



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Utilizando objetos parte 1

Curso Técnico Integrado de Informática
Fundamentos de Programação

Objetos

- São como micro programas sendo capazes de armazenar e processar dados
- Os dados são armazenados em variáveis conhecidas como **atributos**
- **Métodos** são funções que realizam o processamento dos dados, sejam estes atributos e/ou parâmetros
- Assim como uma variável, um objeto possui um tipo de dado conhecido como **classe**
 - Atributos e métodos de um objeto são definidos na sua classe
- Objetos de mesma classe (mesmo tipo) possuem o mesmo conjunto de atributos e métodos

Objetos em C#

- Objetos são criados a partir das classes
 - Matrizes e vetores também são objetos

```
string str = "Olá Mundo!";
```

```
int[] m = new int[10];
```

- Membros (atributos e métodos) são chamados através do operador ponto

```
str.Length;
```

```
str.ToUpper();
```

```
m.Length;
```

Exemplo 1

```
static void Main(string[] args){  
    string str = "Diga não às drogas";  
    string str2 = str.ToUpper();  
    Console.WriteLine(str2);  
}
```

Objetos e referências

- Objetos são manipulados através de referências (variáveis cujo tipo é uma classe)
- Referências podem apontar para um objeto nulo
- Um objeto é atribuído a uma referência através do operador =

```
string s1 = "IFRN";
```

```
string s2 = s1;
```

```
s1 = null;
```

```
s1.Length; //gera NullReferenceException
```

Classe String

- **String** é um tipo de dado utilizado para representar texto
- Em C#, uma string é na verdade um vetor imutável de caracteres
 - Colchetes podem ser utilizados para acessar os elementos (caracteres) da string
 - Não é possível alterar os caracteres de uma string

```
string s = "Olá Mundo!";  
Console.Write(s[2]); //imprime á  
s = "algoritmo"; //ok
```

Concatenação

- Strings podem ser concatenadas através dos operadores + e +=

```
string s1 = "Olá";  
string s2 = " Mundo!";  
Console.Write(s1 + s2); //Olá Mundo!  
Console.Write(s2 + s1); //Mundo!Olá  
s1 += s2;  
Console.Write(s1); //Olá Mundo!
```

Comparando strings

- Os operadores lógicos de igualdade e diferença podem ser utilizados na comparação de strings

```
string s1 = "Olá";  
string s2 = " Mundo!";  
Console.WriteLine(s1 == s2); //false  
Console.WriteLine(s1 != s2); //true  
Console.WriteLine(s1 == "Olá"); //true  
Console.WriteLine(s1 == "olá"); //false
```


Métodos de manipulação

- A classe **String** disponibiliza um conjunto de métodos (funções) para manipulação de strings
- Note que os métodos nunca alteram a string original, e sempre geram como resultado uma nova string
- **string Substring(int i)**: retorna uma substring que inicia no caractere localizado na posição **i** da string original

```
string s1 = "goiaba";  
string s2 = s1.Substring(3);  
Console.WriteLine(s2); //imprime aba
```

Métodos de manipulação

- **string Substring(int i, int n)**: retorna uma substring com **n** caracteres e que começa no caractere localizado na posição **i** da string original

```
string s1 = "goiaba";  
string s2 = s1.Substring(1, 3);  
Console.WriteLine(s2); //imprime oia
```

- **int IndexOf(string sub)**: retorna a posição da substring **sub** na string original. Caso **sub** não seja encontrada, retorna **-1**.

```
string s1 = "goiaba";  
int i = s1.IndexOf("aba");  
Console.WriteLine(i); //imprime 3
```

Métodos de manipulação

- **int IndexOf(string sub, int i):** como o método anterior, sendo que a busca é feita a partir do caractere na posição **i**

```
string s1 = "banana";  
int i = s1.IndexOf("ana", 2);  
Console.WriteLine(i); //imprime 3
```

- **string ToUpper():** retorna uma cópia da string com todos os caracteres em maiúsculo

```
string s1 = "banana";  
string s2 = s1.ToUpper();  
Console.WriteLine(s2); //BANANA
```

Métodos de manipulação

- **string ToLower()**: retorna uma cópia da string com todos os caracteres em minúsculo

```
string s1 = "BANANA";  
string s2 = s1.ToLower();  
Console.WriteLine(s2); //banana
```
- **string Trim()**: retorna uma cópia da string sem os espaços em branco localizados no início e no fim da string

Métodos de manipulação

- **string[] Split(char c):** “quebra” a string utilizando o caractere **c** como separador.

```
string s1 = "Bom dia";
```

```
string[] s2 = s1.Split(' ');
```

```
Console.WriteLine(s2[0]); //Bom
```

```
Console.WriteLine(s2[1]); //dia
```

Exercícios

- Utilizando objetos da classe string e seus métodos, resolva as questões a seguir:
 - 1) Faça uma função que receba uma string e retorne a quantidade de palavras da string.
 - 2) Faça uma função que receba uma frase e uma palavra e retorne a quantidade de ocorrências da palavra na frase.
 - 3) Escreva uma função que receba um texto e duas palavras e substitua todas as ocorrências da primeira palavra pela segunda palavra.
 - 4) Faça uma função que receba um nome e apresente-o da seguinte forma:

Nome: João Maria Silva

Resultado: Silva, João

Referências

- <http://msdn.microsoft.com/en-us/library/ms228362%28v=vs.80%29.aspx>
- http://www.codigofonte.net/dicas/csharp/317_manipulando-strings-no-c