

#include <iostream>

using namespace std;

#include <string>

#include <set>

void printSet(set<int>&s)

{

for(set<int>::iterator it = s.begin(); it != s.end(); it++)

{

cout << \*it << " ";

}

cout << endl;

}

void test01()

{

set<int> s1;

s1.insert(20);

s1.insert(30);

s1.insert(40);

s1.insert(10);

s1.insert(30);//set容器按照所有元素插入的时候自动排序

//set你容许插入重复值。

printSet(s1);

set<int>s2(s1);

printSet(s2);

set<int> s3;

s3 = s2;

printSet(s3);

}

int main()

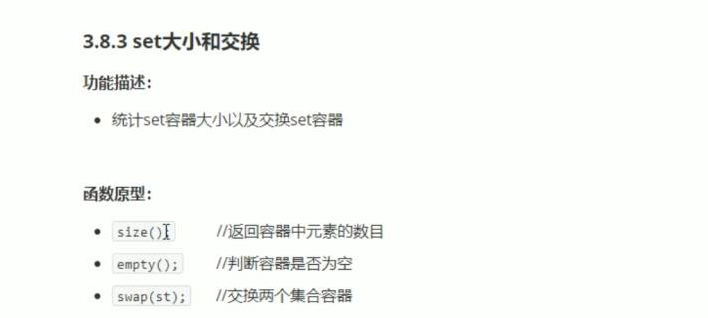
{

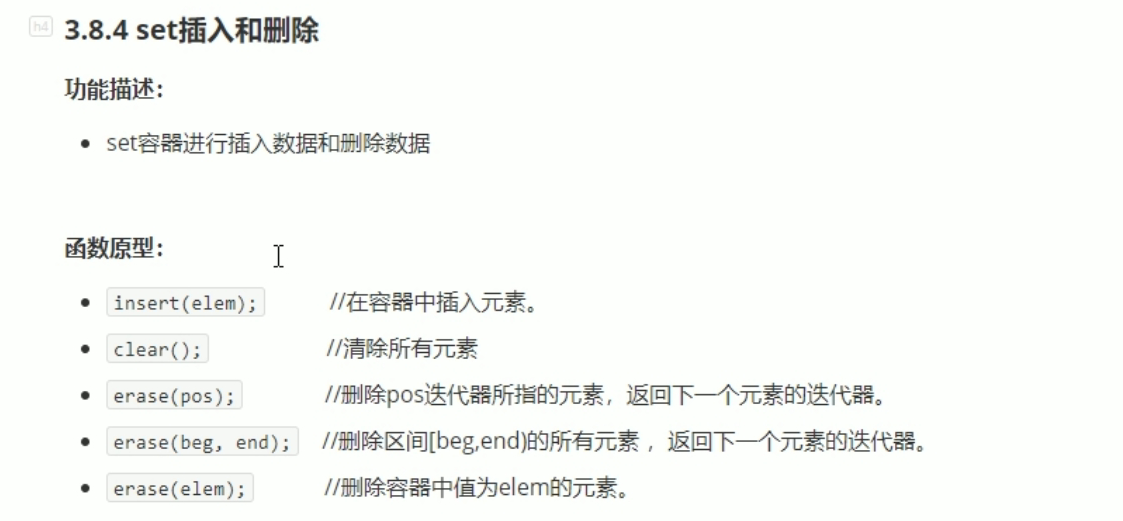
test01();

system("pause");

return 0;

}





#include <iostream>

using namespace std;

#include <string>

#include <set>

void printSet(set<int>&s)

{

for(set<int>::iterator it = s.begin(); it != s.end(); it++)

{

cout << \*it << " ";

}

cout << endl;

}

void test01()

{

set<int> s1;

s1.insert(20);

s1.insert(30);

s1.insert(40);

s1.insert(10);

s1.insert(30);//set容器按照所有元素插入的时候自动排序

//set你容许插入重复值。

printSet(s1);

//s1.erase(s1.begin());

//printSet(s1);

//s1.erase(30);

//printSet(s1);

}

int main()

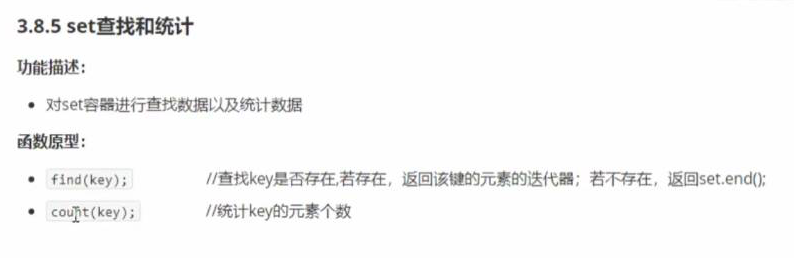
{

test01();

system("pause");

return 0;

}



#include <iostream>

using namespace std;

#include <string>

#include <set>

void printSet(set<int>&s)

{

for(set<int>::iterator it = s.begin(); it != s.end(); it++)

{

cout << \*it << " ";

}

cout << endl;

}

void test01()

{

set<int> s1;

s1.insert(20);

s1.insert(30);

s1.insert(40);

s1.insert(10);

s1.insert(30);

printSet(s1);

set<int>::iterator pos = s1.find(30);

if (pos != s1.end())

{

cout << "找到元素： " << \*pos << endl;

}

else

{

cout << "未找到元素。" << endl;

}

}

void test02()

{

set<int> s1;

s1.insert(20);

s1.insert(30);

s1.insert(40);

s1.insert(10);

s1.insert(30);

printSet(s1);

int num = s1.count(30);

cout << "num = " << num<<endl;//set容器不能查重复的数字只能0和1

}

int main()

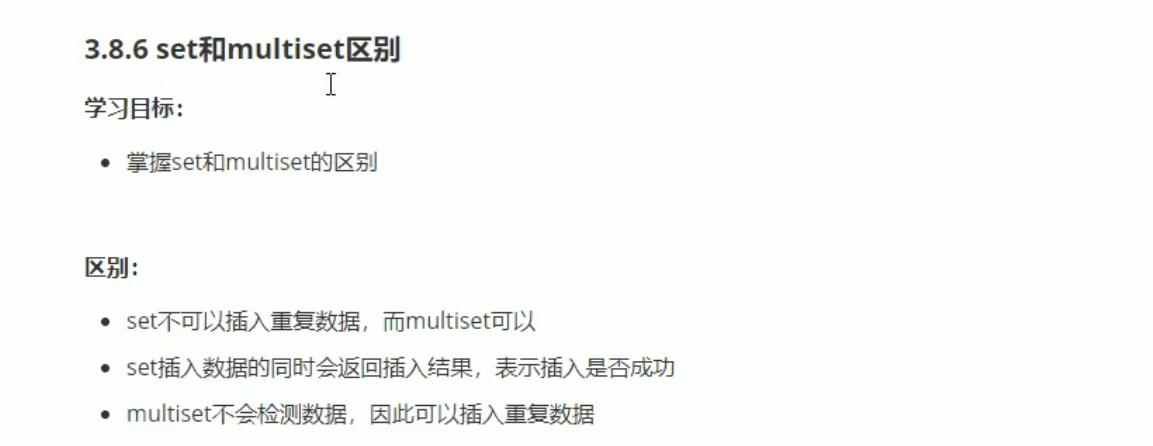
{

test02();

system("pause");

return 0;

}





#include <iostream>

using namespace std;

#include <string>

#include <set>

void test01()

{

set<int> s1;

pair<set<int>::iterator ,bool> ret = s1.insert(20);

if (ret.second)

{

cout << "第一次插入成功。" << endl;

}

else

{

cout << "第一次插入失败。" << endl;

}

ret = s1.insert(20);

if (ret.second)

{

cout << "第一次插入成功。" << endl;

}

else

{

cout << "第一次插入失败。" << endl;

}

multiset<int> ms;

ms.insert(10);

ms.insert(10);

for (multiset<int> ::iterator it = ms.begin();it != ms.end(); it++)

{

cout << \*it << " ";

}

cout << endl;

}

int main()

{

test01();

system("pause");

return 0;

}



#include <iostream>

using namespace std;

#include <string>

void test01()

{

pair<string, int> p("tom", 20);

cout << "姓名是 ： " << p.first << " 年龄是： " << p.second << endl;

pair<string, int>p2 = make\_pair("jerry", 18);

cout << "姓名是 ： " << p2.first << " 年龄是： " << p2.second << endl;

}

int main()

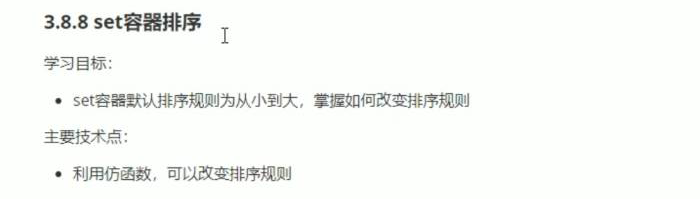
{

test01();

system("pause");

return 0;

}



#include <iostream>

using namespace std;

#include <string>

#include<set>

class MyCompare

{

public:

bool operator()(int v1,int v2)

{

return v1 > v2;

}

};

void test01()

{

set<int> s1;

s1.insert(10);

s1.insert(20);

s1.insert(30);

s1.insert(40);

s1.insert(50);

s1.insert(60);

for (set<int>::iterator it = s1.begin(); it != s1.end(); it++)

{

cout << \*it << " ";

}

cout << endl;

set<int, MyCompare> s2;

s2.insert(10);

s2.insert(20);

s2.insert(30);

s2.insert(40);

s2.insert(50);

s2.insert(60);

for (set<int, MyCompare>::iterator it = s2.begin(); it != s2.end(); it++)

{

cout << \*it << " ";

}

cout << endl;

}

int main()

{

test01();

system("pause");

return 0;

}