

فرض کنید قبلا کلاسی بنام CollectionClass را داشته‌اید که در آن یک آرایه از نوع String[] تعریف کرده‌اید. همچنین n تا کلاس هم دارید که از آرایه‌ی تعریف شده‌ی در CollectionClass استفاده می‌کنند. تا اینجا مشکلی نیست. مشکل زمانی شروع می‌شود که متوجه می‌شوید دیگر این آرایه کارایی ندارد و باید آن را با List<string> جایگزین کنید. واضح است که نمی‌توانید همه کلاس‌هایی را که از CollectionClass استفاده کرده‌اند، بیابید و آنها را تغییر دهید؛ چرا که شاید برخی از کلاس‌ها اصلا در دسترس شما نباشند یا هر دلیل دیگری.

راهگشای این مشکل، استفاده از الگوی طراحی Iterator است. در این الگو، باید کلاس CollectionClass ابتدا واسط IEnumerable را پیاده سازی نماید. این واسط متدی بنام GetEnumerator دارد که می‌توان به کمک آن، درون آرایه یا هر نوع کالکشن دیگری حرکت کرده و آیتم‌های آن را برگرداند. ([مطالعه بیشتر](#))

اول این الگو را پیاده سازی می‌کنیم و در ادامه توضیح می‌دهیم که چگونه مشکل ما را حل میکند:

ابتدا باید کلاس CollectionClass واسط IEnumerable را پیاده سازی نماید. در ادامه بدنه متد GetEnumerator را می‌نویسیم:

```
public class CollectionClass : IEnumerable
{
    private string[] mySet = { "Array of String 1", "Array of String 2", "Array of String 3" };
    public IEnumerator GetEnumerator()
    {
        //return arrayStrings.GetEnumerator();
        foreach (var element in mySet )
        {
            yield return element;
        }
    }
}
```

در اینجا یک آرایه رشته‌ای را بنام mySet تعریف کرده‌ایم و مقادیر مختلفی را در آن قرار داده‌ایم. سپس در متد GetEnumerator اعضای این آرایه را خوانده و return می‌کنیم. ([yield چیست؟](#))

وقتی از این کلاس می‌خواهیم استفاده کنیم، داریم:

```
CollectionClass c = new CollectionClass();
foreach (var element in c)
{
    Console.WriteLine(element);
}
```

در این حالت مهم نیست که مجموعه‌ی مورد نظر، آرایه هست یا هر نوع کالکشن دیگری. لذا وقتی بخواهیم نوع mySet را تغییر دهیم، نگران نخواهیم بود؛ چراکه فقط کافی‌است کلاس CollectionClass را تغییر دهیم. بصورت زیر:

```
public class CollectionClass : IEnumerable
{
    //private readonly string[] arrayStrings = { "Array of String 1", "Array of String 2", "Array of String 3" };
    private List<string> mySet= new List<string>() { "Array of String 1", "Array of String 2", "Array of String 3" };
    public IEnumerator GetEnumerator()
    {
        foreach (var element in mySet )
        {
            yield return element;
        }
    }
}
```