

DESENVOLVIMENTO DE PROJETOS COM C#

GILBERTO S. T. JUNIOR
GILBERTOTAKEYA@YAHOO.COM.BR

Métodos

- São rotinas criadas pelo desenvolvedor de software, essas rotinas serão utilizadas mais de 1x no projeto.
- Tipos:
 - Retornam valor ou não retornam valor.
- Vantagens:
 - **Modularização** das informações, com isso quando ocorrer uma alteração será realizado uma alteração em um só lugar com isso **evitando códigos repetidos**

Métodos

□ Exemplo

escopo tipo NomeMetodo(<lista de argumento>)

Onde: escopo – public ou protected ou private

 tipo – int, double, float, string

Exemplo:

```
public void validarCNPJCPF( string CNPJCPF )
```

```
{
```

```
    if( !( CNPJCPF.Text.Length == 11 ) && !(CNPJCPF.Text.Length == 14 ) )
```

```
        throw new ApplicationException(“CNPJ/CPF inválido”);
```

```
}
```

Métodos

□ Lista de argumentos

- **Passagem por parâmetro:** Parametro de entrada. Ocorre a copia de valores, ou seja o valor da variavel não poderá ser alterado.

Ex:

```
validarCNPJCPF( txtCNPJCPF.Text); // Copia o valor que esta  
//no objeto de tela txtCNPJCPF e envia para a função de  
//validar CNPJCPF
```

```
public void validarCNPJCPF( string CNPJCPF )// CNPJCPF  
//recebe algum valor por parâmetro
```

Métodos

- Passagem por referência

É um parametro de entrada e saída.Ou seja o valor pode ser apenas enviado como parametro como pode ser alterado.

Ex:

```
validarCNPJCPF( ref txtCNPJCPF.Text); // Envia o endereço da variavel
```

```
public void validarCNPJCPF( ref string CNPJCPF )
```

```
// CNPJCPF recebe o endereço da onde esta localizado txtCNPJCPF e caso  
ocorra alguma alteração no CNPJCPF automaticamente será alterado o texto  
do campo txtCNPJCPF
```

Métodos

- Out

- É um parâmetro de saída. Sempre que um parâmetro out é passado, somente a referência da variável é passada para o método.

```
// Cria a nossa classe de exemplo
class MinhaClasse
{
    public void Metodo(out int v1)
    {
        v1 = 100;
        Console.WriteLine("Valor em Metodo = " + v1);
    }
}

// classe para execução do aplicativo
class Class1
{
    static void Main()
    {
        // cria uma nova instância da classe MinhaClasse
        MinhaClasse obj = new MinhaClasse();
        int valor = 5;
        obj.Metodo(out valor);
        Console.WriteLine("Valor na Main = " + valor);
    }
}
```

Métodos

- Params

É um array unidimensional. Isso possibilita ter argumentos de tamanho variavel

```
// Cria a nossa classe de exemplo
class MinhaClasse
{
    public int Soma(params int[] v1)
    {
        int v2 = 0;
        foreach(int v in v1)
        {
            v2 = v2 + v;
        }
        return v2;
    }
}

// classe para execução do aplicativo
class Class1
{
    static void Main()
    {
        // cria uma nova instância da classe MinhaClasse
        MinhaClasse obj = new MinhaClasse();
        Console.WriteLine(obj.Soma(10, 20, 30));
        Console.WriteLine(obj.Soma(10, 20, 30, 40, 50));
    }
}
```