# Лекция № 08

***Указатели***

***Массивы***

# Ключевые слова

Массив, элемент массива, индекс, объявление массивов, инициализация массивов, нумерация массивов, доступ к элементам массива.

# Проблема ???

Каждая из простых переменных способна хранить лишь один элемент информации.

Чтобы сохранить второй элемент информации, необходимо создать еще одну переменную.

Переменные такого рода называются скалярными переменными.

Но что делать, если  необходимо хранить множество элементов однородных данных?

Будет весьма неудобно создавать для каждого элемента переменную.

А что если требуется работать со многими тысячами записей сотрудников?

***Задача очень быстро становится невыполнимой !!!***

# Решение проблемы …

По счастью, у большинства проблем есть решения.

В нашем случае таким решением являются массивы.

# Массивы

Массив –

это специальная группа переменных, которая позволяет хранить много однотипных значений.

Элемент –

отдельная переменная (значение) в массиве.

Индекс –

это число, которое указывает, к какому из элементов массива обращается пользователь.

# Массивы

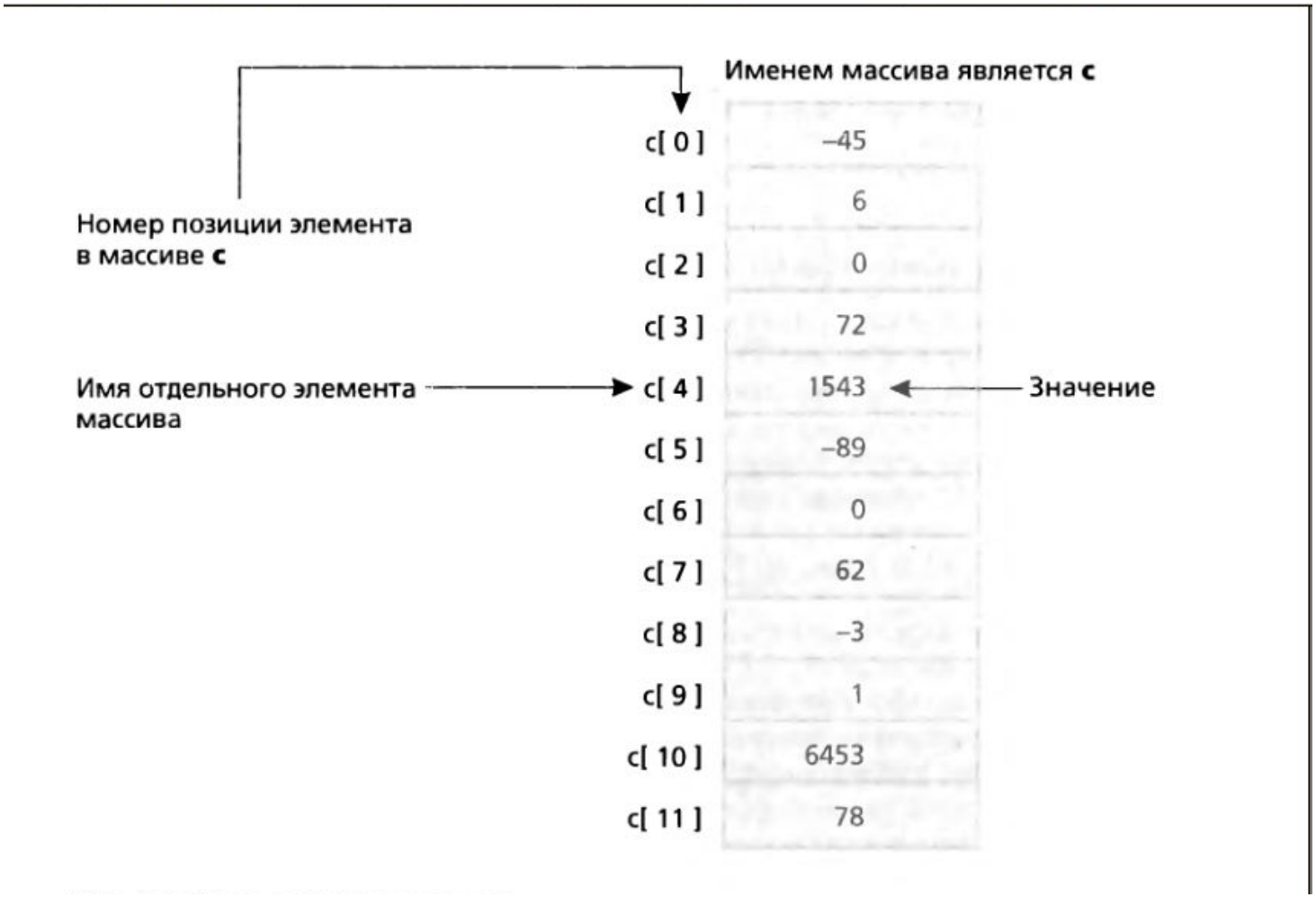
Массивы-

не **изменяют своего размера** в течение всего времени исполнения программы.

Массив -

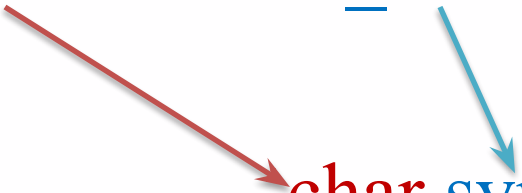
это группа последовательных ячеек памяти, имеющих один и тот же тип.

# Схема массива из 12 элементов



# Объявление массивов

тип_данных имя_массива[число_элементов];



char symb[15];

int number[20];

# Помните !!!

Нумерация массива начинается с **0 (нуля)**

**Пример**

**string month[12];**

**индекс меняется**

**от 0 до 11**

# Помните !!!

1. Число в квадратных скобках должно быть **константным** выражением,
2. то есть для определения числа элементов массива нельзя использовать простую **переменную**.
3. Размер массива определяется на этапе компиляции.

# Инициализация массивов

**тип_данных имя_массива[число_элементов] = {знач1, знач2, знач3, ...};**

float num[3] = {0.25, .876, 3.0};
char symb[6] = {‘П,'р','и','в','е','т'};
int a[10] = {0}; - целочисленный массив из 10 нулевых элементов

**число_элементов** можно опускать. Компьютер самостоятельно определит размер массива исходя из числа элементов в списке инициализации.

char letter[] = {‘a’,’b’,’c’,’d’,’e’};

# Доступ к элементам массива

Каждый элемент массива можно считать самостоятельной скалярной переменной.

Чтобы получить доступ к отдельному элементу, следует воспользоваться именем массива и оператором индекса, указав номер элемента:

имя_массива[номер_индекса];

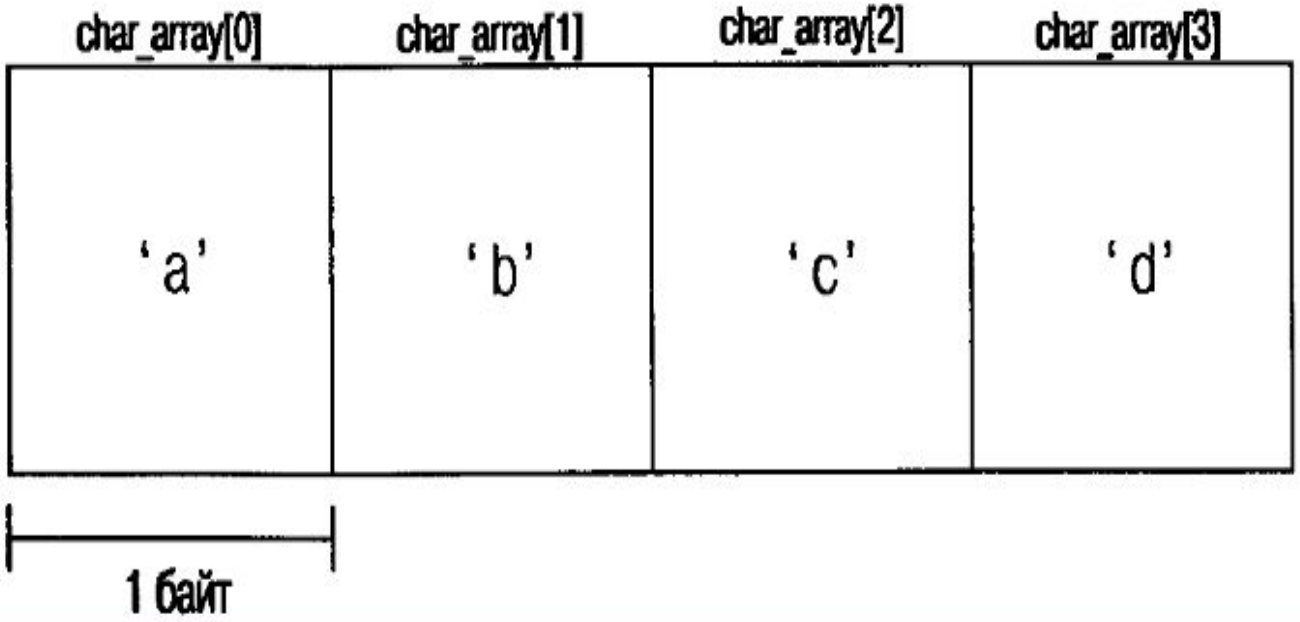
arr[9] ;

# Пример

```
char  symb[6] = {‘П’, ‘р’, ‘и’, ‘в’, ‘е’, ‘т’};

cout << symb[2] ;        => и

float  num[3] = {0.25, .876, 3.0};

cout << num[0] ;         => 0.25
```

# Пример

**char char_array = {‘a’,’b’,’c’,’d’}**



# Осторожно !!!

int  array[10];
0, ……., 9 - изменение индекса массива

***Выход за пределы массива ни во время компиляции, ни во время исполнения не контролируется !!!***

array[-1];
array[12];

Что будет в процессе выполнения программы никто не знает.

В C# и java такие моменты контролируются и программа аварийно завершается.