

Consultando sequências de elementos

Transcrição

Começando deste ponto? Você pode fazer o [download \(https://github.com/marcelooliveira/csharp-collections-2/archive/08e64b2f995a12120794b6b3c9fa43f350a73bca.zip\)](https://github.com/marcelooliveira/csharp-collections-2/archive/08e64b2f995a12120794b6b3c9fa43f350a73bca.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Neste vídeo, trabalharemos com alguns problemas relacionados a obtenção de sequências de elementos de coleções.

A partir da criação de um novo projeto *console application*, utilizaremos a mesma lista de meses dos vídeos anteriores, bem como a mesma classe `class Mes : IComparable`.

No primeiro problema, iremos isolar o primeiro trimestre. Temos os 12 meses do ano, e queremos selecionar apenas os três primeiros.

Utilizaremos o método "Take" para obter elementos, que em tradução para o Português é a palavra "pegar".

Ele exige um parâmetro, que é a quantidade de elementos, ou seja, neste caso, três elementos. Este método retornará um `IEnumerable`.

Para imprimirmos o resultado no console, utilizaremos o método `foreach`.

```
//Pegar o primeiro trimestre
var consulta = meses.Take(3);
foreach (var item in consulta)
{
    Console.WriteLine(item);
}
Console.WriteLine();
```

Executaremos a aplicação, com o atalho "Ctrl + F5".

No console, serão exibidos os meses de janeiro, fevereiro e março, ou seja, os três primeiros meses do ano.

Em seguida, isolaremos os meses subsequentes ao primeiro trimestre.

Ao montar a consulta, precisamos ter em mente que será necessário "pular" os três primeiros meses. Para isso, utilizaremos o método Linq "Skip", cuja tradução seria "pular", em Português.

Esta nova consulta será denominada "consulta2".

```
//Pegar os meses depois do primeiro trimestre
var consulta2 = meses.Skip(3);
foreach (var item in consulta2)
{
    Console.WriteLine(item);
}
```

```
}  
Console.WriteLine();
```

Executaremos o programa.

No console, vemos primeiro a listagem que fizemos no exemplo anterior e, em seguida, a lista dos meses após o primeiro trimestre, evidenciando a consulta que acabamos de criar.

A seguir, faremos com que sejam exibidos somente os meses do terceiro trimestre.

Interpretando o problema, vemos que, para isso, precisamos isolar os meses de julho, agosto e setembro. Precisaremos "pular" o primeiro semestre.

```
//Pegar os meses do terceiro trimestre  
var consulta3 = meses.Skip(6).Take(3);  
foreach (var item in consulta3)  
{  
    Console.WriteLine(item);  
};  
Console.WriteLine();
```

Para vermos o resultado, executaremos a aplicação.

Podemos observar que foram impressos os meses desejados.

A seguir, isolaremos todos os meses do ano, até o primeiro que inicie com a letra "S".

Para isso, não utilizaremos o método `Take`. Este método só é capaz de receber números inteiros como parâmetro, que representam a quantidade de elementos que queremos selecionar.

Utilizaremos, em vez disso, o método `TakeWhile`, que nos permite **obter** algo, **enquanto** alguma coisa for verdadeira.

Como parâmetro, faremos com que não sejam selecionados meses que iniciem com a letra "S". Como é um *predicate*, ele pode ser representado com uma expressão lambda, como veremos abaixo.

O `StartsWith` indicará se uma string inicia, ou não, com uma determinada sequência de caracteres. No caso, a letra "S".

```
//Pegar os meses até que o mês comece com a letra 'S'  
var consulta4 = meses.Take(m => !m.Nome.StartsWith("S"));  
foreach (var item in consulta4)  
{  
    Console.WriteLine(item);  
}  
Console.WriteLine();
```

Executaremos a aplicação.

Como podemos observar, foram impressos os nomes dos meses que não iniciam com a letra "S", até aquele que inicia, ou seja, a partir de setembro nenhum nome foi impresso.

Por fim, isolaremos os meses até que haja um que comece com a letra "S".

Criaremos uma nova consulta `consulta5`, que utilizará a lista de meses, como base para utilizarmos os filtros.

Para ignorar certos meses, não utilizaremos método `Skip` porque, como vimos com o método `Take`, estes só aceitam números inteiros como parâmetros.

Assim, utilizaremos o método `SkipWhile`, que pulará informações considerando uma condição verdadeira.

A expressão lambda, neste caso, será a mesma do problema anterior.

```
//Pular os meses até que o mês comece com a letra 'S'  
var consulta5 = meses.SkipWhile(m => !m.Nome.StartsWith("S"));  
foreach (var item in consulta5)  
{  
    Console.WriteLine(item);  
}  
Console.WriteLine();
```

Executaremos a aplicação.

No console, foram impressos todos os meses, a partir de setembro, que é o único que inicia com a letra "S".

Com isso, aprendemos a utilizar operadores Linq para obter sequências de elementos de coleções do .NET Framework.