

SEMINARIO DE LENGUAJES – OPCION DELHI - 30/04/2014

- 1) Implementar la aplicación con la interfaz de la Figura 1. Esta cuenta con un único formulario en el que se disponen adecuadamente 9 controles *TImage* con frutas y 1 *TButton* ubicados sobre un panel y un *TImage* con la imagen de una mascota (fuera del panel).

Las frutas deben ser arrastradas y soltadas sobre la mascota para alimentarla, desapareciendo del panel.



Figura 1

El botón Reponer Frutas vuelve a mostrar las frutas eliminadas. Implementar:

- *procedure TForm1.Button1Click(Sender: TObject)*
- *procedure TForm1.ImageDragOver(Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean)*
- *procedure TForm1.ImageDragDrop(Sender, Source: TObject; X, Y: Integer)*

```
procedure TForm1.Button1Click(Sender: TObject);
var i:Integer;
begin
  for i := 0 to Panel1.ControlCount-1 do
    if Panel1.Controls[i] Is TImage then
      TImage(Panel1.Controls[i]).Visible:=True;
      //puede ser (Panel1.Controls[i] As TImage).Visible
  end;

  procedure TForm1.ImageDragDrop(Sender, Source: TObject; X, Y: Integer);
  begin
    (Source As TImage).Visible:=False; // puede ser TImage(Source).Visible
  end;

  procedure TForm1.ImageDragOver(Sender, Source: TObject; X, Y: Integer; State:
  TDragState; var Accept: Boolean);
  begin
    Accept:= True;
  end;
```

- 2) Implementar la aplicación de la Figura 2. La misma cuenta con un formulario principal (FormPpal) que tiene un *TLabel* (LblNro) donde se muestra un número entero entre 0 y 100. También posee un botón para seleccionar el valor del número utilizando un formulario especial (FormValidar) que se abre de manera modal. Este formulario contiene un *TEdit* (EdNro) que utiliza un manejador de excepciones para controlar el ingreso de un número y asegurar un valor entre 0 y 100. El botón *Aceptar*, que cierra el formulario, debe deshabilitarse y habilitarse según el número cumpla o no con las condiciones de validación. En caso de cerrar el formulario de validación utilizando la cruz ubicada en el extremo superior derecho, no debe modificarse el número del formulario principal.

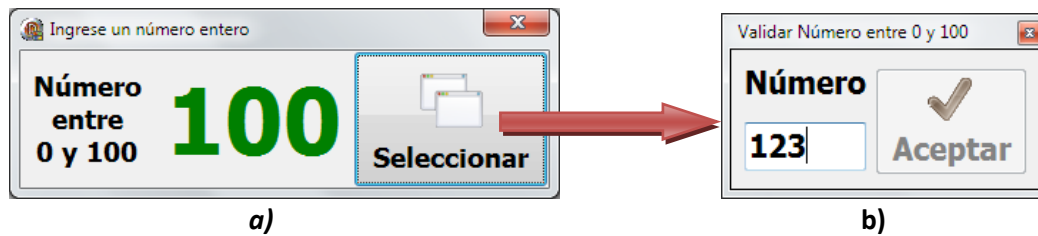


Figura 2: a) Formulario principal cuya propiedad Name contiene el valor FormPpal;
b) Formulario que se visualiza al presionar el botón Seleccionar. Su propiedad Name contiene el valor FormValidar

Implementar los siguientes métodos de los correspondientes formularios:

- *procedure TFormPpal.btSeleccionarClick(Sender: TObject)*
- *procedure TFormValidar.edNroChange(Sender: TObject)*
- *procedure TFormValidar.btOkClick(Sender: TObject)*

```
procedure TFormPpal.btSeleccionarClick(Sender: TObject);  
begin  
    if formValidar.ShowModal = mrOk then  
        lblNro.Caption:= formValidar.edNro.Text  
    end;  
  
procedure TFormValidar.btOkClick(Sender: TObject);  
begin  
    ModalResult:= mrOk;  
end;  
  
procedure TFormValidar.edNroChange(Sender: TObject);  
var nro:Integer;  
begin  
    try  
        nro:= StrToInt(edNro.Text);  
        btOk.Enabled:= (0 <= nro) and (nro <= 100);  
    except  
        btOk.Enabled:= False;  
    end;  
end;
```

- 3) Utilizando la declaración de variables existente, corrija los errores que encuentre en el siguiente código indicando si se trata de errores de compilación o de ejecución. Luego indique que imprime.

SEMINARIO DE LENGUAJES - OPCION DELHI - 30/04/2014

```
var texto1, texto2, texto3 : array of string;
    linea1 : string;
    linea2 : string[30];
begin
    texto1[low(texto1)] := 'Hoy es miércoles';
    texto1[length(texto1)] := 'tenemos examen de Delphi'; (1)

    texto2 := texto1;
    texto2[2] := 'Arreglo dinamico';
    writeln(sizeof(texto1[2])); (3)
    writeln('texto1[2] = ', texto1[2]); (4)

    texto3 := copy(texto1, 0, length(texto1));
    texto2[2] := 'Resultado del COPY';
    writeln('texto1[2] = ', texto1[2]); (5)
    writeln('texto3[2] = ', texto3[2]); (6)

    writeln('ANTES de asignar :');
    writeln(sizeof(linea1), ' - ', sizeof(linea2)); (7)
    linea1 := texto3[1];
    linea2 := linea1;
    writeln('DESPUES de asignar :');
    writeln(sizeof(linea1), ' - ', sizeof(linea2)); (8)
end.
```

Rta:

Las variables `texto1`, `texto2`, `texto3` son arreglos dinámicos por lo que antes de utilizarlos para almacenar información es preciso reservarles memoria utilizando la instrucción `SetLength`. En este caso, luego del `begin` de la línea (1) debería decir `SetLength(texto1,3)`. La cantidad de elementos debería ser 3 como mínimo por las distintas asignaciones que se observan en el programa.

Independientemente de la cantidad de elementos que se espere en el arreglo, el elemento `texto1[length(texto1)]` no existe ya que el primer elemento tiene índice 0. En este caso, en la línea (2) en lugar de la función `length` debería utilizar `High`.

La línea (3) imprime 4.

La línea (4) imprime `texto1[2] = Arreglo Dinámico`

La línea (5) imprime `texto1[2] = Resultado del COPY`

La línea (6) imprime `texto3[2] = Arreglo dinámico` si se utilizó `setlength(texto1,3)` o un valor indefinido si se utilizó una cantidad de elementos mayor y se cambió `length` por `high` en la línea (2)

La línea (7) imprime `4 - 31`

La línea (8) imprime `4 - 31`

SEMINARIO DE LENGUAJES – OPCION DELHI - 30/04/2014

4) Indique el valor de verdad de las siguientes afirmaciones:

- a) Sobre un formulario se encuentran pegados dos paneles: uno contiene 5 TImages y el otro 3 TLabel; es decir que el formulario contiene 10 controles.

FALSO. El formulario contiene 2 controles (cada uno de los paneles)

- b) Las componentes TOpenDialog y TSaveDialog no son controles.

VERDADERO.

- c) Para saber si una componente TOpenDialog se cerró al presionar el botón abrir es preciso configurar su propiedad Options.

FALSO. La propiedad Options permitía realizar validaciones sobre el nombre del archivo (ej: que exista la ruta o que exista el archivo). Para saber con que botón se cerró es preciso analizar lo que retorna el método EXECUTE del componente de diálogo.

- d) La instrucción StrToFloat('10') levanta una excepción.

FALSO porque los números reales incluyen a los enteros.

- e) Es posible asignar el contenido de un TListBox a un TMemo.

VERDADERO. Ambos componentes tienen una propiedad TString con la lista de strings correspondientes y por lo tanto puede copiarse el contenido de uno en otro.

- f) Si un panel contiene 3 botones y está deshabilitado, no es posible clickear sobre ninguno de ellos. **VERDADERO**

- g) En la propiedad ModalResult de un formulario es posible asignar un número entero.

VERDADERO

- h) Los componentes TMemo, TComboBox y TListbox son componentes contenedoras. **FALSO**

- i) La propiedad Left de un control puede contener un valor negativo. **VERDADERO**

- j) Es posible guardar la lista de opciones de un TComboBox en un archivo.

VERDADERO. La lista es un TString, es decir que puede guardarse en un archivo utilizando el método SaveToFile.