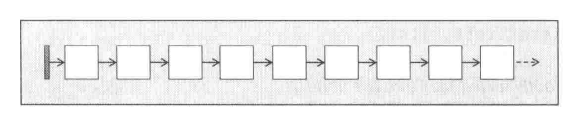
forward\_list内部以singly linked list管理元素，没有双向迭代器，不提供随机访问。



例：程序stl\_test47

void PrintLists(const *string*& str, const *forward\_list*<int>& list1, const *forward\_list*<int>& list2)

{

*cout* << str << *endl*;

*cout* << "list1: ";

*copy*(list1.*cbegin*(), list1.*cend*(), *ostream\_iterator*<int>(*cout*, " "));

*cout* << *endl* << "list2: ";

*copy*(list2.*cbegin*(), list2.*cend*(), *ostream\_iterator*<int>(*cout*, " "));

*cout* << *endl* << *endl*;

}

int *main*(int argc, char\* argv[])

{

*forward\_list*<int> list1 = {1, 2, 3, 4};

*forward\_list*<int> list2 = {77, 88, 99};

PrintLists("initial: ", list1, list2);

// 头部插入元素

list2.*insert\_after*(list2.*before\_begin*(), 99);

list2.*push\_front*(10);

list2.*insert\_after*(list2.*before\_begin*(), {10, 11, 12, 13});

PrintLists("6 new elems: ", list1, list2);

list1.*insert\_after*(list1.*before\_begin*(), list2.*begin*(), list2.*end*());

PrintLists("list2 into list1: ", list1, list2);

// 删除第2个元素

list2.*erase\_after*(list2.*begin*());

// 删除第1个99后的所有元素

list2.*erase\_after*(*find*(list2.*begin*(), list2.*end*(), 99), list2.*end*());

PrintLists("delete 2nd and after 99:", list1, list2);

list1.*sort*(*less*<int>());

list2 = list1;

list2.*unique*(); // 删除相邻的相同元素，只保留1个

PrintLists("sorted and unique: ", list1, list2);

// 合并并且保持排序

list1.*merge*(list2);

PrintLists("merged: ", list1, list2);

return 0;

}

输出为：

list1: 1 2 3 4

list2: 10 11 12 13 10 99 77 88 99

list2 into list1:

list1: 10 11 12 13 10 99 77 88 99 1 2 3 4

list2: 10 11 12 13 10 99 77 88 99

delete 2nd and after 99:

list1: 10 11 12 13 10 99 77 88 99 1 2 3 4

list2: 10 12 13 10 99

sorted and unique:

list1: 1 2 3 4 10 10 11 12 13 77 88 99 99

list2: 1 2 3 4 10 11 12 13 77 88 99

merged:

list1: 1 1 2 2 3 3 4 4 10 10 10 11 11 12 12 13 13 77 77 88 88 99 99 99

list2: