Implementing a custom loop

I love the C#/C++/Java **for** loop. It lets you define the value by which you increment the loop variable, e.g.:

```
for (int index = 0; index <= 1000; index += 2) { }
```

Delphi does not have an equal loop, but you can use a while/repeat loop to implement something similar. Anyways here's my trial code.

Create a new VCL application, drop a button and a spin edit on the form, the define a new type like so:

```
type
  TurboLoopCallback = procedure (Index: Integer) of Object;
```

Now the loop procedure:

```
procedure TurboLoop(
  Index, (* loop start index *)
  ToIndex: Integer; (* until this value *)
  Callback: TurboLoopCallback; (* the callback procedure *)
  const Step: Integer = 2); (* the step value *)
begin
  (* is it a TO or DOWNTO loop? *)
  if Index > ToIndex then begin
    (* this is a DOWNTO loop
      for index := VALUE downto VALUE do... *)
    while Index >= ToIndex do begin
      (* callback procedure *)
      Callback(Index);
      (* decrement the value of index by STEP value *)
      Dec(Index, Step);
    end;
  end
  else
  begin
    (* this is a TO loop *)
    while Index <= ToIndex do begin
      (* callback procedure *)
      Callback(Index);
      (* increment the value of index by STEP value *)
      Inc(Index, Step);
    end;
  end;
  (* tadam! that's it *)
end;
```

Add a new public procedure to form:

```
...
  public
    procedure MessageLoop(Index: Integer);
  end;
...
```

And it's implemention:

```
procedure TForm1.MessageLoop(Index: Integer);
begin
  ShowMessageFmt('this is my %d message', [Index]);
end;
```

Finally in the button's *OnClick* event handler write this code:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  TurboLoop(1, 10, MessageLoop, seIndex.Value);
end;
```

This is just a proof of concept, it is not very productive.

Author: Dorin Duminica
Contributor: topellina
Added: 2013-09-06
Last updated: 2013-09-06