C++标准库提供了特殊迭代器，称为迭代器适配器。

1. Insert iterator（插入型迭代器）

可以使算法以insert方式而非overwrite方式运作。

例：程序stl\_test33

*list*<int> coll1 = {1, 2, 3, 4, 5, 6, 7, 8, 9};

*vector*<int> coll2;

// 插入容器末端

// 在内部调用push\_back执行插入

*copy*(coll1.*cbegin*(), coll1.*cend*(), *back\_inserter*(coll2));

for (const auto& elem : coll2)

{

*cout* << elem << ' ';

}

*cout* << *endl*;

//vector<int> coll3; // error，vector不能执行push\_front操作

*deque*<int> coll3;

// 插入容器前端

// 在内部执行push\_front操作

*copy*(coll1.*cbegin*(), coll1.*cend*(), *front\_inserter*(coll3));

for (const auto& elem : coll3)

{

*cout* << elem << ' ';

}

*cout* << *endl*;

*set*<int> coll4;

// 一般性插入，内部调用insert()插入

// 可以用于关联式容器和无序容器

*copy*(coll1.*cbegin*(), coll1.*cend*(), *inserter*(coll4, coll4.*begin*()));

for (const auto& elem : coll4)

{

*cout* << elem << ' ';

}

*cout* << *endl*;

输出为：

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9

使用这种插入时迭代器适配器，被拷贝的容器不需要提前分配空间了。

1. Stream Iterator（串流迭代器）