

# Программирование (Python)

Решение задач  
с одномерными массивами

# Конструкция программы

1 этап: Задание массива

2 этап: Заполнение массива элементами

3 этап: Вывод массива на экран

4 этап: Работа с массивом (перебор элементов)

5 этап: Вывод измененного массива (при необходимости)

## 4 этап: Работа с массивом (перебор элементов)

Перебор элементов массива:

просматриваем все элементы массива и, если нужно, выполняем с каждым из них некоторую операцию.

```
for i in range(N):  
    # здесь работаем с A[i]
```



**A[i]** – элемент массива

**i** – номер (индекс) элемента массива (с 0)

# Пример программы

Найти произведение положительных элементов массива.

0 этап	<code>P = 1</code>	# объявление переменной для хранения произведения
1 этап	<code>N = 10</code>	# размер (длина) массива
	<code>A = [0]*N</code>	# выделить память
2 этап	<code>from random import randint</code>	# подключение генератора случайных чисел
	<code>for i in range(N):     A[i] = randint(-10,10)</code>	# заполнение массива
3 этап	<code>for i in range(N):     print("A[" ,i, "]=", A[i])</code>	# вывод массива на экран
4 этап	<code>for i in range(N):     if A[i]&gt;0:         P=P*A[i] print("Произведение + эл-ов=", P)</code>	# работа с массивом, отбор положительных элементов и нахождение их произведения

# **Слайды для дополнительного изучения**

Алгоритмы обработки  
массивов

## Сумма элементов массива

Задача. Найти сумму элементов массива из N элементов.

**?** Какие переменные нужны?

5	2	8	3	1
---	---	---	---	---

```
Sum = 0
for i in range(N):
    Sum += A[i]
print( Sum )
```

i	Sum
	0
0	5
1	7
2	15
3	18
4	19

В стиле Python:

```
print( sum(A) )
```

## Сумма элементов массива (Python)

Задача. Найти сумму элементов массива A.

```
Sum = 0
for x in A:
    Sum += x
print( Sum )
```

для всех  
элементов из A



Не нужно знать размер!

или так:

```
print( sum(A) )
```

встроенная  
функция

## Сумма не всех элементов массива

Задача. Найти сумму чётных элементов массива.

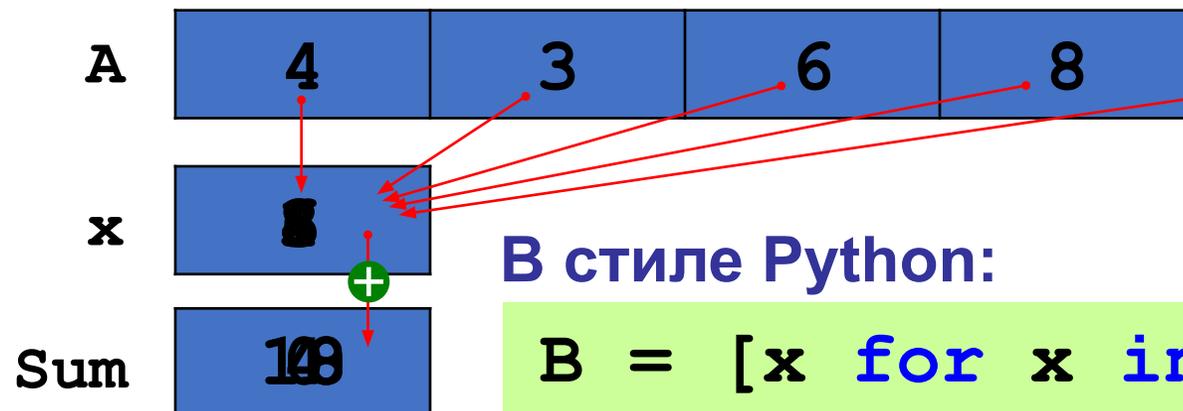
**?** Что делаем с нечётными?

```
Sum = 0
for i in range(N):
    if A[i] % 2 == 0:
        Sum += A[i]
print( Sum )
```

## Сумма не всех элементов массива

Задача. Найти сумму чётных элементов массива.

```
Sum = 0
for x in A:
    if x % 2 == 0:
        Sum += x
print( Sum )
```



отбираем в новый массив все нужные значения

В стиле Python:

```
B = [x for x in A
      if x % 2 == 0]
print( sum(B) )
```

## Подсчёт элементов по условию

Задача. Найти количество чётных элементов массива.

**?** Какие переменные нужны?

```
count = 0
for i in range(N):
    if A[i] % 2 == 0:
        count += 1
print(count)
```

переменная-  
счётчик

**?** Что тут делаем?

## Подсчёт элементов по условию (Python)

Задача. Найти количество чётных элементов массива.

```
count = 0
for x in A:
    if x % 2 == 0:
        count += 1
print( count )
```

В стиле Python:

```
B = [x for x in A
      if x % 2 == 0]
print( len(B) )
```

размер массива

## Среднее арифметическое

Задача. Найти среднее арифметическое элементов массива, которые больше 180 (рост в см).

```
Sum = 0
for x in A:
    if x > 180:
        Sum += x
print( Sum / N )
```



Что плохо?

## Среднее арифметическое

*Задача.* Найти среднее арифметическое элементов массива, которые больше 180 (рост в см).

**?** Какие переменные нужны?

```
Sum = 0
count = 0
for x in A:
    if x > 180:
        count += 1
        Sum += x
print( Sum/count )
```

**?** Что тут делаем?

## Среднее арифметическое (Python)

---

*Задача.* Найти среднее арифметическое элементов массива, которые больше 180 (рост в см).

```
B = [ x for x in A
      if x > 180 ]
print ( sum(B) / len(B) )
```

отбираем нужные

## Задачи

---

- «А»: Напишите программу, которая заполняет массив из 20 элементов случайными числами на отрезке  $[0; 200]$  и считает число элементов, которые делятся на 10.
- «В»: Напишите программу, которая заполняет массив из 20 элементов случайными числами на отрезке  $[0; 200]$  и считает число двузначных чисел в массиве.
- «С»: Напишите программу, которая заполняет массив из 20 элементов случайными числами на отрезке  $[10; 100]$  и считает число пар соседних элементов, сумма которых делится на 3.