# Bài 7: HẰNG TRONG C#

*Xem bài học trên website để ủng hộ Kteam:* Hằng 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 BIẾN TRONG C# chúng ta đã tìm hiểu về biến và có một khái niệm tương tự với biến – Đó là hằng. Cho nên bài học hôm nay chúng ta sẽ cùng tìm hiểu chi tiết về **HẰNG 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 , KIỂU DỮ LIỆU & TOÁN TỬ trong C#.

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

- Hằng là gì? Tại sao phải có hằng?
- Có mấy loại hằng? Ý nghĩa và cách sử dụng từng hằng.

---

## Hằng là gì? Tại sao phải có hằng?

Khái niệm về hằng:

- Là một biến những giá trị không thay đổi trong suốt chương trình.
- Bắt buộc phải khởi tạo giá trị khi khai báo.

Tại sao phải có hằng?


Copyright © Howkteam.com

- Để ngăn chặn việc gán giá trị khác vào biến.
- Hằng làm cho chương trình dễ đọc hơn bằng cách biến những con số vô cảm thành những tên có nghĩa.
- Hằng giúp cho chương trình dễ nâng cấp, dễ sửa chữa hơn.
- Hằng giúp cho việc tránh lỗi dễ dàng hơn. Nếu bạn vô ý gán giá trị cho một biến hằng ở đâu thì trình biên dịch sẽ báo lỗi ngay lập tức.

---

# Có mấy loại hằng? Ý nghĩa và cách sử dụng từng hằng

Trong C#, hằng được chia làm 3 loại:

- Giá trị hằng.
- Biểu tượng hằng.
- Kiểu liệt kê.

---

## Giá trị hằng

Ta có câu lệnh gán sau: x = **10**;

Giá trị **10** là **giá trị hằng**. Giá trị **10** luôn là **10** và ta không thể gán giá trị khác cho **10** được.

## Biểu tượng hằng

Việc gán một tên cho **giá trị hằng** được xem là một **biểu tượng hằng**.

Xét lại câu lệnh: **x** = 10; thì **x** được xem là **biểu tượng hằng**.

Cú pháp để tạo một biểu tượng hằng:

```
const <kiểu dữ liệu> <tên biến> = <giá trị hằng>;
```


Copyright © Howkteam.com

Có thể thấy cú pháp này khá giống với cú pháp khai báo nhưng có 2 điểm các bạn cần lưu ý:

- Phải có từ khóa const phía trước khai báo.
- Phải khởi tạo giá trị ngay tại khai báo.

# Kiểu liệt kê

Kiểu liệt kê là tập hợp các tên hằng có giá trị không thay đổi, sẽ được trình bày chi tiết trong bài ENUM TRONG C#.

---

# Cách sử dụng hằng

Vì bản chất hằng cũng là một biến nhưng giá trị không thay đổi nên hằng được sử dụng tương tự như biến (xem lại cách sử dụng biến ở bài BIẾN TRONG C#).

Một số lưu ý khi sử dụng hằng:

- Hằng bắt buộc phải được khởi tạo giá trị ngay khi khai báo và không được thay đổi giá trị trong suốt chương trình.
- Giá trị của hằng được tính toán vào lúc biên dịch nên không thể gán trực tiếp giá trị của biến vào hằng. Nếu muốn làm điều đó thì phải sử dụng từ khóa readonly trước khai báo biến (readonly là từ khóa chỉ cho trình biên dịch biết rằng biến này chỉ được đọc, lấy giá trị chứ không được gán giá trị)
- Hằng bao giờ cũng static nhưng ta không được đưa từ khóa này vào khai báo hằng (chi tiết về từ khóa static sẽ được trình bày trong bài TỪ KHÓA STATIC TRONG C#)

Những kiến thức về hằng cũng khá đơn giản và ứng dụng của hằng thì rất nhiều nên các bạn cần nắm vững để không bỡ ngỡ ở những bài sau.

---


Copyright © Howkteam.com

# Kết luận

Nội dung bài này giúp các bạn nắm được:

- Khái niệm về hằng và tại sao phải có hằng.
- Phân loại hằng và ý nghĩa và cách sử dụng từng hằng.

Bài học sau chúng ta sẽ cùng tìm hiểu một khái niệm tiếp theo đó là ÉP KIỂU 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**".


Copyright © Howkteam.com