

عنوان: تعریف نوع جنریک به صورت متغیر

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۱۲/۱۲ ۱۱:۳۹:۰۰

آدرس: [www.dotnettips.info](http://www.dotnettips.info)

برچسب‌ها: #C

در تهیه مثال [Auto Mapping](#) به کمک امکانات توکار NH 3.2 به این مورد نیاز پیدا کردم:  
بتوان نوع متد جنریک را به صورت متغیر تعریف کرد و این نوع در زمان کامپایل برنامه مشخص نباشد. مثلاً چیزی شبیه به این مثال:

```
using System;
namespace GenericsSample
{
    class TestGenerics
    {
        public static void Print<T>(T data)
        {
            Console.WriteLine("Print<T>");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            var type = typeof(Nullable<int>);
            TestGenerics.Print<type>(1);
        }
    }
}
```

این نوع فراخوانی متد Print در دات نت به صورت پیش فرض غیرمجاز است و نوع جنریک را نمی‌توان به صورت متغیر معرفی کرد.  
که البته این هم راه حل دارد و به کمک Reflection قابل حل است:

```
using System;
namespace GenericsSample
{
    class TestGenerics
    {
        public static void Print<T>(T data)
        {
            Console.WriteLine("Print<T>");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            var nullableIntType = typeof(Nullable<>).MakeGenericType(typeof(int));
            var method = typeof(TestGenerics).GetMethod("Print");
            var genericMethod = method.MakeGenericMethod(new[] { nullableIntType });
            genericMethod.Invoke(null, new object[] { 1 });
        }
    }
}
```

دو متد [MakeGenericType](#) و [MakeGenericMethod](#) برای ساخت پویای نوع‌های جنریک و همچنین ارسال آن‌ها به متدهای جنریک در دات نت وجود دارند که مثالی از نحوه استفاده از آن‌ها را در بالا ملاحظه می‌کنید.

مثال دوم:

اگر کلاس TestGenerics نسخه غیرجنریک متد Print را هم داشت، چطور؟ مثلاً:

```
class TestGenerics
{
    public static void Print<T>(T data)
    {
        Console.WriteLine("Print<T>");
    }

    public static void Print(object data)
    {
        Console.WriteLine("Print");
    }
}
```

اینبار اگر برنامه فوق را اجرا کنیم، پیغام Ambiguous match found را حین فراخوانی GetMethod دریافت خواهیم کرد؛ چون دو متد با یک نام در کلاس یاد شده وجود دارند. برای حل این مشکل باید به نحو زیر عمل کرد:

```
using System;
using System.Linq;

namespace GenericsSample
{
    class TestGenerics
    {
        public static void Print<T>(T data)
        {
            Console.WriteLine("Print<T>");
        }

        public static void Print(object data)
        {
            Console.WriteLine("Print");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            var nullableIntType = typeof(Nullable<>).MakeGenericType(typeof(int));
            var method = typeof(TestGenerics).GetMethod()
                .First(x => x.Name == "Print" &&
                    (x.GetParameters()[0]).ParameterType.IsGenericParameter);
            var genericMethod = method.MakeGenericMethod(new[] { nullableIntType });
            genericMethod.Invoke(null, new object[] { 1 });
        }
    }
}
```

GetMethod تمام متدها را بازگشت داده و سپس بر اساس متادیتای متدها، می‌توان تشخیص داد که کدام یک جنریک است.

## نظرات خوانندگان

نویسنده: ZB

تاریخ: ۲۲:۰۲:۲۱ ۱۳۹۰/۱۲/۱۲

سلام آقای نصیری .

ممنون از مطلب مفیدتون. من این کد رو نوشتم برای اینکه یک کلاس رو به متد جنریک بفرستم ولی نمیدونم چطور میتونم T رو به کلاسی که میخوام Cast کنم. ممنون میشم من رو راهنمایی بفرمایید:

```
var clsType = typeof(MyClass);var method = typeof(TestGenerics).GetMethods().First(x => x.Name == "Print" &&
(x.GetParameters()[0]).ParameterType.IsGenericParameter);genericMethod = method.MakeGenericMethod(new[] {
clsType });genericMethod.Invoke(null, new object[] { new MyClass{P1=100,P2="Name"} });Console.Read();var
clsType = typeof(MyClass);var method = typeof(TestGenerics).GetMethods().First(x => x.Name == "Print" &&
(x.GetParameters()[0]).ParameterType.IsGenericParameter);genericMethod = method.MakeGenericMethod(new[] {
clsType });genericMethod.Invoke(null, new object[] { new MyClass{P1=100,P2="Name"} });Console.Read();var
method = typeof(TestGenerics).GetMethods().First(x => x.Name == "Print" &&
(x.GetParameters()[0]).ParameterType.IsGenericParameter);genericMethod = method.MakeGenericMethod(new[] {
clsType });genericMethod.Invoke(null, new object[] { new MyClass{P1=100,P2="Name"} });Console.Read();new[] {
clsType });genericMethod.Invoke(null, new object[] { new MyClass{P1=100,P2="Name"} });Console.Read();null, new
public static void Print(T data) { و ();object[] { new MyClass{P1=100,P2="Name"} });Console.Read();Console.Read
static void Print(T data) { if (typeof(T) == typeof(MyClass)){var cls = (MyClass)data
if (typeof(T) == typeof(MyClass)){var cls = (MyClass)data
if (typeof(T) == typeof(MyClass)){var cls = (MyClass)data
var cls = (MyClass)data
که از خط آخر ایراد میگیره
که از خط آخر ایراد میگیره
که از خط آخر ایراد میگیره
که از خط آخر ایراد میگیره
```

نویسنده: وحید نصیری

تاریخ: ۲۲:۳۰:۴۲ ۱۳۹۰/۱۲/۱۲

لطفا از سایت <http://pastebin.com> برای ارسال کد طولانی استفاده کنید. به نظر می‌رسه قسمتی از کد شما توسط بلاگر حذف شده.

نویسنده: ZB

تاریخ: ۲۳:۰۹:۳۸ ۱۳۹۰/۱۲/۱۲

بله بخشید چون پست بلافاصله نمایش پیدا نمیکند متوجه نشدم که بهم ریخته.

در این آدرس کد رو قرار دادم : <http://pastebin.com/Q4bu1Vv1>

ولی از این خط 14 ارور میگیرم. در کل میخوام بدونم چطور T رو به هر کلاسی Cast کنم. ممنون

نویسنده: ZB

تاریخ: ۲۳:۲۹:۱۶ ۱۳۹۰/۱۲/۱۲

آقای نصیری اون مشکل حل شد . کد رو در این آدرس قرار دادم <http://pastebin.com/jKcnp6eb>

نویسنده: وحید نصیری

تاریخ: ۲۳:۴۹:۲۴ ۱۳۹۰/۱۲/۱۲

این دو روش هم هست:

```
var cls = data as MyClass
```

Or

```
((var cls = (MyClass)Convert.ChangeType(data, typeof(T
```

نویسنده: ZB

تاریخ: ۱۳۹۰/۱۲/۱۳ ۰۸:۳۱:۱۰

سلام آقای نصیری. ببخشید من یه چیزی رو متوجه نشدم. در مثالی که فرمودین در قسمت اول که گفتین ایراد داره اگر ما اون قسمت رو به این صورت بنویسیم:

```
;(TestGenerics.Print(1
```

برنامه کار میکنه. من حتی با کلاس هم امتحان کردم جواب میگیرم. در چه مواردی شما جواب نگرفتین و از رفلکشن استفاده کردین؟ با تشکر

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۱۲/۱۳ ۱۰:۱۷:۳۰

موردی که من بهش برخورد کردم این است:

فرض کنید یک متد جنریک دارید که T آن باید به صورت متغیر ساخته شود. مثلاً نوع آن EnumStringType of T است. این T هم هر بار در زمان اجرا بر اساس Enum های تعریف شده در اسمبلی فرق می‌کند. متد مورد نظر هم آرگومانی نداره. Map.Type EnumStringType of T است. اینجا است که نیاز خواهید داشت مراحلی که عنوان شد طی شود. یک مرحله خاصیت جنریک پویا تولید شود. یک مرحله متد پویای جنریک تولید شود و بعد هم فراخوانی آن.

نویسنده: Iman Abidi Ashtiani

تاریخ: ۱۳۹۰/۱۲/۱۳ ۱۱:۰۳:۵۸

```
اگر شما 1;(TestGenerics.Print(1
رو با 1;(TestGenerics.Print(1
```

عوض کنید که صورت مسئله هم عوض میشه و تبدیل میشه به یک مطلب آموزشی ابتدایی تویه جنریک ها. اصلاً اینجا دنبال اینیم که مشکل اینکه نوع جنریک را نمی‌توان به صورت متغیر معرفی کرد رو حل کنیم. و یکم رفلکشن یاد بگیریم و حواسمون به امکانات عالیش باشه

نویسنده: ZB

تاریخ: ۱۳۹۰/۱۲/۱۳ ۱۴:۳۷:۱۵

من منظورم این بود که بهر حال این متغیر یک نوع داره. بجای اینکه ما اول مقدار رو بریزیم داخل یک متغیر و بعد ازش استفاده کنیم مستقیم نوع متغیر رو استفاده کنیم.

نویسنده: Amgh\_12

تاریخ: ۱۳۹۰/۱۲/۱۷ ۱۵:۲۱:۲۲

سلام آقای نصیری

ممنون از سایت مفیدتون

می‌خواستم خواهش کنم برای نرم افزار subtitle tools امکان تصحیح یونیکد گروهی زیرنویس ها را هم فراهم کنید

البته می‌خواستم تو پست مربوطه نظر بگذارم ولی نظرات بسته بود

ممنون

نویسنده: وحید نصیری

سلام، امکان انتخاب یک پوشه را هم به همان قسمت تبدیل encoding اضافه کردم: [\(^\)](#)

پ.ن.  
ایمیل من پایین صفحه هست. همچنین هر پروژه موجود در CodePlex یک قسمت discussion هم دارد برای طرح این نوع موارد.