UNIDADE 13 – STRING

Nessa unidade veremos resumo bem objetivo de como usar os recursos da classe String.

SUMÁRIO

1 STRING

A classe String é utilizada em praticamente todas as aplicações .NET. Consequentemente, os programadores .NET devemconhecer bem o funcionamento dela. A documentação da classe String pode ser consultada na url http://msdn.microsoft.com/pt-br/library/system.string(v=vs.110).aspx

1.1 IMUTABILIDADE

Umacaracterística fundamental dos objetos da classe String é que eles são imutáveis. Emoutras palavras, o conteúdo de uma string não altera.

Alguns métodos das strings podem dar a impressão errada de que o conteúdo do objeto será alterado. Por exemplo, o método ToUpper() que é utilizado para obter uma string comletras maiúsculas. Esse método não altera a string original, ele cria uma nova string com o conteúdo diferente.

```
1  string nome = "Thiago Sartor";
2
3  nome.ToUpper();
4
5  // imprime Rafael Cosentino
6  Console.WriteLine(nome);
```

Tabela 1: Pegadinha. . .

```
1  string nome = "Thiago Sartor";
2
3  string nomeAlterado = nome.ToUpper();
4
5  // imprime THIAGO SARTOR
6  Console.WriteLine(nomeAlterado);
```

Tabela 2: Guardando o resultado do ToUpper()

1.2 MÉTODOS E PROPRIEDADES

Todos os métodos e propriedades da classe String podem ser consultados na url http://msdn.microsoft.com/en-us/library/system.string.aspx. Discutiremos aqui o funcionamento dos principais métodos e propriedades dessa classe.

1.2.1 Length

A propriedade Length armazena a quantidade de caracteres de uma string.

```
1  string nome = "Academia do Programador";
2
3  // imprime 23
4  Console.WriteLine(nome.Length);
```

Tabela 3: Length

1.2.2 ToUpper()

O método ToUpper() é utilizado para obter uma cópia de uma string com letras maiúsculas.

```
1  string nome = "Alexandre Rech";
2
3  string nomeAlterado = nome.ToUpper();
4
5  // imprime ALEXANDRE RECH
6  Console.WriteLine(nomeAlterado);
```

Tabela 4: ToUpper()

1.2.3 ToLower()

O método ToLower() é utilizado para obter uma cópia de uma string com letras minúsculas.

```
1  string nome = "Alexandre Rech";
2
3  string nomeAlterado = nome.ToLower();
4
5  // imprime alexandre rech
6  Console.WriteLine(nomeAlterado);
```

Tabela 5: ToLower()

1.2.4 Trim()

O método Trim() é utilizado para obter uma cópia de uma string sem os espaços em braco do início e do final.

```
1  string nome = "          Formação Desenvolvedor . NET";
2
3  string nomeAlterado = nome.Trim();
4
5  // imprime '' Formação Desenvolvedor .NET ''
6  Console.WriteLine(nomeAlterado);
```

Tabela 6: Trim()

1.2.5 Split (char[])

O método Split(char[]) divide uma string em várias de acordo com um delimitador e devolve um array com as strings obtidas.

```
1  string[] cursos = texto.Split(new char[] {','});
2
3  // imprime NDDA
4  Console.WriteLine(cursos[0]);
5  // imprime NDDB
6  Console.WriteLine(cursos[1]);
```

Tabela 7: Split(char[])

1.2.6 Replace()

O método Replace() cria uma cópia de uma string substituindo “pedaços” internos por outro conteúdo.

```
1  string texto = " Curso de CSharp da NDD , Curso de ASP . NET MVC da NDD ";
2
3  string textoAlterado = texto.Replace(" Curso ", " Treinamento ");
4
5  // imprime Treinamento de CSharp da NDD , Treinamento de ASP . NET MVC da NDD
6  Console.WriteLine(textoAlterado);
```

Tabela 8: Replace ()

1.2.7 ToCharArray()

O método ToCharArray() converte strings para arrays de caracteres. Ele é chamado em uma cadeia e retorna uma nova matriz de char. Você pode manipular essa matriz no local, o que você não pode fazer com uma corda. Isso faz com que muitas otimizações de desempenho possível.

```
1  string palavra = "THIAGO SARTOR";
2
3  char[] letras = palavra.ToCharArray();
4
5  foreach (var letra in letras)
6  {
7      Console.WriteLine(letras);
   }
```

Tabela 9:ToCharArray()

1.3 Exercícios de Fixação

1) Crie um projeto no chamado **Unidade 13** e uma pasta **ExercicioFixacao**.

2) Teste a imutabilidade das strings.

```
1  public class TestaImutabilidade
2  {
3      public static void Main()
4      {
5          string nome = "Thiago Sartor";
6
7          string nomeAlterado = nome.ToUpper();
8
9          // imprime Rafael Cosentino
10         Console.WriteLine(nome);
11
12         // imprime RAFAEL COSENTINO
13         Console.WriteLine(nomeAlterado);
14     }
15 }
```

Tabela 7: TestaImutabilidade.cs

1.4 Exercícios Complementares

1) Crie uma pasta **ExerciciosComplementares** na **Unidade 13**.

2) Faça um vetor de string com o nome e sobrenome de todos do laboratório. E implemente as seguintes funcionalidades:

- Busque a quantidade de letras de cada nome imprima na tela;
- Mostre todos os nomes em letra maiúscula e minúscula;
- Tire os espaços entre o nome e sobrenome;
- Substitua a primeira letra pelo caracter especial >- “#”;
- Substitua todas as vogais de 15 nomes por ‘*’.

Bons Estudos!