

vector

郭 炜 刘家瑛





vector

- ▀ 可变长的动态数组
- ▀ 必须包含头文件 `#include <vector>`
- ▀ 支持 随机访问迭代器
 - 根据下标随机访问某个元素时间为常数
 - 在尾部添加速度很快
 - 在中间插入慢
- ▀ 所有STL算法 都能对vector操作



vector的成员函数

构造函数初始化

成员函数	作 用
<code>vector();</code>	无参构造函数, 将容器初始化成空的
<code>vector(int n);</code>	将容器初始化成有n个元素
<code>vector(int n, const T & val);</code>	假定元素类型是T, 将容器初始化成有n个元素, 每个元素的值都是val
<code>vector(iterator first, iterator last);</code>	将容器初始化为与别的容器上区间[first, last)一致的内容



vector的成员函数

其他常用函数

成员函数	作 用
<code>void pop_back();</code>	删除容器末尾的元素
<code>void push_back(const T & val);</code>	将val添加到容器末尾
<code>int size();</code>	返回容器中元素的个数
<code>T & front();</code>	返回容器中第一个元素的引用
<code>T & back();</code>	返回容器中最后一个元素的引用



例1:

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    int i;
    int a[5] = {1,2,3,4,5};
    vector<int> v(5);
    cout << v.end() - v.begin() << endl;
    for( i = 0; i < v.size(); i ++ ) v[i] = i;
    v.at(4) = 100;
    for( i = 0; i < v.size(); i ++ )
        cout << v[i] << "," ;
    cout << endl;
    vector<int> v2(a,a+5); //构造函数
    v2.insert(v2.begin() + 2,13); //在begin()+2位置插入13
    for( i = 0; i < v2.size(); i ++ )
        cout << v2.at(i) << "," ;
}
```

输出:

5

0,1,2,3,100,

1,2,13,3,4,5,




二维动态数组

```
vector< vector<int> > v(3);
```

//v有3个元素,

//每个元素都是vector<int> 容器



例2:

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<vector<int >> v(3);
    for(int i=0; i<v.size(); ++i)
        for(int j=0; j<4; ++j)
            v[i].push_back(j);
    for(int i=0; i<v.size(); ++i){
        for(int j=0; j<v[i].size(); ++j)
            cout<<v[i][j]<<" ";
        cout<<endl;
    }
    return 0;
}
```

输出:

0 1 2 3

0 1 2 3

0 1 2 3



In-Video Quiz

vector没有以下哪个成员函数：

- ▲ A) push_back
- ▲ B) front
- ▲ C) pop_back
- ▲ D) push_front