عنوان: **درباره Iterator methodها و yield return در C#**
نویسنده: شهره مرتضوی
تاریخ: ۱۳۹۲/۰۶/۲۴ ۱۲:۳۵
آدرس: www.dotnettips.info
برچسب‌ها: C#.NET, Itrator methods, IEnumerator, IEnumerable, Yeild return

هنگامیکه می‌خواهید در متدهای خود مقداری (از هر نوع datatype دلخواه) را return نمایید، در حالت عادی قادر خواهید بود که فقط از یک return در بدنه متد خود استفاده نمایید:

```
public int Sum(int a, int b)
{
     return a + b;
}
```

اما چنانچه از متدهای تکرار شونده استفاده نمایید، چطور؟

متدهای تکرار شونده یا Iterator methodها ، در داخل یک collection به صورت دلخواه iterate کرده یا به اصلاح می‌چرخند. این متدها از کلمه کلیدی Yield در هنگام return کردن مقادیر استفاده می‌کنند. (در C# از Yield return و در VB از Yield استفاده می‌شود) به عبارت دیگر یک متد با خروجی از نوع قابل چرخش (مانند IEnumerable)، با استفاده از چند yield return، دارای قابلیت چرخش و بازگرداندن چندین مقدار به جای یک مقدار واحد می‌گردد.

برای درک بهتر مسئله از مثالی برای ادامه توضیحات استفاده می‌کنم. متد چرخشی (Iterate method) زیر را در نظر بگیرید که خروجی IEnumerable دارد:

```
public static IEnumerable SomeNumbers()
{
     yield return 3;
     yield return 5;
     yield return 8;
}
```

برای استفاده از مقادیر بازگشتی متد بالا از حلقه foreach زیر استفاده می‌نماییم:

```
static void Main()
{
    foreach (int number in SomeNumbers())
    {
        Console.Write(number.ToString() + " ");
    }
    // Output: 3 5 8
    Console.ReadKey();
}
```

حلقه foreach فوق ، در پایان اولین چرخش، عدد 3 را باز گردانده و مکان این return را حفظ می‌کند. در چرخه بعدی عدد 5 را باز می‌گرداند و این نقطه را نیز نگه می‌دارد و در چرخه پایانی عدد 8 را برگردانده و سپس حلقه با رسیدن به نقطه پایانی متد، خاتمه می‌یابد.

برای خاتمه چرخش در Iterator methodها ، میتوانید از foreach استفاده کنید و یا اینکه عبارت yield break را بعد از تمامی yield returnها به کار گیرید:

```
public static IEnumerable SomeNumbers()
{
   yield return 3;
   yield return 5;
   yield return 8;
   yeild break;
}
```

نکات:

- در هنگام ایجاد Iterator method ها، نوع مقادیر خروجی متد ، باید یکی از انواع IEnumerable, IEnumerable<T>, IEnumerator, و یا IEnumerator<T>. باشد.
- در هنگام declare کردن ، نمی‌توانید از پارامترهای ref و out استفاده نمایید.
- در Anonymous methodها (متدهای بی نام) و Unsafe blockها نمی‌توانید از yield return (yield در VB) استفاده نمایید.
- نمی‌توانید از Yield return در بلوکهای try-catch استفاده کندی. اما می‌توانید در قسمت try بلوک try-finally استفاده نمایید.
- از yield break می‌توانید در بلوک try و یا بلوک catch استفاده نمایید ، اما در بلوک finally خیر.
- هنگام بروز خطا در foreach هایی که خارج از iterator methodها استفاده می‌شوند، بلوک finally داخل این متدها اجرا می‌گردد.

مثالی دیگر با استفاده Iterator methodها و yield return جهت بازگرداندن روزهای هفته:

```
static void Main()
{
  DaysOfTheWeek days = new DaysOfTheWeek();
  foreach (string day in days)
    {
        Console.Write(day + " ");
    }
    // Output: Sun Mon Tue Wed Thu Fri Sat
    Console.ReadKey();
}

public class DaysOfTheWeek : IEnumerable
{
  private string[] days = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
  public IEnumerator GetEnumerator()
    {
        for (int index = 0; index < days.Length; index++)
        {
            // Yield each day of the week.
            yield return days[index];
        }
    }
}
```

منابع:

yield ، Iterators