

## Cap. 9 - Texto

Aprenda a Programar com C# - 2ª Edição (2020)

Edições Sílabo

<https://bit.ly/36nyKFm>

António Trigo, Jorge Henriques  
{antonio.trigo,jmvhenriques}@gmail.com

1 de outubro de 2020

Strings

Strings - Exercícios

Passagem de parâmetros pela linha de comando

## Definição

- ▶ As strings são sequências de caracteres, ex.: "Bom dia";
- ▶ Em C# utiliza-se a classe String para armazenar a sequência de caracteres. A **keyword** string utilizada na declaração de strings é um **alias** para a classe String.
- ▶ A string é um tipo referenciado e não uma variável do tipo primitivo como o int ou o float.
- ▶ Em C# ao contrário do C as strings são imutáveis, ou seja, uma vez definidas não podem ser alteradas.
- ▶ Em C# ao contrário do C um string não termina com o carácter '\0', podendo conter uma série de caracteres '\0';
- ▶ A propriedade Length, à semelhança dos vetores, dá-nos o tamanho da string.
- ▶ Declaração:

```
string nome_da_string;
```

# Inicialização

- ▶ Em C# as strings podem ser inicializados da seguinte forma:

```
string nome_da_string =  
    'carater1carater2carater2...caratern'  
char[dimensao] nome_do_vetor =  
    {'carater1','carater2','...','caratern'};  
string nome_da_string = new string(new  
    char[]{'carater1 ','carater2 ','...','caratern'});
```

- ▶ Exemplo da mesma string frase inicializada de duas formas diferentes:

```
string frase = 'Bom dia\n';  
char[] char_frase = {'B','o','m','  
    ','d','i','a','\n'};  
string frase = new string(char_frase);
```

# Utilização

- ▶ Dado serem um vetor de caracteres as strings podem ser escritas no ecrã tal como os vetores:

```
static void Main(string[] args)
{
    string frase = "Bom dia\n";
    for (int i = 0; i < frase.Length; i++)
        Console.Write("{0}", frase[i]);
}
```

- ▶ Não obstante, também se pode escrever a string utilizando o nome do vetor:

```
static void Main(string[] args)
{
    string frase = "Bom dia\n";
    Console.Write(frase);
}
```

## Comparação de strings

- ▶ Para saber se duas strings são iguais utiliza-se o símbolo “==” ou método (função) Equals
- ▶ O símbolo “==” faz uso do método Equals.

```
static void Main(string[] args)
{
    string frase1 = "C#";
    string frase2 = "c#";
    Console.WriteLine(frase1.Equals("C#"));
    Console.WriteLine(frase1.Equals(frase2));
    Console.WriteLine(frase1 == "C#");
    Console.WriteLine(frase1 == frase2);
}
```

## Comparação de strings

- ▶ Para comparar alfabeticamente duas ou mais strings utiliza-se o método `CompareTo`, que devolve os valores apresentados no exemplo.

```
static void Main(string[] args)
{
    string strA = "A";
    string strB = "B";
    Console.WriteLine(strA.CompareTo(strB)); // -1
    Console.WriteLine(strB.CompareTo(strA)); // 1
    Console.WriteLine(strB.CompareTo(strB)); // 0
}
```

## Mostrar a primeira palavra de uma frase introduzida pelo utilizador

```
static void Main(string[] args)
{
    string frase;
    int i = 0;

    Console.Write("Insira uma frase: ");
    frase = Console.ReadLine();
    while (frase[i] != ' ')
    {
        Console.Write(frase[i]);
        i++;
    }
    Console.WriteLine();
}
```



## Remover todos os espaços de uma string introduzida pelo utilizador

```
static void Main(string[] args){  
    string frase1;  
    int i = 0, j = 0;  
    Console.Write("Insira uma frase: ");  
    frase1 = Console.ReadLine();  
    char[] charsFrase = new char[frase1.Length];  
    for (i = 0; i < frase1.Length; i++)  
    {  
        if (frase1[i] != ' ' )  
        {  
            charsFrase[j] = frase1[i];  
            j++;  
        }  
    }  
    frase1 = new string(charsFrase);  
    Console.WriteLine(frase1);  
}
```

## Substituir todos os 'a' de uma string por um '\*' (utilizando o método ToArray())

```
static void Main(string[] args) {  
    string frase1;  
    int i = 0, j = 0;  
    Console.Write("Insira uma frase: ");  
    frase1 = Console.ReadLine();  
    char[] charsFrase = frase1.ToArray();  
    // Tambem pode ser utilizado o ToCharArray()  
    // char[] charsFrase = frase1.ToCharArray();  
    for (i = 0; i < charsFrase.Length; i++)  
    {  
        if (charsFrase[i] == 'a')  
        {  
            charsFrase[i] = '*';  
        }  
    }  
    Console.WriteLine(charsFrase);  
}
```

## Outras funções

- ▶ A classe `String` disponibiliza um conjunto de métodos para trabalhar com strings, dos quais destacamos:
  - ▶ **`String.Concat(string string1, string string2)`**, para concatenação de strings; Também se podem utilizar os operadores “+” e “+=”;
  - ▶ **`ToUpper()`**, para colocar todos os caracteres de uma string em letra maiúscula;
  - ▶ **`ToLower()`**, para colocar todos os caracteres de uma string em letra minúscula;
  - ▶ **`IndexOf(string string1)`**, pesquisar a `string1` numa outra `string1`; Devolve a posição onde começa a string a pesquisar (`string1`) na string pesquisada.
  - ▶ **`Substring(int inicio, int comprimento)`**, retira uma string de outra string, a partir de uma determinada posição e com um determinado comprimento.

## Outras funções

- ▶ Mais informações sobre a classe String podem ser obtidas online: [https://msdn.microsoft.com/en-us/library/system.string\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string(v=vs.110).aspx)

## Definição

- ▶ A linguagem C# permite a passagem de parâmetros pela linha de comando para um programa a executar;
- ▶ Para isso temos de definir a função principal da seguinte forma:

```
public static void Main(string[] args){  
  
}
```

- ▶ Em que args é uma variável do tipo vetor que contém os parâmetros passados pelo linha de comando.
- ▶ No C# ao contrário do C o primeiro parametro (args[0]) não corresponde ao nome do programa, mas sim ao primeiro parâmetro.

## Utilização

- ▶ Para utilizar esta funcionalidade basta invocar nome do programa a executar e os respetivos parâmetros separados por espaços: `linhadecomando> nome_programa arg1 arg2 ... argn`.
- ▶ Exemplo:

```
static void Main(string[] args){  
    Console.WriteLine("A soma de {0} com {1} e  
        {2}.", args[0],  
        args[1], Convert.ToInt32(args[0]) +  
        Convert.ToInt32(args[1]));  
}
```

```
c:\> soma 2 3
```

```
c:\> A soma de 2 com 3 e 5.
```