



# LINGUAGEM DE PROGRAMAÇÃO

---

Orientada a objeto

# Roteiro da aula

- Tratamento de erros
- Debug
- Comandos do Windows Form

# Tratamento de erros

- Localizar e corrigir os erros é chamado de depuração
- Diversos Erros: Erros impedem que o programa seja iniciado, interrompem ou travam a execução ou não fornecem os resultados esperados.
- Exemplo projeto calculadora com digitação de caracter
- Tipos de erro:
  - Erro de compilação: não executa o programa(duble)
  - Erro de Tempo de Execução: durante execução.(char)
  - Erro de lógica: mais difíceis de corrigir

# Debug

- Criação de um ponto de interrupção (breakpoint) para executar passo a passo as instruções do programa
- Clique na barra cinza, a esquerda no numero da linha para marcar o ponto de interrupção.
- Execute e use F11 ou “Step Into” para prosseguir.

# Tratamento de Exceções

- Maneira de lidar com qualquer situação inesperada que apareça durante a execução do programa (acesso ao Banco, impressora não localizada)
- Try
- Catch
- Finally
- Exemplo calculadora  
valor2 = 0

```
private void cmdDividir_Click(object sender, EventArgs e)
{
    try
    {
        int v1 = Convert.ToInt32(txtValor1.Text);
        int v2 = Convert.ToInt32(txtValor2.Text);
        int r = v1 / v2;
        lblResultado.Text = "Resultado: " + r;
    }
    catch
    {
        lblResultado.Text = "Valores incorretos";
    }
}
```

# Classe Exception

- A classe Exception contém informações detalhadas sobre o erro.

...

```
catch(Exception x)
```

```
{
```

```
    lblStatus.Text = x.Message;
```

```
}
```

# Criando as próprias exceções

- Método para Validação

```
private static void ValidarNome(string nome)
{
    if (nome.Trim().Length == 0)
    {
        string msg = "O nome não pode estar em branco";
        ApplicationException e = new ApplicationException(msg);
        throw e;
    }
}

private static void ValidarNota(double nota)
{
    if ((nota < 0) || (nota > 10))
    {
        string msg = "A nota deve estar entre 0 e 10";
        ApplicationException e = new ApplicationException(msg);
        throw e;
    }
}
```

# Criando as próprias exceções

- Chamada aos métodos

```
private void cmdIncluir_Click(object sender, EventArgs e)
{
    try
    {
        string nome = txtNome.Text;
        ValidarNome(nome);
        double nota = Convert.ToDouble(txtNota.Text);
        ValidarNota(nota);
        lblStatus.Text = "Nota registrada!";
    }
    catch (ApplicationException x)
    {
        lblStatus.Text = x.Message;
    }
    catch (Exception x)
    {
        lblStatus.Text = x.Message;
    }
}
```



# Bloco Finally

- Instruções desse bloco sempre serão executadas, independente de ocorrer ou não um erro (fechar BD)

# Exercicio

- Exercicio 1

# MessageBox

```
private void mensagemToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Confirmação", "Confirma Cadastrado ?", MessageBoxButtons.YesNo );
}
```

# Chamar outro Formulário

```
Form2 proximaTela = new Form2();  
    proximaTela.Show();  
Form1 home = new Form1();  
    home.ShowDialog();  
    // ShowDialog - não deixa clicar em outra tela
```

Obs.: Para iniciar o projeto com uma outra form:

→ no **program.cs** altere a linha de comando **applicationRun()**

# Exercicio

- Exercício 2 e 3

# Windows form

- Atalhos em botões → propriedade Text - &Fechar (ctrl+F)
- Fechar janela → Close() ou this.Close() ou Application.Exit();
- Propriedade TextBox → Multiline permite varias linhas.

ScrollBars para barras de rolagem

- CheckBox e RadioButton → Exemplos:

```
string permite= " ";  
if ( chkAlterar.Checked)  
    permite+= chkAlterar.Text;  
if (chkConsulta.Checked)  
    permite+= chkConsulta.Text
```

```
string sexo = " ";  
if ( rbFeminino.Checked)  
    sexo = rbFeminino.Text;  
else  
    sexo = rbMasculino.Text
```

# Windows form

- ListBox e ComboBox → exibe vários itens. Com um duplo clique na lista ou combo tenho o exemplo

`lblRes.Text = "Selecionado " + lbCarro.SelectedItem.ToString()`

- Obter item → `int indice=0   indice = lbCarro.SelectedIndex`
- Propriedade Sorted → Lista ordenada
- Metodos :

`Items.Add` → adiciona à lista

`Items.RemoveAt(1)` → remove o índice indicado

`Items.Clear()` → remove todos items.

`Items.Contains()` → verifica se contem

`If (! lbCarro.Items.Contains("Fiat") )`

`lbCarro.Items.Add("Fiat");`

# Windows Form

- **MenuStrip** → criação de menus.

Para linha separadora clique na seta a direita do TypeHere.

Acesso rapido use “&” antes da letra (Alt + Letra).

Atalho através da propriedade ShortcutKeys.

Adicionar imagem no menu, propriedade Image

- **ContextMenuStrip** → menu de atalho com tecla direita do mouse



# Windows Form

- Evento Load → executa todos os comandos ao iniciar o formulário  
→ com 2 cliques na tela do formulário ou no em propriedades - evento – Load

# Exercício

- Exercício 4 e 5