

# Поиск элемента с заданными свойствами

## **Ты узнаешь:**

функции и методы для поиска элементов с заданными свойствами;  
структуру программы для поиска элементов.

## **Ты научишься:**

записывать условия для поиска элементов с заданными свойствами;  
находить минимальный и максимальный элементы в одномерном массиве;  
вычислять количество и сумму элементов с заданными свойствами;  
писать программы для поиска элементов.

# Функции и методы для поиска заданного элемента массива

Название	Назначение
<code>max()</code>	максимальный элемент массива
<code>min()</code>	минимальный элемент массива
<code>count(x)</code>	количество элементов со значением x
<code>index(x)</code>	номер первого вхождения элемента x

# Пример 1

Условие задачи:

Найти максимальный и минимальный элементы массива из десяти случайных чисел в диапазоне от 10 до 20.

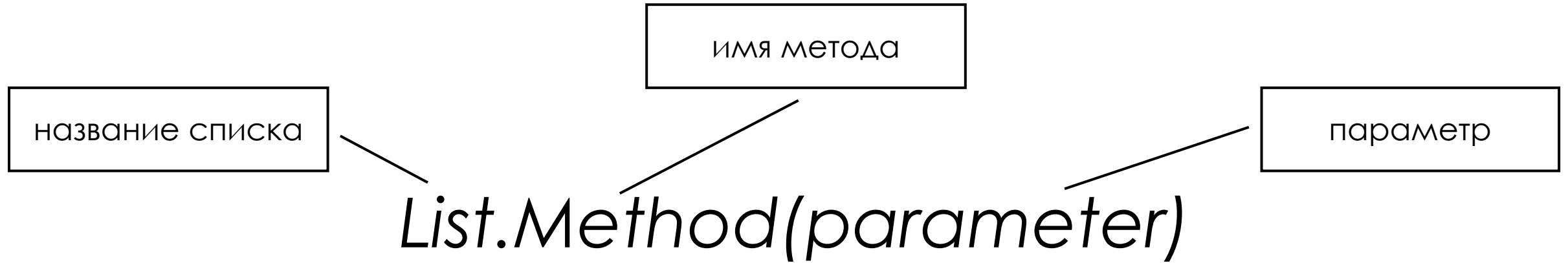
Алгоритм решения:

1. заполнить массив *number* случайными числами;
2. вывести на экран максимальный элемент массива;
3. вывести на экран минимальный элемент массива.

# Решение задачи в Google Colab

```
▶ from random import randint  
number = [randint(10, 20) for x in range(10)]  
print(number)  
print('max = ', max(number))  
print('min = ', min(number))
```

# ИСПОЛЬЗОВАНИЕ МЕТОДОВ



# Пример 2

Условие задачи:

Дан массив итоговых оценок за II четверть для 15 учащихся 9-го класса по информатике. Нужно определить количество отличников, ударников и троечников.

Алгоритм решения:

1. Сформировать массив *mark* из оценок для 15 учащихся;
2. Вывести на экран количество вхождений элемента 5;
3. Вывести на экран количество вхождений элемента 4;
4. Вывести на экран количество вхождений элемента 3.

# Решение задачи в Google Colab

```
[4] mark = [4, 3, 5, 4, 5, 3, 4, 5, 5, 3, 4, 3, 5, 5, 4]
print('кол-во отличников', mark.count(5))
print('кол-во ударников', mark.count(4))
print('кол-во троечников', mark.count(3))
```

# Пример 3

Условие задачи:

*Дан массив цветов.  
Определить индекс цвета,  
название которого  
вводится с клавиатуры.*

Алгоритм решения:

- 1. Сформировать массив цветов;*
- 2. Организовать ввод с клавиатуры названия цвета;*
- 3. Вывести на экран индекс цвета в массиве.*



# Решение задачи в Google Colab

```
color = ['red', 'green', 'blue', 'pink', 'gold', 'orange']  
x = input()  
print(color.index(x))
```

# Пример 4

Условие задачи:

Ученик 9-го класса собирал данные о температуре воздуха в течение 10 дней. Результаты он записал в массив `temp`. Нужно составить программу, которая определяет количество дней с температурой воздуха выше 0.

Алгоритм решения:

1. Присвоить счетчику положительных чисел значение 0;
2. Организовать ввод с клавиатуры массива из 10 целых чисел;
3. В цикле сравнить элементы с 0, если элемент больше 0, то счетчик увеличить на 1;
4. Вывести значение счетчика.

# Решение задачи в Google Colab

```
k = 0--
temp = []
temp = [int(input()) for i in range(10)]
print(temp)
for i in range(10):
    if temp[i] > 0:
        k = k + 1
print('Дни с температурой воздуха выше 0', k)
```

# Пример 5

Условие задачи:

В массиве из 20 случайных чисел в диапазоне от 0 до 50 найти сумму четных элементов.

Алгоритм решения:

1. Присвоить сумме значение 0;
2. Заполнить массив случайными числами, вывести его на экран;
3. В цикле проверить четность элемента, если элемент четный, то прибавить его к сумме;
4. Вывести значение суммы.

# Решение задачи в Google Colab

```
s = 0
from random import randint

number = [randint(0, 50) for x in range(20)]
print(number)
for x in range(20):
    if x % 2 == 0:
        s = s + number[x]
print("сумма четных =", s)
```

# Пример 6

Условие задачи:

В массив записана последовательность целых чисел от 10 до 50. Нужно найти в массиве числа, которые делятся на 3 и на 7 без остатка.

Алгоритм решения:

1. Сформировать массив, вывести его на экран;
2. В цикле вывести на экран элементы, удовлетворяющие условию.

# Решение задачи в Google Colab

```
number = [n + 10 for n in range(40)]
print(number)
for n in range(40):
    if number[n] % 3 == 0 and number[n] % 7 == 0:
        print(number[n])
```