

Bài: Từ khóa ref và out trong C#

Xem bài học trên website để ủng hộ Kteam: [Từ khóa ref và out trong C#](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở các bài học trước, chúng ta đã cùng nhau tìm hiểu về [BIẾN TOÀN CỤC VÀ BIẾN CỤC BỘ TRONG C#](#). Chúng ta đã biết cách sử dụng hàm cùng parameter. Có một vấn đề phát sinh như sau: mong muốn giá trị của parameter truyền vào, sau khi thực thi các câu lệnh bên trong hàm giá trị đó có thể thay đổi.

Chúng ta sẽ cùng đi đến cách sử dụng **từ khóa ref và out trong C#** để giải quyết vấn đề này nhé.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- [BIẾN](#) và [KIỂU DỮ LIỆU](#) trong C#
- [TOÁN TỬ TRONG C#](#)
- [CÂU ĐIỀU KIỆN TRONG C#](#)
- [CẤU TRÚC CƠ BẢN VÒNG LẶP](#)

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Từ khóa **ref** và **out** trong C#

Từ khóa ref và out trong C#

Từ khóa ref

Ta xét ví dụ sau:

C#:

```
static void Main(string[] args)
{
    int value = 5;

    Console.WriteLine("Value before increase: {0}", value);

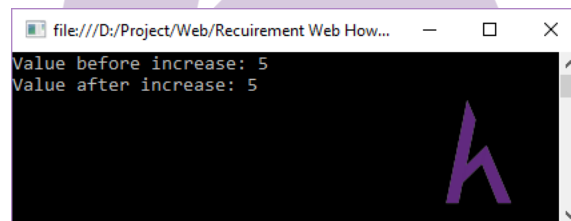
    IncreaseValue(value);

    Console.WriteLine("Value after increase: {0}", value);

    Console.ReadKey();
}

static void IncreaseValue(int value)
{
    value++;
}
```

Chúng ta sẽ thấy kết quả màn hình in ra hai giá trị 5. Vì sau khi kết thúc hàm **IncreaseValue** giá trị của **value** vẫn không thay đổi.



Với mong muốn giá trị của **value** sẽ thay đổi sau khi kết thúc lời gọi hàm **IncreaseValue** thì chúng ta sẽ thêm từ khóa **ref** phía trước kiểu dữ liệu của parameter mong muốn thay đổi giá trị khi khai báo hàm. Đồng thời phải thêm từ khóa **ref** ngay trước biến parameter truyền vào khi sử dụng hàm.

C#:

```

static void Main(string[] args)
{
    int value = 5;

    Console.WriteLine("Value before increase: {0}", value);

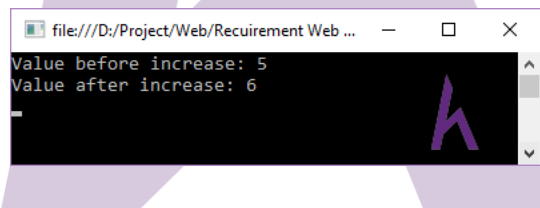
    IncreaseValue(ref value);

    Console.WriteLine("Value after increase: {0}", value);

    Console.ReadKey();
}

static void IncreaseValue(ref int value)
{
    value++;
}
    
```

Chúng ta có thể thấy kết quả in ra màn hình giá trị 5 và 6. Do giá trị của value đã thay đổi sau khi kết thúc lời gọi hàm **IncreaseValue**.



Lưu ý:

- Từ khóa **ref** phải có trước tên parameter của hàm và trước tên biến truyền vào khi gọi hàm sử dụng.
- Truyền parameter có từ khóa **ref** bắt buộc phải là một biến (không thể truyền vào một hằng vì hằng là giá trị không thay đổi).
- Có thể có một hoặc nhiều parameter với từ khóa **ref** trong lời khai báo hàm.
- Biến truyền vào có từ khóa **ref** thì phải được khởi tạo giá trị trước khi truyền vào.
- Hàm sử dụng sẽ thao tác trực tiếp với vùng nhớ của các parameter trên RAM. Cho nên kết thúc lời gọi hàm giá trị các parameter sẽ bị thay đổi.

Từ khóa out

Từ khóa **out** cũng tương tự từ khóa **ref**. Đó là:

- Vùng nhớ của các parameter sẽ được hàm sử dụng thao tác trực tiếp, dẫn đến khi kết thúc lời gọi hàm giá trị của các parameter có thể bị thay đổi.
- Phải có từ khóa **out** trước tên parameter của hàm và trước tên biến truyền vào khi gọi hàm sử dụng.

Nhưng có một sự khác biệt đó là:

- Biến truyền vào có từ khóa **out** sẽ **không** cần khởi tạo giá trị ban đầu.
- Parameter đó chỉ như một thùng chứa kết quả trả về khi kết thúc gọi hàm.
- Đồng thời parameter đó phải được khởi tạo ngay bên trong lời gọi hàm.

C#:

```
static void Main(string[] args)
{
    int value = 5;

    Console.WriteLine("Value before increase: {0}", value);

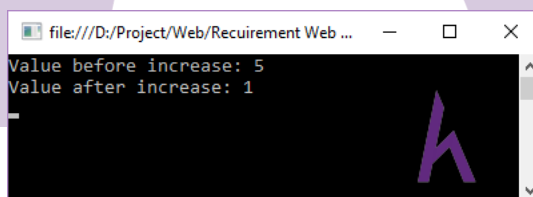
    IncreaseValue(out value);

    Console.WriteLine("Value after increase: {0}", value);

    Console.ReadKey();
}

static void IncreaseValue(out int value)
{
    value = 0;
    value++;
}
```

Chúng ta có thể thấy. Trong thân hàm **IncreaseValue** bắt buộc phải khởi tạo giá trị cho biến **value**. Kết quả màn hình xuất ra giá trị 5 ban đầu và 1 là kết quả cuối cùng của biến **value** sau khi kết thúc lời gọi hàm **IncreaseValue**.



Kết luận

Qua bài này chúng ta đã nắm được cách sử dụng từ khóa **ref** và **out** trong C#.

Bài sau chúng ta sẽ tìm hiểu về [MẢNG 1 CHIỀU TRONG C#](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".

