Bài: Enum trong lập trình C#

Xem bài học trên website để ủng hộ Kteam: Enum trong lập trình 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

Trong bài HẰNG TRONG C# mình đã có giới thiệu về kiểu liệt kê trong C# là 1 dạng hằng. Và hôm nay chúng ta sẽ tìm hiểu xem **kiểu liệt kê** này là gì? Dùng nó như thế nào 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 CỦA VÒNG LẶP TRONG C#
- CẤU TRÚC CƠ BẢN CỦA HÀM TRONG C#
- MẢNG 1 CHIỀU TRONG C#

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

- Enum là gì? Đặc điểm của Enum.
- Khai báo và sử dụng Enum.

Enum là gì? Đặc điểm của Enum

Enum là từ khoá dùng để khai báo một kiểu liệt kê (Enumeration). Kiểu liệt kê là một tập hợp các hằng số do người dùng tự định nghĩa.

Nói cách khác, enum là cách mà C# hỗ trợ người dùng gom nhóm các hằng số lại với nhau và có chung một tên gọi (thường các hằng số này sẽ có liên quan với nhau ví dụ như các trạng thái của 1 sự vật, các tính chất của 1 sự vật, . . .)

Đặc điểm của enum:

- Là một kiểu dữ liệu tham trị (kiểu dữ liệu tham trị đã được trình bày trong bài KIỂU DỮ LIỆU)
- Enum không được phép kế thừa (khái niệm về kế thừa sẽ trình bày trong bài KẾ THỪA TRONG C#).

Khai báo và sử dụng enum

Khai báo

Cú pháp:

```
enum <tên enum>
{
        <danh sách các biểu tượng hằng>
}
```

Trong đó:

- <tên enum> là tên kiểu liệt kê do mình tự đặt và tuân thủ theo quy tắc đặt tên (đã trình bày trong bài BIẾN TRONG C#).
- <danh sách các biểu tượng hằng> là danh sách các biểu tượng hằng thành phần mỗi biểu tượng hằng cách nhau bằng dấu ",".

Ví dụ:

C#:

```
enum Color
{
RED,
BLUE,
YELLOW
}
```

- Với khai báo này ta đã có 1 kiểu liệt kê tên là Color.
- Về bản chất, các biểu tượng hằng RED, BLUE, YELLOW này đại diện cho các số nguyên lần lượt là 0, 1, 2.
- Như vậy, nếu như chúng ta sử dụng cách khai báo hằng bình thường thì ta có thể khai báo như sau:

C#:

```
public const int RED = 0;

public const int BLUE = 1;

public const int YELLOW = 2;
```

Lưu ý:

- Ta hoàn toàn có thể quy định giá trị cho từng biểu tượng hằng bằng cách trực tiếp khi khai báo. Ví dụ:

C#:

```
enum Color
{
RED = 2,
BLUE = 4,
YELLOW = 6,
}
```

Khi đó các biểu tượng hằng RED, BLUE, YELLOW sẽ đại diện cho các số nguyên lần lượt là 2, 4, 6

- Nếu ta không quy định giá trị cho các biểu tượng hằng thì giá trị của biểu tượng hằng đầu tiên sẽ mặc định là 0 và tăng dần cho các biểu tượng hằng tiếp theo.

Sử dụng

Ta có thể truy xuất đến từng biểu tượng hằng của enum thông qua toán tử "." Kèm theo tên biểu tượng hằng muốn truy xuất.

Ví dụ:

C#:

```
Color.RED;
```

Lưu ý:

- Mặc dù bản chất các biểu tượng hằng là đại diện cho các số nguyên nhưng bạn không thể so sánh trực tiếp chúng với các số nguyên được mà phải ép kiểu. Ví dụ:

:

```
enum Color
{
RED,
BLUE,
YELLOW
}



int Choose = int.Parse(Console.ReadLine());
if (Choose == Color.RED) // lỗi vì không thể so sánh trực tiếp 1 enum với 1 số nguyên
{

Console.WriteLine("Ban vua chon mau do");
}
```

Để chương trình không báo lỗi ta có thể ép kiểu biểu tượng hằng RED về kiểu int.

C#:

```
Choose == (int)Color.RED
```

- Chúng ta cũng có thể ép kiểu ngược lại từ số nguyên sang kiểu liệt kê.

Ví dụ:

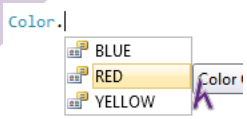
C#:

```
Color Background = (Color)2; // Background sẽ có giá trị là Color.YELLOW
```

- Khi khai báo 1 biến nào đó, các lập trình viên thường cố gắng xây dựng 1 tập các giá trị của biến đó (nếu có thể) và gom nhóm chúng bằng enum. Điều này rất thường gặp trong các bộ thư viện của C# và là sự khác biệt giữa C# và Java. Sự khác biệt này có ảnh hưởng gì đến việc lập trình? Câu hỏi này sẽ được trả lời ngày sau đây.

Sau khi xem qua cách khai báo và sử dụng enum ta có thể thấy rằng enum có những ưu điểm sau đây:

- Chính vì được sử dụng với mục đích gom nhóm các hằng có liên quan với nhau thành 1 tên duy nhất nên khi sử dụng bạn không cần phải nhớ chính xác tên hằng mà chỉ cần nhớ tên enum chứa nó là đủ việc còn lại đã có visual studio hỗ trợ.



  Bạn thấy đấy chỉ cần gõ tên enum và dấu "." Visual studio đã liệt kê sẵn danh sách các biểu tượng hằng bên trong nó. Điều này giúp cho việc lập trình dễ dàng hơn nhiều.

- Hơn thế nữa visual studio còn hỗ trợ giúp bạn tìm ra tên enum phù hợp với biến đang cần gán giá trị (các bài học sau sẽ minh hoạ rõ điều này).

Một chút ngoài lề:

- Nếu bạn nào đã sử dụng Java thì sẽ biết rằng Java dùng các khai báo hằng bình thường và đặt chúng trong class (Khái niệm class sẽ được trình bày trong series LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI C#). Khi đó việc gọi chúng ra rất phiền phức đặc biệt là những bạn mới làm quen với java. Bạn sẽ không chắc là hằng nào dùng cho biến nào. Thì qua C# điều này đã được giải quyết triệt để.
- Không phải tự nhiên mà .NET Framework lại mạnh đến như vậy. Sức mạnh lớn nhất của nó là hỗ trợ một người chưa rành về C# có thể tiếp cận và sử dụng chúng một cách dễ dàng.

Kết luận

Nội dung bài này giúp các bạn nắm được:

- Enum là gì? Đặc điểm của enum.
- Khai báo và sử dụng enum.

Như vậy chúng ta đã kết thúc serial **LẬP TRÌNH C# CƠ BẢN**. Các bạn hãy ôn lại những gì đã học để chuẩn bị bước sang series kế tiếp **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI C#** nào!

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**".