

---

# Массивы в C++

Преп.: Елизавета Голенок  
mail: [golenok-ea5@narod.ru](mailto:golenok-ea5@narod.ru)

---



---

# Массив

---

- ❖ Массив - структура данных, представленная в виде “ячеек” одного типа, объединенных одним единым именем. Массивы используют для большого количества однотипных данных.



---

# Массив

---

|      |      |      |      |      |      |      |      |      |      |       |       |       |       |       |       |
|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|
| 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] |



# Массивы

```
//синтаксис объявления одномерного массива в C++:  
/*тип данных*/ /*имя одномерного массива*/[/*размерность одномерного массива*/];  
//пример объявления одномерного массива, изображенного на рисунке 1:  
int a[16];
```

|   |  |
|---|--|
| 1 | //ещё один способ объявления одномерных массивов |
| 2 | <b>int</b> mas[10], a[16];                       |



---

# Массивы

---

```
// массивы могут быть инициализированы при объявлении:
```

```
int a[16] = { 5, -12, -12, 9, 10, 0, -9, -12, -1, 23, 65, 64, 11, 43, 39, -15 };
```



---

# Массивы

---

```
int a[]={5,-12,-12,9,10,0,-9,-12,-1,23,65,64,11,43,39,-15};
```



# Массивы

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{
    int array1[16] = { 5, -12, -12, 9, 10, 0, -9,
                     -12, -1, 23, 65, 64, 11, 43, 39, -15 }; // объявление и инициализация одномерного массива
    cout << "index" << "\t\t" << "element " << endl; // печать заголовков
    for (int counter = 0; counter < 16; counter++) //начало цикла
    {
        //вывод на экран индекса ячейки массива, а затем содержимого этой ячейки, в нашем случае - это целое число
        cout << "array[" << counter << "]" << "\t\t" << array1[counter] << endl;
    }
    system("pause");
    return 0;
}
```



# Двумерные массивы

|         |         |         |         |       |         |
|---------|---------|---------|---------|-------|---------|
| 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

```
int a[5][3];
int b[5][3] = { {4, 7, 8}, {9, 66, -1}, {5, -5, 0}, {3, -3, 30}, {1, 1, 1} };
```

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{
    // 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}, // инициализация двумерного массива
                        {1,2,1,0,0,1,0,1,2,2,2,1,1,1,1,0,0,0,0,1},
                        {1,2,1,1,0,1,0,1,2,1,2,2,2,2,1,0,1,1,0,1},
                        {1,2,2,2,2,2,2,1,2,1,1,1,1,2,1,0,0,1,0,1},
                        {1,1,1,1,1,1,2,1,2,1,0,0,1,2,1,1,0,1,0,1},
                        {1,0,0,1,0,0,2,2,2,1,1,0,0,2,0,0,0,1,0,1},
                        {1,0,1,1,0,1,1,1,1,1,0,0,1,2,1,1,1,1,0,1},
                        {1,0,0,0,0,0,0,0,0,1,1,1,1,2,1,0,0,0,0,1},
                        {1,1,1,1,1,1,0,1,1,1,2,2,2,2,1,0,1,1,1,1},
                        {1,1,0,0,0,1,0,0,1,1,2,1,1,1,1,0,0,0,0,1},
                        {1,0,0,1,0,0,0,0,0,1,2,2,2,2,1,1,1,1,0,1},
                        {1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,0,0,0,0,1},
```



```

{1,2,1,2,1,2,2,2,1,0,1,1,1,1,1,1,0,1,2,1,},
{1,2,1,2,1,1,1,1,1,0,0,0,1,0,1,0,0,1,2,1,},
{1,2,1,2,2,1,0,0,1,1,1,0,0,0,1,0,1,1,2,1,},
{1,2,1,1,2,1,1,0,0,0,0,0,1,0,1,0,0,1,2,1,},
{1,2,1,1,2,1,0,0,1,1,1,1,1,1,1,1,1,1,2,1,},
{1,2,1,1,2,1,1,0,1,2,2,2,2,2,2,2,2,2,2,1,},
{1,2,1,1,2,1,0,0,1,2,1,1,1,1,1,1,1,1,1,1,},
{1,2,1,1,2,1,0,1,1,2,1,1,1,1,1,1,1,1,2,2,},
{1,2,1,1,2,1,0,0,1,2,1,1,2,2,2,2,2,2,2,1,},
{1,2,1,1,2,1,0,1,1,2,1,1,2,1,1,1,1,1,1,1,},
{1,2,1,1,2,1,0,0,1,2,1,1,2,1,0,0,0,1,0,1,},
{1,2,2,2,2,1,0,1,1,2,2,2,2,0,0,1,0,0,0,1,},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,} };

```

// два цикла - внутренний и внешний, осуществляющие обращение к каждому элементу массива

```

for (int i = 0; i < 33; i++) //переключение по строкам

```

```

{

```

```

    for (int j = 0; j < 20; j++)// переключение по столбцам

```

```

        if (mas[i][j] == 1)

```

```

        {

```

```

            cout << '|';

```

```

            cout << '|';

```

```

        } else

```

```

            cout << "  "; // вывести два пробела

```

```

    cout << endl;

```

```

,

```



---

# Задачи

---

- ❖ Посчитать сумму элементов в массиве из 10 элементов
- ❖ Найти среднее арифметическое в массиве из 20 элементов
- ❖ Найти минимум в массиве из 7 элементов
- ❖ Посчитать сумму элементов по главной диагонали (7x7)
- ❖ Найти сумму четных элементов в массиве из 15 элементов



---

# Домашнее задание

---

- ❖ Найти произведение последних 10 чисел в массиве из 30 элементов
- ❖ Найти сумму чисел, которые делятся на 7
- ❖ Посчитать произведение элементов побочной диагонали (11x11)
- ❖ Найти максимум в массиве из 3 элементов