OBJETOS : Timer1 e RadioGroup

É comum em projetos termos rotinas que devem ser disparadas de tempo em tempo. Para isso, o Delphi dispõe de um controle que permite a execução de procedimentos sempre que o intervalo de tempo definido ocorrer. Esse controle é o Timer, cuja utilidade e propriedades será escarecida no exercício a seguir.

O aplicativo abaixo tem a finalidade de demonstrar a utilidade dos Componentess **Timer** (Aba System do Fichário da Barra de Componentes) e **RadioGroup** (Aba Standard da Barra de Componentes). Siga as instruções para montá-lo.

Componentes	Propriedade	Valor
Timer1	**Interval**	**500**
ListBox1	Items	'2 x 0' '2 x 1' '2 x 2' '2 x 3' '2 x 4' '2 x 5' '2 x 6' '2 x 7' '2 x 8' '2 x 9' '2 x 10'
RadioGroup1	Items Caption	'Liga' 'Desliga' Liga/Desliga

CÓDIGO-FONTE :

```
unit UTabuada;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    ListBox1: TListBox;
    Label1: TLabel;
    RadioGroup1: TRadioGroup;
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure RadioGroup1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
   // Aponta para o primeiro item daCaixa de listagem
   ListBox1.ItemIndex:=0;
end;

procedure TForm1.Timer1Timer(Sender: TObject);

const
 count:Integer = 0;  //  Cria uma variável estática e a inicializa em zero.
begin

  ListBox1.ItemIndex:=count;  // Seleciona o primeiro Item da Caixa de Listagem

(* Atribui ao Label o conteúdo do item selecionado e calcula o resultado da multiplicação *)

  Label1.Caption:=ListBox1.Items[ListBox1.ItemIndex]+ ' = '+InttoStr(2*count);
```

```
  count:=count+1;
    if count > 10 then
       count:=0;

  end;


procedure TForm1.RadioGroup1Click(Sender: TObject);

begin
if RadioGroup1.ItemIndex = 1 then  // Se a opção selecionada pelo usuário for a opção 1
(Desliga) , então
    Timer1.Interval:=0     // Desliga o cronômetro
else                              //  Caso contrário ela é a opção zero, isto é Liga
    Timer1.Interval:=500;   // Liga o Cronômetro - Intervalo de 500 ms
end;

end.
```

ESTUDO DE CASO

1) Brinquedo do Caco Antibes

O mais famoso personagem do seriado "Sai de Baixo" da Rede Globo, descobriu uma maneira de descarregar sua frustração pelo festival de besteirol falado por sua esposa Magna. Tente reproduzir o aplicativo abaixo e descubra como ele resolveu o seu problema. ***(Nota : O arquivo Executável Será Distribuído pela professora durante a aula)***

2) Visualizador de Arquivos Gráficos

Construir um Visualizador de Imagens que percorre os arquivos gráficos existentes em um determinado diretório e os mostra um por um ao usuário,a intervalo de tempo de 500 ms.

DICA : Utilize, juntamente com o objeto Timer1, o componente FileListBox1 (Aba Win 3.1 da Barra de Componentes do Delphi) e o componente Image1 (Aba Additional).

PROPRIEDADES DOS COMPONENTES

object Form1: TForm1

Caption = 'Brinquedinho do Caco Antibes'

object Shape1: Tshape (Aba Aditional)

Shape = stEllipse
Visible = False

object Label1: TLabel

Alignment = taCenter
AutoSize = False
Caption = #39'Você tá com problema na calúnia, mami?!'
Visible = False
WordWrap = True

object Button1: TButton

Caption = '&Fala Magda !'

objet Timer1:Ttimer
Interval=0
objet Timer2:Ttimer
Interval=0
object Image1:Timage
Name=ImgPrincipal

object Image2:Timage
Name=ImgMagdaAbre

object Image3:Timage
Name=ImgMAgdaFecha

CÓDIGO-FONTE DO ESTUDO DE CASO 1

```
unit Umagda;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls;

type
  TForm1 = class(TForm)
    IMGPRINCIPAL: TImage;
    IMGMAGDAABRE: TImage;
    IMGMAGDAFECHA: TImage;
    Timer1: TTimer;
    Button1: TButton;
    Timer2: TTimer;
    Shape1: TShape;
    Label1: TLabel;
    procedure Timer1Timer(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}


procedure TForm1.Timer1Timer(Sender: TObject);

const
count:integer=0;

begin

count:=count+1;
if count > 1 then
   count:=0;

if count=1 then
     imgprincipal.picture := imgmagdafecha.picture
 else
```

```
 imgprincipal.picture := imgmagdaabre.picture;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
if button1.Caption = '&Fala Magda !' then
   begin
     shape1.visible:=true;
     label1.visible:=true;
     Timer1.Interval:=200;
     Timer2.Interval:=1200;
     button1.Caption := '&Cala a Boca Magda !';
   end
else
  begin
    shape1.visible:=false;
    label1.visible:=false;
    Timer1.Interval:=0;
    Timer2.Interval:=0;
    button1.Caption := '&Fala Magda !';
 end;

 end;

procedure TForm1.Timer2Timer(Sender: TObject);

const
contafrase:integer=0;
begin

 contafrase:=contafrase+1;
 if contafrase > 3 then
    contafrase:=1;

  case contafrase of
     1:Label1.Caption :='O telefone é sem fio, como é que pode dar linha?' ;
     2:Label1.Caption :='Você tá com problema na calúnia, mami?!' ;
     3:Label1.Caption :='Ser mãe é padecer na Paraíba.' ;
  end;

end;


end.
```

TRY...EXCEPT

Para administrar um erro à nossa maneira (e não à maneira do Delphi) uma das formas é cercar o conjunto de statements que podem gerar um erro com os comandos : **try, except,end**.

Os tipos de erros mais comuns podem ser relacionados abaixo :

- **Problemas no acesso a arquivos;**
- **Entrada Inválida;**
- **Divisão de integer por zero;**
- **Divisão de não integer por zero;**
- **Dado inadequado para conversão;**
- **Falta de memória.**

Para demonstrar como você pode salvaguardar seu código, faça o seguinte exercício , para criar um programa simples que usa manipulador de exceções. Depois de fazer esse exercício, você deve ser capaz de:

- Escrever manipulador de exceções;
- Salvaguardar programas contra erros de intervalos de listas.
- Alterar opções do ambiente de desenvolvimento.

1) Do menu File, escolha New Application .
2) Acrescente uma caixa de lista, um botão e um rótulo ao formulário;
3) No manipulador de evento OnCreate do formulário, acrescente o código a seguir :

```
ListBox1.Items.add('Item no. 1');
ListBox1.Items.add('Item no. 2');
ListBox1.Items.add('Item no. 3');
ListBox1.Items.add('Item no. 4');
ListBox1.Items.add('Item no. 5');
```

4) No manipulador de evento OnClick do botão digite :

```
TRY
   Label1.Caption :=ListBox1.Items[7];
EXCEPT
   Application.MessageBox (' Índice Excede o Limite','Erro Detectado',MB_ICONEXCLAMATION+MB_OK);
end;
```

5) Selecione **Tools | Environment Options...** para abrir a caixa de Diálogo Environment Options
6) Dê um clique na caixa de seleção **Break on exception** se ela estiver marcada, para desligar essa opção.

7) Escolha OK para salvar a nova configuração de ambiente.

8) Rode o programa e dê um clique no botão. Como a lista só possui 5 itens, tentar referenciar o item 7 (um item não existente) lança um erro de exceção. Ao invés de " derrubar" o programa, no entanto, o manipulador de exceção captura o erro e exibe uma mensagem.

1) 9) Repita os passos 5 e 6 , dessa vez na caixa de seleção **Break on exception** que ela estava marcada, devemos desmarcar essa opção.