عنوان: **توابع تعمیم یافته در #C**
نویسنده: سیدمجتبی حسینی
تاریخ: ۱۳۹۱/۰۷/۰۲ ۱:۳
آدرس: www.dotnettips.info
برچسب‌ها: #C

از توابع تعمیم یافته می‌توان برای توسعه توابع هر کلاس یا اینترفیسی استفاده کرد. یعنی می‌توان یک تابع را به هر کلاسی اضافه کرد.

قبل از C# 3.0 فقط می‌شد یک کلاس را از طریق ارث‌بری از آن توسعه داد و به کلاس مهروموم شده یا Sealed نیز نمی‌شد تابعی افزود که البته ممکن است بگویید که این کار قوانین شیءگرایی را نقض می‌کند اما در پاسخ باید گفت که توابع تعمیم یافته به اعضاء خصوصی کلاسی که تعمیم می‌یابد، دسترسی ندارند.

تعمیم یک کلاس در خارج از بدنه کلاس انجام می‌شود و این کار می‌تواند در فضای نام همان کلاس یا خارج از آن انجام شود و شکل کلی آن به صورت زیر است :

```
public static class  ExtendingClassName
 {
      public static ReturnType MethodName(this   ExtendedMethod  arg)
      {
           //دستورات درون متد
          Return ReturnType;
      }
 }
```

توجه کنید که:
کلاس توسعه‌دهنده و تابع توسعه‌دهنده باید استاتیک باشند.
در داخل آرگومان تابع، کلمه کلیدی this استفاده می‌شود.
بعد از this عنوان کلاسی که قصد توسعه آن را داریم، ذکر می‌کنیم.
در هرجا که خواستیم از قابلیت تعمیم داده شده استفاده کنیم باید فضای نام مربوط به آن را ذکر کنیم.
با کلمه کلیدی static نمی‌توان کلاسی با متدهای abstract ، virtual و override را توسعه داد.

مثلا اگر قصد داریم به کلاس String تابع AddPrefix را اضافه کنیم به این شکل عمل می‌کنیم :

```
public static class  ExtendingString
{
  public static  string  AddPrefix(this   string  arg)
  {
     return String.Format("prefix{0}",arg);
  }
}
```

که نحوه استفاده از آن به شکل زیر است:

```
string  s = "Student";
Console.WriteLine(s.AddPrefix());
```

و خروجی آن نمایش prefixStudent است.

اگر بخواهیم عبارت پیشوند را از طریق آرگومان ارسال کنیم به این شکل عمل می‌کنیم:

```
public static class ExtendingString
{
```

```
    public static string AddPrefix(this   string arg, string prefix)
    {
         return String.Format("{0}{1}", prefix, arg);
     }
}
```

که نحوه استفاده از آن به شکل زیر است:

```
var  s = "Student";
Console.WriteLine(s.AddPrefix("tbl"));
```

و خروجی آن نمایش tblStudent است.

به عنوان مثال دوم کلاس زیر را در نظر بگیرید:

```
public class Test
{
     public int AddOne(int val)
     {
         return val + 1;
     }
  }
```

اگر بخواهیم کلاس فوق را توسعه داده و متد دیگری مثلا با عنوان AddTwo اضافه کنیم، کلاس توسعه دهنده را به این شکل می‌نویسیم:

```
public static class ExtendingTest
{
     public static int AddTwo(this   Test  arg, int val)
     {
         return val + 2;
     }
}
```

و روش استفاده از آن بصورت زیر است:

```
static void Main(string[] args)
{
       var x = new Test();
       Console.WriteLine(x.AddOne(10));
       Console.WriteLine(x.AddTwo(10));
       Console.Read();
}
```

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۷/۰۲ ۹:۲۸

بسیار عالی بیان کردید.ممنون از شما.
موفق باشید.

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۷/۰۲ ۱۶:۲۰

من ترجمه extension methods رو «متدهای الحاقی» هم دیدم.

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۷/۰۳ ۱۳:۳۳

ممنون از مقاله خوبتون.
به نظر من هم متدهای الحاقی ترجمه صحیح Extension Method است. اول اینکه متد با تابع فرق دارد دوم اینکه تعمیم به معنای گسترش چیزی است و توابع تعمیم یافته یعنی اینکه توابع موجود را گسترش میدید در صورتی که در Extension Methodها چیزی که تعمیم می‌یابد در اصل کلاسها هستند نه متدها که این تعمیم کلاسها با الحاق متدهای جدید انجام میشود

نویسنده: علی
تاریخ: ۱۳۹۱/۰۷/۰۳ ۲۰:۲۹

البته توسعه تعداد توابع نیز به نوعی توسعه توابع است و الزامی ندارد فقط یک تابع توسعه یابد تا تعمیم و توسعه صدق کند

البته الحاق هم صحیح است .

نکته دیگر آنکه برای متد و فانکشن و مانند آن در فارسی معادلی غیر تابع ندیده‌ام که متداول باشد. بعلاوه آنکه تابع مفهوم همه اینها را شامل می‌شود خواه نوع بازگشتی داشته باشد یا void باشد، لذا تعبیر غیر صحیحی نیست

باتشکر

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۷/۰۴ ۹:۴۳

بله با شما موافقم که تابع مفهوم همه اینها را شامل می‌شود اما در اینجا دقیقا "متد" مد نظر است. مفهوم "تابع" همه رویه هایی که متعلق به کلاس باشند یا نباشند را در بر میگیرد در صورتی که "متد" قطعا متعلق به یک کلاس است. با توجه به ذکر صریح کلمه Method در Extension Method و مفهوم آن به نظر من استفاده از متد صحیح است.
در فارسی مدتی کلمه "روش" به جای "متد" در ترجمه‌ها رایج شد که زیاد جا نیفتاد و بیشتر مترجمین جدیدا از همون کلمه "متد" استفاده می‌کنن