

Chương 18: Set (Tập hợp) trong C++

Chinh phục tri thức cùng Learning How to Learn

Slide Learning C++

2026

Lời chào dẫn nhập

Người đồng hành

Chào bạn! Tôi là người đồng hành cùng bạn trên con đường chinh phục tri thức. Chúng ta sẽ cùng khám phá công cụ thông minh mang tên **Set (Tập hợp)**.

Triết lý học tập

Chúng ta sẽ không học vẹt code. Chúng ta xây dựng các **mô hình tâm trí (mental models)** để thực sự nhìn thấy cách dữ liệu vận hành.

- ❶ **Chương 1:** Set là gì? – Chiếc túi thần kỳ của sự độc nhất.
- ❷ **Chương 2:** Khai báo và Khởi tạo – Chuẩn bị túi.
- ❸ **Chương 3:** Thêm và Xóa – Quy tắc gác cửa.
- ❹ **Chương 4:** Tìm kiếm (C++20) – Kỹ năng thám tử.
- ❺ **Chương 5:** Duyệt Set – Cuộc diễu hành các con số.
- ❻ **Chương 6:** Iterator – Ngón tay chỉ đường.

Chương 1: Set là gì?

Hãy tưởng tượng một chiếc túi thần kỳ có hai khả năng:

- **Chống trùng lặp:** Bạn bỏ 2 viên bi giống hệt nhau, túi sẽ tự động nhỏ viên thứ hai ra. Chỉ giữ lại duy nhất một bản sao.
- **Tự động sắp xếp:** Bỏ số 5, 1, 9 vào túi, khi mở ra chúng sẽ đứng theo hàng: 1, 5, 9.

Bức tranh toàn cảnh

Trong Set, mỗi thành viên là **duy nhất** và luôn được **sắp xếp** ngay ngắn từ nhỏ đến lớn.

Câu hỏi: Nếu tôi có một Set $\{2, 4, 8\}$ và thêm số 4 vào một lần nữa. Kết quả sẽ là gì?

- A. $\{2, 4, 4, 8\}$
- B. $\{2, 4, 8\}$
- C. $\{4, 2, 8\}$

Câu hỏi: Nếu tôi có một Set $\{2, 4, 8\}$ và thêm số 4 vào một lần nữa. Kết quả sẽ là gì?

- A. $\{2, 4, 4, 8\}$
- B. $\{2, 4, 8\}$
- C. $\{4, 2, 8\}$

Đáp án: B

Chính xác! Vì quy luật gác cửa, Set từ chối nhận thêm số 4 đã tồn tại.

Chương 2: Khai báo và Khởi tạo

Để dùng Set, bạn cần `#include <set>`.

```
1 #include <iostream>
2 #include <set>
3
4 using namespace std;
5
6 int main() {
7     // Cách 1: Khai báo một chiếc túi rỗng
8     set<int> túi_so_nguyen;
9
10    // Cách 2: (C++17) Khởi tạo có sẵn phần tử
11    set<int> túi_than_ky = {1, 5, 2, 5, 1};
12    // Kết quả thực tế: {1, 2, 5}
13
14    return 0;
15 }
16
```

Câu hỏi: Với dòng code `set danh_sach = {10, 20, 10, 30};`

- 1 Kiểu dữ liệu máy tính tự hiểu là gì?
- 2 Các phần tử thực tế trong túi là gì?

Câu hỏi: Với dòng code `set danh_sach = {10, 20, 10, 30};`

- 1 Kiểu dữ liệu máy tính tự hiểu là gì?
- 2 Các phần tử thực tế trong túi là gì?

Giải đáp

1. Kiểu `int`.
2. Kết quả: `{10, 20, 30}`.

Chương 3: Thêm và Xóa

Lệnh `.insert()` - Thêm vào

Người gác cổng kiểm tra: Nếu đã có thì từ chối, nếu chưa có thì cho vào và **xếp đúng vị trí thứ tự**.

Lệnh `.erase()` - Đuổi ra

Tìm đúng giá trị và mời ra khỏi hàng. Hàng ngũ tự động khép kín lại.

Tốc độ

Thêm/Xóa trong Set rất nhanh nhờ cấu trúc **Cây (Tree)** bên dưới.

Thực hành Thêm và Xóa

```
1 #include <iostream>
2 #include <set>
3 using namespace std;
4
5 int main() {
6     set<int> tap_hop;
7     tap_hop.insert(40);
8     tap_hop.insert(10);
9     tap_hop.insert(20);
10    tap_hop.insert(10); // Bi tu choi
11
12    // XOA PHAN TU
13    tap_hop.erase(20);
14
15    // Ket qua con lai: {10, 40}
16    return 0;
17 }
```

Chương 4: Tìm kiếm (C++20)

Ảnh hưởng từ điển

Tìm chữ "M" không lật từng trang. Bạn mở đôi sách, thấy "K", biết "M" ở nửa sau. Cách "chặt đôi" này giúp Set tìm kiếm cực nhanh.

Lệnh .contains() mới

Từ C++20, thay vì dùng các lệnh phức tạp, ta chỉ cần:

```
if (tui_do_choi.contains(30)) { ... }
```

Code Tìm kiếm hiện đại

```
1 #include <iostream>
2 #include <set>
3 using namespace std;
4
5 int main() {
6     set tui_do_choi = {10, 20, 30, 40, 50};
7
8     if (tui_do_choi.contains(30)) {
9         cout << "Tim thay so 30!";
10    } else {
11        cout << "Khong co 30.";
12    }
13    return 0;
14 }
```

Chương 5: Duyệt Set

Duyệt (Iteration) là đi qua từng phần tử để in ra hoặc tính toán.

Lưu ý quan trọng

Set không dùng chỉ số `[i]` như Vector. Vì nó không lưu kiểu ngăn kéo đánh số mà lưu theo kiểu **Cây**. Bạn phải đi từ gốc đến ngọn.

Cách hiện đại (For-each)

```
for (const auto& ten : biet_doi) { ... }
```

Thực hành Duyệt Set

```
1 #include <iostream>
2 #include <set>
3 #include <string>
4 using namespace std;
5
6 int main() {
7     set<string> biet_doi = {"Ironman", "Thor", "Hulk", "Thor"};
8
9     for (const auto& ten : biet_doi) {
10         cout << "- " << ten << endl;
11     }
12     return 0;
13 }
14 // Ket qua: Hulk, Ironman, Thor (Sap xep chu cai)
```

Chương 6: Iterator - Ngón tay chỉ đường

- `s.begin()`: Ngón tay chỉ vào **người đầu tiên**.
- `s.end()`: Ngón tay chỉ vào **vị trí sau người cuối cùng**.
- `++it`: Dịch ngón tay sang người kế tiếp.
- `*it`: Lấy giá trị tại nơi ngón tay đang chỉ.

Chương 6: Iterator - Ngón tay chỉ đường

- `s.begin()`: Ngón tay chỉ vào **người đầu tiên**.
- `s.end()`: Ngón tay chỉ vào **vị trí sau người cuối cùng**.
- `++it`: Dịch ngón tay sang người kế tiếp.
- `*it`: Lấy giá trị tại nơi ngón tay đang chỉ.

Nguy hiểm: Vòng lặp vô tận

Nếu dùng `while` mà quên `++it`, ngón tay sẽ đứng yên một chỗ mãi mãi!

Nhiệm vụ: Chiếc máy lọc thông minh

```
1 #include <iostream>
2 #include <vector>
3 #include <set>
4 using namespace std;
5
6 int main() {
7     // 1. Danh sách lỗi trùng lặp
8     vector<int> diem_loi = {8, 5, 9, 8, 7, 5, 10, 7};
9
10    // 2. Đưa vào Set để lọc
11    set<int> diem_sach(diem_loi.begin(), diem_loi.end());
12
13    // 3. In kết quả: 5 7 8 9 10
14    for (int x : diem_sach) cout << x << " ";
15
16    return 0;
17 }
```

Tổng kết và Bài tập tốt nghiệp

Tóm tắt

1. Duy nhất và Tự sắp xếp.
2. `.insert()`, `.erase()`.
3. `.contains()` (C++20).
4. Duyệt bằng For-each hoặc Iterator.

Câu hỏi tốt nghiệp

Nếu bạn có danh sách điểm {8, 5, 9, 8, 7, 5}, kết quả sau khi qua Set là gì?

Tổng kết và Bài tập tốt nghiệp

Tóm tắt

1. Duy nhất và Tự sắp xếp.
2. `.insert()`, `.erase()`.
3. `.contains()` (C++20).
4. Duyệt bằng For-each hoặc Iterator.

Câu hỏi tốt nghiệp

Nếu bạn có danh sách điểm {8, 5, 9, 8, 7, 5}, kết quả sau khi qua Set là gì?

Đáp án: {5, 7, 8, 9}.