

Bài 5.6: Tìm hiểu thư viện queue

- ✓ Tổng quan
- ✓ Các hàm thông dụng
- ✓ Ví dụ minh họa

Tổng quan

- Queue là container adaptor được thiết kế để vận hành theo quy tắc FIFO, trong đó các phần tử được chèn vào cuối và lấy ra ở đầu.
- Queue được sử dụng phổ biến trong xử lý các lệnh, yêu cầu truy cập tài nguyên dùng chung của hệ thống, truyền dữ liệu liên tiến trình, trong hệ điều hành, giao tiếp qua mạng...

Các hàm thông dụng

| Tên hàm | Mô tả |
|--|--|
| <code>queue() : queue(Container()) {}</code> | Hàm khởi tạo mặc định. |
| <code>explicit queue(const Container& cont = Container());</code> <code>explicit queue(const Container& cont);</code> | Tạo queue với các phần tử nhân bản từ trong container cho trước. |
| <code>explicit queue(Container&& cont);</code> | Tạo queue với các phần tử được move từ container cho trước. |
| <code>queue(const queue& other);</code> | Copy constructor tạo queue từ một queue có sẵn. |
| <code>queue(const queue&& other);</code> | Move constructor tạo queue từ queue cho trước. |
| <code>template<class InputIt> queue(InputIt first, InputIt last);(C++23)</code> | Tạo queue từ nội dung cho trước trong nửa khoảng [first, last). |
| <code>operator=</code> | Thay thế nội dung của container adaptor bằng nội dung mới. |
| <code>reference front();</code> <code>const_reference front() const;</code> | Trả về tham chiếu đến phần tử đầu queue. |

Các hàm thông dụng

| | |
|--|--|
| reference back(); const_reference back() const; | Trả về tham chiếu đến phần tử cuối queue. |
| bool empty() const; [[nodiscard]] bool empty() const;(C++20) | Kiểm tra xem queue có rỗng hay không. |
| size_type size() const; | Trả về số lượng phần tử hiện có trong container. |
| void push(const value_type& value); void push(value_type&& value); | Thêm phần tử mới vào cuối queue. |
| template<class ... Args> void emplace(Args... args); template<class ... Args> decltype(auto) emplace(Args... args);(C++17) | Thêm mới phần tử vào cuối queue. Phần tử của queue được tạo tại chỗ, không sao chép hay move từ phần tử nào. |
| void pop(); | Xóa phần tử ở đầu queue. |
| Void swap(queue& other) noexcept(); | Tráo đổi phần tử của 2 queue. |
| Các hàm nạp chồng toán tử ==, !=, <, <=, >, >=, <=> | So sánh các phần tử trong 2 queue. |

Nội dung tiếp theo

Tìm hiểu thư viện <priority_queue>