Bài 12: TỪ KHÓA DYNAMIC TRONG C#.

Xem bài học trên website để ủng hộ Kteam: Từ khóa Dynamic 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 KIỂU DỮ LIỆU OBJECT TRONG C#. Hôm nay chúng ta sẽ tiếp tục tìm hiểu kiểu dữ liệu tham chiếu tiếp theo đó là **kiểu dữ liệu dynamic**.

Nội dung

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

- Cấu trúc cơ bản của một chương trình C# console application
- BIẾN và KIỂU DỮ LIỆU trong C#
- TOÁN TỬ TRONG C#
- CÂU ĐIỀU KIỆN TRONG C#
- CẤU TRÚC CƠ BẢN CỦA VÒNG LẶP

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

- Từ khóa dynamic trong C#
- Phân biệt object, var và dynamic.

Từ khóa dynamic trong C#

Từ khóa dynamic là từ khóa dùng để khai báo kiểu dynamic. Kiểu dynamic là một khái niệm mới được đưa vào trong C# 4.0.

Cú pháp khai báo kiểu dynamic hoàn toàn tương tự như khai báo biến bình thường:

```
dynamic <tên biến>;
```

Trong đó:

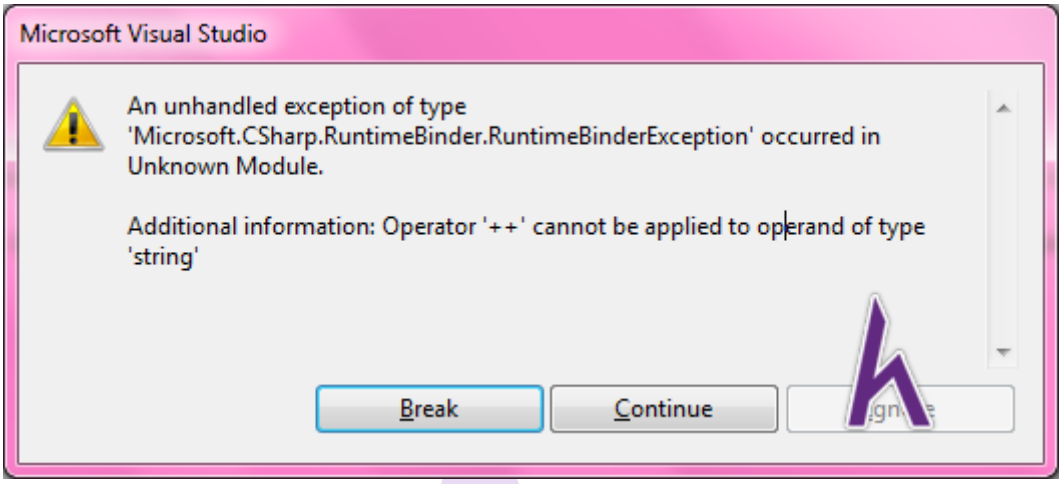
- dynamic là từ khóa.
- <tên biến> là tên do người dùng đặt.

Đặc điểm của kiểu dynamic:

- Các đối tượng thuộc kiểu dynamic sẽ không xác định được kiểu cho đến khi chương trình được thực thi. Tức là trình biên dịch sẽ bỏ qua tất cả lỗi về cú pháp, việc kiểm tra này sẽ thực hiện khi chương trình thực thi.

```
// Khai báo biến StringValue kiểu dynamic và khởi tạo giá trị là một chuỗi kiểu string
dynamic StringValue = "HowKteam";
/*
* Chúng ta biết rằng kiểu chuỗi không hỗ trợ toán tử ++
* Nhưng câu lệnh StringValue++ vẫn không báo lỗi là do ở thời điểm hiện tại trình biên dịch vẫn chưa xác định kiểu dữ liệu cho biến StringValue
* Khi chạy chương trình thì lúc này C# mới phát hiện biến StringValue là kiểu string và không thể thực hiện toán tử ++ lúc đó sẽ xuất hiện lỗi
*/
StringValue++;
```

Khi chạy chương trình trên ta sẽ nhận được lỗi sau:



- Hỗ trợ Dynamic programming (lập trình động) cho ngôn ngữ lập trình sử dụng kiểu dữ liệu tĩnh như C#.
- Giúp cải thiện khả năng tương thích với các ngôn ngữ và nền tảng (frameworks) động.
- Giúp việc viết code đơn giản và nhanh hơn.
- Có thể ép kiểu qua lại với các kiểu dữ liệu khác một cách bình thường (các cách ép kiểu đã trình bày trong bài ÉP KIỂU TRONG C#).

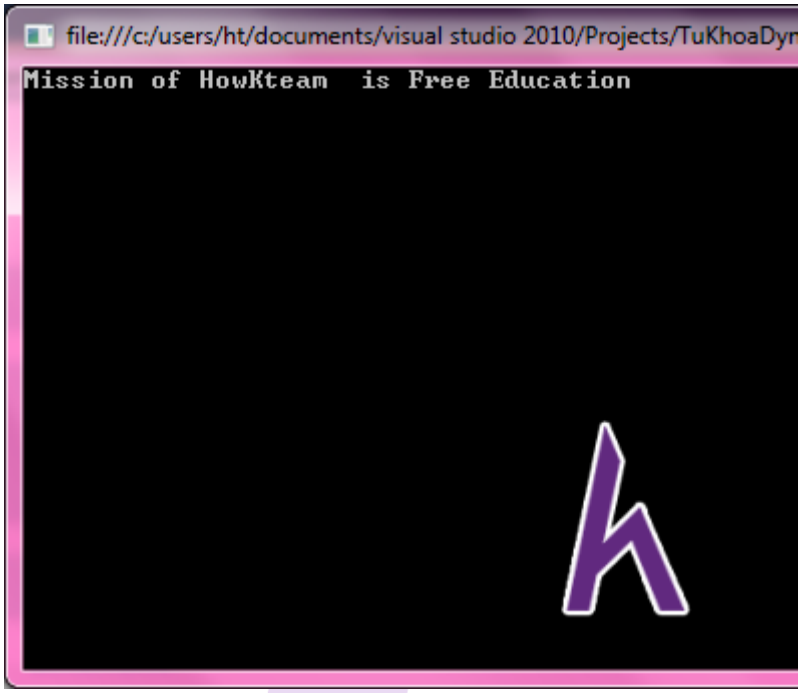
Ví dụ chương trình sử dụng dynamic:

```
// Khai báo 2 biến Name và Mission kiểu string và khởi tạo giá trị.
string Name = "HowKteam ";
string Mission = "Free Education";

/*
 * Thực hiện gán 1 biến kiểu string cho biến kiểu dynamic bằng cách ép kiểu ngầm định (implicit)
 * Sau phép gán này thì biến DynamicValue chứa giá trị là "Free Education" nhưng kiểu dữ liệu của nó vẫn chưa được xác định.
*/
dynamic DynamicName = Name;

// Thực hiện cộng chuỗi và in ra màn hình bình thường
Console.WriteLine("Mission of " + DynamicName + " is " + Mission);
```

Kết quả khi chạy chương trình trên là:



Phân biệt object, var và dynamic

Về khái niệm thì:

- Object là kiểu dữ liệu cơ bản của tất cả kiểu dữ liệu trong C#.
- Var là một từ khóa để khai báo một cách ngầm định kiểu dữ liệu và kiểu anonymous (kiểu anonymous sẽ được trình bày ở những bài sau).
- Dynamic là một từ khóa để khai báo kiểu dynamic. Kiểu dynamic cũng có thể tương tác với mọi kiểu dữ liệu nhưng khác object, biến kiểu dynamic chỉ được xác định kiểu dữ liệu khi chương trình thực thi.

Chúng ta cùng phân biệt object, var và dynamic qua bảng tổng hợp sau

Đặc điểm	Object	Var	Dynamic
Là một kiểu dữ liệu	Phải	Về bản chất thì var và dynamic đều là từ khóa không phải kiểu dữ liệu	
Phải khởi tạo giá trị khi khai	Không bắt buộc	Bắt buộc	Không bắt buộc
Sử dụng để làm kiểu trả về hoặc tham số cho hàm	Có	Không	Có
Có khả năng ép kiểu qua lại với các kiểu dữ liệu khác	Có	Không	Có
Thời điểm xác định kiểu dữ liệu thực sự	Là một kiểu dữ liệu nên không cần xác định gì nữa	Xác định ngay tại khai báo thông	Xác định khi chương trình thực

Kết luận

Qua bài này chúng ta đã nắm được cách sử dụng từ khóa dynamic trong C#.

Bài sau chúng ta sẽ tìm hiểu về CẤU TRÚC LẶP CƠ BẢN 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**".