

Curso C# Completo

Programação Orientada a Objetos + Projetos

Capítulo: Tópicos Especiais em C# - PARTE 2

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Extension methods

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Extension methods

- São métodos que estendem a funcionalidade de um tipo, sem precisar alterar o código fonte deste tipo, nem herdar desse tipo
- Como fazer um extension method?
 - Criar uma classe estática
 - Na classe, criar um método estático
 - O primeiro parâmetro do método deverá ter o prefixo `this`, seguido da declaração de um parâmetro do tipo que se deseja estender. Esta será uma referência para o próprio objeto.

Demo 1

Vamos criar um *extension method* chamado *"ElapsedTime()"* no struct `DateTime` para apresentar um objeto `DateTime` na forma de tempo decorrido, podendo ser em horas (se menor que 24h) ou em dias caso contrário. Por exemplo:

```
DateTime dt = new DateTime(2018, 11, 16, 8, 10, 45);  
Console.WriteLine(dt.ElapsedTime());
```

"4.5 hours"

"3.2 days"

```

using System.Globalization;

namespace System
{
    static class DateTimeExtensions
    {
        public static string ElapsedTime(this DateTime thisObj)
        {
            TimeSpan duration = DateTime.Now.Subtract(thisObj);

            if (duration.TotalHours < 24.0)
            {
                return duration.TotalHours.ToString("F1", CultureInfo.InvariantCulture) + " hours";
            }
            else
            {
                return duration.TotalDays.ToString("F1", CultureInfo.InvariantCulture) + " days";
            }
        }
    }
}

```

Demo 2

Vamos criar um *extension method* chamado "Cut(int)" na classe String para receber um valor inteiro como parâmetro e gerar um recorte do string original daquele tamanho. Por exemplo:

```

String s1 = "Good morning dear students!";
Console.WriteLine(s1.Cut(10));

```

"Good morni..."

```
namespace System
{
    static class StringExtensions
    {
        public static string Cut(this string thisObj, int count)
        {
            if (thisObj.Length <= count)
            {
                return thisObj;
            }
            else
            {
                return thisObj.Substring(0, count) + "...";
            }
        }
    }
}
```