

عنوان: آرگومان‌های نامگذاری شده (named arguments/parameters) در C#4 :

نویسنده: وحید نصیری

تاریخ: ۱۸:۵۸:۰۰ ۱۳۸۹/۰۳/۱۶

آدرس: www.dotnettips.info

برچسب‌ها: #C

سؤال: آیا شما بدون مراجعه به مستندات شیء FormsAuthenticationTicket می‌توانید پاسخ دهید هر کدام از آرگومان‌های ذکر شده در سازنده‌ی این کلاس چه کاربرد و معنایی دارند؟

```
var ticket = new FormsAuthenticationTicket
(
    1,
    principal.Identity.Name,
    DateTime.Now,
    DateTime.Now.AddMinutes(30),
    true,
    string.Empty,
    FormsAuthentication.FormsCookiePath
);
```

الان چطور؟!

```
//using named parameters of C# 4.0
var ticket = new FormsAuthenticationTicket
(
    version: 1,
    name: principal.Identity.Name,
    issueDate: DateTime.Now,
    expiration: DateTime.Now.AddMinutes(30),
    isPersistent: true,
    userData: string.Empty,
    cookiePath: FormsAuthentication.FormsCookiePath
);
```

گاهی از اوقات ویژگی‌هایی که به یک زبان برنامه نویسی اضافه می‌شوند راه و روش برنامه نویسی را ممکن است کاملاً تحت تاثیر قرار دهند؛ برای مثال استفاده از Generics. اما گاهی از اوقات این ویژگی‌های جدید syntax sugar یا nice to have هستند (!) مانند Named arguments.

Named arguments یا آرگومان‌های نامگذاری شده خوانایی کدهای شما را با بیان اینکه هر آرگومان چه معنایی دارد، افزایش می‌دهند. برای مثال به نظر شما کدام روش فراخوانی متد Copy در ذیل خواناتر است؟

```
File.Copy("source.txt", "destination.txt", true);
```

و یا؟

```
File.Copy("source.txt", "destination.txt", overwrite: true);
```

هر دو یک کار را انجام می‌دهند اما درک عملکرد روش اول حتماً نیاز به بررسی مستندات متد Copy را خواهد داشت.

نکات تکمیلی:

متد ساده زیر را در نظر بگیرید:

```
static void Foo(int x, int y) {}
```

الف) در حالت معمولی و بدون استفاده از named argument ، پارامترهای یک متد، positional parameters هم نامیده می‌شوند. به این معنا که محل ذکر یک پارامتر مهم است و برای مثال در روش متداول فراخوانی ذیل، پارامتر دوم همیشه همان y است و پارامتر اول همواره همان x در نظر گرفته می‌شود:

```
Foo(1,2);
```

اما زمانیکه از named parameters استفاده می‌کنید، این ترتیب دیگر اهمیتی نداشته و هر دو فراخوانی ذیل مجاز می‌باشند و به یک معنا هستند:

```
Foo(x:4, y:2);  
Foo(y:2, x:4);
```

ب) همواره positional parameters باید پیش از named parameters ذکر شوند. برای مثال فراخوانی ذیل مجاز است:

```
Foo(4, y:2);
```

اما فراخوانی زیر مجاز نبوده و برنامه کامپایل نخواهد شد:

```
Foo(x:4, 2);    // Compile time error
```

نظرات خوانندگان

نویسنده: Amin

تاریخ: ۱۳۸۹/۰۳/۱۷ ۰۸:۵۲:۱۰

جالب بود. ممنون