

向量(vector)

学习C++ -> 向量(vector)

一、向量的介绍

向量 **vector** 是一种对象实体，能够容纳许多其他类型相同的元素，因此又被称为容器。与 **string** 相同，**vector** 同属于 STL (Standard Template Library, 标准模板库) 中的一种自定义的数据类型，可以广义上认为是数组的增强版。

在使用它时，需要包含头文件 **vector**，**#include**

vector 容器与数组相比其优点在于它能够根据需要随时自动调整自身的大小以便容下所要放入的元素。此外，**vector** 也提供了许多的方法来对自身进行操作。

二、向量的声明及初始化

vector 型变量的声明以及初始化的形式也有许多，常用的有以下几种形式：

```
vector<int> a; //声明一个int型向量a vector<int> a(10); //声明一个初始大小为10的向量
vector<int> a(10, 1); //声明一个初始大小为10且初始值都为1的向量
vector<int> b(a); //声明并用向量a初始化向量b vector<int> b(a.begin(), a.begin()+3); //将a向量中从第0个到第2个(共3个)作为向量b的初始值
```

除此之外，还可以直接使用数组来初始化向量：

```
int n[] = {1, 2, 3, 4, 5}; vector<int> a(n, n+5); //将数组n的前5个元素作为向量a的初值
vector<int> a(&n[1], &n[4]); //将n[1] - n[4]范围内的元素作为向量a的初值
```

三、元素的输入及访问

元素的输入和访问可以像操作普通的数组那样，用 **cin>>** 进行输入，**cout<** 这样进行输出：

示例：

```
1 #include 2 #include 3 4usingnamespace std; 5 6int main() 7 { 8 vector<int> a(10, 0); //大小为10初值为0的向量a 910//对其中部分元素进行输入11 cin >>a[2];12 cin >>a[5];13 cin >>a[6];1415//全部输出16int i;17for(i=0; i)18 cout<<< span style='font-size:12px;font-style:normal;font-weight:normal;font-family:"Courier New";line-height:18px;color:rgb(128, 0, 0);text-align:left;float:none;border-top:0px none rgb(128, 0, 0);border-bottom:0px none rgb(128, 0, 0);border-right:0px none rgb(128, 0, 0);border-left:0px none rgb(128, 0, 0);text-decoration:none solid rgb(128, 0, 0);background:rgba(0, 0, 0, 0) none repeat 0% 0%;' >></span>"
;1920return 0 ;21 }
```

在元素的输出上，还可以使用遍历器(又称迭代器)进行输出控制。在 **vector b(a.begin(), a.begin()+3)**；这种声明形式中，(**a.begin()**、**a.begin()+3**) 表示向量起始元素位置到起始元素+3之间的元素位置。(a.begin(), a.end())则表示起始元素和最后一个元素之外的元素位置。

向量元素的位置便成为遍历器，同时，向量元素的位置也是一种数据类型，在向量中遍历器的类型为：**vector::iterator**。遍历器不但表示元素位置，还可以再容器中前后移动。

在上例中讲元素全部输出部分的代码就可以改写为：

```
//全部输出    vector<int>::iterator t ;    for(t=a.begin(); t!=a.end(); t++)        cout<< tspan style='font-size:12px;font-style:normal;font-weight:normal;font-family:"Courier New";line-height:18px;color:rgb(128, 0, 0);text-align:left;float:none;border-top:0px none rgb(128, 0, 0);border-bottom:0px none rgb(128, 0, 0);border-right:0px none rgb(128, 0, 0);border-left:0px none rgb(128, 0, 0);text-decoration:none solid rgb(128, 0, 0);background:rgba(0, 0, 0, 0) none repeat 0% 0%;' >"< /span>
```

*t 为指针的间接访问形式，意思是访问t所指向的元素值。

四、向量的基本操作

1>.a.size()//获取向量中的元素个数

2>.a.empty()//判断向量是否为空

3>.a.clear()//清空向量中的元素

4>. 复制a = b ;//将b向量复制到a向量中

5>. 比较 保持 ==、!=、>、>=、<< span>、< span>的惯有含义 ; 如: a == b ;//a向量与b向量比较，相等则返回1

6>. 插入 - insert ①、 a.insert(a.begin(), 1000);//将1000插入到向量a的起始位置前 ②、
a.insert(a.begin(), 3, 1000) ;//将1000分别插入到向量元素位置的0-2处(共3个元素) ③、 vector<int> a(5,
1) ; vector<int> b(10) ; b.insert(b.begin(), a.begin(), a.end()) ; //将a.begin(), a.end()之
间的全部元素插入到b.begin()前

7>. 删除 - erase ①、 b.erase(b.begin()) ; //将起始位置的元素删除 ②、
b.erase(b.begin(), b.begin()+3) ;//将(b.begin(), b.begin()+3)之间的元素删除

8>. 交换 - swap b.swap(a) ; //a向量与b向量进行交换

五、二维向量

与数组相同，向量也可以增加维数，例如声明一个m*n大小的二维向量方式可以像如下形式：

```
vector< vector    span style='font-size:12px;font-style:normal;font-weight:normal;font-family:"Courier New";line-height:18px;color:rgb(0, 0, 255);text-align:left;float:none;border-top:0px none rgb(0, 0, 255);border-bottom:0px none rgb(0, 0, 255);border-right:0px none rgb(0, 0, 255);border-left:0px none rgb(0, 0, 255);text-decoration:none solid rgb(0, 0, 255);background:rgba(0, 0, 0, 0) none repeat 0% 0%;' >int> > b(10, vector<int>(5));    //创建一个10*5的  
int型二维向量</span>
```

在这里，实际上创建的是一个向量中元素为向量的向量。同样可以根据一维向量的相关特性对二维向量进行操作。

例：

```
1    #include 2    #include 3 4usingnamespace std ; 5 6int main() 7    { 8        vector< vector    span style='font-size:12px;font-style:normal;font-weight:normal;font-family:"Courier New";line-height:18px;color:rgb(0, 0, 255);text-align:left;float:none;border-top:0px none rgb(0, 0, 255);border-bottom:0px none rgb(0, 0, 255);border-right:0px none
```

```

rgb(0, 0, 255);border-left:0px none rgb(0, 0, 255);text-decoration:none solid rgb(0, 0, 255);background:rgba(0, 0, 0, 0)
none repeat 0% 0%;' >int</span>> > b(10, vector<int>(5, 0)) ; 910//对部分数据进行输入11      cin>>b[1][1] ;12
cin>>b[2][2] ;13      cin>>b[3][3];1415//全部输出16int m, n ;17for(m=0; m//b.size()获取行向量的大小18
{19for(n=0; n//获取向量中具体每个向量的大小20      cout<<< span style='font-size:12px;font-
style:normal;font-weight:normal;font-family:"Courier New";line-height:18px;color:rgb(128, 0, 0);text-
align:left;float:none;border-top:0px none rgb(128, 0, 0);border-bottom:0px none rgb(128, 0, 0);border-
right:0px none rgb(128, 0, 0);border-left:0px none rgb(128, 0, 0);text-decoration:none solid rgb(128, 0,
0);background:rgba(0, 0, 0, 0) none repeat 0% 0%;' >"</span>" ;21      cout<< span style='font-
size:12px;font-style:normal;font-weight:normal;font-family:"Courier New";line-height:18px;color:rgb(128,
0, 0);text-align:left;float:none;border-top:0px none rgb(128, 0, 0);border-bottom:0px none rgb(128, 0,
0);border-right:0px none rgb(128, 0, 0);border-left:0px none rgb(128, 0, 0);text-decoration:none solid
rgb(128, 0, 0);background:rgba(0, 0, 0, 0) none repeat 0% 0%;' >"</span>\n" ;22      }2324return 0;25    }

```

同样，按照这样的思路我们还可以创建更多维的向量，不过维数太多会让向量变得难以灵活控制，三维以上的向量还需酌情使用。

wid, 2013.01.22

上一篇: [学习C++ -> string类字符串](#)