для системы дополнительного образования и основана на профильном изучении данной дисциплины.

Обучающиеся будут знать не только основные алгоритмические конструкции и операторы изучаемого языка программирования, но и иметь опыт самостоятельной записи алгоритмов и программ, использовать не один язык программирования, решать практические задачи методом разработки и отладки компьютерной

программы, проектировать деятельность по решению задач.

**Ценность программы** определяется разнообразием тематики и методов решения задач, на языке программирования Python. Школьники учатся работать с программным кодом, решать и записывать задачи на языке программирования Python, правильно работать с операторами, синтаксисом, переменными, типами данных и т.д.

Изучение основ программирования связано с целым рядом умений и навыков (организация деятельности, планирование ее), которые по праву носят общеинтеллектуальный характер и формирование которых - одна из приоритетных задач современной школы. Очень велика роль изучения программирования для развития мышления школьников, формирования многих приемов умственной деятельности. Здесь роль информатики можно сравнить с математикой в школьном образовании. Поэтому необходимо использовать действительно большие возможности программирования, решения соответствующих задач для развития мышления школьников, формирования многих общеучебных умений и навыков. Курс направлен на изучение основ языка, его идеологии и позволит взглянуть на процесс разработки под новым углом. Во время обучения слушатели будут решать множество интересных и полезных задач.

**Цель программы** - подготовить учеников, способных эффективно и лаконично решать практические задачи на языке программирования Python.

**Основные задачи:**

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- развитие интереса учащихся к изучению программирования;
- приобретение навыков работы в системе программирования Python;
- формирование самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники.
- формирование навыков алгоритмического и логического мышления;
- формирование навыков грамотной разработки программ.

### ***Отличительные особенности:***

В процессе обучения включены необходимые базовые знания по программированию с учетом возрастных особенностей ребенка и уровнем его подготовки.

***Возраст обучающихся: 13-16 лет***

***Срок реализации программы – 1 год(72 часа).***

***Режим и форма занятий.***

Для изучения этой программы отводится 72 часов из расчета 2 часа в неделю.

*Основные формы проведения занятий:* лекции, решение задач, работа с электронным учебником, работа с онлайн приложениями для изучения синтаксиса языка.

***В результате изучения программирования на языке Python ученик будет:***

*Знать/понимать:*

- основные типы и виды конструкций алгоритмов;
- иметь представление о структуре программы,
- содержание этапов разработки программы: алгоритмизация- кодирование- отладка-тестирование;
- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- основы постановки задач в области информационных систем.

*Уметь:*

- записывать основные алгоритмические структуры на языке программирования Python;
- использовать Python для решения задач из области математики;
- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;
- решать нестандартные задачи и задачи повышенной сложности;
- анализировать текст чужих программ, находить в их неточности, оптимизировать алгоритм, создавать собственные варианты решения.

***Способы отслеживания результатов освоения программы учащимися:***

- освоение теоретического и практического материала на занятиях;
- самостоятельная практическая работа: выполнение домашних заданий.

***Механизм оценки результатов.***

Система оценки достижений учащихся включает в себя такие методы, как тестирование, практические и контрольные работы (см. *Приложение 2*).

## 2. Учебно-тематический план

№ п/п	Тема	Часы			Формы аттестации/ контроля
		Теория	Практика	Всего	
1	Синтаксис языка программирования Python	6	10	16	Устный зачет/ решение задач
2	Основные управляющие конструкции линейного алгоритма.	4	12	16	Устный зачет / решение задач
3	Основные управляющие конструкции алгоритма с ветвлением в Python.	4	8	12	Устный зачет / решение задач
4	Основные управляющие конструкции циклического алгоритма в Python.	4	12	16	Устный зачет / решение задач
5	Фундаментальная структура данных - список в Python.	4	8	12	Контрольная работа / решение задач
	<b>ИТОГО:</b>	<b>22</b>	<b>50</b>	<b>72</b>	

Тематическое планирование (смотри *Приложение 1.*)

### **3. Содержание программы.**

#### **Тема 1. Синтаксис языка программирования Python**

Теория: Введение в образовательную программу. Понятие языка программирования, история, структура. Типы данных.

Практика: Простые операторы. Структура программы.

#### **Тема 2. Основные управляющие конструкции линейного алгоритма.**

Теория: Простейшие функции, простые операции над числами, строками.

Практика: Решение линейных задач.

#### **Тема 3: Основные управляющие конструкции алгоритма с ветвлением в Python.**

Теория: Логические операции. Условный оператор, вложенные структуры. Сложные и составные условия.

Практика: Решение задач с ветвлением

#### **Тема 4: Основные управляющие конструкции циклического алгоритма в Python**

Теория: Циклы, структура цикла, виды циклов, особенности и отличия циклов при решении задач.

Практика: Решение задач с циклами

#### **Тема 5: Фундаментальная структура данных - массив в Python**

Теория: Списки, структура, методы и срезы в списках.

Практика: решение задач со списками.



#### **4. Методическое обеспечение программы.**

Различные формы и методы обучения в программе «Мой Python» реализуются различными способами и средствами, способствующими повышению эффективности усвоения знаний и развитию творческого потенциала личности учащегося.

Метод: исследование готовых знаний > Формы: поиск материалов, систематизация знаний, лекций > Приемы: Работа с методической литературой.

Метод объяснительно-иллюстративный > Формы: Лекции, беседы, рассказы, демонстрации > Приемы: Беседа: «Применение компьютеров в жизни человека»

Метод репродуктивный > Формы: Воспроизведение приемов действий, применение знаний на практике > Приемы: Практическая работа по разным направлениям.

Метод проверки знаний и умений > Формы: Игры, выставки по разделам > Приемы: Викторина по пройденным темам.

Методы контроля и управления образовательным процессом - это наблюдение педагога в ходе занятий, анализ подготовки и участия членов коллектива в мероприятиях, оценка членов жюри, анализ результатов выступлений на различных областных, всероссийских мероприятиях, выставках, конкурсах и соревнованиях. При работе по данной программе вводный (первичный) контроль проводится на первых занятиях с целью выявления образовательного и творческого уровня обучающихся, их способностей. Он может быть в форме собеседования или тестирования. Текущий контроль проводится для определения уровня усвоения содержания программы. В конце учебного курса итоговая работа (*Приложение 2*).

**Формы контроля** - традиционные: конференция, фронтальная и индивидуальная беседа, выполнение дифференцированных практических заданий, участие в конкурсах и выставках научно-технической направленности и т.д.

#### ***Оборудование***

#### ***Техническое обеспечение***

1. Компьютерный класс (12 ученических ПК + 1 учительский ПК)

2. Интерактивная доска
3. Проектор
4. Принтер

### ***Программное обеспечение***

1. ОС Windows
2. Пакет Microsoft Office
3. Python (версия 3+)

## 5. Список литературы

1. ФГОС. Примерные программы по информатике для основной и старшей школы. М.: БИНОМ. Лаборатория знаний. 2012.
2. Программа УМК К. Ю. Полякова, М.: Бином
3. «Начало программирования на языке Python» Л.Л.Босова, Н.А.Аквилянов, И.О. Кочергин, Ю.Л.Штепа, Т.А.Бурцева
4. Официальный сайт python –<https://www.python.org/>

Тематическое планирование занятий

№	Тема	Дата проведения
<b>Синтаксис языка программирования Python (16 ч.)</b>		
1	Понятие о языке высокого уровня Python.	
2	Технология разработки программного обеспечения. Среда Python.	
3-4	Структура программы.	
5-6	Знакомство с IDLE. Тестирующая система	
7-8	Переменные и константы	
9-10	Оператор присваивания.	
11-12	Оператор вывода	
13-14	Оператор ввода	
15-16	Типы данных. Числовые типы	
<b>Основные управляющие конструкции линейного алгоритма (16 ч.)</b>		
17-18	Простейшие функции	
19-20	Операции над числами	
21	Приоритет операций	
22-24	Целочисленное деление	
25	Обмен значениями	
26-28	Программы с линейной структурой	
29-30	Операции над строками	
31-32	Решение линейных задач.	
<b>Основные управляющие конструкции алгоритма с ветвлением в Python (12 ч.)</b>		
33-34	Условный оператор(if else), структура, отступы.	
35-36	Логические операции.	
37-38	Вложенные структуры(elif), структура, отступы.	
39-40	Сложные условия.	
41-42	Составные условия(and, or)	
43-44	Решение задач с ветвлением	
<b>Основные управляющие конструкции циклического алгоритма в Python (16 ч.)</b>		
45-46	Повторение линейных задач	
47-48	Цикл for, структура, отступы.	
49-50	Структура range(start, stop, step)	
51-52	Цикл while, структура, отступы.	
53-54	Счётчик, сложные условия	
55-56	Особенности циклов, вложенные циклы	

<b>57-58</b>	Оператор break, continue. Тип bool.	
<b>59-60</b>	Решение задач с циклами	
<b>Фундаментальная структура данных - массив в Python (12 ч.)</b>		
<b>61-62</b>	Списки. Создание списков	
<b>63-64</b>	Индексы. Append	
<b>65-66</b>	Срезы в списках.	
<b>67-68</b>	Методы в списках.	
<b>69-70</b>	Решение задач со списками.	
<b>71-72</b>	Контрольная работа.	

## Итоговая работа

1. Язык программирования - ... \*

1 балл

- это язык, по средствам которого пишутся программы.
- это формальный язык, предназначенный для записи алгоритмов, исполнителем которого будет компьютер.
- это программный язык, предназначенный для записи программ с помощью алгоритмов, с множеством исполнителей.
- все варианты

2. Какая команда запрашивает ввод данных у пользователя? \*

1 балл

- map()
- print()
- input()
- for()

3. Что выведет команда print("Привет друзья!") \*

1 балл

- "Привет друзья!"
- Привет друзья!
- print("Привет друзья!")
- ("Привет друзья!")
- Привет друзья

4. Какая команда печатает данные пользователя? \*

1 балл

- input()
- while()
- for()
- print()
- map()

5. Что выведет следующий код? \*

1 балл

```
a = 10
b = 3
a = a // 2 % b
print(a, b)
```

- 2
- 23
- 2 2

- 2 3
- 3

6. Переменные sum и Sum - это ... \*

1 балл

- разные переменные
- одинаковые переменные
- нет таких переменных
- Sum - не переменная
- sum - не переменная

7. Что выведет следующий код? \*

1 балл

```
a=7  
  
b=a+10  
  
a=b-5  
  
print (a)
```

- 13
- 12
- 5
- 17

8. Что означает логический оператор >= \*

1 балл

- Выполнение команды справа от оператора
- Равно
- Не равно
- Меньше или равно
- Больше или равно

9. Что выведет следующий код? \*

1 балл

```

a = "a"
b = "б"
c = "р"
d = "ы"
e = "о"
f = "н"
s1 = b+a+c+a+b+a+f
s2 = c+d+b+a
s3 = f+a+b+e+c
print(s2, s1, sep = " и ")

```

- рыбаибарабан
- Рыба и Барабан
- рыба и барабан
- барабан и рыба
- Ошибка

10. "Простые" типы данных Python \*

1 балл

- int
- char
- str
- double
- bool
- real
- float

11. Что выведет следующий код? \*

2 балла

```

a = 4
b = a ** 2
b = b ** (1/2)
print(b)

```

- 2
- 4
- 16.2
- 0.5
- 4.0

12. В строке `hours, minutes = input().split(':')` что делает `split`? \*

1 балл

- Указывает какой символ будет введен
- Запрещает ввод символов
- Указывает сообщение для ввода
- Указывает конкретный символ разделителя



13. Что выведет следующий код? \*

1 балл

```
a = True
b = False
c = False
print(a+b+c)
```

- 1
- Ошибка
- 0
- False
- True

14. Как найти корень цифры 9 в Python? \*

1 балл

- 9/2
- 9 \*\* 2
- math.sqrt(9)
- sqrt(p)
- 9 \*\* (1/2)

15. Что делает функция round(x)? \*

1 балл

- Дробная часть числа x
- Квадрат числа x
- Модуль числа x
- Преобразование к целому типу данных
- Округление до ближайшего целого

16. Где ошибка? \*

1 балл

```
a = 10
a = a // 7
a = a + a
print(a)
```

Вариант 1

```
a = b + 7
b = a - 3
print(b)
```

Вариант 2

```
a = 10 % 3
b = 1
a = b // a + a
print(a)
```

Вариант 3

17. Что делает функция `abs(x)`? \*

1 балл

- Модуль числа  $x$
- Дробная часть числа  $x$
- Квадрат числа  $x$
- Округление до ближайшего целого
- Преобразование к целому типу данных

18. Что делает функция `int(x)`? \*

1 балл

- Округление до ближайшего целого
- Модуль числа  $x$
- Квадрат числа  $x$
- Дробная часть числа  $x$
- Преобразование к целому типу данных

19. Чему равна переменная  $a$  ? `a=int(10.8)` \*

1 балл

- 11
- 9
- 8
- 10

20. Что выведет следующий код? \*

1 балл

```
a=3
b=a+41
print(b)
```

- 44
- 41
- 3
- 341

21. Чему равна переменная  $a$  ? `a=round(10.8)` \*

1 балл

- 11
- 9
- 10
- 8

22. Что выведет эта программа, если  $x=123$  ? \*

1 балл

```
x = 321
a = x // 100
b = x % 100 // 10
c = x % 10
s = a + b + c
print ('s =', s)
```

- 6
- 321
- 5
- 32

23. Какие, из перечисленных слов, логические операции? \*

1 балл

- round
- not
- map
- near
- or
- and

24. Используя сколько, минимально возможных, переменных можно решить задачу: "сумма цифр трёхзначного числа 256" ? \*

2 балла

- 4
- 0
- 1
- 2
- 3

**Оценивание:**

Всего 26 баллов(24 вопроса)

13 - 17 баллов - "удовлетворительно"

18 - 22 баллов - "хорошо"

23 – 26 балла - "отлично"