容器库

容器库是类模板与算法的汇集，允许程序员简单地访问常见数据结构，例如队列、链表和栈。有三类容器——顺序容器、关联容器和无序关联容器——每种都被设计为支持不同组的操作。

容器管理为其元素分配的存储空间，并提供直接或间接地通过迭代器（拥有类似指针属性的对象）访问它们的函数。

大多数容器拥有至少几个常见的成员函数，并共享功能。特定应用的最佳容器不仅依赖于提供的功能，还依赖于对于不同工作量的效率。

顺序容器

顺序容器实现能按顺序访问的数据结构。

array (C++11 起)	静态的相接数组 (类模板)
vector	动态的相接数组 (类模板)
deque	双端队列 (类模板)
forward_list (C++11 起)	单链表 (类模板)
list	双链表 (类模板)

关联容器

关联容器实现能快速查找（ *O(log n)* 复杂度）的数据结构。

set	唯一键的集合，按照键排序 (类模板)
map	键值对的集合，按照键排序，键是唯一的 (类模板)
multiset	键的集合，按照键排序 (类模板)
multimap	键值对的集合，按照键排序 (类模板)

无序关联容器

无序关联容器提供能快速查找（均摊 *O(1)* ，最坏情况 *O(n)* 的复杂度）的无序（哈希）数据结构。

unordered_set (C++11 起)	唯一键的集合，按照键生成散列 (类模板)
unordered_map (C++11 起)	键值对的集合，按照键生成散列，键是唯一的 (类模板)
unordered_multiset (C++11 起)	键的集合，按照键生成散列 (类模板)
unordered_multimap (C++11 起)	键值对的集合，按照键生成散列 (类模板)

容器适配器

容器适配器提供顺序容器的不同接口。

stack	堆栈适配器（LIFO） (类模板)
queue	改写容器来提供队列（FIFO数据结构） (类模板)
priority_queue	改写容器来提供优先级队列 (类模板)

span

span 是相接的对象序列上的非占有视图，某个其他对象占有序列的存储。

span (C++20)	相接的对象序列上的非占有视图 (类模板)

迭代器非法化

只读方法决不非法化迭代器或引用。修改容器内容的方法可能非法化迭代器和/或引用，总结于此表格。

<table>
<tr><th rowspan="2">类别</th><th rowspan="2">容器</th><th colspan="2">插入后……</th><th colspan="2">擦除后……</th><th rowspan="2">条件</th></tr>
<tr><th>迭代器合法?</th><th>引用合法?</th><th>迭代器合法?</th><th>引用合法?</th></tr>
<tr><td rowspan="6">顺序容器</td><td>array</td><td colspan="2">N/A</td><td colspan="2">N/A</td><td></td></tr>
<tr><td rowspan="3">vector</td><td colspan="2">否</td><td colspan="2">N/A</td><td>插入更改容量</td></tr>
<tr><td colspan="2">是</td><td colspan="2">是</td><td>被修改元素前</td></tr>
<tr><td colspan="2">否</td><td colspan="2">否</td><td>于被修改元素或其后</td></tr>
<tr><td rowspan="2">deque</td><td rowspan="2">否</td><td>是</td><td>是，除了被擦除元素</td><td></td><td>修改首或尾元素</td></tr>
<tr><td>否</td><td colspan="2">否</td><td>仅修改中部</td></tr>
<tr><td>list</td><td colspan="2">是</td><td colspan="2">是，除了被擦除元素</td><td></td></tr>
<tr><td>forward_list</td><td colspan="2">是</td><td colspan="2">是，除了被擦除元素</td><td></td></tr>
<tr><td>关联容器</td><td>set
multiset
map
multimap</td><td colspan="2">是</td><td colspan="2">是，除了被擦除元素</td><td></td></tr>
<tr><td rowspan="2">无序关联容器</td><td rowspan="2">unordered_set
unordered_multiset
unordered_map
unordered_multimap</td><td>否</td><td rowspan="2">是</td><td>N/A</td><td>插入导致重哈希</td></tr>
<tr><td>是</td><td>是，除了被擦除元素</td><td>无重哈希</td></tr>
</table>

此处**插入**指代任何添加一或多个元素到容器的方法，而**擦除**指代任何从容器移除一或多个元素的方法。

- 插入方法的例子是 `std::set::insert` 、 `std::map::emplace` 、 `std::vector::push_back` 和 `std::deque::push_front` 。
  - 注意 `std::unordered_map::operator[]` 也算，因为它可能插入元素到 map 中。
- 擦除方法的例子是 `std::set::erase` 、 `std::vector::pop_back` 、 `std::deque::pop_front` 和 `std::map::clear` 。
  - `clear` 非法化所有迭代器和引用。因为它擦除所有元素，这在技术上遵照上述规则。

尾后迭代器需要特别留意。通常像指向未被擦除元素的正常迭代器一般非法化此迭代器。故 `std::set::end` 决不被非法化， `std::unordered_set::end` 仅在重哈希时被非法化， `std::vector::end` 始终被非法化（因为它始终出现在被修改元素后），以此类推。

有一个例外：删除 `std::deque` 末元素的擦除操作*会*非法化尾后迭代器，尽管它不是容器的被擦除元素（或者说根本不是元素）。与 `std::deque` 迭代器的通用规则结合后，最终结果是*不*非法化 `std::deque::end` 的唯一修改操作是删除首元素，而非末元素的擦除。

线程安全

1. 能同时在不同容器上由不同线程调用所有容器函数。更广泛而言， C++ 标准库函数不读取能通过其他线程访问的对象，除非这些对象能直接或间接地经由函数参数，包含 this 指针访问。
2. 能同时在同一容器上由不同线程调用 `const` 成员函数。而且，成员函数 `begin()` 、 `end()`, `rbegin()` 、 `rend()` 、 `front()` 、 `back()` 、 `data()` 、 `find()` 、 `lower_bound()` 、 `upper_bound()` 、 `equal_range()` 、 `at()` 和除了关联容器中的 `operator[]` 对于线程安全的目标表现如同 `const` （即它们亦能同时在同一容器上由不同线程调用）。更广泛而言， C++ 标准库函数不修改对象，除非这些对象能直接或间接地经由函数参数，包含 this 指针访问。 (C++11 起)
3. 同一容器中不同元素能由不同线程同时修改，除了 `std::vector<bool>` 的元素（例如， `std::future` 对象的 vector 能从多个线程接收值）。
4. 迭代器操作（例如自增迭代器）读但不修改底层容器，而且能与同一容器上的其他迭代器操作同时由 const 成员函数执行。非法化任何迭代器的容器操作修改容器，且不能与任何在既存迭代器上的操作同时执行，即使这些迭代器未被非法化。
5. 同一容器上的元素可以同时由不指定为访问这些元素的函数修改。更广泛而言， C++ 标准库函数不间接读取能从其参数访问的对象（包含容器的其他对象），除非其规定要求如此。
6. 任何情况下，容器操作（还有算法，或其他 C++ 标准库函数）可于内部并行化，只要不更改用户可见的结果（例如 `std::transform` 可并行化，但指定了按顺序观览序列的每个元素的 `std::for_each` 不行）

成员函数表格

- C++03 起存在的函数
- C++11 起存在的函数
- C++17 起存在的函数
- C++20 起存在的函数

		顺序容器					关联容器				无序关联容器				容器适配器		
头文件		<array>	<vector>	<deque>	<forward_list>	<list>	<set>		<map>		<unordered_set>		<unordered_map>		<stack>	<queue>	
容器		array	vector	deque	forward_list	list	set	multiset	map	multimap	unordered_set	unordered_multiset	unordered_map	unordered_multimap	stack	queue	priority_queue
	(构造函数)	(隐式)	vector	deque	forward_list	list	set	multiset	map	multimap	unordered_set	unordered_multiset	unordered_map	unordered_multimap	stack	queue	priority_queue
	(析构函数)	(隐式)	~vector	~deque	~forward_list	~list	~set	~multiset	~map	~multimap	~unordered_set	~unordered_multiset	~unordered_map	~unordered_multimap	~stack	~queue	~priority_queue
	operator=	(隐式)	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=	operator=
	assign		assign	assign	assign	assign											
迭代器	begin	begin	begin	begin	begin	begin	begin	begin	begin	begin	begin	begin	begin	begin			
迭代器	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin	cbegin			
迭代器	end	end	end	end	end	end	end	end	end	end	end	end	end	end			
迭代器	cend	cend	cend	cend	cend	cend	cend	cend	cend	cend	cend	cend	cend	cend			
迭代器	rbegin	rbegin	rbegin	rbegin		rbegin	rbegin	rbegin	rbegin	rbegin							
迭代器	crbegin	crbegin	crbegin	crbegin		crbegin	crbegin	crbegin	crbegin	crbegin							
迭代器	rend	rend	rend	rend		rend	rend	rend	rend	rend							
迭代器	crend	crend	crend	crend		crend	crend	crend	crend	crend							
元素访问	at	at	at	at					at				at				
元素访问	operator[]	operator[]	operator[]	operator[]					operator[]				operator[]				
元素访问	data	data	data														
元素访问	front	front	front	front	front	front										front	top
元素访问	back	back	back	back		back									top	back	
容量	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty	empty
容量	size	size	size	size		size	size	size	size	size	size	size	size	size	size	size	size
容量	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size	max_size			
容量	resize		resize	resize	resize	resize											
容量	capacity		capacity								bucket_count	bucket_count	bucket_count	bucket_count			
容量	reserve		reserve								reserve	reserve	reserve	reserve			
容量	shrink_to_fit		shrink_to_fit	shrink_to_fit													
修改器	clear		clear	clear	clear	clear	clear	clear	clear	clear	clear	clear	clear	clear			
修改器	insert		insert	insert	insert_after	insert	insert	insert	insert	insert	insert	insert	insert	insert			
修改器	insert_or_assign								insert_or_assign				insert_or_assign				
修改器	emplace		emplace	emplace	emplace_after	emplace	emplace	emplace	emplace	emplace	emplace	emplace	emplace	emplace			
修改器	emplace_hint						emplace_hint	emplace_hint	emplace_hint	emplace_hint	emplace_hint	emplace_hint	emplace_hint	emplace_hint			
修改器	try_emplace								try_emplace				try_emplace				
修改器	erase		erase	erase	erase_after	erase	erase	erase	erase	erase	erase	erase	erase	erase			
修改器	push_front			push_front	push_front	push_front											
修改器	emplace_front			emplace_front	emplace_front	emplace_front											
修改器	pop_front			pop_front	pop_front	pop_front										pop	pop
修改器	push_back		push_back	push_back		push_back									push	push	push
修改器	emplace_back		emplace_back	emplace_back		emplace_back									emplace	emplace	emplace
修改器	pop_back		pop_back	pop_back		pop_back									pop		
修改器	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap	swap
修改器	merge				merge	merge	merge	merge	merge	merge	merge	merge	merge	merge			
修改器	extract						extract	extract	extract	extract	extract	extract	extract	extract			
链表操作	splice				splice_after	splice											
链表操作	remove				remove	remove											
链表操作	remove_if				remove_if	remove_if											
链表操作	reverse				reverse	reverse											
链表操作	unique				unique	unique											
链表操作	sort				sort	sort											
查找	count						count	count	count	count	count	count	count	count			
查找	find						find	find	find	find	find	find	find	find			
查找	contains						contains	contains	contains	contains	contains	contains	contains	contains			
查找	lower_bound						lower_bound	lower_bound	lower_bound	lower_bound							
查找	upper_bound						upper_bound	upper_bound	upper_bound	upper_bound							
查找	equal_range						equal_range	equal_range	equal_range	equal_range	equal_range	equal_range	equal_range	equal_range			
观察器	key_comp						key_comp	key_comp	key_comp	key_comp							
观察器	value_comp						value_comp	value_comp	value_comp	value_comp							
观察器	hash_function										hash_function	hash_function	hash_function	hash_function			
观察器	key_eq										key_eq	key_eq	key_eq	key_eq			
分配器	get_allocator		get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator	get_allocator			
容器		array	vector	deque	forward_list	list	set	multiset	map	multimap	unordered_set	unordered_multiset	unordered_map	unordered_multimap	stack	queue	priority_queue
		顺序容器					关联容器				无序关联容器				容器适配器		

来自“https://zh.cppreference.com/mwiki/index.php?title=cpp/container&oldid=55390”