**ЛЕКЦИЯ 3. МНОГОМЕРНЫЕ МАССИВЫ**

Содержание

[1 Понятие многомерного массива и его определние 1](#_Toc146013139)

[2 Инициализация многомерного массива 2](#_Toc146013140)

[3 Примеры использования многомерных массивов 3](#_Toc146013141)

[4 Советы профессионала 6](#_Toc146013142)

# Понятие многомерного массива и его определние

Кроме одномерных массивов в C++ есть и многомерные. **Элементы таких массивов сами в свою очередь являются массивами, в которых также элементы могут быть массивами**.

Как правило, распространены двухмерные и трехмерные массивы. Например, определим двухмерный массив чисел:

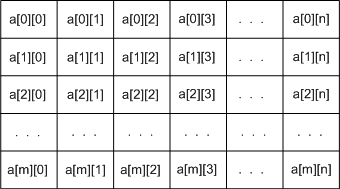
**Тип\_данных имя\_массива[количество\_строк][количество\_столбцов];**

**int numbers[3][2];**

Здесь массив numbers имеет две размерности (две пары квадратных скобок): первая размерность равна 3, а вторая размерность - 2. Такой массив состоит из трех элементов, при этом каждый элемент представляет массив из двух элементов. Двухмерный массив еще можно представить в виде таблицы, где первая размерность представляет количество строк, а вторая размерность - количество столбцов.

В C/C++ прямоугольный двумерный массив чисел реализует математическое понятие «матрица». Однако, в общем случае, двумерный массив — понятие гораздо более широкое, чем матрица, поскольку он может быть и не прямоугольным, и не числовым.

Структура двумерного массива, с именем **a**, размером **m** на **n:**



где, m – количество строк двумерного массива;

n – количество столбцов двумерного массива;

m \* n – количество элементов массива.

Подобным образом можно определять массивы и с большим количеством размерностей, например, трехмерный массив:

**int numbers[3][2][2];**

# Инициализация многомерного массива

Как и в общем случае многомерный массив можно инициализировать некоторыми значениями, например, нулями:

**int numbers[3][2] {};**

Также можно инициализировать все элементы индивидуальными значениями. Так, массив **numbers** состоит из трех элементов (строк), каждый из которых представляет массив из двух элементов (столбцов). Поэтому такой массив можно проинициализировать, например, следующим образом:

**int numbers[3][2]**

**{**

**{1, 2},     // первая строка**

**{4, 5},     // вторая строка**

**{7, 8}      // третья строка**

**};**

Вложенные фигурные скобки очерчивают элементы для каждого подмассива.

Возможна также инициализация не всех элементов, а только некоторых:

**int numbers[3][2] { {1, 2}, {}, {} };**

В этом случае значения присваиваются первым элементам массивов, а остальные элементы инициализируются нулями.

При рассмотрении одномерных массивов мы видели, что компилятор можем автоматически выводить длину массива на основании количества элементов. При инициализации многомерных массивов тоже можно опустить длину массива, но только первую размерность (первые квадратные скобки):

**int numbers[][2] { {1, 2}, {3, 4}, {5, 6} };**

# Примеры использования многомерных массивов

**ПРИМЕР 1**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**setlocale(LC\_ALL, "");**

**const int ArraySize1=3, // размер1 массива(количество строк таблицы)**

**ArraySize2=2; // размер2 массива(количество столбцов)**

**// инициализация массива:**

**int a[ArraySize1][ArraySize2] ={**

**{12, 3}, // a[0][0]=12, a[0][1]=3,**

**{42, 5}, // a[1][0]=42, a[1][1]=5,**

**{10, 34} // a[2][0]=10, a[2][1]=34**

**};**

**//начальные значения можно ввести и с клавиатуры:**

**/\***

**for(int i=0; i<ArraySize1; ++i)**

**for(int j=0; j<ArraySize2; ++j)**

**cin>>a[i][j];**

**\*/**

**for (int i=0; i<ArraySize1;++i){ //вывод значений элементов**

**for (int j=0; j<ArraySize2; ++j) //массива**

**cout<<a[i][j]<<" ";**

**cout<<endl;**

**}**

**//сумма значений элементов массива**

**int total=0;**

**for (int i=0; i<ArraySize1; ++i)**

**for (int j=0; j<ArraySize2; ++j)**

**total+=a[i][j];**

**cout << "Сумма элементов массива: "<<total<<endl;**

**return 0;**

**}**

**ПРИМЕР 2**

Сформировать двумерный массив С из элементов одномерного массива А, расположив их на главной диагонали С и выше, и из элементов В – ниже главной диагонали

**#include <iostream>**

**using namespace std;**

**int main() {**

**setlocale(LC\_ALL, "Russian");**

**const int ArraySize1 = 6, ArraySize2 = 3; // размеры массивов**

**int с[ArraySize2][ArraySize2];**

**// инициализация массивов a и b:**

**int a[ArraySize1] = {1, 2, 3, 4, 5, 6};**

**int b[ArraySize2] = {7, 8, 9};**

**//Формирование массива на главной диагонали и выше**

**for (int i = 0, k=0; i < ArraySize2; ++i)**

**for (int j = i; j < ArraySize2; ++j)**

**с[i][j] = a[k++];**

**//ниже главной диагонали**

**for (int i = 1, k=0; i < ArraySize2; ++i)**

**for (int j = 0; j < i; ++j)**

**с[i][j] = b[k++];**

**//вывод значений элементов массивов**

**cout << "A: ";**

**for (int i = 0; i < ArraySize1; ++i)**

**cout << a[i]<< " ";**

**cout << endl;**

**cout << "B: ";**

**for (int i = 0; i < ArraySize2; ++i)**

**cout << b[i]<< " ";**

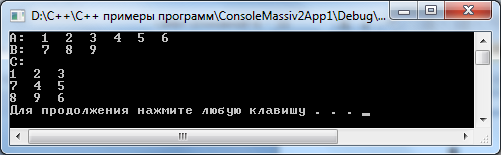
**cout << endl;**

**cout << "C: "<<endl;**

**for (int i = 0; i < ArraySize2; ++i) {**

**for (int j = 0; j < ArraySize2; ++j)**

**cout << с[i][j] << " ";**

** cout << endl;**

**}**

**return 0;**

**}**

**ПРИМЕР 3**

**#include <iostream>**

**using namespace std;**

**int main(){**

**int nums[3][2] {**

**{1, 2},**

**{3, 4},**

**{5, 6}**

**};**

**// получаем значение элемента**

**int n = nums[1][0]; // вторая строка, первый столбец**

**cout << "n =" << n << endl; // n = 3**

**cout << "nums[2][1] = " << nums[2][1] << endl; // nums[2][1] = 6**

**nums[2][1] = 123; // изменяем значение:третья строка, второй столбец**

**cout << "nums[2][1] = " << nums[2][1] << endl; // nums[2][1] = 123**

**const int rows = 3, columns = 2;**

**int numbers[rows][columns]{ {1, 2}, {3, 4}, {5, 6} };**

**for (auto& subnumbers : numbers) {**

**for (int number : subnumbers)**

**{**

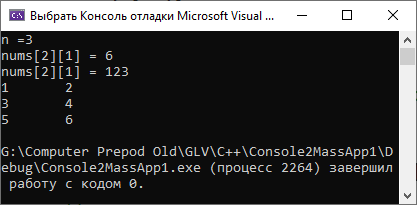
**cout << number << "\t";**

**}**

**cout << std::endl;**

**}**

**}**



Для перебора массивов, которые входят в массив, применяются ссылки. То есть во внешнем цикле **for(auto &subnumbers : numbers) &subnumbers** представляет ссылку на подмассив в массиве. Во внутреннем цикле **for(int number : subnumbers)** из каждого подмассива в **subnumbers** получаем отдельные его элементы в переменную number и выводим ее значение на консоль.

# Советы профессионала

**IconAttention**

**Правильно организуйте обработку квадратной матрицы**

Найти сумму значений элементов главной диагонали двумерного массива.

**Вариант 1. Избыточный, плохой код**

**const ArraySize = 3; // размеры массива**

**int a[ArraySize][ArraySize];**

**// ввод исходных значений смотри в предыдущих примерах**

**int total = 0;**

**for (int i = 0; i < ArraySize; ++i)**

**for (int j = 0; j < ArraySize; ++j) //лишний цикл!!!**

**if(j==i) //ненужное сравнение!!!**

**total += a[i][j];**

**cout << "Сумма элементов массива: " << total << endl;**

**Вариант 2. Правильный код**

**int total = 0;**

**for (int i = 0; i < ArraySize; ++i)**

**total += a[i][i];**

**cout << "Сумма элементов массива: " << total << endl;**