

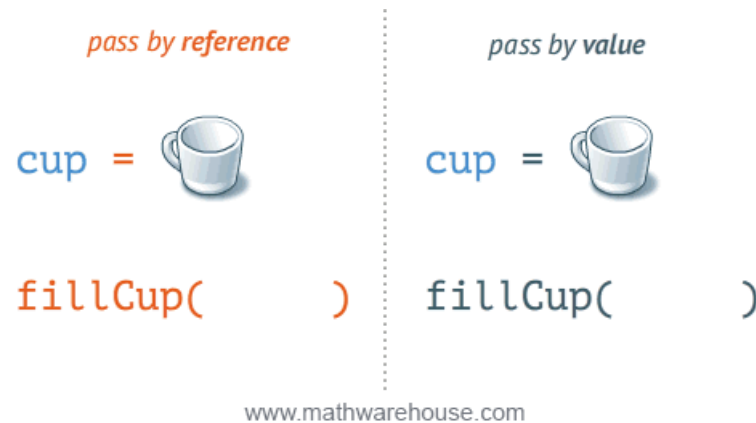
Bài 33. Tham chiếu và tham trị trong C++

Bởi Nguyễn Văn Hiếu -

Bài số 31 trong 69 bài của khóa học [Học C Không Khó](#)

Đã học về [hàm](#) thì chúng ta cần phải nắm rõ khái niệm **tham chiếu** và **tham trị**. Đây là 2 cách truyền tham số vào hàm khác nhau. Do đó, tùy mục đích mà chúng ta sẽ truyền tham số bằng tham trị hoặc tham chiếu. Trong bài học này, Lập trình không khó sẽ cùng bạn đi làm sáng tỏ 2 khái niệm này qua các bài tập thực hành nhé.

Lưu ý: Tham chiếu trong bài này là kiến thức của C++. Còn với ngôn ngữ C, nếu muốn truyền tham chiếu thì bạn cần đọc [tham chiếu trong C](#) sau khi học [con trỏ](#).



Hình ảnh mô phỏng tham chiếu (pass by reference) và tham trị (pass by value) trong C++. Trong đó `cup` là một biến được truyền vào lời gọi hàm `fillCup()`. Khi truyền theo tham chiếu thì hàm có khả năng thay đổi giá trị của tham số của hàm; Còn truyền theo tham trị thì hàm không thể thay đổi giá trị của tham số. Nguồn ảnh: [mathwarehouse](#)

10



NỘI DUNG BÀI VIẾT



1. Video hướng dẫn tham chiếu và tham trị
2. Làm rõ tham trị và tham chiếu
 - 2.1. Tham chiếu là gì?
 - 2.2. Tham trị là gì?
3. Ví dụ về tham chiếu và tham trị trong C++
4. Kết luận về tham chiếu và tham trị trong C++

Video hướng dẫn tham chiếu và tham trị

Source code trong video hướng dẫn về tham trị và tham chiếu:

```
0
1  /*
2     truyền tham trị(giá trị): pass by value
3     truyền tham chiếu: pass by reference
4  */
5
6  /*
7     Viết hàm hoán vị 2 số
8  */
9
10 #include <stdio.h>
11
12 void Swap(int &a, int &b){
13     int tmp = a;
14     a = b;
15     b = tmp;
16 }
17
18 void increment(int &n){
19     n++;
20 }
21
22 int main(){
23     int first, second;
24
25     printf("\nNhập first = "); scanf("%d", &first);
26     printf("\nNhập second = "); scanf("%d", &second);
27
28     printf("\nfirst = %d, second = %d", first, second);
29     increment(first);
```

10



```

30     increment(second);
31     Swap(first, second);
32     printf("\nfirst = %d, second = %d", first, second);
33
34
35 }
36

```

Kết quả chạy chương trình:

```

0
1 PS G:\c_courses\day_33> .\ValueVsReff.exe
2
3 Nhap first = 4
4
5 Nhap second = 5
6
7 first = 4, second = 5
8 first = 5, second = 4
9

```

Làm rõ tham trị và tham chiếu

Bạn đọc cần đọc kết hợp cả tham chiếu và tham trị để dễ dàng phân biệt 2 khái niệm này, cũng như cách dùng và khi nào cần dùng.

Tham chiếu là gì?

- Khai báo hàm sử dụng tham số dưới dạng tham chiếu bằng cách thêm dấu `&` vào trước tham số đó, Ví dụ: `int f(int &x){}`.
- Khi một biến `a` được truyền vào lời gọi hàm `f(int &x)` làm tham số dưới dạng tham chiếu, thì biến `x` của hàm `f(int &x)` và biến `a` thực chất là một. Do đó, nếu hàm mà thay đổi giá trị của `x` trong hàm `f(int &x)` này thì đồng nghĩa tại nơi gọi hàm biến `a` cũng bị thay đổi theo.
- Nếu tham số là mảng hoặc chuỗi, thì tham số này được truyền theo tham chiếu.

Tham trị là gì?

10



- Chúng ta vẫn thường dùng hàm trong hầu hết các bài học trước và truyền tham số theo cách truyền theo tham trị. Ví dụ: `int f(int x){}`.
- Khi một biến `a` được truyền vào lời gọi hàm `f(int x)` làm tham số dưới dạng tham trị, thì biến `x` của hàm `f(int x)` và biến `a` là hai biến độc lập. Bởi vì khi tham số của hàm `f(int x)` là tham trị, hàm này sẽ tạo ra một biến mới và sao chép giá trị của `a` vào. Do đó, nếu hàm mà thay đổi giá trị của `x` trong hàm `f(int x)` này thì không tác động gì tới giá trị của biến `a`.

Ví dụ về tham chiếu và tham trị trong C++

Ở ví dụ này, chúng ta sẽ cùng nhau đi viết 1 hàm kiểu `void` nhận vào 1 biến số nguyên và thân hàm có nhiệm vụ tăng giá trị tham số đó lên 1 đơn vị.

Khi hàm của chúng ta nhận tham số dưới dạng tham trị:

```
0
1 #include <stdio.h>
2
3 void PassByValue(int n){
4     n++; // Tăng n lên 1 đơn vị
5     printf("\nĐịa chỉ của n: %d", &n); // Lấy địa chỉ của biến
6 }
7
8 int main(){
9     int value;
10    printf("\nNhập value = "); scanf("%d", &value);
11    printf("\nĐịa chỉ của n: %d", &value); // Lấy địa chỉ của biến
12    printf("\nTrước khi gọi hàm value = %d", value);
13    PassByValue(value);
14    printf("\nSau khi gọi hàm value = %d", value);
15 }
16
```

Kết quả chạy:

```
0
1 PS G:\c_courses\day_33> g++ .\ValueVsRef.cpp -o .\ValueVsRef
2 PS G:\c_courses\day_33> .\ValueVsRef.exe
3
4 Nhập value = 5
```

10



```

5
6 Địa chỉ của n: 6487580
7 Trước khi gọi hàm value = 5
8 Địa chỉ của n: 6487536
9 Sau khi gọi hàm value = 5
10

```

Như bạn thấy, khi hàm được truyền theo tham trị thì:

- Giá trị của biến `value` trong hàm `main` không bị thay đổi.
- Địa chỉ của `value` trong hàm `main` và địa chỉ của `n` trong hàm `PassByValue` là khác nhau.

Khi hàm của chúng ta nhận tham số dưới dạng tham chiếu:

```

0
1 #include <stdio.h>
2
3 void PassByReference(int &n){ // Truyền theo tham chiếu
4     n++; // Tăng n lên 1 đơn vị
5     printf("\nĐịa chỉ của n: %d", &n); // Lấy địa chỉ của biến
6 }
7
8 int main(){
9     int value;
10    printf("\nNhập value = "); scanf("%d", &value);
11    printf("\nĐịa chỉ của n: %d", &value); // Lấy địa chỉ của biến
12    printf("\nTrước khi gọi hàm value = %d", value);
13    PassByReference(value);
14    printf("\nSau khi gọi hàm value = %d", value);
15 }
16

```

Kết quả chạy chương trình:

```

0
1 PS G:\c_courses\day_33> g++ .\ValueVsRef.cpp -o .\ValueVsRef
2 PS G:\c_courses\day_33> .\ValueVsRef.exe
3
4 Nhập value = 5
5
6 Địa chỉ của n: 6487580
7 Trước khi gọi hàm value = 5

```



```
8 Địa chỉ của n: 6487580
9 Sau khi gọi hàm value = 6
10
```

Như bạn thấy, khi hàm nhận tham số là tham chiếu thì:

- Giá trị của biến `value` trong hàm `main` bị thay đổi đúng theo cách biến `n` bị thay đổi trong hàm `PassByReference`.
- Địa chỉ của `value` trong hàm `main` và địa chỉ của `n` trong hàm `PassByReference` là giống nhau => chúng cùng là 1 biến.

Kết luận về tham chiếu và tham trị trong C++

Qua bài học và các ví dụ mình đã trình bày ở trên, mình có 1 số tóm tắt về tham chiếu và tham trị như sau:

- Dùng tham trị khi bạn không muốn giá trị của biến bị thay đổi.
- Dùng tham chiếu khi bạn muốn hàm thay đổi giá trị biến của bạn.
- Dùng tham chiếu sẽ tiết kiệm bộ nhớ hơn, vì nó không phải tạo ra 1 bản sao.
- Nếu bạn dùng tham chiếu, bạn cũng không muốn hàm thay đổi giá trị của bạn thì bạn cần thêm từ khóa `const` vào tham số đó. Ví dụ: `int f(const int &x)`.
- Nếu tham số là mảng => truyền theo tham chiếu. Chúng ta sẽ tìm hiểu nó trong chương sau.

Các bài viết trong khóa học

Bài trước: Bài 32. Biến cục bộ và biến toàn cục

Bài sau: Bài 34. Lệnh return và hàm exit trong C

10



Nguyễn Văn Hiếu

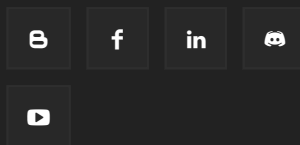


Sáng lập cộng đồng Lập Trình Không Khó với mong muốn giúp đỡ các bạn trẻ trên con đường trở thành những lập trình viên tương lai. Tất cả những gì tôi viết ra đây chỉ đơn giản là sở thích ghi lại các kiến thức mà tôi tích lũy được.



Blog chia sẻ kiến thức lập trình, xây dựng cộng đồng học lập trình. Cho đi kiến thức mà mình có là cách học tập hiệu quả nhất

👍 Báo lỗi / Liên hệ / Hợp tác / Quảng cáo



BÀI VIẾT HAY

Bài 1. Giới thiệu khóa học “Học C Bá Đạo”

21/07/2019

1000 bài tập lập trình C/C++ có lời giải của thầy Khang

25/12/2019

Kiểm tra số nguyên tố sử dụng C/C++ và Java

15/07/2018

CHUYÊN MỤC HAY

Học C/C++	194
Học Python	48
Học Java	45
Học Javascript	37
Khóa học	33
Học Web	26
Học C#	24
Chia sẻ	23

- BẠN BÈ & ĐỐI TÁC -

Luyện Code - Tự Học Đồ Họa - Cách Học Lập Trình - VNTALKING

