

تفاوت بین یک کلاس استاتیک، متدی استاتیک و یا متغیر عضو استاتیک چیست؟ چه زمانی باید از آن‌ها استفاده کرد و لزوم بودن آن‌ها چیست؟

برای پاسخ دادن به این سؤالات باید از نحوه‌ی تقسیم بندی حافظه شروع کرد. RAM برای هر نوع پروسه‌ای که در آن بارگذاری می‌شود به سه قسمت تقسیم می‌گردد: Stack ، Heap و Static (استاتیک در دات نت در حقیقت قسمتی از Heap است که به آن High Frequency Heap نیز گفته می‌شود). این قسمت استاتیک حافظه، محل نگهداری متدها و متغیرهای استاتیک است. آن متدها و یا متغیرهایی که نیاز به وهله‌ای از کلاس برای ایجاد ندارند، به صورت استاتیک ایجاد می‌گردند. در سی شارپ از واژه کلیدی static برای معرفی آن‌ها کمک گرفته می‌شود. برای مثال:

```
class MyClass
{
    public static int a;
    public static void DoSomething();
}
```

در این مثال برای فراخوانی متد DoSomething نیازی به ایجاد یک وهله جدید از کلاس MyClass نمی‌باشد و تنها کافی است بنویسیم:

```
MyClass.DoSomething(); // and not -> new MyClass().DoSomething();
```

نکته‌ی مهمی که در اینجا وجود دارد این است که متدهای استاتیک تنها قادر به استفاده از متغیرهای استاتیک تعریف شده در سطح کلاس هستند. علت چیست؟ به مثال زیر دقت نمائید:

```
class MyClass
{
    // non-static instance member variable
    private int a;
    //static member variable
    private static int b;
    //static method
    public static void DoSomething()
    {
        //this will result in compilation error as "a" has no memory
        a = a + 1;
        //this works fine since "b" is static
        b = b + 1;
    }
}
```

در این مثال اگر متد DoSomething را فراخوانی کنیم، تنها متغیر b تعریف شده، در حافظه حضور داشته (به دلیل استاتیک معرفی شدن) و چون با روش فراخوانی MyClass.DoSomething هنوز وهله‌ای از کلاس مذکور ایجاد نشده، به متغیر a نیز حافظه‌ای اختصاص داده نشده است و نامعین می‌باشد. بر این اساس کامپایلر نیز از کامپایل شدن این کد جلوگیری کرده و خطای لازم را گوشزد خواهد کرد.

اکنون تعریف یک کلاس به صورت استاتیک چه اثری را خواهد داشت؟ با تعریف یک کلاس به صورت استاتیک مشخص خواهیم کرد که این کلاس تنها حاوی متدها و متغیرهای استاتیک می‌باشد. امکان ایجاد یک وهله از آن‌ها وجود نداشته و نیازی نیز به این امر ندارند. این کلاس‌ها امکان داشتن instance variables را نداشته و به

صورت پیش فرض از نوع sealed به حساب خواهند آمد و امکان ارث بری از آنها نیز وجود ندارد. علت این امر هم این است که یک کلاس static هیچ نوع رفتاری را تعریف نمی‌کند.

پس با این تفاسیر چرا نیاز به یک کلاس static ممکن است وجود داشته باشد؟ همانطور که عنوان شد یک کلاس استاتیک هیچ نوع رفتاری را تعریف نمی‌کند بنابراین بهترین مکان است برای تعریف متدهای کمکی که به سایر اعضای کلاس‌های ما وابستگی نداشته، عمومی بوده، مستقل و متکی به خود هستند. عموماً متدهای کمکی در یک برنامه به صورت مکرر فراخوانی شده و نیاز است تا به سرعت در دسترس قرار داشته باشند و حداقل یک مرحله ایجاد و هله کلاس در اینجا برای راندمان بیشتر حذف گردد. برای مثال متدی را در نظر بگیرید که بجز اعداد، سایر حروف یک رشته را حذف می‌کند. این متد عمومی است، وابستگی به سایر اعضای یک کلاس یا کلاس‌های دیگر ندارد. بنابراین در گروه متدهای کمکی قرار می‌گیرد. اگر از افزونه‌ی ReSharper استفاده نمائید، این نوع متدها را به صورت خودکار تشخیص داده و راهنمایی لازم را جهت تبدیل آن‌ها به متدهای استاتیک ارائه خواهد داد.

با کلاس‌های استاتیک نیز همانند سایر کلاس‌های یک برنامه توسط [JIT compiler](#) رفتار می‌شود، اما با یک تفاوت. کلاس‌های استاتیک فقط یکبار هنگام اولین دسترسی به آن‌ها ساخته شده و در قسمت High Frequency Heap حافظه قرار می‌گیرند. این قسمت از حافظه تا پایان کار برنامه از دست garbage collector در امان است (بر خلاف garbage-collected heap یا object heap که جهت instance classes مورد استفاده قرار می‌گیرد)

نکته:

در برنامه‌های ASP.Net از بکارگیری متغیرهای عمومی استاتیک برحذر باشید (از static fields و نه static methods). این متغیرها بین تمامی کاربران همزمان یک برنامه به اشتراک گذاشته شده و همچنین باید مباحث قفل‌گذاری و امثال آن‌را در محیط‌های چند ریسمانی هنگام کار با آن‌ها رعایت کرد ( [thread safe نیستند](#) ).

## نظرات خوانندگان

نویسنده: نیما

تاریخ: ۱۳۸۸/۰۶/۲۸ ۲۲:۰۵:۰۸

با سلام خدمت استاد نصیری  
جناب نصیری طبق نکت آخر پس اینکه ما میام برای استفاده از jQuery Ajax از متدهای استاتیک استفاده میکنیم کار خوبی نمیکنیم؟ آخه این تنها راه هست.  
یه سوال دیگه هم داشتم که MS Ajax چطور میتونه متدهای غیر استاتیک صفحه رو هم صدا بزنه در حالی که ما در jQuery فقط قادر به صدا زدن متدهای استاتیک هستیم؟  
سپاسگزارم

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۶/۲۸ ۲۳:۰۵:۲۹

سلام،  
- همانطور که تاکید کردم فیلد استاتیک و نه متد استاتیک که مشکل همزمانی را ندارد.  
- برای jQuery Ajax چون بسیاری از ملاحظات که توسط MS Ajax انجام می شود مانند کار با ViewState و ارسال و مدیریت آن صورت نمی گیرد، امکان ایجاد وهله ای از کلاس استاندارد صفحه ASP.Net توسط آن میسر نیست. بنابراین باید این متد را استاتیک تعریف کرد تا وابستگی آن را از شء صفحه قطع کرد. به همین جهت jQuery Ajax بسیار بهینه تر از MS Ajax عمل می کند (چون اساسا درکی از ViewState ندارد)

نویسنده: محمد امین شریفی

تاریخ: ۱۳۸۸/۰۶/۲۸ ۲۳:۳۳:۰۴

همانطور که گفته شد از متدهای استاتیک برای کارهایی که زیاد به آن ها نیاز داریم استفاده می شود.  
نکته های خوبی را بیان کردید.  
لطفا درباره کلاس های sealed بیشتر بنویسید.

پیروز باشید.

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۶/۲۸ ۲۳:۵۹:۵۴

الزاما خیر. اگر متد شما صرفا یک متد utility است و مستقل است یا اگر کلا متد شما وابستگی به کلاس جاری ندارد یا نیاز است متد مورد نظر از کلاس جاری گسسته شود، می توان آن را استاتیک تعریف کرد.

نویسنده: محمد امین شریفی

تاریخ: ۱۳۸۸/۰۶/۲۹ ۱۵:۳۶:۳۹

درباره کلاس های static و sealed هم می شود توضیح دهید؟  
در دات نت میکرو برخی کلاس ها را که در رابطه با پورت ها بودند و همچنین برای ساخت اکستنشن برای html helper، دات نت آنها را بصورت استاتیک تعریف کرده است. چگونه می توان از GC در کلاس ها، متغیر ها و تابع های استاتیک سود برد؟

ممنون،

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۶/۲۹ ۱۶:۰۴:۰۵

- همانطور که عنوان شد یک کلاس sealed قابل ارث بری نیست. توضیحات بیشتر: [http://msdn.microsoft.com/en-us/library/88c54tsw\(v5.71\).aspx](http://msdn.microsoft.com/en-us/library/88c54tsw(v5.71).aspx)

- من جایی عنوان نکردم اشیاء غیر استاتیک داخل یک متد استاتیک garbage collected نمی‌شوند. فقط عنوان کردم اگر فیلدی استاتیک بود در آن ناحیه از حافظه تا پایان کار برنامه باقی می‌ماند. یا اگر متدی استاتیک بود، کار ساخت جهت دسترسی سریع به آن فقط یکبار انجام می‌شود.

نویسنده: میثم جوادی  
تاریخ: ۱۳۸۸/۰۷/۰۱ ۰۸:۴۰:۱۸

در واقع زمانیکه ما یک کلاس رو سینگلتون پیاده سازی میکنیم، همواره از یک نقطه آبجکت رو میخونیم؟ کی این آبجکت از حافظه خالی میشود؟(بعد از اتمام برنامه؟) اگر اینطور است در Web-App مشکل ساز نخواهد بود؟(مانند فیلد استاتیک؟)

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۸/۰۷/۰۱ ۱۱:۲۷:۴۱

لطفا به آخرین لینک مقاله فوق مراجعه کنید. قسمت "thread safe نیستند". یک مقاله دو قسمتی است که این مباحث همزمانی رو بررسی کرده.

نویسنده: DotNetCoders  
تاریخ: ۱۳۸۸/۰۷/۰۱ ۱۸:۱۷:۵۸

ممنون جالب بود.فقط جای این نکته خالی موند که معادل static در vb.net کلمه کلیدی share هست.

/

جناب نصیری جای یک مقاله کامل با موضوع event و delegate ها در وبلاگ شما خالیه.

الان همه پروژه های اپن سورس رو که باز می کنی حتمی توش از ایونت ها استفاده شده ولی مفهوم و کارایش برای خیلی ها ( مثل خودم ) گنگه...

ممنون میشم این مورد هم مثل همه مورد های دیگه کامل بررسی کنید.

ممنون و خسته نباشید.