

Тема 1.5

Массивы

скалярных данных

Одномерные массивы.

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]

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

где, `int` — целочисленный тип данных;

`a` — имя одномерного массива;

16 — размер одномерного массива, 16 ячеек.

Работа с одномерными массивами.

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

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

```
1 // массивы могут быть инициализированы при объявлении:
2 int a[16] = { 5, -12, -12, 9, 10, 0, -9, -12, -1, 23, 65, 64, 11, 43, 39, -15 }; // инициализация одномерного массива
```

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

```
1 int a[]={5,-12,-12,9,10,0,-9,-12,-1,23,65,64,11,43,39,-15}; // инициализации массива без определения его размера
```

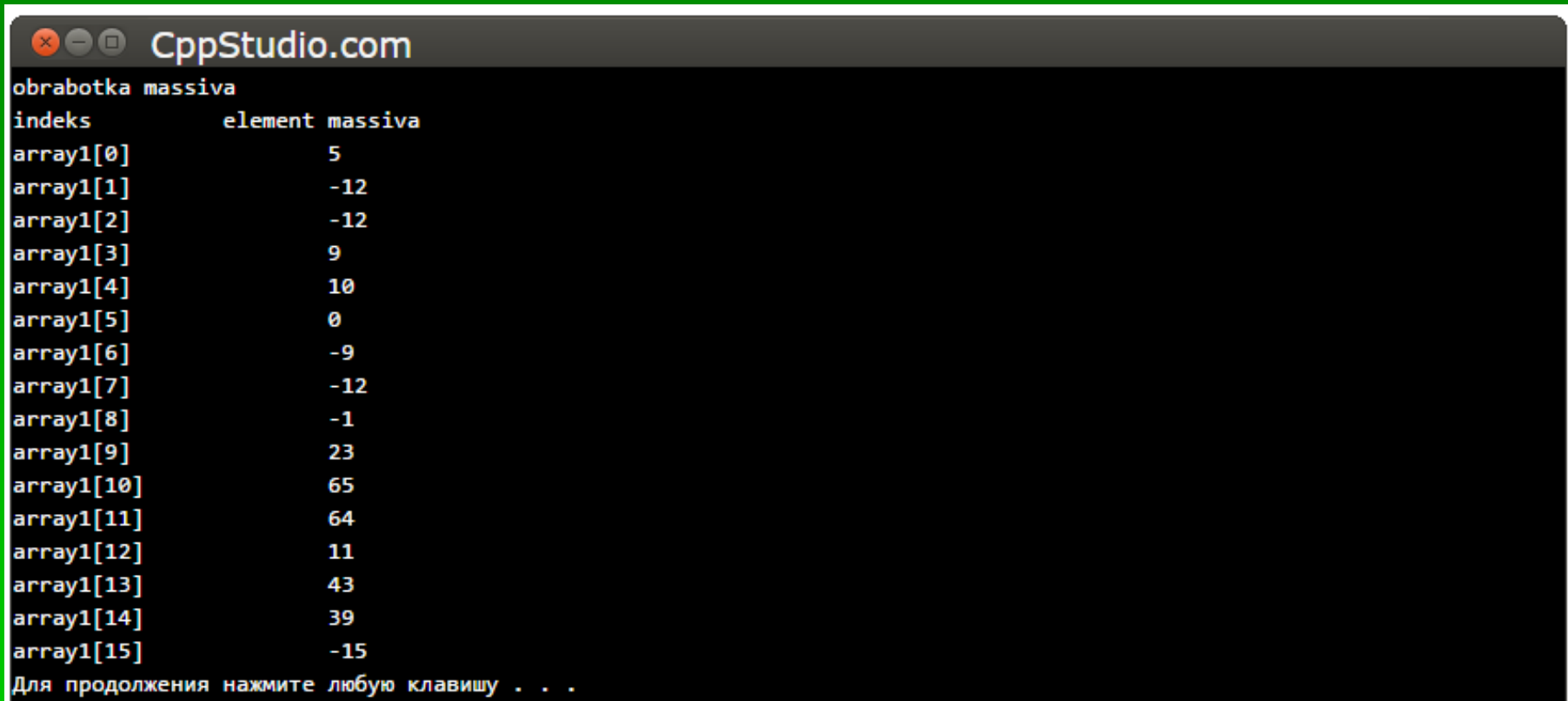
Обработка одномерного массива.

```
// array.cpp: определяет точку входа для консольного приложения.

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

int main(int argc, char* argv[])
{
    cout << "obrabotka massiva" << endl;
    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 << "indeks" << "\t\t" << "element massiva" << 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 << "array1[" << counter << "]" << "\t\t" << array1[counter] << endl;
    }
    system("pause");
    return 0;
}
```

Обработка одномерного массива.

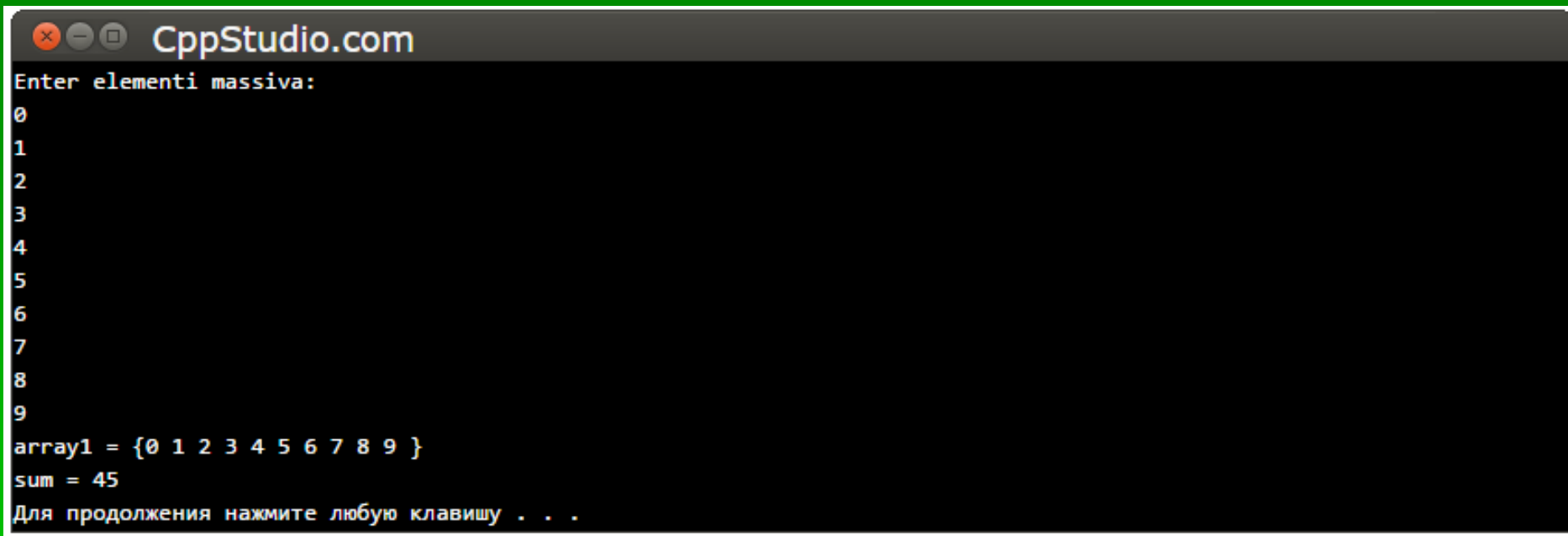


```
CppStudio.com
obrabotka massiva
indeks      element massiva
array1[0]    5
array1[1]   -12
array1[2]   -12
array1[3]    9
array1[4]   10
array1[5]    0
array1[6]   -9
array1[7]  -12
array1[8]   -1
array1[9]   23
array1[10]  65
array1[11]  64
array1[12]  11
array1[13]  43
array1[14]  39
array1[15] -15
Для продолжения нажмите любую клавишу . . .
```

Обработка одномерного массива 2.

```
1 // array_sum.cpp: определяет точку входа для консольного приложения.
2
3 #include "stdafx.h"
4 #include <iostream>
5 using namespace std;
6
7 int main(int argc, char* argv[])
8 {
9     int array1[10]; // объявляем целочисленный массив
10    cout << "Enter elementi massiva: " << endl;
11    int sum = 0;
12    for ( int counter = 0; counter < 10; counter++ ) // цикл для считывания чисел
13        cin >> array1[counter]; // считываем вводимые с клавиатуры числа
14    cout << "array1 = {";
15    for ( int counter = 0; counter < 10; counter++ ) // цикл для вывода элементов массива
16        cout << array1[counter] << " "; // выводим элементы массива на стандартное устройство вывода
17    for ( int counter = 0; counter < 10; counter++ ) // цикл для суммирования чисел массива
18        sum += array1[counter]; // суммируем элементы массива
19    cout << "}\nsum = " << sum << endl;
20    system("pause");
21    return 0;
22 }
```

Обработка одномерного массива 2.

A screenshot of a web-based C++ IDE window titled "CppStudio.com". The terminal area has a black background with white text. It shows a prompt "Enter elementi massiva:" followed by a list of numbers from 0 to 9. Below this, it displays the array initialization "array1 = {0 1 2 3 4 5 6 7 8 9 }" and the sum calculation "sum = 45". At the bottom, there is a Russian instruction: "Для продолжения нажмите любую клавишу . . .".

```
CppStudio.com
Enter elementi massiva:
0
1
2
3
4
5
6
7
8
9
array1 = {0 1 2 3 4 5 6 7 8 9 }
sum = 45
Для продолжения нажмите любую клавишу . . .
```

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]

```
1 // синтаксис объявления двумерного массива
2 /*тип данных*/ /*имя массива*/[/*количество строк*/][/*количество столбцов*/];
```


Работа с двумерными массивами.

```
1 // пример объявление двумерного массива:  
2 int a[5][3];
```

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

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]

Обработка двумерного массива.

[illegible]

Обработка двумерного массива.

