06

Números por Extenso

Transcrição

Vamos ver como trabalhar com números por extenso utilizando o C#.

Adicionaremos um novo projeto do tipo **Console App (.NET Framework)** chamado de "NumerosPorExtenso". Deixaremos esse projeto como um projeto inicial ("Set as StartUp Project").

Primeiro, criaremos uma variável com um valor qualquer:

```
static void Main(string[] args)
{
    double valor = 75;
}
```

Como vamos conseguir imprimir por extenso? Lembrando que vamos continuar usando a biblioteca da Caelum **Stella**. Clicando com o botão direito do mouse no projeto > Manage NuGet Packages... > Browse "caelum". Depois, clicaremos em *Install* e em *I Accept*.

Muito bem, agora já somos capazes e utilizar essa biblioteca para imprimir números por extenso.

Colocaremos o valor `75` , em um outro tipo que existe na biblioteca da Caelum, o tipo **Numero**.

```
static void Main(string[] args)
{
    double valor = 75;
    new Numero(valor);
}
```

Agora chamaremos o método para pegar o extenso desse valor:

```
double valor = 75;
new Numero(valor).Extenso();
```

Armazenaremos a string em uma variável local.

```
double valor = 75;
string extenso = new Numero(valor).Extenso();
```

E também, queremos imprimir no console do Debug:

```
static void Main(string[] args)
{
    double valor = 75;
    string extenso = new Numero(valor).Extenso();
```

```
        Debug.WriteLine(extenso);
    }
```

E o resultado é `setenta e cinco`. Vamos trabalhar com números um pouco maiores agora...

```
static void Main(string[] args)
    {
        double valor = 75;
        string extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);

        valor = 1532987;
    }
```

Temos que declarar um novo `Numero()` para salvar o valor que acabamos de criar, em seguida, chamaremos o método extenso. Armazenaremos o resultado na variável "extenso". Imprimiremos o resultado no console do Debug.

```
static void Main(string[] args)
    {
        double valor = 75;
        string extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);

        valor = 1532987;
        extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);
    }
```

Aqui está o nosso resultado:

```
um milhão, quinhentos e trinta e dois mil novecentos e oitenta e sete
```

Poderíamos também pegar esse valor e imprimir em formato de moeda. Então, vamos adaptar o último Debug para imprimir como se esse valor estivesse em reais. Utilizaremos uma outra classe chamada `MoedaBRL`.

Primeiro, precisamos criar uma nova instância da `MoedaBRL()` passando um "valor" e pegando o método `.Extenso()`. Em seguida, guardaremos esse Extenso em uma outra variável. Por fim, imprimiremos o valor no console do Debug.

```
static void Main(string[] args)
    {
        double valor = 75;
        string extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);

        valor = 1532987;
        extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);

        string extensoBRL = new MoedaBRL(valor).Extenso();
        Debug.WriteLine(extensoBRL);
    }
```

Teremos como resultado:

```
um milhão, quinhentos e trinta e dois mil novecentos e oitenta e sete reais
```

Como reparamos, não precisamos adicionar a palavra "reais" na mão, pois essa classe já faz esse trabalho para nós.

E se o nosso valor conter **centavos**? Adicionaremos *89 centavos*:

```
static void Main(string[] args)
    {
        double valor = 75;
        string extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);

        valor = 1532987;
        extenso = new Numero(valor).Extenso();
        Debug.WriteLine(extenso);

        string extensoBRL = new MoedaBRL(valor).Extenso();
        Debug.WriteLine(extensoBRL);

        valor = 1532987.89;
        extensoBRL = new MoedaBRL(valor).Extenso();
        Debug.WriteLine(extensoBRL);
    }
```

Rodando a aplicação, temos esse resultado:

```
um milhão, quinhentos e trinta e dois mil novecentos e oitenta e sete reais e oitenta e nove cer
```

Vimos como imprimir números grandes e pequenos por extenso, e no formato de moedas, com e sem os centavos. Esses formatos são muito úteis para documentos legais, contratos, escrituras de terreno, e por aí vai.