

# Cấu trúc lặp trong C++

Mục đích của các vòng lặp là lặp lại một thao tác với một số lần nhất định hoặc trong khi một điều kiện nào đó còn thoả mãn.

## 1. Vòng lặp while

Dạng của nó như sau:

```
while (expression) statement
```

và chức năng của nó đơn giản chỉ là lặp lại statement khi điều kiện expression còn thoả mãn.

Ví dụ, chúng ta sẽ viết một chương trình đếm ngược sử dụng vào lặp while:

```
// custom countdown using while
#include <iostream.h>
int main ()
{
    int n;
    cout << "Enter the starting number > ";
    cin >> n;
    while (n>0) {
        cout << n << ", ";
        --n;
    }
    cout << "FIRE! ";
    return 0;
}
```

Kết quả:

```
Enter the starting number > 8
8, 7, 6, 5, 4, 3, 2, 1, FIRE!
```

Khi chương trình chạy người sử dụng được yêu cầu nhập vào một số để đếm ngược. Sau đó, khi vòng lặp `while` bắt đầu nếu số mà người dùng nhập vào thoả mãn điều kiện `n>0` khối lệnh sẽ được thực hiện một số lần không xác định chừng nào điều kiện `(n>0)` còn được thoả mãn.

Chúng ta cần phải nhớ rằng vòng lặp phải kết thúc ở một điểm nào đó, vì vậy bên trong vòng lặp chúng ta phải cung cấp một phương thức nào đó để buộc `condition` trở thành sai nếu không thì nó sẽ lặp lại mãi mãi. Trong ví dụ trên vòng lặp phải có lệnh `--n`; để làm cho `condition` trở thành sai sau một số lần lặp.

## 2. Vòng lặp do-while

Dạng thức:

```
do
    statement
while (condition);
```

Chức năng của nó là hoàn toàn giống vòng lặp `while` chỉ trừ có một điều là điều kiện điều khiển vòng lặp được tính toán sau khi `statement` được thực hiện, vì vậy `statement` sẽ được thực hiện ít nhất một lần ngay cả khi `condition` không bao giờ được thoả mãn.

Ví dụ, chương trình dưới đây sẽ viết ra bất kì số nào mà bạn nhập vào cho đến khi bạn nhập số 0.

```
// number echoer
#include <iostream.h>
int main ()
{
    unsigned long n;
    do {
        cout << "Enter number (0 to end): ";
        cin >> n;
        cout << "You entered: " << n << "\n";
    } while (n != 0);
    return 0;
}
```

Kết quả:

```
Enter number (0 to end): 12345
You entered: 12345
Enter number (0 to end): 160277
You entered: 160277
Enter number (0 to end): 0
You entered: 0
```

## 3. Vòng lặp for

Dạng thức:

```
for (initialization; condition; increase)
    statement;
```

và chức năng chính của nó là lặp lại statement chừng nào condition còn mang giá trị đúng, như trong vòng lặp while. Nhưng thêm vào đó, for cung cấp chỗ dành cho lệnh khởi tạo và lệnh tăng. Vì vậy vòng lặp này được thiết kế đặc biệt lặp lại một hành động với một số lần xác định.

Cách thức hoạt động của nó như sau:

1. `initialization` được thực hiện. Nói chung nó đặt một giá trị ban đầu cho biến điều khiển. Lệnh này được thực hiện chỉ một lần.
2. `condition` được kiểm tra, nếu nó là đúng vòng lặp tiếp tục còn nếu không vòng lặp kết thúc và statement được bỏ qua.
3. `statement` được thực hiện. Nó có thể là một lệnh đơn hoặc là một khối lệnh được bao trong một cặp ngoặc nhọn.
4. Cuối cùng, `increase` được thực hiện để tăng biến điều khiển và vòng lặp quay trở lại bước 2.

Sau đây là một ví dụ đếm ngược sử dụng vòng for.

```
// countdown using a for loop
#include <iostream.h>
int main ()
{
    for (int n=10; n>0; n--) {
        cout << n << ", ";
    }
    cout << "FIRE! ";
    return 0;
}
```

Kết quả:

```
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, FIRE!
```

Phần khởi tạo và lệnh tăng không bắt buộc phải có. Chúng có thể được bỏ qua nhưng vẫn phải có dấu chấm phẩy ngăn cách giữa các phần. Vì vậy, chúng ta có thể viết for (;n<10;) hoặc for (;n<10;n++).

Bằng cách sử dụng dấu phẩy, chúng ta có thể dùng nhiều lệnh trong bất kỳ trường nào trong vòng for, như là trong phần khởi tạo. Ví dụ chúng ta có thể khởi tạo một lúc nhiều biến trong vòng lặp:

```
for ( n=0, i=100 ; n!=i ; n++, i-- )  
{  
    // command...  
}
```

Vòng lặp này sẽ thực hiện 50 lần nếu như n và i không bị thay đổi trong thân vòng lặp:

image not found or type unknown



---

Revision #1

Created 5 October 2019 04:07:46 by Laptrinh.vn

Updated 5 October 2019 04:16:44 by Laptrinh.vn