LESSON 04

I. TỪ KHÓA DYNAMIC TRONG C#

❖ Định nghĩa

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:**

dynamic **<tên biến>;**

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

➢ **Đặc điểm:**

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

Là 1 kiểu dữ liệu tham chiếu.

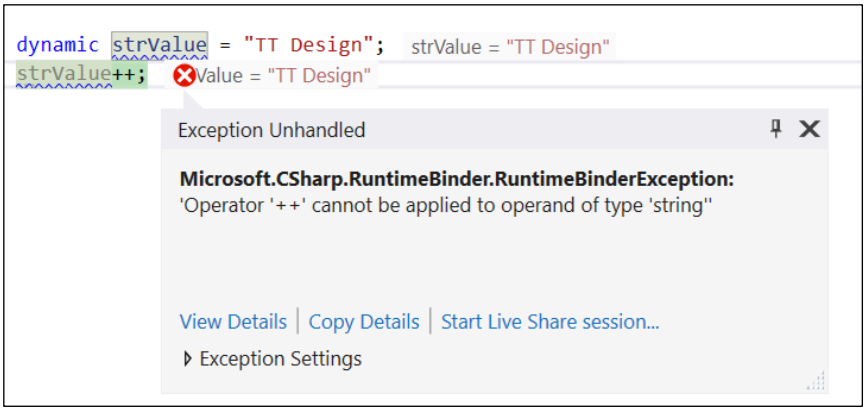
➢ **Ví dụ:**

```
0 references
static void Main(string[] args)
{
    C.WriteLine("**** Test dynamic program *****");

    dynamic strValue = "TT Design";
    strValue++;

}
```

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



➢ **Ý nghĩa:**

- 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
- Qúa trình boxing và unboxing được thực hiện ngầm định

➢ **Ví dụ:**

```
0 references
static void Main2(string[] args)
{
    string nameCom =  "TT Design";
    string description = "This is our company!";

    dynamic dynamicName = nameCom;

    Console.WriteLine($"We are in {nameCom}, {description}");
    //=> We are in TT Design, This is our company!
}
```

❖ 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 ()
- 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.

Bảng tổng hợp object, var và dynamic

Đặc điểm	Object	Var	Dynamic
Có phải kiểu dữ liệu	Đúng	Sai (là từ khóa)	
Khởi tạo giá trị khi khai báo	Ko bắt buộc	Bắt buộc	Ko bắt buộc
Có thể dùng để ép qua lại giữa các kiểu dữ liệu	Có	Không	Có
Thời điểm xác định KDL thực	Là KDL	Xác định sau khai báo	Xác định khi chương trình chạy

II. GIỚI THIỆU CẤU TRÚC VÒNG LẶP TRONG C#

Một vòng lặp là một chuỗi các sự kiện, hành động lặp lại đến khi thỏa mãn điều kiện dừng nào đó.

Vòng lặp vô tận là một chuỗi các sự kiện, hành động lặp lại vô tận do không bao giờ thỏa mãn điều kiện dừng.

Ví dụ:

- Chúng ta muốn chương trình viết ra "Hello guy!" 100 lần.
- Chúng ta muốn chương trình chạy đi chạy lại cho đến bao giờ người dùng nhập đúng input.

❖ Phân loại các vòng lặp trong C#

Trong C#, có nhiều cách để thực hiện vòng lặp. Chúng ta sẽ điểm qua tên của chúng nhé:

- Vòng lặp For
- Vòng lặp While
- Vòng lặp Do While
- Vòng lặp Foreach
- Vòng lặp goto
- Vòng lặp không chính quy khác

CẤU TRÚC LẶP GOTO TRONG C#:

➢ **Cú pháp:**

goto <label>;

- Trong đó label là một nhãn đích đến trong code. Nơi mà code sẽ tiếp tục được thực thi từ đó. Cấu trúc của một label: **<Tên label>:**
- goto là từ khóa thông báo cho trình biên dịch biết sẽ đi đến nhãn ngay sau để tiếp tục thực thi code.

➢ **Ý nghĩa:**

Trình biên dịch đi đến câu lệnh "goto <label> thì sẽ nhảy đến vị trí <label> để thực hiện biên dịch tiếp.

➢ **Ví dụ 01:**

```
0 references
static void Main3(string[] args)
{
    int a = 1;
    if (a == 2)
    {
        goto a_Is_2;
    }
    else
    {
        goto  a_Is_1;
    }
    C.WriteLine("Do nothing here!");

    a_Is_1:
    Console.WriteLine("A == 1");
    a_Is_2:
    C.WriteLine("A == 2");

    C.ReadKey();
}
```

➢ **Ví dụ 02: Lồng vào cú pháp của switch:**

```
//VD02 Ket hop vs Switch;
0 references
static void Main(string[] args)
{
    int a = 2;
    switch (a)
    {
        case 1:
            C.WriteLine("Case 1");
            break;
        case 2:
            C.WriteLine("Case 2");
            goto case 1;
            C.WriteLine("Case 2 was choose");
            break;
        case 3: // label case 3
            C.WriteLine("Case 3");
            break;
    }
    Console.ReadKey();
}
```

➢ **Kết luận:**

- Việc tạo label sẽ không ảnh hưởng gì đến code thông thường.
- label có thể được tạo ra một cách dễ dàng và lồng vào trong cấu trúc switch.

➢ **Ví dụ 02: Vòng lặp vô tận:**

```
//VD 03 Vong lap vo tan
0 references
static void Main(string[] args)
{
    string strValue = "Hello guy!";

    Start:

    // Do something here
    Console.WriteLine("I am Hoan.");
    Console.WriteLine(strValue + "\n\n");

    goto Start;
    C.ReadKey();
}
```

VÒNG LẶP FOR TRONG C#:

- **Cú pháp:**

```
for ([Khởi tạo]; [Điều kiện lặp]; [Bước lặp lại])
{

    // Khối lệnh được lặp lại. Có thể bỏ trống

}
```

- **Trong đó:**
  - Các phần [Khởi tạo]; [Điều kiện lặp]; [Bước lặp lại] hoàn toàn có thể để trống như ví dụ sau.
  - Mỗi đoạn [Khởi tạo]; hay [Điều kiện lặp]; hay [Bước lặp lại] là một câu lệnh riêng
- **Tiến trình:**
  - Ban đầu trình biên dịch sẽ di vào phần khởi tạo chạy đoạn lệnh khởi tạo
  - Tiếp theo kiểm tra điều kiện lặp. Rồi thực hiện khối code bên trong vòng lặp for. Khi đến ký hiệu } thì sẽ quay lên bước lặp lại.
  - Sau đó lại kiểm tra điều kiện lặp rồi tiếp tục thực hiện đoạn code trong khối lệnh. Đến khi điều kiện lặp không còn thõa mãn thì sẽ kết thúc vòng lặp for.
- **Vòng lặp for vô tận:**

```
for (; ;) // lưu ý dấu ;
{
   // Khối lệnh được lặp lại. Có thể bỏ trống
}
```

➢ **Ví dụ:**

```
//VD02 Lặp vô tận:
0 references
static void Main7(string[] args)
{
    for(; ; )
    {
        C.WriteLine("TT Design hello you!");
        C.WriteLine("************************\n");

    }
}
```

- **Bài tập.**
    1. Viết chương trình liệt kê n số nguyên tố đầu tiên.
    2. Ví dụ nếu bạn nhập số hàng là 5 thì vẽ tam giác sao có dạng:

        *
        **
        ***
        ****
        *****
    3. Kiểm tra số hoàn hảo: *(VD: 6 có các ước số ngoại trừ chính nó là 1, 2, 3 và có tổng các ước là 1+2+3= 6 --> 6 là số hoàn hảo)*
    4. Cho người dùng chơi lại trò “Kéo, Bao, Búa” đến khi thắng máy 3 lần thì mới exit chương trình.