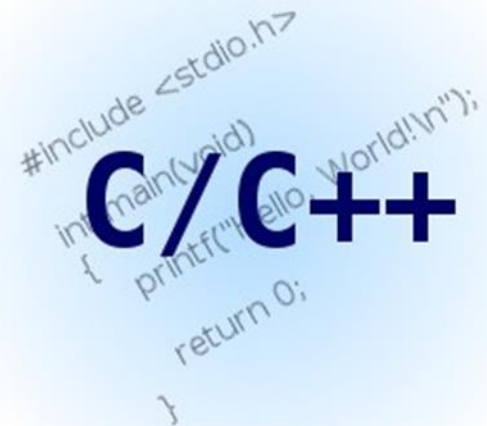


Алгоритмизация и программирование

Программирование на C/C++
(ч.5 – массивы)



```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>

Массивы - основы

- Массив это структура данных, представленная в виде группы ячеек одного типа, объединенных под одним общим именем.
- Массивы используются для обработки большого количества однотипных данных.
- Имя массива является указателем, что такое указатели – рассмотрим в следующих лекциях.

Массивы - основы



Массивы - основы

- Отдельная ячейка данных массива называется элементом массива.
- Элементами массива могут быть данные любого типа.
- Массивы могут иметь как одно, так и более одного измерений.
- В зависимости от количества измерений массивы делятся на одномерные массивы, двумерные массивы, трёхмерные массивы и так далее...

Одномерные массивы в C++

- Одномерный массив — массив, с одним параметром, характеризующим количество элементов одномерного массива.
- Фактически одномерный массив — это массив, у которого может быть только одна строка, и n-е количество столбцов.
- Столбцы в одномерном массиве — это элементы массива.
- На рисунке показана структура целочисленного одномерного массива **a**.
- Размер этого массива — 16 ячеек.

5	-12	-12	9	10	0	-9	-12	-1	23	65	64	11	43	39	-15
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

Одномерные массивы в C++

5	-12	-12	9	10	0	-9	-12	-1	23	65	64	11	43	39	-15
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

- Заметьте, что максимальный индекс одномерного массива **a** равен 15, но размер массива 16 ячеек, потому что нумерация ячеек массива всегда начинается с 0.
- Индекс ячейки – это целое неотрицательное число, по которому можно обращаться к каждой ячейке массива и выполнять какие-либо действия над ней (ячейкой).

```
//синтаксис объявления одномерного массива в C++:  
/*тип данных*/ /*имя массива*/[/*размерность*/];  
//пример объявления одномерного массива, изображенного на рисунке  
int a[16];
```

Одномерные массивы в C++

```
//синтаксис объявления одномерного массива в C++:  
/*тип данных*/ /*имя массива*/[/*размерность*/];  
//пример объявления одномерного массива, изображенного на рисунке  
int a[16];
```

- Всегда сразу после имени массива идут квадратные скобки, в которых задаётся размер одномерного массива, этим массив и отличается от всех остальных переменных.

```
//ещё один способ объявления одномерных массивов  
int mas[10], a[16];
```

- Объявлены два одномерных массива **mas** и **a** размерами 10 и 16 соответственно.
- При таком способе объявления все массивы будут иметь одинаковый тип данных, в нашем случае - **int**.

Одномерные массивы в C++

```
// массивы могут быть инициализированы при объявлении:  
int a[10] = { 5, -12, -12, 9, 10, 0, -9, -12, -1, 7};
```

- Инициализация одномерного массива выполняется в фигурных скобках после знака **равно**, каждый элемент массива отделяется от предыдущего запятой.

```
int a[] = { 5, -12, -12, 9, 10, 0, -9, -12, -1, 7};
```

- В данном случае компилятор сам определит размер одномерного массива.
- Размер массива можно не указывать только при его инициализации, при обычном объявлении массива обязательно нужно указывать размер массива.

Рассмотрим пример программы:

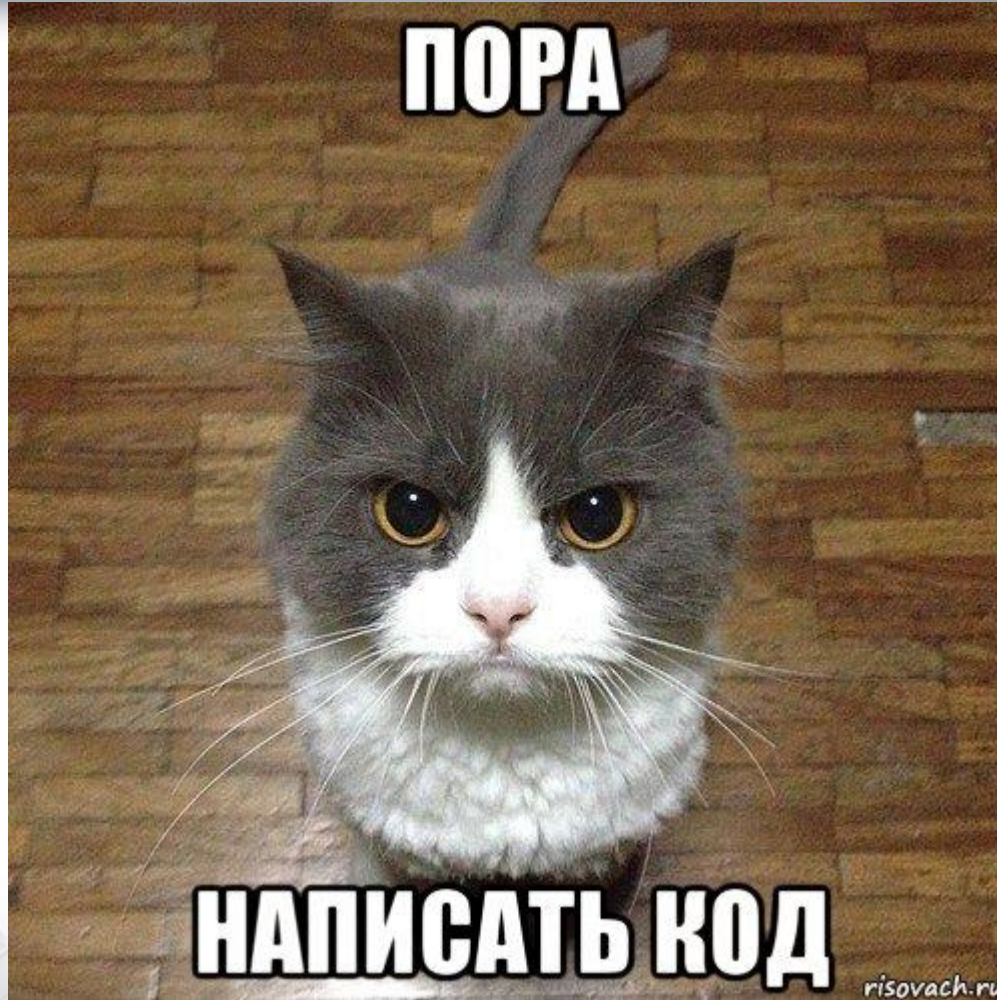
```
#include <iostream>
using namespace std;

int main()
{
    cout << "obrabotka massiva" << endl;
    int array1[16] = { 5, -12, -12, 9, 10, 0, -9,
                    -12, -1, 23, 65, 64, 11, 43, 39, -15 };
    cout << "index" << "\t\t" << "element massiva" << endl;
    for (int counter = 0; counter < 16; counter++)
    {
        cout << "array1[" << counter << "]" << "\t\t"
             << array1[counter] << endl;
    }
    return 0;
}
```



НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрація



Одномерные массивы в C++

- Разработаем ещё одну программу на обработку одномерного массива в C++.
- Программа должна последовательно считывать десять введённых чисел с клавиатуры.
- Все введённые числа просуммировать, результат вывести на экран.



```
#include <iostream>
using namespace std;

int main()
{
    int arr[10];
    cout << "elementi massiva: " << endl;
    for ( int counter = 0; counter < 10; counter++ )
        cin >> arr[counter];
    int sum = 0;
    for ( int counter = 0; counter < 10; counter++ )
        sum += arr[counter];
    cout << "arr = {";
    for ( int counter = 0; counter < 10; counter++ )
        cout << arr[counter] << " ";
    cout << "}\nsum = " << sum << endl;
    return 0;
}
```



Демонстрація



Двумерные массивы в C++

- До этого момента мы рассматривали одномерные массивы, которыми не всегда можно ограничиться.
- Допустим, необходимо обработать некоторые данные из таблицы. В таблице есть две характеристики: количество строк и количество столбцов.
- И в двумерном массиве, кроме количества элементов массива, есть такие характеристики как, количество строк и количество столбцов двумерного массива.
- То есть, визуально, двумерный массив — это обычная таблица, со строками и столбцами.

Двумерные массивы в C++

- Фактически двумерный массив — это одномерный массив одномерных массивов.
- Структура двумерного массива, с именем a , размером m на n показана ниже

$a[0][0]$	$a[0][1]$	$a[0][2]$	$a[0][3]$...	$a[0][n]$
$a[1][0]$	$a[1][1]$	$a[1][2]$	$a[1][3]$...	$a[1][n]$
$a[2][0]$	$a[2][1]$	$a[2][2]$	$a[2][3]$...	$a[2][n]$
...
$a[m][0]$	$a[m][1]$	$a[m][2]$	$a[m][3]$...	$a[m][n]$

Двумерные массивы в C++

```
// синтаксис объявления двумерного массива  
/*тип данных*/ /*имя массива*/ [/*к-во  
строк*/] [/*количество столбцов*/];
```

В объявлении двумерного массива, также как и в объявлении одномерного массива, нужно указать:

- тип данных;
- имя массива.

После чего, в первых квадратных скобках указывается количество строк двумерного массива, во вторых квадратных скобках — количество столбцов двумерного массива. Двумерный массив визуально отличается от одномерного второй парой квадратных скобок.

Двумерные массивы в C++

```
// пример объявления двумерного массива:  
int a[5][3];
```

- **a** - имя целочисленного массива
- число в первых квадратных скобках указывает количество строк двумерного массива, в данном случае их 5;
- число во вторых квадратных скобках указывает количество столбцов двумерного массива, в данном случае их 3.

Двумерные массивы в C++

```
// инициализация двумерного массива:  
int a[5][3] = { {4, 7, 8}, {9, 66, -1}, {5, -5, 0},  
               {3, -3, 30}, {1, 1, 1} };
```

- В данном массиве 5 строк, 3 столбца.
- После знака «присвоить» ставятся общие фигурные скобки, внутри которых ставится столько пар фигурных скобок, сколько должно быть строк в двумерном массиве, причём эти скобки разделяются запятыми.
- В каждой паре фигурных скобок записывать через запятую элементы двумерного массива.

Двумерные массивы в C++

- Так как в массиве пять строк, то и внутренних пар скобок тоже пять.
- Во внутренних скобках записаны по три элемента, так как количество столбцов — три.
- Графически наш массив будет выглядеть, как двумерная таблица

4 a[0][0]	7 [0][1]	8 [0][2]
9 a[1][0]	66 [1][1]	-1 [1][2]
5 a[2][0]	-5 [2][1]	0 [2][2]
3 a[3][0]	-3 [3][1]	30 [3][2]
1 a[4][0]	1 [4][1]	1 [4][2]

Пример программы

```
#include <iostream>
using namespace std;
int main()
{
// 1-условно "стенки лабиринта"
// 2-"правильный путь, выход из лабиринта"
// 0-"ложный путь"
    int mas[33][20] = {
{1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,},
{1,2,1,0,0,1,0,1,2,2,2,1,1,1,1,0,0,0,0,1,},
.....
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,} };
    for (int i = 0; i < 33; i++) {
        for (int j = 0; j < 20; j++)
            if (mas[i][j] == 1) {
                cout << static_cast<char>(176);
                cout << static_cast<char>(176);
            } else cout << "  "; // вывести два пробела
        cout << endl;
    }
    return 0;
}
```



Демонстрація



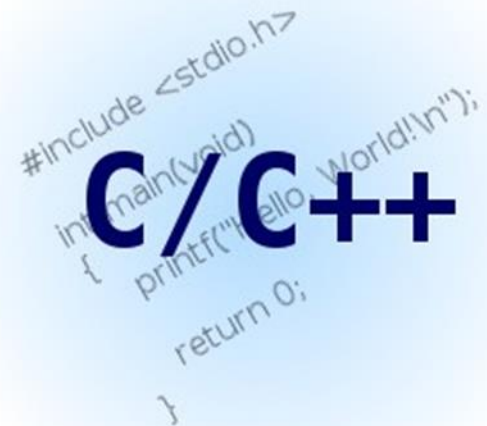


НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА



Алгоритмізація і програмування

Програмування на C/C++
(ч.5 – масиви)



```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>