

# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

Estrutura de uma aplicação C#

***Prof.***

***Samuel Ribeiro (Samuka)***



<http://www.processware.com.br>

**E-MAIL:** professor.samuka@gmail.com 2004



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

keyword

Namespace

```
using System;  
using System.Web;
```

A keyword "using" importa o Namespace 'System' para nosso programa

```
public partial class Ex1 : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
        Response.Write("Olá Mundo C#");  
    }  
}
```

Um Namespace é um agrupamento lógico de funcionalidades.

Cada espaço de nome na .Net Framework é essencialmente uma coleção de bibliotecas.

As classes são organizadas em Namespaces.

# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

modificador de acesso

```
using System;
using System.Web;

public partial class Ex1 : System.Web.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Response.Write("Olá Mundo C#");
    }
}
```

**public** : Estes atributos e métodos são sempre acessíveis em todos os métodos de todas as classes. Este é o nível menos rígido de encapsulamento, que equivale a não encapsular.

**private** : Estes atributos e métodos são acessíveis somente nos métodos (todos) da própria classe. Este é o nível mais rígido de encapsulamento.

**protected** : Estes atributos e métodos são acessíveis no pacote, nos métodos da própria classe e suas subclasses, o que será visto em Herança.

**Internal** : É um modificador de acesso para tipos internos ou membros acessíveis somente nos arquivos no mesmo conjunto.



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

partial class

```
using System;  
using System.Web;  
  
public partial class Ex1 : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
        Response.Write("Olá Mundo C#");  
    }  
}
```

partial class:

Partial class nada mais do que uma característica que possibilita dividir seus tipos tais como classes e interfaces através de múltiplos arquivos.

Quando se é compilada uma partial class (classe parcial) no VS 2005 o compilador simplesmente agrupa todas os arquivos da classe parcial e o torna uma entidade única.

# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Ex1.aspx.cs" Inherits="Ex1" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
```

```
<head runat="server">
```

```
<title>Untitled Page</title>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

**Page Language="C#":**

Indica qual linguagem a ser utilizada

**AutoEventWireup="true":**

faz com que a página de ASP.NET associe eventos e métodos de maneira automática.

**CodeFile="Ex1.aspx.cs":**

Indica qual o arquivo associado que contém o código

**Inherits="Ex1":**

Indica quais as classes este projeto herda automaticamente



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Ex1.aspx.cs" Inherits="Ex1" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
```

```
<head runat="server">
```

```
<title>Untitled Page</title>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

<html xmlns="http ... especifica o namespace usado para as tags customizadas em um documento Html

<!DOCTYPE html PUBLIC ... :

De acordo com os padrões *HTML* e *XHTML*, um DOCTYPE informa ao *validador* qual é a versão do *HTML*, ou *XHTML*, que está sendo usada.

Isto valida sua marcação e seus *CSS's*

# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

**class**

...

**class Hello**

{

...

}

A lógica de um programa em C# deve sempre estar dentro de uma definição de tipo nesse caso uma classe. Veja que a definição de nossa classe começa na abertura da chave({) e termina em seu fechamento(}).



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Comentários

// Comentário de uma linha apenas

/\* Comentário  
de várias  
linhas \*/



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Geração de Documentação XML

```
using System;
/// <summary>
/// A classe LeitorDeTexto lê uma frase do utilizador
/// </summary>
public class LeitorDeTexto
{
    /// <remarks>
    /// Utiliza-se um I/O baseado em consola
    /// Para mais informação sobre sobre o WriteLine, ver
    /// <seealso cref="System.Console.WriteLine()"/>
    /// </remarks>
    public static void Main()
    {
        string input = Console.ReadLine();
        Console.WriteLine("A linha escrita foi: {0}", input);
    }
}
```



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Tratamento de Exceções

```
using System;

class Hello
{
    public static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Hello, World");
        }
        catch (Exception e)
        {
            Console.WriteLine("Exception at {0}", e.StackTrace);
        }
    }
}
```



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Variáveis na linguagem C#.

Na linguagem C# as variáveis estão agrupadas em algumas categorias como:

**Static:** Existe apenas uma única cópia desta variável para todas as instancias de uma classe.

**Instance:** existe uma cópia para cada instancia de uma classe.