

## Bài 4.4: Giới thiệu thư viện stack

---

- ✓ Khái niệm và đặc điểm
- ✓ Các hàm thông dụng và mô tả
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

# Khái niệm và mô tả

- Stack là một container adaptor được thiết kế để vận hành theo quy tắc đặc biệt: LIFO, trong đó các phần tử được chèn và xóa chỉ từ một đầu của container.
- Đây là một cấu trúc dữ liệu đặc biệt được ứng dụng phổ biến trong các phép tính toán, thực hiện undo, redo, tiến và lùi khi duyệt web trên trình duyệt, đảo thứ tự string, các thuật toán quay lui, đồ thị, quản lý cấp phát bộ nhớ...

# Các hàm thông dụng và mô tả

Tên hàm	Mô tả
<code>stack() : stack(Container()) {}</code>	Hàm khởi tạo mặc định.
<code>explicit stack(const Container&amp; cont = Container());</code> <code>explicit stack(const Container&amp; cont);</code>	Tạo stack với các phần tử nhân bản từ trong container cho trước.
<code>explicit stack(Container&amp;&amp; cont);</code>	Tạo stack với các phần tử được move từ container cho trước.
<code>stack(const stack&amp; other);</code>	Copy constructor tạo stack từ một stack có sẵn.
<code>stack(const stack&amp;&amp; other);</code>	Move constructor tạo stack từ stack cho trước.
<code>template&lt;class InputIt&gt; stack(InputIt first, InputIt last);(C++23)</code>	Tạo stack 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 top();</code> <code>const_reference top() const;</code>	Trả về tham chiếu đến phần tử trên đỉnh stack. Đây là phần tử vừa được push vào stack.

# Các hàm thông dụng và mô tả

<code>bool empty() const;</code> <code>[[nodiscard]] bool empty() const; (C++20)</code>	Kiểm tra xem stack có rỗng hay không.
<code>size_type size() const;</code>	Trả về số lượng phần tử hiện có trong container.
<code>void push(const value_type&amp; value);</code> <code>void push(value_type&amp;&amp; value);</code>	Thêm phần tử mới vào đỉnh stack.
<code>template&lt;class ... Args&gt; void emplace(Args... args);</code> <code>template&lt;class ... Args&gt; decltype(auto)</code> <code>emplace(Args... args); (C++17)</code>	Thêm mới phần tử vào đỉnh stack. Phần tử này được tạo tại chỗ, không move hay copy từ phần tử nào.
<code>void pop();</code>	Xóa phần tử trên đỉnh stack.
<code>Void swap(stack&amp; other) noexcept();</code>	Tráo đổi phần tử của 2 stack.
Các hàm nạp chồng toán tử <code>==, !=, &lt;, &lt;=, &gt;, &gt;=, &lt;=&gt;</code>	So sánh các phần tử trong 2 stack.

# Nội dung tiếp theo

Tìm hiểu về hàng đợi