

Это
Эта банка
банка



Одна банка – это
переменная.

А это много
банок.



Много банок – это уже **МАССИВ**
переменных.

В жизни мы часто встречаемся с массивами



Массивы

Массив переменных

- Каждая из переменных, рассмотренных в предыдущих уроках способна одновременно хранить лишь один элемент информации.
- Чтобы сохранить второй, необходимо создать еще одну переменную. Но что делать, если нужно хранить множество элементов однородных типов данных. Будет весьма неудобно создавать для каждого элемента переменную.
- Для решения этой задачи придуманы **МАССИВЫ** переменных.

Массив переменных



Массив переменных

- **Массив** — это совокупность переменных, которая позволяет хранить несколько однотипных значений, объединённых общим именем.

Массив переменных



Массив переменных

- Все значения массива объединены под одним именем;
- При этом каждая переменная в массиве является самостоятельной единицей под названием — **элемент**.
- Каждый **элемент** имеет свой порядковый номер — **индекс**.
- По индексу можно обращаться к конкретному элементу массива.

- Нумерация элементов в массиве начинается с **нуля**.

Массив переменных

Исходя из вышеописанных утверждений — общая схема представления массива будет выглядеть так:



Массив переменных



Массив переменных

0	1	2	3	4	5	6	7
4	5	3	4	5	2	5	5

Расположение массива в памяти

- Массив располагается в памяти **последовательно**, элемент за элементом. Сначала лежит **нулевой**, потом первый и т.д
- Элементы располагаются **по возрастанию адреса**: один элемент массива отстоит от другого на количество байт, равное базовому типу массива.

Расположение массива в памяти

- Каждый элемент массива имеет свою собственную размерность, которая напрямую зависит от типа всего массива.
- Например, если массив имеет тип данных `int` — размер каждого элемента в нем — 4 байта.
- Таким образом, общий размер всего массива вычисляется по формуле:

ОБЩИЙ_РАЗМЕР = РАЗМЕР_ТИПА_ДАННЫХ *
КОЛИЧЕСТВО_ЭЛЕМЕНТОВ_В_МАССИВЕ

Синтаксис объявления массива

- Общий синтаксис объявления массива:

`тип_данных имяМассива[количество_элементов];`

Синтаксис объявления массива

- Пример объявления целого массива:

```
int numberArray[10];
```

Синтаксис объявления массива

- Пример объявления целого массива:

```
const int NUMBER_ARRAY_SIZE = 10;  
int numberArray[NUMBER_ARRAY_SIZE ];
```

Обращение к элементам массива

```
const int NUMBER_ARRAY_SIZE =5;
int numberArray [NUMBER_ARRAY_SIZE]; // создание массива
numberArray[2]=25; // запись значения 25 в элемент с индексом 2
cout<< numberArray [2] <<"\n";
// вывод на экран значения элемента с индексом 2
```

Варианты инициализации массива

Первый способ — инициализация **при создании**.

```
const int NUMBER_ARRAY_SIZE =3;
```

```
int numberArray[NUMBER_ARRAY_SIZE] = {1 , 30, 2};
```

Варианты инициализации массива

При инициализации в момент создания массива, размер массива можно не указывать, компилятор автоматически определит размер массива по числу элементов в списке инициализации.

Например:

```
int numberArray[] = {1 , 30, 2};
```

Варианты инициализации массива

Если число элементов в списке инициализации меньше чем число элементов массива, то оставшиеся значения автоматически заполняются нулями

Например:

```
int numberArray[5] = {1 , 30, 2};
```

такая запись эквивалентна записи:

```
int numberArray[5] = {1 , 30, 2, 0, 0};
```

Варианты инициализации массива

Если значений в списке инициализации больше чем количество элементов массива, то происходит ошибка на этапе компиляции:

```
int numberArray[2] = {1 , 30, 7};
```

Варианты инициализации массива

Второй способ — инициализация массива при **ПОМОЩИ ЦИКЛА**.

```
const int NUMBER_ARRAY_SIZE =5;
int numberArray[NUMBER_ARRAY_SIZE ];
for (int i=0; i< NUMBER_ARRAY_SIZE ; i++)
{
    cout<<«\nВведите элемент\n";
    cin>> numberArray[i];
}
```

Вывод содержимого массива на экран

```
for (int i=0; i< NUMBER_ARRAY_SIZE; i++)  
{  
    cout<<"Значение элемента массива " << i << " = ";  
    cout << numberArray[i] << "\n";  
}
```

Копирование элементов массива

```
const int NUMBER_ARRAY_SIZE =5;
int numberArray[NUMBER_ARRAY_SIZE];
int secondNumberArray[NUMBER_ARRAY_SIZE];
for (int i=0; i< NUMBER_ARRAY_SIZE; i++)
{
    cout<<«\nВведите элемент\n";
    cin>> numberArray[i];
    secondNumberArray[i] = numberArray[i];
    cout << secondNumberArray[i] <<"\n";
}
```

Копирование элементов массива

```
int numberArrayC[NUMBER_ARRAY_SIZE];  
for (int i=0; i< NUMBER_ARRAY_SIZE; i++)  
{  
    numberArrayC [i] = numberArray[i];  
}  
for (int i=0; i< NUMBER_ARRAY_SIZE; i++)  
{  
    cout<< numberArrayC [i] << "\t";  
}  
cout << endl;
```

Задание

- Создайте массив размером 10 ячеек;
- Попросите пользователя заполнить массив отрицательными и положительными значениями, вперемешку;
- Подсчитайте количество отрицательных и положительных элементов массива и выведите эту информацию на экран.

Подсчет количества отрицательных и положительных элементов массива

```
int negativeNumberCount = 0, positiveNumberCount = 0;
for (int i=0;i<size;i++)
{
    if(numberArray[i]<0)
    {
        negativeNumberCount++;
    }
    else positiveNumberCount ++;
}
```

Задание

- Создайте массив произвольного размера;
- Подсчитайте сумму отрицательных и сумму положительных элементов массива и выведите эту информацию на экран.

Подсчет количества отрицательных и положительных элементов массива

```
int negativeNumberSum = 0, positiveNumberSum = 0;
for (int i=0;i<size;i++)
{
    if (numberArray[i]<0)
    {
        negativeNumberSum += numberArray[i];
    }
    else positiveNumberSum += numberArray[i];
}
```

Задача

Продолжение предыдущей задачи. Найти номера и значения элементов массива содержащих максимальное и минимальное значение массива.

Нахождение максимального и минимального элементов

массива

```
int maxValue = numberArray[0], minValue = numberArray[0];
```

```
int maxElement = 0, minElement = 0;
```

```
for (int i=1; i<numberArraySize; i++) {
```

```
    if(minValue > numberArray[i]) {
```

```
        minValue = numberArray[i];
```

```
        minElement = i;
```

```
    }
```

```
    if(maxValue < numberArray[i]) {
```

```
        maxValue = numberArray[i];
```

```
        maxElement = i;
```

```
    }
```

```
}
```

Задание

- Продолжение предыдущего задания;
- Подсчитайте количество и сумму чётных элементов массива и выведите эту информацию на экран.
- Найдите среднее арифметическое значение чётных чисел

Подсчёт суммы чётных элементов массива

```
int parityNumberCount = 0;
const int PARITY_CHECK = 2;
for (int i=0;i<size;i++)
{
    if(numberArray[i] % PARITY_CHECK == 0)
    {
        parityNumberCount ++;
    }
}
```

Задание

- Замените ручной ввод массива, заполнение массива случайными значениями от -20 до +20;

Задание

```
srand(time(0));
```

```
const int NUMBER_ARRAY_SIZE = 10;
```

```
const int MIN_RANDOM_VALUE = -10;
```

```
const int MAX_RANDOM_VALUE = 10;
```

```
const float FACTOR_FOR_DECIMAL = 10;
```

```
float numberArray[NUMBER_ARRAY_SIZE];
```

```
for (int i = 0; i < NUMBER_ARRAY_SIZE; i++)
```

```
{
```

```
    int countPossibleValue = (MAX_RANDOM_VALUE - MIN_RANDOM_VALUE) * FACTOR_FOR_DECIMAL;
```

```
    float randomNumber = MIN_RANDOM_VALUE + rand() % countPossibleValue / FACTOR_FOR_DECIMAL;
```

```
    numberArray[i] = randomNumber;
```

```
}
```

```
for (int i = 0; i < NUMBER_ARRAY_SIZE; i++)
```

```
{
```

```
    cout << numberArray[i] << "\t";
```

```
}
```

```
cout << endl;
```