

C# REVISAO

1 Operadores Booleanos

A seguir temos o exemplo de lista de operadores os operadores Booleanos do C# tem a sintaxe semelhante aos da linguagem C.

```
int a = 4;
int b = 5;
bool result;
result = a < b; // a menor que b - true
result = a > b; // a maior que b - false
result = a <= 4; // a menor ou igual 4 - true
result = b >= 6; // b maior ou igual to 6 - false
result = a == b; // a igual b - false
result = a != b; // a diferente de b - true
result = a > b || a < b; // || operador logico OU condicao um OU condicao dois - true (apenas uma condicao verdadeira)
result = 3 < a && a < 6; // && operador logico `e` condicao um E condicao dois - true (ambas as condicoes tem que ser verdadeiras)
result = !result; // operador logico negacao !(true) = false
```

Exercicio 1

Faca um programa que possua as seguinte informacoes o numero de convidados do noivo e da noiva para festa, e o numero de convidados presente do noivo e da noiva, utilize ao menos uma vez cada um dos operadores condicionais para comparacoes entre as informacoes exibir o resultados sobre essas comparacoes.

Ex

```
using System;

public class Exercicio1
{
    public static void Main()
    {
        int convidadosNoivo = 300;
        int convidadosNoiva = 600;
        int convidadosPresentesNoivo = 400;
        int convidadosPresentesNoiva = 400;
```

```
        //inserir codigo do aqui...

    }
};
```

Ex de saidas

O numero de convidados da noiva e maior que do noivo

O numero de convidados da noivo e menor que do noiva

Existem penetras pois o numero convidados presentes [e maior que o numero de convidados do noivo ou da noiva.

2 Arrays vs Lists

Usando array

```
int[] nums = new int[10];
nums[0] = 1000; // nota o indice comeca em 0
nums[1] = 1001; // atribuindo valor

Console.WriteLine(nums.Length); // imprime tamanho do array
```

Usando Lista

Lista em C# e um objeto que permite guardar um conjunto de variaveis em uma ordem especifica em muitos casos pode ser mais vantajoso trabalhar com lista ao invéz de manipular arrays diretamente.

Listas usam uma sintaxe generica semelhante ao conceito de templates do C++

```
List<int> numbers = new List<int>(); // lista de inteiros <int>
List<string> strings = new List<string>(); // lista de strings <string>
```

Assim podemos criar listas de diferentes tipos de variaveis ou objetos

Nota – não e preciso definir um tamanho para a lista (o tamanho e dinamico)

```
List<string> fruits = new List<string>();  
// adicionando elementos a lista  
fruits.Add("maca");  
fruits.Add("banana");  
fruits.Add("tomate");  
  
// removendo o elemento tomate  
fruits.RemoveAt(2);  
  
// .Count retorna o tamanho atual da lista  
Console.WriteLine("tamanho da lista e "+fruits.Count);
```

Exercicio 2

a)Desenvolvam um programa que armazene uma lista de convidados utilizando um array.

B)Desenvolvam uma outra versao do programa de lista de convidados agora usando List.

```
using System;  
using System.Collections.Generic;  
  
public class Exerc2B  
{  
    public static void Main()  
    {  
        // TODO: inserir codigo aqui  
  
        // teste exemplo  
        Console.WriteLine(SuaLista.Count);  
        Console.WriteLine(SuaLista [0]);  
        Console.WriteLine(SuaLista [1]);  
  
        //ultimo elemento da lista  
        //Console.WriteLine(SuaLista [SuaLista.Count-1]);    }
```

```
}
```

3 Strings

String no C# tratase uma classe por isso a declaracao esperada seria com S maiusculo no inicio `String` entretanto podemos usar ambas as formas `string` em minusculo sera traduzido pelo compilador para a classe String tambem.

```
string myString1 = "String 1";  
String myString2 = "String 2";// ambas as formas estao corretas
```

Existem diferentes metodos implementados na classe String alguns dos que podem ser uteis na manipulacao de string são –

.ToUpper() e .ToLower() retorna uma copia da string maiuscula ou minuscula

```
string myString1 = "String 1";  
String myString2 = "String 2";// ambas as formas estao corretas  
myString1 = myString1.ToUpper(); // "STRING 1";  
myString2 = myString2.ToLower(); // "string 2";
```

.Replace() // substitui parte de uma string por outra

```
string str = "Dr. Vitor";  
string newStr = str.Replace("Dr.", "Sr.");  
Console.WriteLine(newStr);
```

.IndexOf() // retorna a posicao de uma string dentro de outra

```
string frutas = "maca,banana,limao,abacaxi";  
Console.WriteLine("pos: " + frutas.IndexOf("banana"));  
Console.WriteLine("pos: " + frutas.IndexOf("abacaxi"));
```

.substring()// retorna parte de uma string

```
string fullString = "full string";  
string str1 = fullString.Substring(5); // substring da pos 5 ate o final  
string str2 = fullString.Substring(5, 3); // substring da pos 5 a 7 tamanho 3  
Console.WriteLine(str1);
```

```
Console.WriteLine(str2);
```

Exercicio 3

Desenvolva um programa que a entrada seja uma lista de frutas (entrada pode estar fixa no código ou vir pelo console) de modo que as frutas devem estar escritas em minúsculo e separados por vírgula, para cada fruta DIFERENTE na entrada adicione num objeto lista o nome da fruta com o primeiro caractere Maiúsculo e o restante minúsculo e após isso utilizando a lista imprima cada uma das strings na lista.

OBS: frutas repetidas não devem entrar na lista

```
using System;

public class Exerc3
{
    public static void Main()
    {
        // Entrada fixa ou ler do console
        //string entrada = Console.ReadLine();
        string entrada = "maca,banana,limao,abacaxi,banana";
        List<string> frutas = new List<string>();

        // TODO: inserir código aqui
    }
}
```