

# Programação Orientada à Objetos

Aula 10

Collections / Lists

Henrique Poyatos

[henrique.poyatos@bandtec.com.br](mailto:henrique.poyatos@bandtec.com.br)

# Namespace System.Collections

- A namespace System.Collections fornece uma variedade de classes mais flexíveis e mais sofisticadas de trabalho com arrays.

- Vamos estudar uma destas classes: a classe List

- **Declaração de uma Lista:**

```
List<tipo> colecao = new List<tipo>;
```

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();
```

- Veja que neste caso não é necessário destacar o tamanho da lista como era feito com array.

# Adicionando um Item

- Para adicionar um item na coleção podemos utilizar duas funções: Add() ou Insert()

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();

//Insere item na próxima posição
numerosInteiros.Add(10);

//Insere item na próxima posição
numerosInteiros.Add(20);

Console.WriteLine("Primeiro Item: " + numerosInteiros[0]);
```

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();

//Insere item na primeira posição
numerosInteiros.Insert(0, 10);

//Insere item na primeira posição
numerosInteiros.Insert(0, 20);

Console.WriteLine("Primeiro Item: " + numerosInteiros[0]);
```

# Adicionando Itens

- Para adicionar N itens na coleção podemos utilizar a função `AddRange()`

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();
List<int> numerosPares = new List<int>();
List<int> numerosImpares = new List<int>();

//Insere Pares
numerosPares.Add(2);
numerosPares.Add(4);

//Insere números Ímpares
numerosImpares.Add(1);
numerosImpares.Add(3);

//Insere Lista Final
numerosInteiros.AddRange(numerosPares);
numerosInteiros.AddRange(numerosImpares);

Console.WriteLine("Qtd itens: " + numerosInteiros.Count);
```

# Limpando a Coleção Inteira

- Para limpar todos os itens na coleção podemos utilizar a função `Clear()`

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();
List<int> numerosPares = new List<int>();
List<int> numerosImpares = new List<int>();

//Insere Pares
numerosPares.Add(2);
numerosPares.Add(4);

//Insere números Ímpares
numerosImpares.Add(1);
numerosImpares.Add(3);

//Insere Lista Final
numerosInteiros.AddRange(numerosPares);
numerosInteiros.AddRange(numerosImpares);

numerosPares.Clear();
numerosImpares.Clear();

Console.WriteLine("Qtd itens: " + numerosInteiros.Count);
```

# Removendo um Item da Coleção

- Para remover itens da coleção podemos utilizar as funções RemoveAt() ou Remove()

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();

//Insere Números
numerosInteiros.Add(9);
numerosInteiros.Add(8);
numerosInteiros.Add(7);
numerosInteiros.Add(6);

//Remove o Item da Posição 2
numerosInteiros.RemoveAt(2);

foreach (int item in numerosInteiros)
{
    Console.WriteLine("L1 - Número: " + item);
}

//Remove o Número 9
numerosInteiros.Remove(9);

foreach (int item in numerosInteiros)
{
    Console.WriteLine("L2 - Número: " + item);
}
```

# Contando número de itens da Coleção

- Para saber quantos itens a coleção possui, podemos utilizar o .Count

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();
```

```
//Insere Números  
numerosInteiros.Add(9);  
numerosInteiros.Add(8);  
numerosInteiros.Add(7);  
numerosInteiros.Add(6);  
numerosInteiros.RemoveAt(1);
```

```
//Conta quantos itens tem na coleção  
numerosInteiros.Count // Deve mostrar 3
```

# Realizando buscas

## Método Exists()

- **Exemplo:**

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<int> list = new List<int>();
        list.Add(7);
        list.Add(11);
        list.Add(13);

        // See if any elements with values greater than 10 exist
        bool exists = list.Exists(element => element > 10);
        Console.WriteLine(exists);

        // Check for numbers less than 7
        exists = list.Exists(element => element < 7);
        Console.WriteLine(exists);
    }
}
```



# Realizando buscas

## Método IndexOf()

- Exemplo:

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<int> primes = new List<int>(new int[] { 19, 23, 29 });

        int index = primes.IndexOf(23); // Existe na lista, retorna o índice
1
        Console.WriteLine(index);

        index = primes.IndexOf(10); // Não existe na lista, retorna -1
        Console.WriteLine(index);
    }
}
```

### Output

```
1
-1
```

# Realizando buscas

## Método Contains()

- Exemplo:

```
using System;
using System.Collections.Generic;
using System.Linq;

class Program
{
    static void Main()
    {
        // 1. Create List with three elements.
        var list = new List<string>();
        list.Add("cat");
        list.Add("dog");
        list.Add("moth");

        // 2. Search for this element.
        if (list.Contains("dog"))
        {
            Console.WriteLine("dog was found");
        }
    }
}
```

# Realizando buscas

## Método Contains()

### •Exemplo:

```
// 3. Search for this element in any string case.  
//     This is the LINQ method with the same name.  
if (list.Contains("MOTH", StringComparison.OrdinalIgnoreCase))  
{  
    Console.WriteLine("MOTH was found (insensitive)");  
}  
  
// 4. This element is not found.  
Console.WriteLine(list.Contains("fish"));  
}  
}
```

### Output

dog was found	[Part 2]
MOTH was found (insensitive)	[Part 3]
False	[Part 4]

# Ordenando a Coleção

- Também podemos ordenar os itens através da função Sort()

- 

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();

//Insere Números
numerosInteiros.Add(8);
numerosInteiros.Add(6);
numerosInteiros.Add(9);
numerosInteiros.Add(5);
numerosInteiros.Add(7);

//Ordena lista
numerosInteiros.Sort();

foreach (int item in numerosInteiros)
{
    Console.WriteLine("L1 - Número: " + item);
}
```

# Revertendo a Coleção

## (Ordenação Decrescente)

- Também podemos reverter os itens através da função Reverse()

- 

- **Exemplo:**

```
List<int> numerosInteiros = new List<int>();

//Insere Números
numerosInteiros.Add(6);
numerosInteiros.Add(7);
numerosInteiros.Add(8);
numerosInteiros.Add(9);

//Reverte a lista
numerosInteiros.Reverse();

foreach (int item in numerosInteiros)
{
    Console.WriteLine("L1 - Número: " + item);
}
```