

6.6 关联容器之 map、multimap_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 6.6 关联容器之 map、multimap 涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

6. 关联容器之 map、multimap

- map 容器提供一个键值对 (key/value) 容器，map 与 multimap 差别仅仅在于 multiple 允许一个键对应多个值。
- map 内部自建一棵红黑树（一种自平衡二叉树），这棵树具有数据自动排序的功能，所以在 map 内部所有的数据都是有序的，以二叉树的形式进行组织。

```
#include <iostream>

#include <map>

using namespace std;

int main(int argc, const char *argv[])
{
    map<string,int> map_table;

    map_table["one"] = 1;
    map_table["two"] = 100;
    map_table.insert(pair<string,int>("three",300));

    cout << "one key value   : " << map_table["one"]   << endl;
    cout << "two key value   : " << map_table["two"]   << endl;
    cout << "three key value : " << map_table["three"] << endl;

    for(map<string,int>::iterator it = map_table.begin();
        it != map_table.end();
        it ++){
        cout << (*it).first << ":" << (*it).second << endl;
    }

    return 0;
}

#include <iostream>

#include <map>
```

```
using namespace std;

int main(int argc, const char *argv[])
{
    multimap<char,int> mymultimap;
    multimap<char,int>::iterator it;

    mymultimap.insert (pair<char,int>('a',10));
    mymultimap.insert (pair<char,int>('b',20));
    mymultimap.insert (pair<char,int>('b',150));
    mymultimap.insert (pair<char,int>('b',200));

    for ( it=mymultimap.begin() ; it != mymultimap.end(); it++)
        cout << (*it).first << " => " << (*it).second << endl;

    cout << "key   : " << mymultimap.find('b')->first << endl;
    cout << "value : " << mymultimap.find('b')->second << endl;

    cout << "-----" << endl;

    pair<multimap<char,int>::iterator,multimap<char,int>::iterator> ret;

    ret = mymultimap.equal_range('b');
    for(it = ret.first; it != ret.second;it++){
        cout << (*it).second << endl;
    }

    return 0;
}
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

