





#include <iostream>

using namespace std;

#include <string>

#include <map>

void printMap(map<int, int> &m)

{

for (map<int, int>::iterator it = m.begin(); it != m.end(); it++)

{

cout << "key: " <<(\*it).first<<" value = "<< it->second << endl ;

}

cout << endl;

}

void test01()

{

map<int, int> m;

m.insert(pair<int, int>(1, 10));

m.insert(pair<int, int>(3, 20));

m.insert(pair<int, int>(2, 30));

m.insert(pair<int, int>(4, 40));

printMap(m);

map<int, int> m2(m);

printMap(m2);

map<int, int> m3;

m3 = m2;

printMap(m3);

}

int main()

{

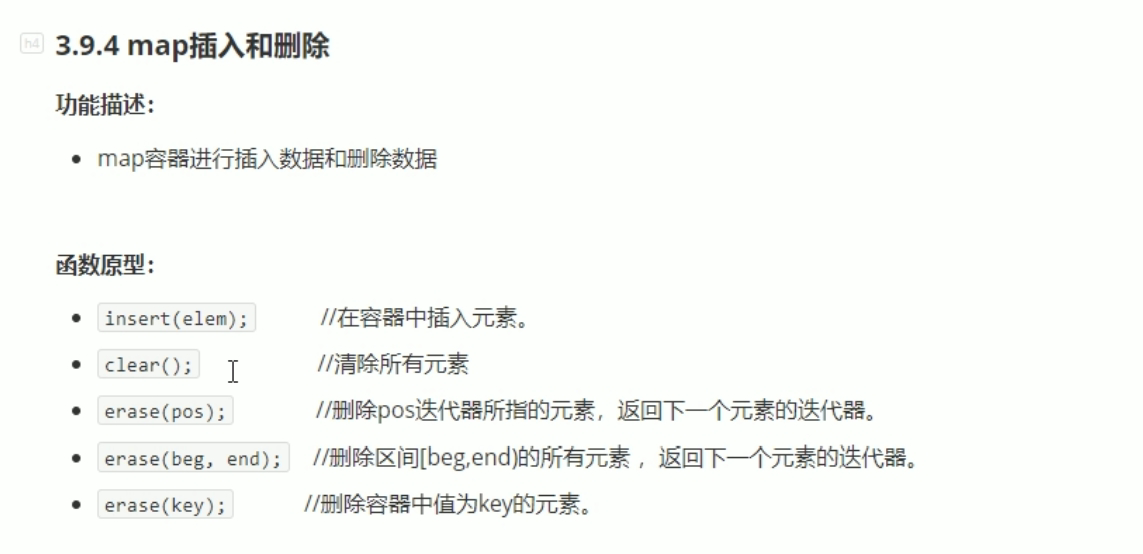
test01();

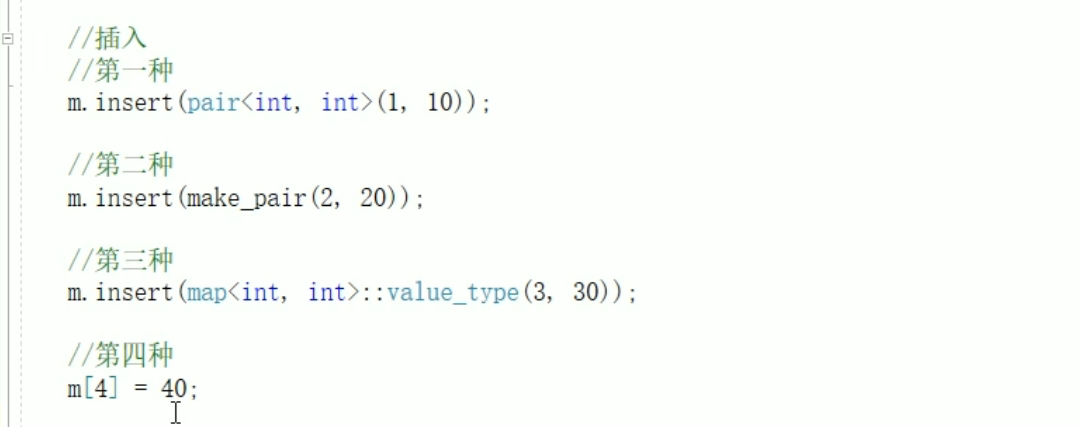
system("pause");

return 0;

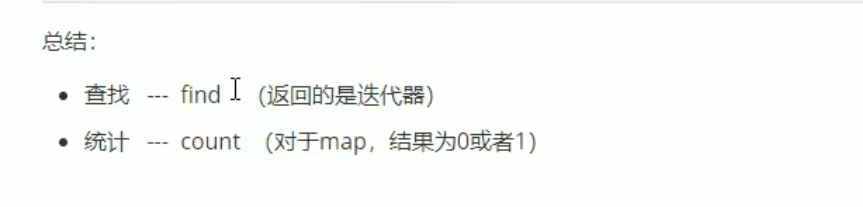
}

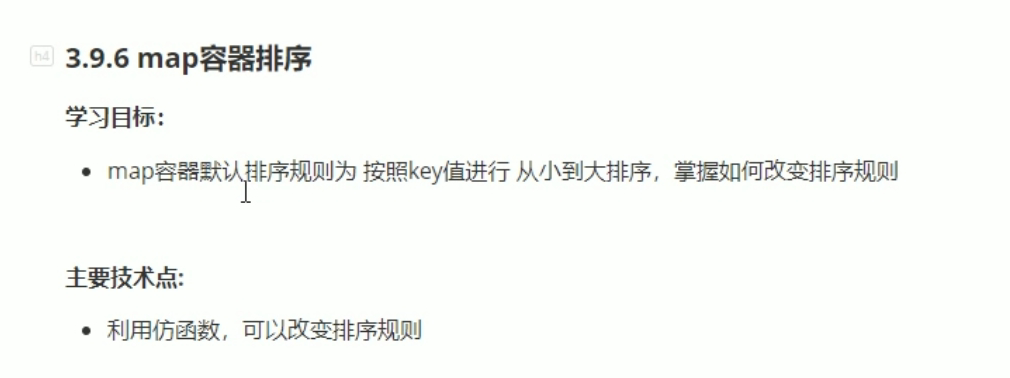


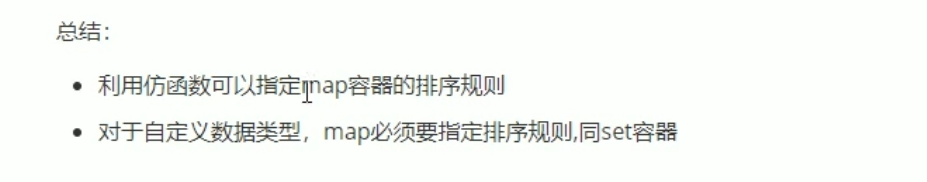




**Map与set用法基本一样。**







#include <iostream>

using namespace std;

#include <string>

#include<map>

class MyCompass

{

public:

bool operator()(int v1, int v2)

{

return v1 > v2;

}

};

void test01()

{

map<int, int, MyCompass>m;

m.insert(make\_pair(1, 10));

m.insert(make\_pair(2, 20));

m.insert(make\_pair(5, 50));

m.insert(make\_pair(3, 30));

m.insert(make\_pair(4, 40));

for (map<int, int, MyCompass>::iterator it = m.begin(); it != m.end(); it++)

{

cout << "key: " << (\*it).first << " value = " << it->second << endl;

}

cout << endl;

}

int main()

{

test01();

system("pause");

return 0;

}