Язык программирования Python

Массивы в языке Python

С. В. Чайченков, МБОУ Грушевская СОШ

Массив – совокупность пронумерованных величин одного типа, объединённых общим именем.

В языке Python нет такой структуры данных, как массив.

Для хранения группы однотипных объектов используют ***списки*** (тип данных **`list`**).

Индекс – порядковый номер элемента в массиве.

Нумерация элементов массива всегда начинается с нуля.

Каждый элемент массива обозначается *индексированным именем*:

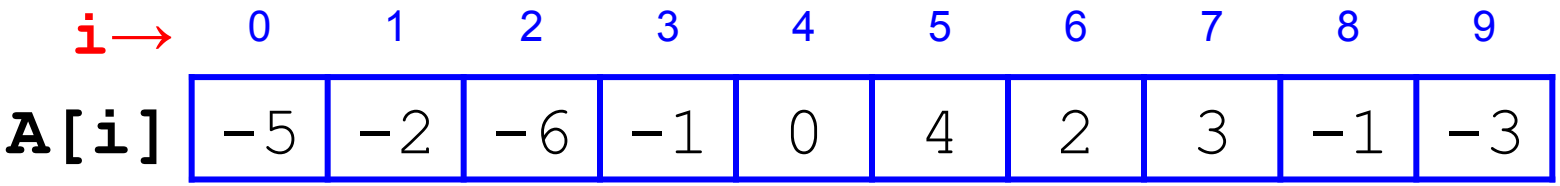
`Имя[индекс]`

Например:

`A[1]` – второй элемент массива A (с индексом 1).

Массивы бывают *одномерные* (*линейные*) и *двумерные* (*прямоугольные*). Далее рассматриваются одномерные массивы.

Одномерный (линейный) массив **A**



A – *имя массива*

i – *индекс элемента*

A[0] = -5, A[1] = -2, . . . A[9] = -3

Индексом может быть не только целое число, но и целое значение переменной или арифметического выражения.

Например: **A[2*i-1] = -2** (при **i=1**).

Перед использованием в программе массив необходимо создать. Обращение к несуществующему элементу вызовет ошибку.

Количество элементов в массиве определяется с помощью функции **len** (length – «длина») . Например: `N = len(A)`

С. В. Чайченков, МБОУ Грушевская СОШ

Вывод массива на экран

1 способ. Весь массив выводится как один объект в квадратных скобках, элементы разделяются запятыми.

```
print (A)
```

На экране:

```
[1, 2, 3, 4, 5]
```

2 способ. Вывод элементов с помощью цикла в одной строке через пробел.

```
for i in range(len(A)):
    print (A[i], end=" ")
print()   # переход на новую строку
```

На экране:

```
1 2 3 4 5
```

Вывод массива на экран

3 способ. Вывод элементов с помощью цикла в столбик.

```
for i in range(len(A)):
   print (A[i])
```

На экране:

```
1
2
3
4
5
```

4 способ. Вывод элементов с помощью цикла в столбик с указанием индексов.

```
for i in range(len(A)):
   print ("A[", i, "]=", A[i])
```

На экране:

```
A[ 0 ]= 1
A[ 1 ]= 2
A[ 2 ]= 3
A[ 3 ]= 4
A[ 4 ]= 5
```

Заполнение массивов

1 способ. Создание массива указанием значений элементов

Массив создается перечислением элементов через запятую в квадратных скобках.

```
A = [1, -2, -3, 5, 7]
```

Если все элементы одинаковые, используется следующий оператор.

```
# массив из 5 элементов
# заполненный нулями
A = [0] * 5
```

Заполнение массивов

2 способ. Ввод с клавиатуры (при небольшом количестве элементов)

```
N = 5                       # размер массива в переменной
B = [0] * N                   # заполнение массива нулями
print ("Введите", N, "элементов массива:")
for i in range(N):              # перебор индексов
    B[i] = int(input())       # ввод числа с клавиатуры
```

Можно в цикл добавить подсказку с индексом вводимого элемента.

```
for i in range(N):                # перебор индексов
    print ("B[", i, "] = ", end="")  # вывод подсказки
    B[i] = int(input())           # ввод числа
```

На экране:

```
B[ 0 ] = 1
B[ 1 ] = 2
B[ 2 ] = 3
B[ 3 ] = 4
B[ 4 ] = 5
```

Заполнение массивов

3 способ. Вычисление элементов по формуле (функция от индекса)

```
N = 5           # размер массива в переменной
C = [0] * N       # заполнение массива нулями
for i in range(N):   # перебор индексов
    C[i] = i**2      # индекс в квадрате
print (C)            # вывод массива
```

На экране:

```
[0, 1, 4, 9, 16]
```

Заполнение массивов

4 способ. Заполнение случайными числами

Функция **randint(a, b)** создаёт случайное целое число из отрезка [a, b].

```
N = 5                       # размер массива в переменной
D = [0] * N                   # заполнение массива нулями
from random import randint  # подключение функции randint
for i in range(N):            # перебор индексов
    D[i] = randint(-5, 5)   # случайные числа от -5 до 5
print (D)                     # вывод массива
```

Возможный результат на экране:

```
[0, -4, -2, 1, 5]
```

Задача 1

Определить средний балл 10 учеников, сдававших ЕГЭ по информатике.

```
# Средний балл учеников
N = 10              # размер массива
A = [0] * N         # заполнение массива нулями
# Ввод значений элементов массива с клавиатуры
print ("Введите оценки:")
for i in range(N):
    print (i+1, "оценка: ", end="")
    A[i] = int(input())
s = 0                       # нач. знач. суммы
for i in range(N):          # перебор индексов
    s = s + A[i]            # добавление к сумме
sb = s/10                   # среднее арифметиче
print ("Средний балл:", sb)
```

```
Введите оценки:
1 оценка: 3
2 оценка: 4
3 оценка: 4
4 оценка: 3
5 оценка: 3
6 оценка: 5
7 оценка: 4
8 оценка: 5
9 оценка: 5
10 оценка: 3
Средний балл: 3.9
```

Задача 2

Подсчитать количество элементов массива, которые больше заданного значения.

```
# Количество элементов массива, соответствующих условию
N = 10
A = [0]*N           # создание массива
from random import randint  # подключение функции randint
for i in range(N):          # заполнение массива
    A[i] = randint(0, 99)   # случайными числами от 0 до 99
print (A)           # вывод массива
x = int(input("x = "))   # ввод значения для условия
k = 0               # начальное значение счетчика
for i in range(N):          # просмотр всех элементов массива
    if A[i] > x:        # если очередной соответ. условию
        k = k+1         # увеличиваем счетчик
print ("Количество элементов больше данного", k)
```

```
[30, 81, 28, 35, 35, 94, 9, 76, 25, 40]
x = 50
Количество элементов больше данного 3
```

Задача 3

В таблице значений среднесуточной температуры за декаду месяца найти самый холодный день и указать его номер.

```
# Минимальный элемент массива
T = [3,5,6,4,2,9,7,5,4,7]  # создание и вывод массива
print ("Среднесуточная температура за декаду")
print (T)
imin = 0                   # считаем первый элемент минимальн.
for i in range(1, 10):  # просмотр элементов со второго
    if T[i] < T[imin]:  # если очередной меньше минимальн.
        imin = i        # сохраняем его индекс
# вывод максимального элемента и его номера
print ("Минимальная температура: ", T[imin])
print ("День номер: ", imin+1)
```

```
Среднесуточная температура за декаду
[3, 5, 6, 4, 2, 9, 7, 5, 4, 7]
Минимальная температура:  2
День номер:  5
```

Задача 4

Определить, есть ли в данном массиве элемент, значение которого равно заданному числу.

```
# Поиск элемента массива, равного заданному значению
N = 10; A = [0]*N          # создание массива
from random import randint # подключение функции randint
for i in range(N):         # заполнение массива
    A[i] = randint(0, 99)  # случайными числами от 0 до 99
print (A)                  # вывод массива
x = int(input("x = "))     # ввод значения для поиска
nx = -1                    # несуществующее значение индекса
for i in range(N):         # просмотр всех элементов массива
    if A[i] == x:          # если очередной равен заданному
        nx = i             # сохраняем его индекс
if nx == -1:               # если индекс не изменился
    print ("Такого значения нет")
else:                      # иначе
    print ("Элемент под номером", nx+1) # вывод номера элемента
```

```
[24, 26, 14, 17, 8, 7, 12, 39, 50, 64]
x = 50
Элемент под номером 9
```

```
[8, 27, 34, 72, 18, 91, 74, 51, 90, 58]
x = 50
Такого значения нет
```

Сортировка массива

Задача. Упорядочить массив в порядке возрастания значений его элементов.

Алгоритм сортировки методом обмена (метод «пузырька»)

Последовательно сравнить пары соседних чисел и при необходимости поменять их местами, и т.д. для каждой пары. За один проход самое большое число окажется на последнем месте. Затем повторить проход до элемента, уже находящегося на своем месте. И т. д.

	A[1]	A[2]	A[3]	A[4]	A[5]
Исходные значения	5	4	2	3	1
1-й проход					
2-й проход					
3-й проход					
4-й проход					

Сортировка массива

Задача. Упорядочить массив в порядке возрастания значений его элементов.

Алгоритм сортировки методом обмена (метод «пузырька»)

Последовательно сравнить пары соседних чисел и при необходимости поменять их местами, и т.д. для каждой пары. За один проход самое большое число окажется на последнем месте. Затем повторить проход до элемента, уже находящегося на своем месте. И т. д.

	A[1]	A[2]	A[3]	A[4]	A[5]
Исходные значения	5	4	2	3	1
1-й проход	5	4	2	3	1
2-й проход					
3-й проход					
4-й проход					

Сортировка массива

Задача. Упорядочить массив в порядке возрастания значений его элементов.

Алгоритм сортировки методом обмена (метод «пузырька»)

Последовательно сравнить пары соседних чисел и при необходимости поменять их местами, и т.д. для каждой пары. За один проход самое большое число окажется на последнем месте. Затем повторить проход до элемента, уже находящегося на своем месте. И т. д.

	A[1]	A[2]	A[3]	A[4]	A[5]
Исходные значения	5	4	2	3	1
1-й проход	4	2	3	1	5
2-й проход	4	2	3	1	5
3-й проход					
4-й проход					

Сортировка массива

Задача. Упорядочить массив в порядке возрастания значений его элементов.

Алгоритм сортировки методом обмена (метод «пузырька»)

Последовательно сравнить пары соседних чисел и при необходимости поменять их местами, и т.д. для каждой пары. За один проход самое большое число окажется на последнем месте. Затем повторить проход до элемента, уже находящегося на своем месте. И т. д.

	A[1]	A[2]	A[3]	A[4]	A[5]
Исходные значения	5	4	2	3	1
1-й проход	4	2	3	1	5
2-й проход	2	3	1	4	5
3-й проход	2	3	1	4	5
4-й проход					

С. В. Чайченков, МБОУ Грушевская СОШ

Сортировка массива

Задача. Упорядочить массив в порядке возрастания значений его элементов.

Алгоритм сортировки методом обмена (метод «пузырька»)

Последовательно сравнить пары соседних чисел и при необходимости поменять их местами, и т.д. для каждой пары. За один проход самое большое число окажется на последнем месте. Затем повторить проход до элемента, уже находящегося на своем месте. И т. д.

	A[1]	A[2]	A[3]	A[4]	A[5]
Исходные значения	5	4	2	3	1
1-й проход	4	2	3	1	5
2-й проход	2	3	1	4	5
3-й проход	2	1	3	4	5
4-й проход	2	1	3	4	5

С. В. Чайченков, МБОУ Грушевская СОШ

Сортировка массива

Задача. Упорядочить массив в порядке возрастания значений его элементов.

Алгоритм сортировки методом обмена (метод «пузырька»)

Последовательно сравнить пары соседних чисел и при необходимости поменять их местами, и т.д. для каждой пары. За один проход самое большое число окажется на последнем месте. Затем повторить проход до элемента, уже находящегося на своем месте. И т. д.

	A[1]	A[2]	A[3]	A[4]	A[5]
Исходные значения	5	4	2	3	1
1-й проход	4	2	3	1	5
2-й проход	2	3	1	4	5
3-й проход	2	1	3	4	5
4-й проход	1	2	3	4	5

С. В. Чайченков, МБОУ Грушевская СОШ

Сортировка массива

```
# Сортировка элементов массива (метод пузырька по неубыванию)
N = 10; A = [0]*N          # создание массива
from random import randint # подкл. генератора случайных чисел
for i in range(N):         # заполнение массива
    A[i] = randint(0, 99)  # случайными числами от 0 до 99
print (A)              # вывод массива
for k in range(1, N-1):    # номер прохода
    for i in range(N-k):   # просмотр за один проход
        if A[i] > A[i+1]:  # если соседние неупорядочены
            A[i], A[i+1] = A[i+1], A[i]    # меняем их местами
    print (A)          # вывод текущих значений массива
```

```
[92, 47, 84, 49, 24, 73, 98, 19, 65, 90]
[47, 84, 49, 24, 73, 92, 19, 65, 90, 98]
[47, 49, 24, 73, 84, 19, 65, 90, 92, 98]
[47, 24, 49, 73, 19, 65, 84, 90, 92, 98]
[24, 47, 49, 19, 65, 73, 84, 90, 92, 98]
[24, 47, 19, 49, 65, 73, 84, 90, 92, 98]
[24, 19, 47, 49, 65, 73, 84, 90, 92, 98]
[19, 24, 47, 49, 65, 73, 84, 90, 92, 98]
[19, 24, 47, 49, 65, 73, 84, 90, 92, 98]
```

Используемые материалы:

- *Босова Л.Л. Информатика. 8-9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам – М. : БИНОМ. Лаборатория знаний, 2020.*
- *Поляков К.Ю. Информатика. 10 класс. Базовый и углубленный уровни : в 2ч. Ч. 2 – М. : БИНОМ. Лаборатория знаний, 2018.*