عنوان: **چگونگی دسترسی به فیلد و خاصیت غیر عمومی**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۰۳/۱۵ ۱۶:۱۰
آدرس: www.dotnettips.info
گروه‌ها: C#, Reflection, read-only

یک از ابتدایی‌ترین مواردی که در یادگیری دات نت آموزش داده می‌شود مباحث مربوط به کپسوله سازی است. برای مثال فیلدها و خواص Private که به صورت خصوصی هستند یا Protected هستند از خارج کلاس قابل دسترسی نیستند. برای دسترسی به این کلاس‌ها باید از خواص یا متدهای عمومی استفاده کرد.

```
public class Book
    {
        private int code = 10;

        public int GetCode()
        {
            return code;
        }
    }
```

یا فیلدها و خواصی که به صورت فقط خواندنی هستند،(RealOnly) امکان تغییر مقدار برای اون‌ها وجود ندارد. برای مثال کد پایین کامپایل نخواهد شد.

```
public class Book
    {
        private readonly int code = 10;

        public int GetCode()
        {
            return code = 20;
        }
    }
```

اما در دات نت با استفاده از Reflectionها می‌تونیم تمام قوانین بالا رو نادیده بگیریم. یعنی می‌تونیم هم به خواص و فیلدهای غیر عمومی کلاس دسترسی پیدا کنیم و هم می‌تونیم مقدار فیلدهای فقط خواندنی رو تغییر بدیم. به مثال‌های زیر دقت کنید.
#مثال اول

```
using System.Reflection;

 public class Book
 {
        private int code = 10;
 }

 public class Program
 {
        static void Main( string[] args )
        {
            Book book = new Book();
            var codeField = book.GetType().GetField( "code", BindingFlags.NonPublic |
BindingFlags.Instance );
            codeField.SetValue( book, 20 );
            var value = codeField.GetValue( book );
        }
    }
```

ابتدا یک کلاس که دارای یک متغیر به نام کد است ساخته ایم که مقدار 10 را دارد. فیلد به صورت private است. بعد از اجرا به راحتی مقدار Code را به دست می‌آوریم.

```
class Program
{
   static void Main( string[] args )
   {
      Book book = new Book();
      var codeField = book.GetType().GetField( "code", BindingFlags.NonPublic | BindingFlags.Instance );
      var value = codeField.GetValue( book );
   }
}
```

value 10

حتی امکان تغییر مقدار فیلد private هم امکان پذیر است.

```
class Program
{
   static void Main( string[] args )
   {
      Book book = new Book();
      var codeField = book.GetType().GetField( "code", BindingFlags.NonPublic | BindingFlags.Instance );
      codeField.SetValue( book, 100 );
      var value = codeField.GetValue( book );
   }
}
```

value 100

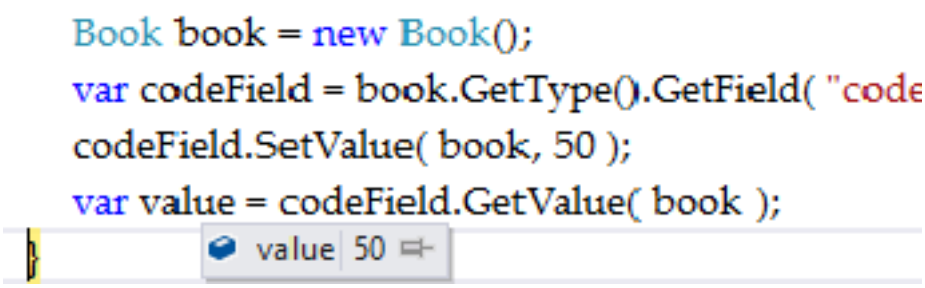
#مثال دوم.
در این مثال قصد داریم مقدار یک فیلد، از نوع فقط خواندنی رو تغییر دهیم.

```
using System.Reflection;

 public class Book
 {
        private readonly int code = 10;
 }

 public class Program
 {
        static void Main( string[] args )
        {
            Book book = new Book();
            var codeField = book.GetType().GetField( "code", BindingFlags.NonPublic |
BindingFlags.Instance );
            codeField.SetValue( book, 50);
            var value = codeField.GetValue( book );
        }
    }
```

بعد از اجرا مقدار متغیر code به 50 تغییر می‌یابد.



مطالب تکمیلی

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۷ ۱۰:۱۰

البته مباحث Reflection، تابع سطح دسترسی کد فراخوان است (همان لینک آخر بحث جهت تاکید بیشتر و همچنین تنویر مقدمه):

« Security Considerations for Reflection »

برای نمونه در حالت medium trust، گزینه ReflectionPermission غیرفعال است.

برای آزمایش این مسایل می‌شود از دو برنامه Permview و Permcalc استفاده کرد.

نویسنده: سالار خلیل زاده
تاریخ: ۱۳۹۲/۰۳/۱۸ ۹:۱۸

ReflectionMagic جهت همین کار طراحی شده

http://nuget.org/packages/ReflectionMagic

نویسنده: reza
تاریخ: ۱۳۹۳/۰۵/۲۵ ۱۷:۹

آیا می‌توان به کمک رفلکشن به خصوصیتی که مثلا Set ندارد مقدار دهی کرد. بعنوان مثال

```
class Program
{
  static void Main(string[] args)
  {
      Book book = new Book();
      var codeprop = book.GetType().GetProperty("Code", BindingFlags.Public | BindingFlags.Instance);
      codeprop.SetValue(book, 20, null);
      var value = codeprop.GetValue(book, null);
  }
}

public class Book
{
  public int code;
  public int Code
  {
      get { return code; }
  }
}
```

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۳/۰۵/۲۶ ۱۴:۰

خیر! با یک ArgumentException و پیغام Property set method not found مواجه خواهید شد. اما در مثال بالا می‌توان مقدار فیلد code را تغییر داد که در نتیجه خاصیت Code نیز مقدار جدید را برگشت می‌دهد.