

**STL：Standard Template Library** C++标准库的子集

容器Containers （Basis operation --> 增删改查） 【各类数据结构】

序列Sequence：不会自动排序，如：vector向量、deque双向队列、list双向链表

关联式Associative：按元素值大小自动升序排序，如：set、multiset、map、multimap

适配器Adaptor：stack、queue、priority\_queue（heap）

所谓的Adapter(适配器)，就是更具体化，更专门化的

算法Algorithms

各种高效的泛型算法，适用于各种容器

迭代器Iterators

算法通过迭代器来定位和操作容器中的元素（与指针类似）

每个容器内嵌迭代器 -iterator const\_只读 reverse\_逆向的

Container Iterator  Algorithm

容器

(内存自动管理)

函数对象：重载了operator()，比函数指针更好地实现回调函数

Sequence Containers

Vector: 动态调整的数组

Deque: 双向队列 （小片内是连续的，小片之间用链表相连）

List: 双向链表

Associative Containers

Set/Multiset: 集合 / 多重集合(允许元素重复)

Map/Multimap: 映射/多重映射(允许key相同)

key-value: 键值对，通过关键字可以找到完整的信息

【由pair结构体的定义 or make\_pair()函数生成】

Container Adaptor

string、stack栈、queue队列 、priority\_queue优先队列（default: max heap）

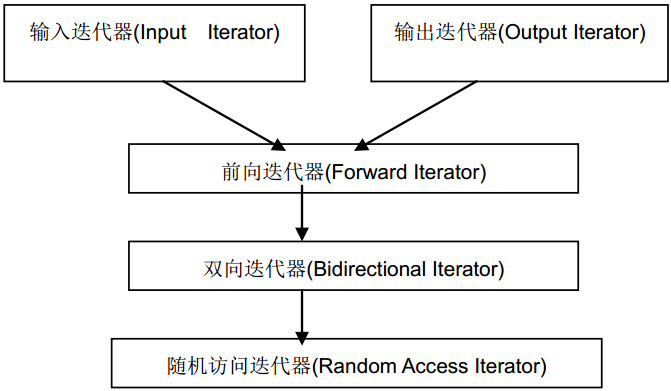
迭代器

(指向容器中元素的位置)

1 每个容器自带定制的迭代器

2 可以访问容器中的任意一个元素

3 操作符： ①+-指向的平移 ②=赋值 ③->指向 ④==/!=判断是否指向相同



begin() 指向首元素

end() 指向尾元素之后的位置

（前缀： c = constant, r = reverse）

advance() 迭代器的前进

distance() 迭代器之间的距离

iter\_swap() 迭代器的值交换

Insert Iterator

back\_inserter、front\_inserter、inserter

Stream Iterator

ostream\_iterator、istream\_iterator

算法

数值计算 <numeric>

日常算法 <algorithm>

**原数据不变**

for\_each 遍历

merge 归并有序序列

find\_if 附加条件查找

binary\_search 二分查找

mismatch 找出两个序列的相异的第一个位置

search 搜索子序列

**改写原数据**

copy 复制

sort 快速排序

stable\_sort 稳定性排序

transform 变换映射

reverse 逆序

unique 去重

generate 按规则生成数据

replace 替换

fill 填充

remove 删除

prev/next\_permutation 前/后一个排列

函数对象

**适配器**

bind/binder 确定函数的部分参数

not 返回值取反

mem\_fun 将成员函数转化为对象函数

ptr\_fun