

1. Sequence Containers

1.1 array

1.1.1 使用须知

- 包含头文件 `#include<array>`
- `using std::array`
- `array` 中元素个数不可变

1.1.2 成员函数

- 构造函数, `array<T, SIZE> a;`
 - `T` 为容器中元素类型, `SIZE` 必须是常量。
- 析构函数
- 赋值运算符

1.1.3 元素接口

- `at`, `a.at(i)`, 获取下标为 `i` 的元素。
- `operator[]`, `a[i]`, 获取下标为 `i` 的元素。
- `front`, `a.front()`, 获取第一个元素。
- `back`, `a.back()`, 获取最后一个元素。
- `data`, `a.data()`, 获取数组地址。

1.1.4 迭代器

- `begin`, `a.begin()`, 返回指向第一个元素的迭代器。
- `end`, `a.end()`, 返回最后一个元素后的迭代器。
- `cbegin`, `a.cbegin()`, 返回指向第一个元素的迭代器, `c` 是 `const`, 表示不能修改容器。
- `cend`, `a.cend()`, 返回最后一个元素后的迭代器。
- `rbegin`, `a.rbegin()`, 返回指向第一个元素的反转迭代器, `r` 是 `reverse`, 表示反转容器。
- `rend`, `a.rend()`, 返回最后一个元素后的反转迭代器。
- `crbegin`, `a.crbegin()`, 返回指向第一个元素的反转迭代器, `c` 是 `const`, 表示不能修改容器, `r` 是 `reverse`, 表示反转容器。
- `crend`, `a.crend()`, 返回最后一个元素后的反转迭代器。

1.1.5 容量

- `empty, a.empty()`, 检查数组是否为空, 为空返回 `true`, 否则返回 `false`。
- `size, a.size()`, 返回数组中元素个数。
- `max_size, a.max_size()`, 返回数组的容量, 即最大可放元素个数。

1.1.6 操作

- `fill, a.fill(const T& t)`; 用类型为 `T` 的 `t` 填充数组 `a`。
- `swap, a.swap(array& b)`, 交换两个数组内容。

1.1.7 非成员函数

- `operator==, a == b`, 判断两个数组是否相等, 相等返回 `true`, 否则返回 `false`。

1.1.8 案例

参考文件 `array.cpp`

1.2 vector

1.3 deque

1.4 list

1.5 forward_list

2. Associative Containers

2.1 set

2.2 multiset

2.3 map

2.4 multimap

3. Unordered Associative Containers

3.1 unordered_set

3.2 unordered_multiset

3.3 unordered_map

3.4 unordered_multimap

4. Container Adaptors

4.1 stack

4.2 queue

4.3 priority_queue