

LP2 – Aula 04

TEÓRICA

Profª Mª Denilce Veloso

- denilce.veloso@fatec.sp.gov.br
 - denilce@gmail.com

Conversões entre tipos

1. Conversão implícita: é aquela em que não é necessário nenhum código especial, pois se trata de uma conversão segura, sem risco de perda de dados.

```
sbyte idade=40;
```

```
short idade2=idade
```

```
byte numAluno=35;
```

```
Float numAluno2=numAluno;
```

2. Utilizando Casts

```
double num = 123.4;
```

```
int num2;
```

```
num2 = (int)num; →será arredondado para 123.
```

***Só pode realizar conversões significativas dessa maneira, como converter um longo para um **int**. Não pode usar **cast** se o formato dos dados tiver que mudar fisicamente, como se você estiver convertendo uma **string** para um número inteiro. Para executar esses tipos de conversões usar outras formas.*

Conversões entre tipos

3. Conversões com classes auxiliares

```
long numCasas=120;
```

```
short numCasas2=Convert.ToInt16(numCasas);
```

```
int numCasas3=Convert.ToInt32(numCasas);
```

```
byte numCasas4=Convert.ToByte(numCasas);
```

```
double num1=124.5;
```

```
double num2=123.5;
```

```
int num3 = Convert.ToInt32(num1) → retorna 124
```

```
int num4 = Convert.ToInt32(num2) → retorna 124
```

**** Conversões de ponto flutuante para inteiro quando a parte decimal é 0,5 serão arredondadas para baixo, quando o número antes da vírgula for par, e para cima quando o número antes da vírgula for ímpar.**

Conversões entre tipos

4. Parse e TryParse

```
byte num;  
num = Byte.Parse("123");
```

// se quiser testar se uma conversão é válida (recomendado) é possível usar o TryParse

```
string num = "600";  
byte num2;
```

```
if (Byte.TryParse(num, out num2))  
    MessageBox.Show("Conversão ok");  
else  
    MessageBox.Show("Problema na conversão");
```

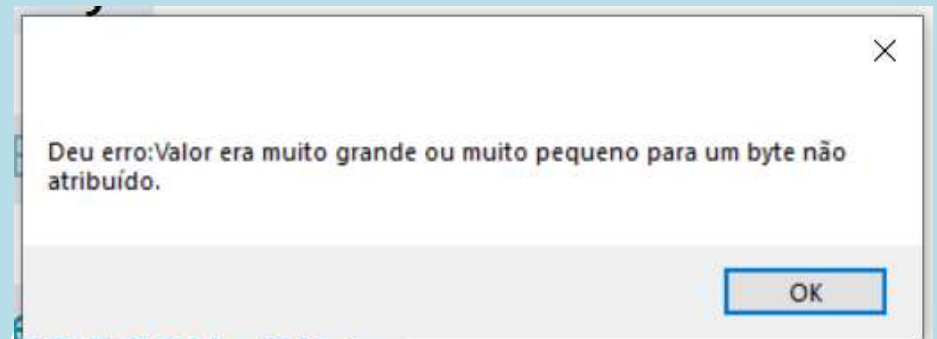
Conversões entre tipos

5. Uso de Try/Catch

Quando não está sendo utilizado o TryParse (retorna boolean), é possível **NOS DEMAIS CASOS (CAST, PARSE, CONVERT.) fazer uma tratamento de exceção (caso ocorra algum erro).**

```
try
{
    byte num;

    num = Byte.Parse("1263");
}
catch (Exception ex)
{
    MessageBox.Show("Deu erro:" + ex.Message);
}
```



EXERCÍCIOS

1) Teste rápido

a) Qual é o resultado da variável Check no final desse trecho de código?

```
int A, B, C;  
boolean Check;  
A = 5;  
B = 3;  
C = 5;  
Check = Convert.ToBoolean(A == C);  
Check = Convert.ToBoolean(B);
```

b) Qual é o resultado da variável *myChar* no final desse trecho de código?

```
string myString;  
char myChar ;  
myString = "XYZ";  
myChar = Convert.ToChar(myString);
```

Outros métodos/propriedades

Métodos e Propriedades para tratar Strings e outros

Método específico quando o método vem a partir da variável que foi criada. Ex.: nomeCliente.Substring() ou método compartilhado quando vem direto da classe. Ex.: Char.IsNumber(). Todos os métodos terminam com ().

As propriedades são métodos especiais para se trabalhar com atributos, não precisam terminar com (), exemplo nomeCliente.Length.

Método / Propriedade	Descrição	Exemplo
Length	Retorna o comprimento(tamanho) de uma string.	<pre>string stringona="Fatec"; int tamanho; tamanho=stringona.Length => 5</pre>
Substring	Retorna uma string contendo um número definido de caracteres. Sintaxe: "string".Substring(inicio, número de caracteres)	<pre>string stringona="sonolentos"; string parte; parte = stringona.Substring(4,6) => "lentos"</pre>

Outros métodos /propriedades

IndexOf	<p>Volta a posição do primeiro int x; caracter da string buscada.</p> <p><code>"string".IndexOf(stringbuscada)</code></p> <p>** pode ignorar case sensitive usando <code>StringComparison</code></p>	<pre>string stringona="Fatec"; x = stringona.IndexOf("tec"); => 2 // se não for encontrado retorna -1 x = stringona.IndexOf("TEC", StringComparison.OrdinalIgnoreCase); => 2</pre>
ToUpper	<p>Converte os caracteres de uma string para caracteres maiúsculos.</p>	<pre>string stringona="Fatec"; MessageBox.Show(stringona.ToUpper()); => "FATEC";</pre>
ToLower	<p>Converte os caracteres de uma string para caracteres minúsculos.</p>	<pre>string stringona="Fatec"; MessageBox.Show(stringona.ToLower()); => "fatec";</pre>
Trim	<p>Remove todos os espaços contidos nos lados (esquerdo e direito) de uma string.</p>	<pre>stringona = " Faculdade de Tecnologia "; stringona = stringona.Trim(); MessageBox.Show(stringona); => Faculdade de Tecnologia (sem os espaços em branco nos lados)</pre>

Outros métodos /propriedades

Replace	Substitui um texto/caracter por outro em uma string str= str.Replace(StrProcurada,StrparaSubstituir)	Stringona = "Faculdade de Tecnologia"; stringona = stringona.Replace("de", "*"); MessageBox.Show(stringona); =>Faculda* * Tecnologia
Insert	Inserere uma string em algum lugar da instância	string sNome= "Faculdade Tecnologia"; sNome = sNome.Insert(9, " de"); MessageBox.Show(sNome); → imprime Faculdade de Tecnologia

EXERCÍCIOS (anotar no caderno vamos usar no laboratório)

Considere um form com dois TextBoxs (txtPalavra1, txtPalavra2) e dois botões, escreva os seguintes códigos para os clicks do:

Botão 1 - Inserir o texto do primeiro TextBox no meio do segundo (usando Length e Substring) e colocar o resultado no segundo TextBox.

Botão 2 - Inserir 2 asteriscos no meio do texto do primeiro TextBox (usando Length e Insert) e colocar o resultado no segundo TextBox.

Outros métodos /propriedades

Split

Pode ser usada em muitos cenários quando é necessário separar strings.

Exemplo1:

```
string nomes = "joao maria pedro";  
string[] vetor = nomes.Split();// ou nomes.Split(' ')  
MessageBox.Show(vetor[0]); => joao  
MessageBox.Show(vetor[1]); => maria  
MessageBox.Show(vetor[2]); => Pedro
```

Exemplo2:

```
string numStr = "40,12,19";  
string[] vetor= numStr.Split(',');  
MessageBox.Show(vetor[0]); =>40  
MessageBox.Show(vetor[1]); =>12  
MessageBox.Show(vetor[2]); =>19
```

```
string frase = "o,rato ,roeu,a,roupa";  
string[] vetor = frase.Split(',');  
foreach (var str in vetor)  
    MessageBox.Show("<"+str+">");
```

=> <o> <rato > <roeu> <a> <roupa> → na vertical

Outros métodos /propriedades

Contains	Pesquisa uma substring	<pre>string sNome = "Faculdade de Tecnologia"; if (sNome.Contains("Tecnologia")) MessageBox.Show("Está contido Tecnologia");</pre>
-----------------	------------------------	--

Remove	Remove um determinado número de caracteres a partir de uma posição especificada	<pre>string sNome2 = "Faculdade de Tecnologia"; sNome2 = sNome2.Remove(10,3); MessageBox.Show(sNome2); → imprime Faculdade Tecnologia</pre>
---------------	---	---

Outros métodos /propriedades

PadLeft, PadRight	Coloca caracteres Unicode à esquerda ou direita OU cria string com caracteres <i>** o número informado é o total de caracteres geral, incluindo o que havia antes</i>	→ espaços em branco string st3 = "Teste"; string st4 = "Teste"; String st5 = "Teste"; st3 = st3.PadLeft(10); → " Teste" st4 = st4.PadRight(7); → "Teste " st5 = st5.PadLeft(13,"*"); → "*****Teste";
TrimEnd, Trim Start	Remove espaços em branco do final ou do início	string stl = " Teste"; string str = "Teste "; stl = stl.TrimStart(); str = str.TrimEnd(); MessageBox.Show("stl=" + stl + " str=" + str); → stl=Teste str=Teste
EndsWith	Verifica se uma string termina com uma determinada sequência de caracteres.	string texto = "Aula C#"; bool resultado1 = texto.EndsWith("C#"); → true

Outros métodos

Atenção: Os métodos e propriedades a seguir utilizam a própria classe DateTime, String, etc e não a variável criada pelo usuário.

Now

Retorna a data e hora atual

```
DateTime dataNiver;  
dataNiver = DateTime.Now;  
  
MessageBox.Show(dataNiver.ToLongTimeString());  
→ imprime hh:mm:ss data atual
```

Today

Retorna a data atual

```
DateTime dataNiver;  
dataNiver = DateTime.Today;  
  
MessageBox.Show(dataNiver.ToShortDateString());  
→ imprime dd/mm/yyyy
```

Empty

Retorna uma string vazia

```
string nome = String.Empty;
```

Outros métodos

Compare

Retorna -1, 0 ou 1, com base no resultado de uma comparação de cadeia de caracteres.

`String.Compare(st1, st2, ignorecase boolean)`
Se ignore case = true, não é case sensitive
Casa = casa

```
string st1 = "Bola";
string st2 = "CASA";
if (String.Compare(st1, st2, true) == 0)
    MessageBox.Show("Sao iguais");
else
    MessageBox.Show("São diferentes");

if (String.Compare(st1, st2, true) > 0)
    MessageBox.Show("Bola vem depois de CASA");
else
    MessageBox.Show("Bola Não vem depois de CASA");

if (String.Compare(st1, st2, true) < 0)
    MessageBox.Show("Bola vem antes de CASA");
else
    MessageBox.Show("Bola Não vem antes de CASA");
```

Exemplos:

Casa e Bola – retorna > 0

Casa e Casa – retorna 0

Bola e Casa – retorna < 0

Casa e Casas – retorna < 0

EXERCÍCIO (anotar no caderno vamos usar no laboratório)

Considere um form com dois TextBoxs (txtPalavra1, txtPalavra2) e um botão, escreva o código para o evento click do botão:

- Verificar se os dois textos são iguais (usando o método `String.compare`).

Outros métodos /propriedades

isNumber	Indica se o Unicode é número	<pre>char ch1 = '1'; if (Char.IsNumber(ch1)) MessageBox.Show("é número"); else MessageBox.Show("não é número");</pre>
-----------------	------------------------------	---

Outros métodos /propriedades

isLetter	Indica se o Unicode é Letra	<pre>if (Char.IsLetter(',')) { MessageBox.Show("é letra"); } else { MessageBox.Show("não é letra"); }</pre>
isPunctuation	Indica se o Unicode é pontuação	<pre>if (Char.IsPunctuation('.') // ? ! . : ; , @ # \$ % & * ... { MessageBox.Show("é pontuação"); } else { MessageBox.Show("não é pontuação"); }</pre>

Outros métodos /propriedades

isWhiteSpace

Indica se o Unicode é espaço em branco

```
if (Char.IsWhiteSpace(' '))
{
    MessageBox.Show("é espaço em branco");
}
else
{
    MessageBox.Show("não é espaço em branco");
}

// outro exemplo

string str = "teste do espaço";

if (Char.IsWhiteSpace(str,5))
{
    MessageBox.Show("é espaço em branco");
}
else
{
    MessageBox.Show("não é espaço em branco");
}
```

Outros métodos /propriedades

Reverse

Inverte o array

```
string sInicio= "Fatec";  
char[] vetor = sInicio.ToCharArray();  
Array.Reverse(vetor);  
string sFinal = new string(vetor);  
// poderia ter usado for  
MessageBox.Show(sFinal); → cetaF
```

Outros métodos /propriedades

```
string s = "Socorram me em Marrocos";  
char[] arr = s.ToCharArray();  
Array.Reverse(arr);
```

```
s="";  
foreach (char c in arr)  
{  
    s=s+c.ToString();  
}
```

MessageBox.Show(s); → socorraM me em marrocoS

*** Quando for testar se 1 texto é palíndromo, deve ToLower/ToLower e tirar os espaços em branco.*

CLASSE RANDOM

DV0

Random É uma classe que permite gerar números aleatórios. O método next gera o próximo número entre o mínimo e o máximo. Random.Next(mínimo,máximo) – mínimo inclusivo e máximo exclusivo

Mínimo e máximo são do tipo inteiro

- Se mínimo e máximo forem iguais vai retornar o próprio número
- ** aceita números negativos

```
Random ObjR = new Random();  
int r = ObjR.Next(0,1000);  
MessageBox.Show(r.ToString());
```

Existe NextDouble que retorna entre 0,0 e 1,0 (exclusive)

DV0

O método Next() da classe Random em C# não gera números aleatórios verdadeiros a partir do "nada". Ele usa um algoritmo determinístico que, a partir de uma semente inicial, calcula o próximo número em uma sequência que foi projetada para parecer aleatória. A "exatidão" do computador reside na forma como ele executa esse algoritmo, sempre produzindo a mesma sequência para a mesma semente. A "aleatoriedade" é, portanto, uma ilusão estatística criada por algoritmos bem projetados.

Se você precisar de números com um nível de aleatoriedade mais alto para aplicações sensíveis à segurança (como criptografia), o .NET oferece outras classes, como System.Security.Cryptography.RandomNumberGenerator, que utilizam fontes de entropia do sistema operacional para gerar números mais imprevisíveis.

Ver:

<https://learn.microsoft.com/en-us/dotnet/api/system.random?view=net-7.0>

DENILCE DE ALMEIDA OLIVEIRA VEL; 2025-03-26T01:26:26.940

EXERCÍCIO (anotar no caderno vamos usar no laboratório)

Considere um form com 2 TextBoxs (txtNumero1, txtNumero2) e um botão. Escreva um código para o click do botão, que realize um sorteio entre o primeiro e o segundo número. (usando random). Não esqueça de validar os dados, lembrando que o segundo número deve ser maior do que o primeiro.

EXERCÍCIOS – testes de mesa (testes manuais)

1) Supondo o txtTexto1 cujo conteúdo é "a" e o txtTexto2 cujo conteúdo é "personalização", qual será o resultado final da variável novaSt nesse código?

***Obs.: vazio ("") é diferente de branco(" ")

```
int xx;
```

```
string novaSt = "";
```

```
bool achou = true;
```

```
while (achou)
```

```
{
```

```
    xx = txtTexto2.Text.IndexOf(txtTexto1.Text);
```

```
    achou = (xx >= 0);
```

```
    if (achou)
```

```
{
```

```
        novaSt = txtTexto2.Text.Substring(0, xx) +  
                txtTexto2.Text.Substring(xx + txtTexto1.Text.Length,  
                txtTexto2.Text.Length - xx - 1);
```

```
        txtTexto2.Text = novaSt;
```

```
    }
```

```
}
```

```
MessageBox.Show($"novaSt={novaSt}")
```

achou	xx	novaSt	txtTexto2.Text



2) Supondo que em uma aplicação no C# .NET há um componente txtTexto1 (tipo TextBox) que recebe qualquer palavra e no onclick de um componente btnTestar (tipo Button) está sendo executado o código abaixo. Diga o quê (qual o objetivo) esse código está fazendo.

```
int numero=0;
char letra = '\0';
for (var xx = 0; xx <= txtTexto1.Text.Length - 1; xx++)
{
    if ((txtTexto1.Text[xx]!='\0') && (txtTexto1.Text[xx]
== letra))
    {
        numero += 1;
    }
    letra = txtTexto1.Text[xx];
}
←
MessageBox.Show(numero.ToString());
```

xx	letra	numero

3) Supondo que exista na aplicação um componente `txtTexto1` (tipo `textbox`) cujo valor é “Tecnologia”, qual será o resultado da variável `numero` no final desse trecho desse código?

```
int numero = 0;
```

```
char letra = Convert.ToChar("s"); // ou char letra = 's';
```

```
foreach (Char c in txtTexto1.Text)
```

```
{
```

```
    if ((c != '\0') && (c == letra))
```

```
    {
```

```
        numero += 1;
```

```
    }
```

```
    letra = c;
```

```
}
```

```
MessageBox.Show(numero.ToString());
```

4) Diga o quê o código abaixo está fazendo:

```
private void button16_Click(object sender, EventArgs e)
```

```
{
```

```
    int soma = 0;
```

```
    System.Random gerador = new System.Random();
```

```
    for(int i = 1; i <= 100; i++)
```

```
    {
```

```
        MessageBox.Show("Lançamento: " + i);
```

```
        int numero = (gerador.Next(0,100) * 6 + 1);
```

```
        soma += numero;
```

```
        if(soma > 3600)
```

```
        {
```

```
            MessageBox.Show("Você ganhou com " + i + " lançamentos");
```

```
            break;
```

```
        }
```

```
    }
```

```
    MessageBox.Show("Jogue Novamente");
```

```
}
```

Dica MessageBox.Show

```
int i = 10;
```

```
// número
```

```
MessageBox.Show(i.ToString());
```

```
// texto + numero (conversao implicita)
```

```
MessageBox.Show("O número é:"+i);
```

Dica Tipo String

null: Significa que a variável não referencia nenhum objeto string na memória. Ela não possui nenhum valor. Tentar acessar métodos ou propriedades de uma variável string que é null resultará em um `NullReferenceException` em tempo de execução.

"" (string vazia): É um objeto string válido que não contém nenhum caractere. Ela possui um valor (vazio) e mas pode chamar métodos e acessar propriedades dela sem gerar uma exceção.

```
string s1; // null
```

```
MessageBox.Show(s1.Length.ToString()); --> Da erro
```

```
string s2 = "";
```

```
MessageBox.Show(s2.Length.ToString());
```