C++ STL unordered_map删除元素：erase()和clear()

C++ STL 标准库为了方便用户可以随时删除 unordered_map 容器中存储的键值对，unordered_map 容器类模板中提供了以下 2 个成员方法：

- erase()：删除 unordered_map 容器中指定的键值对；
- clear()：删除 unordered_map 容器中所有的键值对，即清空容器。

本节就对以上 2 个成员方法的用法做详细的讲解。

unordered_map erase()方法

为了满足不同场景删除 unordered_map 容器中键值对的需要，此容器的类模板中提供了 3 种语法格式的 erase() 方法。

1) erase() 方法可以接受一个正向迭代器，并删除该迭代器指向的键值对。该方法的语法格式如下：

```
iterator erase ( const_iterator position );
```

其中 position 为指向容器中某个键值对的迭代器，该方法会返回一个指向被删除键值对之后位置的迭代器。

举个例子：

```
#include <iostream>
#include <string>
#include <unordered_map>
using namespace std;
int main()
{
    //创建 umap 容器
    unordered_map<string, string> umap{
        {"STL教程", "http://c.biancheng.net/stl/"},
        {"Python教程", "http://c.biancheng.net/python/"},
        {"Java教程", "http://c.biancheng.net/java/"} };
    //输出 umap 容器中存储的键值对
    for (auto iter = umap.begin(); iter != umap.end(); ++iter) {
        cout << iter->first << " " << iter->second << endl;
    }

    cout << "erase:" << endl;
    //定义一个接收 erase() 方法的迭代器
    unordered_map<string,string>::iterator ret;
    //删除容器中第一个键值对
    ret = umap.erase(umap.begin());
    //输出 umap 容器中存储的键值对
    for (auto iter = umap.begin(); iter != umap.end(); ++iter) {
```

```
        cout << iter->first << " " << iter->second << endl;
    }
    cout << "ret = " << ret->first << " " << ret->second << endl;
    return 0;
}
```

程序执行结果为：

```
STL教程 http://c.biancheng.net/stl/
Python教程 http://c.biancheng.net/python/
Java教程 http://c.biancheng.net/java/
erase:
Python教程 http://c.biancheng.net/python/
Java教程 http://c.biancheng.net/java/
ret = Python教程 http://c.biancheng.net/python/
```

可以看到，通过给 erase() 方法传入指向容器中第一个键值对的迭代器，该方法可以将容器中第一个键值对删除，同时返回一个指向被删除键值对之后位置的迭代器。

> 注意，如果erase()方法删除的是容器存储的最后一个键值对，则该方法返回的迭代器，将指向容器中最后一个键值对之后的位置（等同于 end() 方法返回的迭代器）。

2) 我们还可以直接将要删除键值对的键作为参数直接传给 erase() 方法，该方法会自行去 unordered_map 容器中找和给定键相同的键值对，将其删除。erase() 方法的语法格式如下：

```
size_type erase ( const key_type& k );
```

其中，k 表示目标键值对的键的值；该方法会返回一个整数，其表示成功删除的键值对的数量。

举个例子：

```
#include <iostream>
#include <string>
#include <unordered_map>
using namespace std;
int main()
{
    //创建 umap 容器
    unordered_map<string, string> umap{
        {"STL教程", "http://c.biancheng.net/stl/"},
        {"Python教程", "http://c.biancheng.net/python/"},
        {"Java教程", "http://c.biancheng.net/java/"} };
    //输出 umap 容器中存储的键值对
    for (auto iter = umap.begin(); iter != umap.end(); ++iter) {
```

```
        cout << iter->first << " " << iter->second << endl;
    }
    int delNum = umap.erase("Python教程");
    cout << "delNum = " << delNum << endl;
    //再次输出 umap 容器中存储的键值对
    for (auto iter = umap.begin(); iter != umap.end(); ++iter) {
        cout << iter->first << " " << iter->second << endl;
    }
    return 0;
}
```

程序执行结果为：

```
STL教程 http://c.biancheng.net/stl/
Python教程 http://c.biancheng.net/python/
Java教程 http://c.biancheng.net/java/
delNum = 1
STL教程 http://c.biancheng.net/stl/
Java教程 http://c.biancheng.net/java/
```

通过输出结果可以看到，通过将 "Python教程" 传给 erase() 方法，就成功删除了 umap 容器中键为 "Python教程" 的键值对。

3) 除了支持删除 unordered_map 容器中指定的某个键值对，erase() 方法还支持一次删除指定范围内的所有键值对，其语法格式如下：

```
iterator erase ( const_iterator first, const_iterator last );
```

其中 first 和 last 都是正向迭代器，[first, last) 范围内的所有键值对都会被 erase() 方法删除；同时，该方法会返回一个指向被删除的最后一个键值对之后一个位置的迭代器。

举个例子：

```
#include <iostream>
#include <string>
#include <unordered_map>
using namespace std;
int main()
{
    //创建 umap 容器
    unordered_map<string, string> umap{
        {"STL教程", "http://c.biancheng.net/stl/"},
        {"Python教程", "http://c.biancheng.net/python/"},
        {"Java教程", "http://c.biancheng.net/java/"} };
    //first 指向第一个键值对
```

```
    unordered_map<string, string>::iterator first = umap.begin();
    //last 指向最后一个键值对
    unordered_map<string, string>::iterator last = umap.end();
    //删除[fist,last)范围内的键值对
    auto ret = umap.erase(first, last);
    //输出 umap 容器中存储的键值对
    for (auto iter = umap.begin(); iter != umap.end(); ++iter) {
        cout << iter->first << " " << iter->second << endl;
    }
    return 0;
}
```

执行程序会发现，没有输出任何数据，因为 erase() 方法删除了 umap 容器中 [begin(), end()) 范围内所有的元素。

unordered_map clear()方法

在个别场景中，可能需要一次性删除 unordered_map 容器中存储的所有键值对，可以使用 clear() 方法，其语法格式如下：

```
void clear()
```

举个例子：

```
#include <iostream>
#include <string>
#include <unordered_map>
using namespace std;
int main()
{
    //创建 umap 容器
    unordered_map<string, string> umap{
        {"STL教程", "http://c.biancheng.net/stl/"},
        {"Python教程", "http://c.biancheng.net/python/"},
        {"Java教程", "http://c.biancheng.net/java/"} };
    //输出 umap 容器中存储的键值对
    for (auto iter = umap.begin(); iter != umap.end(); ++iter) {
        cout << iter->first << " " << iter->second << endl;
    }
    //删除容器内所有键值对
    umap.clear();
    cout << "umap size = " << umap.size() << endl;
    return 0;
}
```

程序执行结果为：

```
STL教程 http://c.biancheng.net/stl/
Python教程 http://c.biancheng.net/python/
Java教程 http://c.biancheng.net/java/
umap size = 0
```

显然，通过调用 clear() 方法，原本包含 3 个键值对的 umap 容器，变成了空容器。

> 注意，虽然使用 erase() 方法的第 3 种语法格式，可能实现删除 unordered_map 容器内所有的键值对，但更推荐使用 clear() 方法。