

عنوان: اشتباهات متداول برنامه‌نویس‌های دات نت

نویسنده: وحید نصیری

تاریخ: ۱۳۸۷/۱۲/۱۹ ۲۰:۱۵:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: Tips

اشتباه 1:

استفاده از

```
throw ex;
```

بجای استفاده از

```
throw;
```

در حالت اول، تمام stack trace موجود تا نقطه‌ی فراخوانی دستور ذکر شده پاک خواهند شد، اما در حالت دوم stack trace حفظ شده و دیدگاه کردن کد را ساده می‌کند.

اشتباه 2:

درک اشتباه عملکرد متد replace :

```
string s = "Take this out";  
s.Replace("this", "that"); //wrong
```

بجای استفاده از :

```
s = s.Replace("this", "that"); //correct
```

اگر از fxCop استفاده کنید، اینگونه خطاها را (عدم استفاده از مقدار بازگشتی) گوشزد می‌کند.

اشتباه 3:

استفاده‌ی بی دقت از متغیرهای استاتیک در یک برنامه وب. دو مثال زیر را در نظر بگیرید:

```
public static string GetCookieName(Cookie c)  
{  
    return c.Name;  
}
```

```
static List<string> cookieList = new List<string>();  
public static void AddToCookieList(Cookie c)  
{  
    cookieList.Add(c.Name);  
}
```

برنامه‌های وب ذاتاً چند ریسمانی هستند و زمانیکه یک متغیر را از نوع استاتیک تعریف می‌کنید، این متغیر، هنگام مراجعه‌ی کاربران به آن‌ها به اشتراک گذاشته می‌شود و امکان تخریب یا استفاده‌ی ناصحیح از مقادیر آن‌ها وجود خواهد داشت. در حالت اول نیازی

به مباحث همزمانی و قفل کردن منابع نیست زیرا متغیری که در متد استفاده می‌شود، thread safe است اما cookieList در مثال دوم خیر و حتما هنگام استفاده از آن باید مباحث قفل کردن منابع را در نظر داشت. (حتی اگر برنامه‌ی شما از نوع وبی هم نیست اما چند ریسمانی است این مطلب باز هم صادق می‌باشد)

اشتباه 4:

مقابله با خطاها به شکلی نادرست: (اصطلاحا خفه کردن خطاها!)

```
try
{
    //something
    File.Delete(blah);
}
catch{}
```

در این حالت اگر خطایی رخ دهد کاربر متوجه نخواهد شد دقیقا مشکلی وجود داشته یا خیر و علت آن چیست. هرچند گاهی از اوقات اگر خطای حاصل برای ما اهمیتی نداشت می‌توان از آن استفاده نمود، در غیراینصورت باید حتما از این روش پرهیز کرد.

اشتباه 5:

ارائه‌ی برنامه‌های ASP.Net با گزینه‌ی پیش فرض Debug=true در web.config که پیشتر در مورد آن در این سایت [بحث شده](#) است.

اشتباه 6:

عدم استفاده از امکانات ویژه‌ی دات نت فریم ورک هنگام کار با رشته‌ها:

```
string s = "This ";
s += "is ";
s += "not ";
s += "the ";
s += "best ";
s += "way.";
```

```
StringBuilder sb = new StringBuilder();
sb.Append("This ");
sb.Append("is ");
sb.Append("much ");
sb.Append("better. ");
```

زمانیکه نیاز به کنار هم قرار دادن رشته‌های مختلف وجود داشت و تعداد آن‌ها نیز زیاد بود، مثال دوم توصیه می‌شود. البته در مثال فوق که تعداد کمی رشته قرار است با هم جمع شوند، کامپایلر به اندازه‌ی کافی هوشمند خواهد بود که تمام آن‌ها را کنار هم قرار دهد و تفاوتی در کارآیی احساس نشود، حتی روش اول سریع‌تر از روش دوم خواهد بود، اما زمان استفاده از هزاران رشته، این تفاوت [محسوس](#) است.

اشتباه 7:

عدم استفاده از عبارت using هنگام استفاده از اشیایی از نوع IDisposable. [مثالی](#) در این مورد.

```
using (StreamReader reader=new StreamReader(file))
{
    //your code here
}
```

اشتباه 8:

فراموش کردن بررسی نال بودن یک شیء هنگام استفاده از آن.

```
string val = Session["xyz"].ToString();
```

این نوع کد نویسی یکی از اشتباهات متداول تمامی تازه واردان به ASP.Net است. حتما باید پیش از استفاده از متد ToString بررسی شود که آیا این سشن نال است یا نه. در غیراینصورت حاصل کار فقط یک exception خواهد بود. (استفاده از افزونه‌ی ری شارپر در این موارد کمک بزرگی است، زیرا به محض فرار گرفتن مکان نما روی شیء‌ایی که احتمال نال بودن آن میسر است، یک راهنما را به شما ارائه خواهد کرد)

اشتباه 9:

بازگشت دادن یک property عمومی از نوع لیست‌های جنریک.

با توجه به اینکه این نوع لیست‌ها فقط خواندنی نیستند و امکان دستکاری اطلاعات آن توسط فراخوان وجود دارد، [توصیه می‌شود](#) از نوع جنریک IEnumerable استفاده شود. همچنین توصیه شده است هنگام انتخاب نوع پارامترهای ورودی یک متد نیز به این مورد دقت شود.

نظرات خوانندگان

نویسنده: کورش اسلام زاده
تاریخ: ۱۳۸۷/۱۲/۱۹ ۲۱:۰۰:۰۰

سلام

خبرخوان (تکست و گرافیکی)، لیست و فید وبلاگ‌های برنامه‌نویسی ارائه شده است که وبلاگ شما نیز جز آنها قرار گرفته است.

<http://persianbloggers.blogspot.com/2009/03/programming-p.html>

پرشین بلاگرز شما را به بازدید و استفاده از این خبرخوان و 23 خبرخوان تخصصی دیگر موجود دعوت می‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۱۹ ۲۱:۳۶:۰۰

با تشکر.

البته در تایتل لینک هم قرار گرفته که فید نهایی را ناخوانا کرده. لینک و تایتل باید مجزا تعریف شوند.

<http://feeds2.feedburner.com/PersianBloggers/Programming>

نویسنده: کورش اسلام زاده
تاریخ: ۱۳۸۷/۱۲/۲۰ ۱۹:۵۴:۰۰

سلام

ممنون از نظرتان. آیا شما فید را در گوگل ریدر یا هر خوراک خوان دیگر هم ملاحظه کرده اید؟ حرف شما درست که بصورت استاندارد باید لینک و تایتل جدا باشند اما هدف مشاهده در خوراک خوان است که گمان میکنم با روش موجود برتری های بیشتری هم بر فید مرسوم داشته باشد.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۲۰ ۲۰:۱۱:۰۰

سلام،

لطفا یکبار لینک فید فوق را در فایرفاکس و یکبار هم با IE ملاحظه کنید. متن تایتل‌ها به هم ریخته است (مخلوطی است از لینک مطلب و تایتل مطلب).

نویسنده: کورش اسلام زاده
تاریخ: ۱۳۸۷/۱۲/۲۰ ۲۰:۵۶:۰۰

سلام

یک مقایسه. تصویر بدون به هم ریختگی هم در ie و هم فایرفاکس

<http://tinyurl.com/ahkw5u>

<http://tinyurl.com/asohbo>

و تصویری که در ie کاملن سالم است و در فایرفاکس فقط یک کلمه آن جلو و عقب شده است

<http://tinyurl.com/cf2yos>

در هر حال در خوراک خوان عنوان با آبی مشخص شده است و در موارد به هم ریختگی در هر صورت مشخص است. در ضمن در مواردی که انگلیسی و فارسی در عنوان - با هم استفاده شده است در بعضی موارد مشکل بروز میکند که گمان نمیکنم خود فید اصلی هم همیشه و با یک رویه نشان داده شود. موارد بسیاری را شاهدیم که نام وبلاگ یا نویسنده گاهی در راست و گاهی در

چپ است و ...

به نظر من نمی‌رسد مشکل حادی در این رابطه وجود داشته باشد و شاید هم بنده کاملن مقصود شما را متوجه نشده ام که ممنون میشوم توضیح بدهید. در عین حال یک پست در رابطه با فید پرشین بلاگرز را در حال آماده کردن هستم که بزودی پست میکنم. در این پست مقایسه بیشتری به عمل خواهد آمد.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۲۰ ۲۱:۳۴:۰۰

به لطف isp ها، تصاویر بلاگر فیلتر شده ...

<http://www.tinypic.info/files/9m815lhvn499dlmegy2l.png>

نویسنده: Kourosh
تاریخ: ۱۳۸۷/۱۲/۲۰ ۲۱:۴۵:۰۰

واقعن اینجا راز من دیوانه می‌شوم. واقعن نمیدانم کجا باید آپلود کنم.....

نویسنده: Kourosh
تاریخ: ۱۳۸۷/۱۲/۲۰ ۲۱:۴۶:۰۰

در این رابطه

<http://weblogcrawler.blogspot.com/2009/03/feed-in-poersian-bloggers.html>