04

O Laço Foreach

Transcrição

Veremos agora como a instrução `foreach` trabalha para selecionar elementos de determinada coleção.

Declararemos uma coleção de meses:

```
class Program
{
    static void Main(strin[] args)
    {
            var meses = new List<string>
            {
                    "Janeiro", "Fevereiro", "Março",
                    "Abril", "Maio", "Junho",
                    "Julho", "Agosto", "Setembro",
                    "Outubro", "Novembro", "Dezembro"
            };

                foreach (var mes in meses)
                {
                        Console.WriteLine(mes);
                }
        }
}
```

Ao executarmos a aplicação, percebemos que foram impressos cada um dos elementos da coleção.

Veremos a seguir, em detalhes, como a instrução `foreach` faz para exibir cada um dos elementos de uma coleção.

Clicaremos sobre a classe `List` e utilizaremos o comando "Alt + F12". Feito isso, poderemos observar a sua definição.

Ela implementa a interface `IEnumerable`, que é fundamental para a instrução.

Utilizando o atalho "Alt + F12" sobre esta interface vemos que, dentro dela, há um método chamado `GetEnumerator()`, que por sua vez retorna um `IEnumerator<T>`.

Voltaremos a observar a definição da classe `List`.

Nela, vemos que há um enumerador que é um `struct`, e tem uma propriedade `Current` - que é o elemento atual de uma varredura com a instrução `foreach` -, e um método `MoveNext`, que é utilizado para mover o ponteiro para o próximo elemento desta varredura.

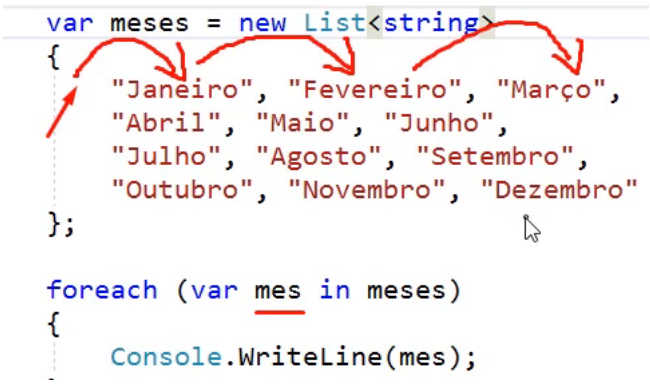
Retornaremos ao nosso código, para vermos o funcionamento disto na prática.

Temos um laço `foreach` e o enumerador será posicionado no índice "-1", que está antes do elemento de índice "0". Está, portanto, fora da faixa do array, mas, assim que o laço passa a pegar elemento por elemento, ele utilizará o método `MoveNext`.

Este método será movido para a primeira posição, ou seja, o índice "0". Com isso, a variável mês, que está dentro do laço, assumirá o valor "janeiro".

Assim que passa pelo código, dentro do `foreach`, novamente o `MoveNext` passará para o próximo elemento, ou seja, "fevereiro".

Isso seguirá até que se atinja o último elemento.



Tentaremos então trocar o valor da variável `mes`, para que passe a ser escrito em letras maiúsculas.

```
foreach (var mes in meses)
{
        mes = mes.ToUpper();
        Console.WriteLine(mes);
}
```

Ao fazermos isso, o Visual Studio aponta um erro, porque não podemos modificar a variável de interação do `foreach`, ou seja, `mes`. Isto é uma proteção da instrução, evitando que hajam modificações sem sentido.

A seguir, tentaremos mudar o valor diretamente na coleção de meses, dentro do laço `foreach`.

```
foreach (var mes in meses)
{
        meses[0] = meses.ToUpper();
        Console.WriteLine(mes);
}
```

Ao executarmos a aplicação, surgirá uma caixa de diálogo, indicando que há um erro:

"Exceção sem tratamento: (...) Coleção foi modificada; talvez a operação de enumeração não seja executada.

Isto acontece, novamente, como uma forma de proteção, não permitindo a modificação da coleção que está sendo enumerada.

Isto é feito por meio de um código, dentro do indexador, que verifica a versão da coleção. Ou seja, a cada modificação, a versão interna de uma coleção - que é o número - é modificada.

A primeira vez que o código entra no laço `foreach` ele armazena qual é a versão da coleção naquele ponto, se dentro do laço a coleção for modificada, foi alterado também o número da versão.

Portanto, a cada vez que passar pela coleção, haverá a verificação do número anterior com o atual. Caso haja diferença, será lançada uma exceção, para proteção.

Trocaremos o tipo de coleção de meses, para um array de strings.

```
var meses = new string []
                {
                        "Janeiro", "Fevereiro", "Março",
                        "Abril", "Maio", "Junho",
                        "Julho", "Agosto", "Setembro",
                        "Outubro", "Novembro", "Dezembro"
                };

                foreach (var mes in meses)
                {

                        meses[0] = meses[0].ToUpper();
                        Console.WriteLine(mes);
                }
```

Para ver o resultado, executaremos a aplicação.

Veremos que foram impressos todos os meses, de janeiro até dezembro.

Não foi identificado nenhum erro desta vez, porque o laço `foreach` trabalha com arrays de uma forma diferente. Ele não trabalha com o enumerador da classe array, mas sim transforma em um laço `for` .

Como se estivesse realizando a seguinte operação:

```
                foreach (var mes in meses)
                {

                        meses[0] = meses[0].ToUpper();
                        Console.WriteLine(mes);
                }

                for (int i = 0; i < lenght; i++)
```

Em que temos um laço `for` , e, para cada índice dentro do tamanho do array, no caso `meses` , ele armazenará uma variável `var mes = meses[i]` e continuará fazendo o laço, do modo como ele existe dentro do `foreach` .

Isto é a implementação, que o compilador cria, quando encontra uma instrução `foreach` dentro de um array.

Portanto, vimos um pouco sobre a instrução `foreach` , e como ela aciona o enumerador de um `IEnumerable` , para poder selecionar os elementos. Também vimos qual é a diferença do laço `foreach` para uma lista, e para um array.