

# Bài: Cấu trúc rẽ nhánh If else trong C#

Xem bài học trên website để ủng hộ Kteam: [Cấu trúc rẽ nhánh If else 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 bài [TOÁN TỬ TRONG C#](#) chúng ta đã tìm hiểu về toán tử 3 ngôi (? :). Đó có thể xem là một cấu trúc rẽ nhánh đơn giản. Vậy "cấu trúc rẽ nhánh" là gì? Sử dụng chúng như thế nào? Bài học hôm nay sẽ giúp chúng ta trả lời những câu hỏi này – **Cấu trúc rẽ nhánh If else trong C#**.

## Nội dung

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

- [CẤU TRÚC LỆNH CỦA C# viết trên nền Console Application](#).
- [CẤU TRÚC NHẬP XUẤT CỦA C# trên nền Console Application](#).
- [BIẾN TRONG C#](#).
- [KIỂU DỮ LIỆU TRONG C#](#).
- [TOÁN TỬ TRONG C#](#).
- [ÉP KIỂU TRONG C#](#).

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

- Cấu trúc rẽ nhánh là gì? Có mấy loại cấu trúc rẽ nhánh?
- Cấu trúc If else dạng thiếu và dạng đủ.
- Ví dụ chương trình sử dụng cấu trúc If else.

## Cấu trúc rẽ nhánh là gì? Có mấy loại cấu trúc rẽ nhánh

Xét 2 mệnh đề sau:

- Mệnh đề 1: "Nếu trời mưa thì đường trơn".
- Mệnh đề 2: "Nếu bạn rảnh thì đi chơi ngược lại thì thôi"

Các bạn để ý những chữ in đậm trong 2 mệnh đề trên. Cấu trúc **Nếu... thì... hoặc Nếu... thì... ngược lại thì...** được gọi là **cấu trúc rẽ nhánh**.

Trong hầu hết các ngôn ngữ lập trình đều có loại cấu trúc rẽ nhánh:

- .. else ..
- .. case .. (sẽ nghiên cứu ở bài [CẤU TRÚC RẪN NHÁNH SWITCH – CASE TRONG C#](#))

Cấu trúc rẽ nhánh If... else... còn có tên gọi khác là "câu lệnh điều kiện".

## Cấu trúc If else dạng thiếu và dạng đủ

### Dạng thiếu

**Cú pháp:**

If ([**Biểu thức điều kiện**]) <**Câu lệnh thực hiện**>

- If là từ khóa bắt buộc.

- <Biểu thức điều kiện> là biểu thức dạng boolean (trả về **true** hoặc **false**).
- <Câu lệnh thực hiện> là câu lệnh muốn thực hiện nếu <Biểu thức điều kiện> là đúng.

**Ý nghĩa:**

Nếu <Biểu thức điều kiện> trả về **true** thì thực hiện <Câu lệnh thực hiện> ngược lại thì không làm gì cả.

**Ví dụ:****C#:**

```
string K = "Kteam";
if (K == "Kteam")
// Biểu thức điều kiện sử dụng toán tử == để so sánh xem giá trị biến K có bằng "Kteam" hay không. Nếu bằng trả thì trả về true
ngược lại thì trả về false.
    Console.WriteLine("Free Education"); // In ra màn hình chữ "Free Education" nếu biểu thức trên đúng.
```

Kết quả khi chạy chương trình trên là:

 Cấu trúc rẽ nhánh If else trong C#, C# cơ bản, tự học C#, Kteam, HowKteam

**Dạng đủ****Cú pháp:**

**If** <Biểu thức điều kiện>

<Câu lệnh thực hiện 1>

**else**

<Câu lệnh thực hiện 2>

- **If, else** là từ khóa bắt buộc.
- <Biểu thức điều kiện> là biểu thức dạng boolean (trả về **true** hoặc **false**).
- <Câu lệnh thực hiện 1> là câu lệnh muốn thực hiện nếu <Biểu thức điều kiện> là đúng.
- <Câu lệnh thực hiện 2> là câu lệnh muốn thực hiện nếu <Biểu thức điều kiện> là sai.

**Ý nghĩa:**

Nếu <Biểu thức điều kiện> trả về **true** thì thực hiện <Câu lệnh thực hiện 1> ngược lại thì thực hiện <Câu lệnh thực hiện 2>.

**Ví dụ:****C#:**

```
string K = "Kteam";

if (K == "Kteam") // Nếu giá trị K bằng "Kteam" thì
    Console.WriteLine("Free Education"); // In ra màn hình "Free Education"
else // Ngược lại thì
    Console.WriteLine("Connecting to HowKteam. . ."); // In ra màn hình "Connecting to HowKteam. . ."
```

Kết quả khi chạy chương trình trên là:

 Cấu trúc rẽ nhánh If else trong C#, C# cơ bản, tự học C#, Kteam, HowKteam

- Vì biểu thức điều kiện `K == "Kteam"` là đúng nên màn hình vẫn in ra `"Free Education"`.

## Một số lưu ý khi sử dụng câu lệnh điều kiện

<Biểu thức điều kiện> có thể chứa nhiều biểu thức con bên trong và các biểu thức con liên kết với nhau bằng các toán tử quan hệ (toán tử quan hệ đã được trình bày trong bài [TOÁN TỬ TRONG C#](#)) nhưng tất cả phải trả về kiểu boolean (`true` hoặc `false`).

Nếu muốn thực hiện nhiều câu lệnh thì ta có thể nhóm chúng vào trong cặp ngoặc nhọn `{ }`.

### Ví dụ:

**C#:**

```
If <Biểu thức điều kiện>
{
    <Câu lệnh thực hiện>
}
```

Hoặc

**C#:**

```
If <Biểu thức điều kiện>
{
    <Câu lệnh thực hiện 1>
}

Else
{
    <Câu lệnh thực hiện 2>
}
```

Lời khuyên là cho dù chỉ thực hiện 1 câu lệnh vẫn để trong cặp ngoặc nhọn `{ }` để code rõ ràng, dễ đọc, dễ nâng cấp sửa chữa.

Trong câu lệnh có thể chứa một câu lệnh điều kiện con nữa. Bạn có thể vận dụng điều này để làm giải quyết những vấn đề phức tạp.

### Ví dụ:

**C#:**

```

If <Biểu thức điều kiện 1>
{
    If <Biểu thức điều kiện 2>
    {
        <Câu lệnh thực hiện 1>
    }

    else
    {
        <Câu lệnh thực hiện 2>
    }
    <Câu lệnh thực hiện 3>
}

Else
{
    If <Biểu thức điều kiện 3>
    {
        <Câu lệnh thực hiện 4>
    }
    If <Biểu thức điều kiện 4>
    {
        <Câu lệnh thực hiện 5>
    }
    <Câu lệnh thực hiện 6>
}

```

Ngoài cấu trúc If...else... cơ bản trên còn có cấu trúc nhỏ khác là If...else if...else. **Ví dụ:**

**C#:**

```

If <Biểu thức điều kiện 1>
{
    <Câu lệnh thực hiện 1> //Thực hiện khi <biểu thức điều kiện 1> đúng
}

else if <Biểu thức điều kiện 2>
{
    <Câu lệnh thực hiện 2> // Thực hiện khi <biểu thức điều kiện 1> sai và <biểu thức điều kiện 2> đúng
}

else
{
    <Câu lệnh thực hiện 3> //Thực hiện khi cả 2 biểu thức điều kiện trên đều sai (các trường hợp còn lại)
}

```

- Có thể hiểu ngắn gọn là "Nếu... thì... Ngược lại, nếu... thì... Trường hợp khác...".
- Các biểu thức điều kiện được kiểm tra từ trên xuống dưới và không kiểm tra lại.
- Nếu biểu thức điều kiện đang kiểm tra trả về **true** thì
  - Thực hiện khối lệnh bên trong nó.
  - Thoát ra khỏi cấu trúc.
  - Không kiểm tra các biểu thức điều kiện còn lại.

So với toán tử 3 ngôi thì:

- Câu lệnh điều kiện nhìn trực quan hơn và có thể thực hiện nhiều câu lệnh hơn.
- Nhưng nếu chỉ thực hiện 1 câu lệnh điều kiện đơn giản thì dùng toán tử 3 ngôi sẽ làm cho code ngắn gọn và viết nhanh hơn.

## Ví dụ chương trình sử dụng cấu trúc If else

**Ví dụ:** Viết chương trình giải phương trình bậc 1:  $Ax + B = 0$ .

C#:

```

string strA, strB;
int A, B;
double Nghiem;

Console.WriteLine("*****");
Console.WriteLine(" *                                     *");
Console.WriteLine(" *   Chương trình giải phương trình Ax + B = 0 *");
Console.WriteLine(" *                                     *");
Console.WriteLine("*****");

Console.Write("Moi nhap so A: ");
strA = Console.ReadLine();
Console.Write("Moi nhap so B: ");
strB = Console.ReadLine();

if (int.TryParse(strA, out A) == false || int.TryParse(strB, out B) == false) // kiểm tra người dùng có thực sự nhập số
nguyên vào hay không. Nếu ép kiểu thành công sẽ trả về true, ngược lại trả về false
{
    Console.WriteLine("Du lieu nhap sai !");
    return; // Lệnh này tạm hiểu là dừng và thoát chương trình mà không thực hiện những câu lệnh sau nó nữa. Sẽ được
    tìm hiểu chi tiết trong bài 16 Hàm
}
else
{
    Console.WriteLine("\n Phuong trinh cua ban vua nhap la: {0}x + {1} = 0", A, B);

    if (A == 0)
    {
        Console.WriteLine("\n Phuong trinh co vo so nghiem !");
    }
    else if (B == 0)
    {
        Console.WriteLine("\n Phuong trinh co nghiem x = 0");
    }
    else
    {
        Nghiem = (double)-B / A; // Ép kiểu để cho ra kết quả chính xác
        Console.WriteLine("\n Phuong trinh co nghiem x = {0}", Nghiem);
    }
}
}

```

Trong ví dụ mình đã sử dụng cả 2 dạng câu lệnh điều kiện là:

- .. else ...
- .. else if ... else ...

Kết quả khi chạy chương trình trên là:

 Cấu trúc rẽ nhánh If else trong C#, C# cơ bản, tự học C#, Kteam, HowKteam

- Các bạn thử test khi nhập dữ liệu sai chương trình sẽ như thế nào nhé !

## Bài tập tự luyện

- Viết chương trình nhập vào một số tương ứng là năm. Xuất ra màn hình năm vừa nhập và tuổi của một người tương ứng với năm vừa nhập đó.
- Từ câu 1. Thêm chức năng:

- Nếu tuổi người đó < 16 thì hiện thông báo theo format: Bạn <Tuổi>, tuổi vị thành niên.
- Nếu tuổi người đó >= 16 và <18 thì hiện thông báo theo format: Bạn <Tuổi>, tuổi trưởng thành.

Nếu tuổi người đó  $\geq 18$  thì hiện thông báo theo format: Bạn <Tuổi>, già rồi.

3. Viết trò chơi Kéo Búa Bao với cách chơi: Người dùng sẽ nhập vào các số 1 hoặc 2 hoặc 3 tương ứng với kéo hoặc búa hoặc bao. Máy sẽ ngẫu nhiên sinh ra một số trong 3 số và tính toán máy hoặc người chiến thắng. Nhớ phải in kết quả ra màn hình.

Hãy khoe thành quả của bạn vào fanpage hoặc phần bình luận nhé!

---

## Kết luận

Nội dung bài này giúp các bạn nắm được:

- Khái niệm về cấu trúc rẽ nhánh. Các loại cấu trúc rẽ nhánh.
- Cấu trúc If else dạng thiếu và dạng đủ.
- Viết chương trình sử dụng cấu trúc If else.

Bài học sau chúng ta sẽ cùng tìm hiểu một khái niệm tiếp theo đó là [CẤU TRÚC RẼ NHANH SWITCH - CASE 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"**.

