Chinh Phục C++ For Loop

Xây dựng Cỗ máy Tự động hóa

Slide Learning CPP

Ngày 14 tháng 1 năm 2026

Lộ trình bài học

Chào bạn! Hôm nay chúng ta sẽ biến khái niệm For Loop thành một trò chơi lắp ráp tư duy thú vị: **"Cỗ máy tự động hóa"**.

3 Bước chinh phục

1. **Bức tranh toàn cảnh:** Tại sao lại cần For Loop? (Phép ẩn dụ về Robot chép phạt).
2. **Giải phẫu cỗ máy:** 3 công tắc bí mật vận hành vòng lặp.
3. **Thực chiến:** Code ví dụ và các bài tập kiểm tra tư duy.

Phần 1: Bức tranh toàn cảnh (The Big Picture)

Vấn đề: Chép phạt thủ công

Hãy tưởng tượng bạn phải chép 100 dòng chữ: "Em hứa sẽ làm bài tập đầy đủ".

- **Cách thủ công:** Viết từng dòng một $\rightarrow$ Mỏi tay, chán ngắt. Trong code, tương đương việc copy-paste lệnh `cout` 100 lần.

Giải pháp: Robot For Loop

Bạn chế tạo một con Robot và ra lệnh: *"Hãy bắt đầu từ dòng 0. Hễ số dòng còn nhỏ hơn 100 thì viết tiếp. Mỗi lần xong một dòng thì tự đếm thêm 1 số."*

$\rightarrow$ Vòng lặp `for` giúp thực hiện công việc lặp lại hàng nghìn lần chỉ với vài dòng code.

Phần 2: Giải phẫu cỗ máy (Syntax)

Cú pháp chuẩn của vòng lặp `for` như một bảng điều khiển với 3 công tắc:

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Chúng ta sẽ đi sâu vào từng "công tắc"(Statements).

Chi tiết 3 công tắc điều khiển

- **Statement 1: Vạch Xuất Phát (Khởi tạo)**
  - Nhiệm vụ: Thiết lập biến đếm ban đầu.
  - Ví dụ: `int i = 0;` (Bắt đầu từ 0).
- **Statement 2: Trọng Tài (Điều kiện)**
  - Nhiệm vụ: Kiểm tra xem có được chạy tiếp không.
  - Ví dụ: `i < 5;` (Nếu đúng thì chạy tiếp, sai thì dừng).
- **Statement 3: Bước Nhảy (Tăng/Giảm)**
  - Nhiệm vụ: Thay đổi biến đếm sau mỗi lần lặp để tránh vòng lặp vô tận.
  - Ví dụ: `i++` (Tăng i lên 1 đơn vị).

Phần 3: Thực chiến (Code Example)

Nhiệm vụ: In ra các số từ 0 đến 4.

```cpp
#include <iostream>
using namespace std;

int main() {
  // Bat dau: i = 0; Dieu kien: i < 5; Buoc nhay: i++
  for (int i = 0; i < 5; i++) {
    cout << "Gia tri cua i la: " << i << "\n";
  }
  return 0;
}
```

Kết quả màn hình

```
Gia tri cua i la: 0
...
Gia tri cua i la: 4
```

Phân tích dòng chảy (Flow)

Quá trình máy tính thực hiện:

1. **Khởi tạo:** Máy tạo ra $i = 0$.
2. **Kiểm tra:** 0 có nhỏ hơn 5 không? $\rightarrow$ **CÓ**.
3. **Thực thi:** In ra "Gia tri cua i la: 0".
4. **Bước nhảy:** Tăng i lên 1 (lúc này $i = 1$).
5. **Lặp lại:** Quay lại bước kiểm tra...
6. ... Lặp liên tục đến khi $i = 5$.
7. **Dừng:** 5 có nhỏ hơn 5 không? $\rightarrow$ **KHÔNG**. Thoát vòng lặp.

Kiểm tra sự hiểu biết

Câu hỏi

Trong đoạn code trước, nếu thay đổi **Statement 3** từ `i++` thành `i = i + 2`, chuyện gì sẽ xảy ra?

A. Nó vẫn in từ 0 đến 4 bình thường.

B. Nó sẽ in ra các số: 0, 2, 4.

C. Nó sẽ chạy mãi mãi không dừng.

Kiểm tra sự hiểu biết

Câu hỏi

Trong đoạn code trước, nếu thay đổi **Statement 3** từ `i++` thành `i = i + 2`, chuyện gì sẽ xảy ra?

A. Nó vẫn in từ 0 đến 4 bình thường.

B. Nó sẽ in ra các số: 0, 2, 4.

C. Nó sẽ chạy mãi mãi không dừng.

Đáp án: B

Chính xác! Robot bây giờ sẽ "nhảy cóc"2 bước (bỏ qua số lẻ).

- Vạch xuất phát: $i = 0$
- Đích đến: $i < 5$
- Di chuyển: Nhảy $+2$ ($0 \to 2 \to 4$).

Thử thách: Đảo ngược thời gian

Nhiệm vụ: Lập trình tên lửa đếm ngược từ 10 về 0.
Hãy điền vào chỗ trống:

```
for (int i = ???; i >= 0; ???) {
    cout << i << "\n";
}
```

Thử thách: Đảo ngược thời gian

Nhiệm vụ: Lập trình tên lửa đếm ngược từ 10 về 0.
Hãy điền vào chỗ trống:

```
for (int i = ???; i >= 0; ???) {
    cout << i << "\n";
}
```

Lời giải

```
for (int i = 10; i >= 0; i--) {
    cout << i << "\n";
}
```

- Xuất phát: 10.
- Điều kiện: Lớn hơn hoặc bằng 0 ($>= 0$).
- Bước nhảy: Giảm đi 1 (`i-`).

Vòng lặp lồng nhau (Nested Loops)

Phép ẩn dụ: Chiếc Đồng Hồ

- **Vòng lặp ngoài (Outer Loop):** Kim Giờ.
- **Vòng lặp trong (Inner Loop):** Kim Phút.

Quy luật bất biến

"Chỉ khi nào kim phút chạy đủ một vòng (hết nhiệm vụ của vòng trong), thì kim giờ mới được phép nhích lên 1 nấc (bước tiếp theo của vòng ngoài)."

Code mô phỏng đồng hồ

```cpp
// VONG LAP NGOAI (Kim Gio): i tu 1 den 2
for (int i = 1; i <= 2; ++i) {
    cout << "Gio thu: " << i << "\n";

    // VONG LAP TRONG (Kim Phut): j tu 1 den 3
    // Chay het tu 1-3 moi lan 'i' xuat hien
    for (int j = 1; j <= 3; ++j) {
        cout << "    Phut thu: " << j << "\n";
    }
}
```

Kết quả

Gio thu: 1 $\rightarrow$ Phut thu: 1, 2, 3

Gio thu: 2 $\rightarrow$ Phut thu: 1, 2, 3

Thử thách tư duy (Mental Gym)

Câu hỏi

Nếu vòng ngoài chạy từ 1 đến **3**, vòng trong chạy từ 1 đến **5**. Dòng chữ "Phut thu: ..."sẽ được in ra tổng cộng bao nhiêu lần?

A. 8 lần (lấy $3 + 5$)

B. 15 lần (lấy 3×5)

C. 5 lần (chỉ tính vòng trong)

Thử thách tư duy (Mental Gym)

Câu hỏi

Nếu vòng ngoài chạy từ 1 đến **3**, vòng trong chạy từ 1 đến **5**. Dòng chữ "Phut thu: ..."sẽ được in ra tổng cộng bao nhiêu lần?

A. 8 lần (lấy $3 + 5$)

B. 15 lần (lấy 3×5)

C. 5 lần (chỉ tính vòng trong)

Đáp án: B (15 lần)

Đây chính là phép nhân trong lập trình:

$$3 \text{ (vòng ngoài)} \times 5 \text{ (vòng trong)} = 15 \text{ lần}$$

Mảnh ghép cuối: Foreach Loop

Ẩn dụ: Băng chuyền hành lý Thay vì chạy đi tìm đồ vật (dùng index i), bạn đứng yên và băng chuyền (Array) đưa từng món đồ đến tay bạn.

Cú pháp

```
for (type variableName : arrayName) {
  // code block
}
```

- Không cần biến đếm i.
- Không cần điều kiện dừng phức tạp.
- Đọc là: "Với mỗi `variableName` nằm trong `arrayName`".

Ví dụ Foreach

```cpp
int main() {
  // Bang chuyen chua 4 con so
  int soMayMan[] = {10, 20, 30, 40};

  // Voi moi "so" nam trong "soMayMan"
  for (int so : soMayMan) {
    cout << so << "\n";
  }
  return 0;
}
```

Code gọn gàng, tự động chạy từ đầu đến cuối mảng.

Bài tập tốt nghiệp For Loop

Cho mảng: `int diemSo[] = {5, 8, 10};`
Đoạn code sau thực hiện:

```
1 for (int d : diemSo) {
2   cout << d + 1 << "\n";
3 }
```

Câu hỏi: Biến d đại diện cho cái gì?

A. **Vị trí** của điểm số (0, 1, 2).
B. **Giá trị** thực sự của điểm số (5, 8, 10).
C. Tổng số lượng điểm số (3).

Bài tập tốt nghiệp For Loop

Cho mảng: `int diemSo[] = {5, 8, 10};`
Đoạn code sau thực hiện:

```
1 for (int d : diemSo) {
2   cout << d + 1 << "\n";
3 }
```

Câu hỏi: Biến d đại diện cho cái gì?

A. **Vị trí** của điểm số (0, 1, 2).
B. **Giá trị** thực sự của điểm số (5, 8, 10).
C. Tổng số lượng điểm số (3).

Đáp án: B (Giá trị)

Foreach lấy trực tiếp giá trị (Value). Kết quả in ra sẽ là: 6, 9, 11.

Tổng kết hành trình

Chúng ta đã đi qua 3 trạm dừng chân:

1. **For Loop cơ bản:** Robot chép phạt (Lặp theo số lần).
2. **Nested Loop:** Chiếc đồng hồ (Vòng trong xong, vòng ngoài mới bước).
3. **Foreach Loop:** Băng chuyền hành lý (Duyệt qua danh sách, không cần đếm).

Bước tiếp theo: Quyền năng kiểm soát

Làm sao để phanh gấp hoặc bỏ qua chướng ngại vật? Hẹn gặp lại ở bài học về **Break** và **Continue**!