数组的特征

- 数组是由一定数目的同类元素顺序排列而成的结构类型数据
- 一个数组在内存占有一片连续的存储区域
- 数组名是存储空间的首地址
- 数组的每个元素用下标变量标识

一维数组定义

当数组中每个元素只带有一个下标时，我们称这样的数组为一维数组。

说明：

①数组名的命名规则与变量名的命名规则一致。

②常量表达式表示数组元素的个数。可以是常量和符号常量，但不能是变量。

例如：

```
int a[10];        //数组a定义是合法的
int b[n];         //数组b定义是非法的
```

其中，a是一维数组的数组名，该数组有10个元素，依次表示为：a[0],a[1],a[2],a[3],a[4],a[5]，a[6],a[7],a[8],a[9]。需要注意的是：a[10]不属于该数组的空间范围。当在说明部分定义了一个数组变量之后,C++编译程序为所定义的数组在内存空间开辟一串连续的存储单元，每个数组第一个元素的下标都是0，因此第一个元素为第0个数组元素。例如：上例中的a数组在内存的存储如表所示：

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

a数组共有10个元素组成，在内存中10个数组元素共占10个连续的存储单元。a数组最小下标为0，最大下标9。按定义a数组所有元素都是整型变量。

黑猫编程 blackcat1995.com

一维数组初始化

- 数组是由一定数目的同类元素顺序排列而成的结构类型数据
  数组的初始化可以在定义时一并完成。格式：
  类型标识符 数组名[常量表达式]={值1，值2，…}
  例如：
  int a[5]={1,2,3,4,5}
- 说明：
  - (1)在初值列表中可以写出全部数组元素的值，也可以写出部分。例如，以下方式可以对数组进行初始化：
    int x[10]={0,1,2,3,4};
    该方法仅对数组的前5个元素依次进行初始化，其余值为0。
  - (2)对数组元素全部初始化为0，可以简写为：{}。
    例如：
    int a[5]={}; 将数组a的5个元素都初始化为0。

黑猫编程 blackcat1995.com

与普通变量一样，可以在数组定义的同时，对数组元素赋初值

```
例：  int a[5] = {1, 3, 5, 7, 9};      // 各元素分别赋初始值
      int b[5] = {};                   // 全部元素初始化为0
      int c[5] = {1, 2, 3};            // b2[3]，b2[4]自动赋0
      int d[] = {1, 2, 3, 4, 5, 6, 7}; // 自动定义数组长度为7
      int e[5] = {1, 2, 3, 4, 5, 6, 7}; // 错误，初始化数据过多
```

黑猫编程 blackcat1995.com

- int a[5]; 系统会随机分配一个数，数值不确定；如果定义在全局变量，所有数组元素都初始化为0。

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

int main() {

    int a[10];

    // 随机数据
    for(int i = 0; i < 10; i++)
        cout << a[i] << " ";

    return 0;
}
```

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

int a[10];

int main() {

    // 全局区初始值都为0
    for(int i = 0; i < 10; i++)
        cout << a[i] << " ";

    return 0;
}
```

黑猫编程 blackcat1995.com 一维数组访问

通过给出的数组名称和这个元素在数组中的位置编号(即下标)，程序可以访问这个数组中的任何一个元素。

一维数组元素的访问格式:

数组名[下标]

例如：若i、j都是int型变量，则

a[5]

a[i+j]

a[i++]

都是合法的元素。

说明:

(1)下标可以是任意值为整型的表达式，该表达式里可以包含变量和函数调用。访问时，下标值应在数组定义的下标值范围内。

(2)数组的精妙在于下标可以是变量，通过对下标变量值的灵活控制，达到灵活处理数组元素的目的。

(3)C++语言只能逐个引用数组元素，而不能一次引用整个数组。

(4)数组元素可以像同类型的普通变量那样使用，对其进行赋值和运算的操作，和普通变量完全相同。

例如： c[10]=34;实现了给c[10]赋值为34。

黑猫编程 blackcat1995.com 一维数组的赋值

例： int arr [10] , i = 3, j = 5 ;

arr [0]	10	arr [0] = 10
arr [1]		
arr [2]	10	arr [arr [i]] = arr [0]
arr [3]	2	arr [i] = 2
arr [4]		
arr [5]	2	arr [j] = arr [i]
arr [6]		
arr [7]	31	arr [2+j] = 31
arr [8]		
arr [9]		

注意

C++ 不提供对数组的下标范围检查

黑猫编程 blackcat1995.com 数组越界

C++语言规定，使用数组时，要注意:

(1)、数组元素的下标值为非负整数。

(2)、在定义元素个数的下标范围内使用。

然而，当在程序中把下标写成负数、大于数组元素的个数时，程序编译的时候是不会出错的。例如:

```
int a[10];
a[-3]=5;
a[20]=15;
a[10]=20;
int k=a[30]
```

这些语句的语法是正确的，能够通过程序的编译。然而，它们要访问的数组元素并不在数组的存储空间的，这种现象叫数组越界。

快乐刷题

- P240 数组逆序重存放
- P241 向量点积计算