عنوان: **آشنایی با Implicit Casting و Explicit Casting**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۰۳/۱۸ ۲۳:۳۰
آدرس: www.dotnettips.info
گروه‌ها: C#, implicit casting, explicit casting

همه ما به نحوی در پروژه‌های خود مجبور به تبدیل انوع داده شده ایم و یک نوع از داده یا Object رو به نوع دیگری از داده یا Object تبدیل کرده ایم. در این پست دو روش دیگر برای تبدیل انواع داده‌ها بررسی میکنیم. برای شروع دو کلاس زیر رو در نظر بگیرید.

#1کلاس Book

```
public class Book
    {
        public int Code { get; set; }
        public string Title { get; set; }
        public string Category { get; set; }
    }
```

#2کلاس NoteBook

```
public class NoteBook
    {
        public int Code { get; set; }
        public string Title { get; set; }
    }
```

این دو کلاس هیچ ارتباطی با هم ندارند در نتیجه امکان تبدیل این دو نوع وجود ندارد یعنی اجرای هر دو دستور زیر باعث ایجاد خطای کامپایلری می‌شود.

```
static void Main( string[] args )
        {
            Book book = new Book()
            {
                Code = 1,
                Title = "Book1",
                Category = "Default"
            };

            NoteBook noteBook = new NoteBook();

            noteBook = (NoteBook)book;//Compile error
            noteBook = book as NoteBook;//Compile error
        }
```

برای حل این مشکل و تبدیل این دو نوع از Objectها می‌تونیم از دو نوع ImplicitCasting و Explicit Casting استفاده کنیم.

Explicit Casting#

```
public class Book
    {
        public int Code { get; set; }
        public string Title { get; set; }
        public string Category { get; set; }

        public static explicit operator NoteBook( Book book )
        {
            return new NoteBook()
            {
                Code = book.Code,
                Title = book.Title
            };
        }
    }
```

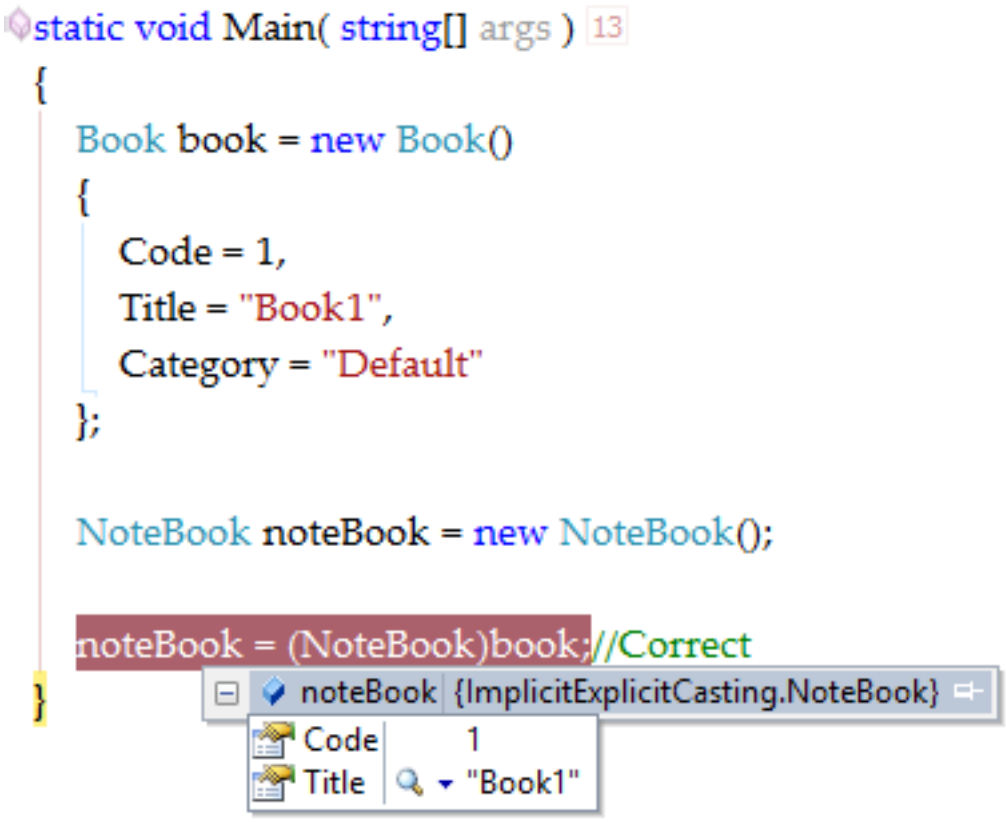
در Explicit یک Operator به صورت Explicit تعریف می‌کنیم که ورودی اون از نوع خود کلاس book و خروجی اون از نوع

مورد دلخواه است. Converter مورد نظر رو در بدنه این Operator می‌نویسیم. حالا به راحتی دستور زیر کامپایل می‌شود.

```
static void Main( string[] args )
        {
            Book book = new Book()
            {
                Code = 1,
                Title = "Book1",
                Category = "Default"
            };

            NoteBook noteBook = new NoteBook();

            noteBook = (NoteBook)book;//Correct
        }
```



در بالا مشاهده می‌کنید که حتما باید به طور صریح عملیات Cast رو انجام دهیددر غیر این صورت همچنان خطا خواهید داشت. اما می‌توان این مراحل رو هم نادیده گرفت و تبدیل رو به صورت Implicit انجام داد.

Implicit Casting#

```
public class Book
    {
        public int Code { get; set; }
        public string Title { get; set; }
        public string Category { get; set; }

        public static implicit operator NoteBook( Book book )
        {
```

```
            return new NoteBook()
            {
                Code = book.Code,
                Title = book.Title
            };
        }
    }
```

تنها تفاوت این روش با روش قبلی، در نوع تعریف operator است. بعد از تعریف نوع استفاده به صورت زیر خواهد بود.

```
static void Main( string[] args )
        {
            Book book = new Book()
            {
                Code = 1,
                Title = "Book1",
                Category = "Default"
            };

            NoteBook noteBook = new NoteBook();

            noteBook = book;//Correct
        }
```

در این روش نیاز به ذکر نوع Object برای Cast نیست و Object مورد نظر به راحتی به نوع داده قبل از اپراتور = تبدیل می‌شود.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۸ ۲۳:۵۸

مطلب جالبی است که کمتر مورد استفاده قرار می‌گیره. شاید ذکر مثال‌های دنیای واقعی اون در اینجا مکمل خوبی باشه. اگر دوستان هم مثالی مدنظرشون بود، لطفا عنوان کنند.

An easier way to manage INotifyPropertyChanged

#Custom Explicit and Implicit Operators in C

?Why does this code work

ASP.NET MVC Model binding with implicit operators

Forgotten C# language features: implicit operator

How to fake Enums in EF 4

نویسنده: میثم هوشمند
تاریخ: ۱۳۹۲/۰۹/۰۲ ۱۰:۲

با سلام
من این کار را میخواهم بر روی آرایه ای از یک کلاس انجام دهم؛
یعنی در واقع آرایه ای از یک کلاس را به آرایه ای از کلاسی دیگر تخصیص دهم
با جستجویی که کردم گویا امکانش نیست و نیاز به استفاده از تکنیک هایی هست
اما مثلا در دات نت ورژن دو گویا پشتیبانی نمی‌شوند.
دقیق خاطرم نیست اما فکر میکنم از LINQ استفاده شده بود
با تشکر