

Занятие 9 — Статические массивы

Программирование, численные методы и информатика

Александр Валерьевич Позднеев

Кафедра автоматизации научных исследований
Факультет вычислительной математики и кибернетики
Московский государственный университет имени М. В. Ломоносова
<http://ani.cmc.msu.ru/geol>

Осенний семестр 2010/2011



Содержание занятия

- 1 Статические одномерные массивы
 - Общие понятия
 - Объявление массивов
 - Инициализация массивов
- 2 Статические многомерные массивы
- 3 Массивы как аргументы функций

Общие понятия о массивах

- ▶ *Массив* — это структура однотипных данных, занимающих непрерывную область памяти
- ▶ Массив имеет *размер* — количество элементов в нем.
- ▶ Каждый элемент массива имеет свой *номер* (также называемый *индексом*), обращение к элементу массива осуществляется путем указания его индекса
- ▶ В языке C++ элементы нумеруются начиная с 0, поэтому последний элемент массива имеет номер на 1 меньше размера массива

Объявление массивов

Массив в языке C++ задается следующим образом:

- ▶ `тип_элементов идентификатор[размер];`

где

- ▶ `тип_элементов` — произвольный тип данных языка C++, который будут иметь элементы массива, например, `int`, `double` и т. д.;
- ▶ `идентификатор` — имя массива;
- ▶ `размер` — число элементов в нем.

К элементу массива можно обращаться, как

- ▶ `идентификатор[индекс]`

Например, если было сделано объявление массива

- ▶ `double A[5];`

то таким образом создается 5 элементов массива типа `double`: `A[0]`, `A[1]`, `A[2]`, `A[3]`, `A[4]`.

Массивы — пример

Заполнить массив десятью случайными числами из полусегмента $[-0,5; 0,5]$ и вывести его на печать.

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

int main() {
    const int N = 10;
    double a[N];

    for (int i = 0; i < N; ++i)
        a[i] = rand() / (double)RAND_MAX - 0.5;

    for (int i = 0; i < N; ++i)
        cout << i << "\t" << a[i] << endl;

    system("pause");
    return 0;
}
```

Массивы — упражнения

1. Выведите на экран все элементы массива с четными индексами (т. е. $A[0]$, $A[2]$, $A[4]$, ...)
2. Найдите количество положительных элементов в массиве
3. Найдите значение наибольшего элемента в массиве
4. Переставьте элементы массива в обратном порядке без использования дополнительного массива
5. Циклически сдвиньте элементы массива вправо (0-й элемент становится 1-м, 1-й становится 2-м, ..., последний становится 0-м)
6. Переставьте соседние элементы массива (0-й элемент поменять с 1-м, 2-й с 3-м и т. д.)
7. Определите, есть ли в данном массиве два соседних элемента с одинаковыми знаками и укажите их индексы

Инициализация массивов

```
int leap [] =
{
    31, 29, 31, 30, 31, 30,
    31, 31, 30, 31, 30, 31
};

int non_leap [] =
{
    31, 28, 31, 30, 31, 30,
    31, 31, 30, 31, 30, 31
};

int N = sizeof(leap) / sizeof(non_leap[0]);
```

Многомерные массивы

Массивы могут быть и двумерные (матрицы), трехмерные и т.д. Размерность массивов в C++ никак не ограничивается.

Двумерный вещественный массив (3 строки, 2 столбца):

- ▶ `double matrix[3][2];`

Обращаются к элементам такого массива, указывая два индекса:

- ▶ `matrix[1][2];`

Многомерные массивы — пример

Заполнить массив размера 2×3 случайными числами из полусегмента $[-0,5; 0,5]$ и вывести его на печать.

```
int main() {
    const int N = 2, M = 3;
    double a[N][M];

    for (int i = 0; i < N; ++i)
        for (int j = 0; j < M; ++j)
            a[i][j] = rand() / (double)RAND_MAX - 0.5;

    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < M; ++j)
            cout << a[i][j] << "\t";
        cout << endl;
    }

    system("pause");
    return 0;
}
```

Двумерные массивы — упражнения

1. Вывести на печать значения, расположенные на главной диагонали квадратной матрицы размера 10×10
2. Распечатать элементы, стоящие на побочной диагонали квадратной матрицы размера 10×10
3. Найти сумму элементов каждой строки матрицы размера 4×3 и записать эти значения в одномерный массив длины 4. Найти максимальную из этих сумм

Массивы как аргументы функций

```
void Init_Array(double a[], int N) {  
    for (int i = 0; i < N; ++i)  
        a[i] = rand() / (double)RAND_MAX - 0.5;  
}
```

```
void Print_Array(double a[], int N) {  
    for (int i = 0; i < N; ++i)  
        cout << i << "\t" << a[i] << endl;  
}
```

```
int main() {  
    const int N = 10;  
    double a[N];  
  
    Init_Array(a, N);  
    Print_Array(a, N);  
  
    system("pause");  
    return 0;  
}
```

Массивы-аргументы — упражнения

1. Напишите функцию

```
int Search (int A[], int n, int x)
```

, которая находит в массиве `int A[n]` элемент, значение которого равно `x`. Функция возвращает индекс найденного элемента или `-1`, если такого элемента в массиве нет.

2. Напишите функцию

```
int CountMax (int A[], int n)
```

, которая подсчитывает, сколько раз в массиве встречается значение, являющееся максимальным. Функция должна выполнять однократный просмотр массива.