**C++ Vectors**

**Vectors 包含着一系列连续存储的元素,其行为和数组类似。访问Vector中的任意元素或从末尾添加元素都可以在**[**常量级时间复杂度**](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/complexity.html)**内完成，而查找特定值的元素所处的位置或是在Vector中插入元素则是**[**线性时间复杂度**](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/complexity.html)**。**

|  |  |
| --- | --- |
| [Constructors](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#Constructors) | 构造函数 |
| [Operators](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#Operators) | 对vector进行赋值或比较 |
| [assign()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#assign) | 对Vector中的元素赋值 |
| [at()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#at) | 返回指定位置的元素 |
| [back()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#back) | 返回最末一个元素 |
| [begin()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#begin) | 返回第一个元素的迭代器 |
| [capacity()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#capacity) | 返回vector所能容纳的元素数量(在不重新分配内存的情况下） |
| [clear()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#clear) | 清空所有元素 |
| [empty()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#empty) | 判断Vector是否为空（返回true时为空） |
| [end()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#end) | 返回最末元素的迭代器(译注:实指向最末元素的下一个位置) |
| [erase()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#erase) | 删除指定元素 |
| [front()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#front) | 返回第一个元素 |
| [get\_allocator()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#get_allocator) | 返回vector的内存分配器 |
| [insert()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#insert) | 插入元素到Vector中 |
| [max\_size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#max_size) | 返回Vector所能容纳元素的最大数量（上限） |
| [pop\_back()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#pop_back) | 移除最后一个元素 |
| [push\_back()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#push_back) | 在Vector最后添加一个元素 |
| [rbegin()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#rbegin) | 返回Vector尾部的逆迭代器 |
| [rend()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#rend) | 返回Vector起始的逆迭代器 |
| [reserve()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#reserve) | 设置Vector最小的元素容纳数量 |
| [resize()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#resize) | 改变Vector元素数量的大小 |
| [size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#size) | 返回Vector元素数量的大小 |
| [swap()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppvector_details.html#swap) | 交换两个Vector |

**C++ Stacks（堆栈）**

C++ Stack（堆栈） 是一个容器类的改编，为程序员提供了堆栈的全部功能，——也就是说实现了一个先进后出（FILO）的数据结构。

|  |  |
| --- | --- |
| [操作](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppstack_details.html#Operators) | 比较和分配堆栈 |
| [empty()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppstack_details.html#empty) | 堆栈为空则返回真 |
| [pop()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppstack_details.html#pop) | 移除栈顶元素 |
| [push()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppstack_details.html#push) | 在栈顶增加元素 |
| [size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppstack_details.html#size) | 返回栈中元素数目 |
| [top()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppstack_details.html#top) | 返回栈顶元素 |

**C++ Sets**

集合(Set)是一种包含已排序对象的关联容器

|  |  |
| --- | --- |
| [begin()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#begin) | 返回指向第一个元素的迭代器 |
| [clear()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#clear) | 清除所有元素 |
| [count()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#count) | 返回某个值元素的个数 |
| [empty()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#empty) | 如果集合为空，返回true |
| [end()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#end) | 返回指向最后一个元素的迭代器 |
| [equal\_range()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#equal_range) | 返回集合中与给定值相等的上下限的两个迭代器 |
| [erase()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#erase) | 删除集合中的元素 |
| [find()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#find) | 返回一个指向被查找到元素的迭代器 |
| [get\_allocator()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#get_allocator) | 返回集合的分配器 |
| [insert()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#insert) | 在集合中插入元素 |
| [lower\_bound()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#lower_bound) | 返回指向大于（或等于）某值的第一个元素的迭代器 |
| [key\_comp()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#key_comp) | 返回一个用于元素间值比较的函数 |
| [max\_size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#max_size) | 返回集合能容纳的元素的最大限值 |
| [rbegin()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#rbegin) | 返回指向集合中最后一个元素的反向迭代器 |
| [rend()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#rend) | 返回指向集合中第一个元素的反向迭代器 |
| [size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#size) | 集合中元素的数目 |
| [swap()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#swap) | 交换两个集合变量 |
| [upper\_bound()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#upper_bound) | 返回大于某个值元素的迭代器 |
| [value\_comp()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppset_details.html#value_comp) | 返回一个用于比较元素间的值的函数 |

**C++ Maps**

C++ Maps是一种关联式容器，包含“关键字/值”对

|  |  |
| --- | --- |
| [begin()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#begin) | 返回指向map头部的迭代器 |
| [clear()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#clear) | 删除所有元素 |
| [count()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#count) | 返回指定元素出现的次数 |
| [empty()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#empty) | 如果map为空则返回true |
| [end()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#end) | 返回指向map末尾的迭代器 |
| [equal\_range()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#equal_range) | 返回特殊条目的迭代器对 |
| [erase()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#erase) | 删除一个元素 |
| [find()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#find) | 查找一个元素 |
| [get\_allocator()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#get_allocator) | 返回map的配置器 |
| [insert()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#insert) | 插入元素 |
| [key\_comp()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#key_comp) | 返回比较元素key的函数 |
| [lower\_bound()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#lower_bound) | 返回键值>=给定元素的第一个位置 |
| [max\_size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#max_size) | 返回可以容纳的最大元素个数 |
| [rbegin()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#rbegin) | 返回一个指向map尾部的逆向迭代器 |
| [rend()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#rend) | 返回一个指向map头部的逆向迭代器 |
| [size()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#size) | 返回map中元素的个数 |
| [swap()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#swap) | 交换两个map |
| [upper\_bound()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#upper_bound) | 返回键值>给定元素的第一个位置 |
| [value\_comp()](mk:@MSITStore:F:\mycode\C++库函数参考.chm::/cppreference.com/cppmap_details.html#value_comp) | 返回比较元素value的函数 |