MAY 20, 2023

DELPHI OOP - PHONE CLASS
MILAN

1.1 PHONECLASS_U

```pascal
unit PhoneClass_U;

interface

uses sysutils;

type
   // start of class
   TPhone = class
   private
     // attributes / properties
     fBrand: string;
     fModel: string;
     fOwner: string;
     fPurchaseDate: tDatetime;
     fprice: double;

   public
     // create an instance
     constructor create(sBrand: string; sModel: string; sOwner: string;
       tPurchaseDate: tDatetime; dPrice: double); overload;

     constructor create(sBrand: string; sModel: string); Overload;

     // class methods
     function getprice: double;
     procedure setprice(dPrice: double);
     function calculateAge: integer;
     function tostring: string;
     function ostring: string;
   end; // end of class
```

Class Name = TPhone

Attributes tell us about the class. For example, a "Phone" has a Brand, and Model

Parameter A value passed to a Function or Procedure to use.

Constructor: Used to create a "instance" of the class

Overload: Used when there is more than one function/procedure with the same name. ie: create

Function: Returns a value. Use's the Result keyword.
Procedure: Similar to function, but no return value.
toString: A function that outputs the classes attributes in string format.

```pascal
implementation

constructor TPhone.create(sBrand: string; sModel: string);
begin
  fBrand := sBrand;
  fModel := sModel;
end;

constructor TPhone.create(sBrand: string; sModel: string; sOwner: string;
  tPurchaseDate: tDatetime; dPrice: double);
begin
  fBrand := sBrand;
  fModel := sModel;
  fOwner := sOwner;
  fPurchaseDate := tPurchaseDate;
  fprice := dPrice;

end;

function TPhone.getprice: double;
begin
  result := fprice;

end;


function TPhone.ostring: string;
```

```pascal
begin
  result := 'Brand: ' + fBrand + #10 + 'Model: ' + fModel;

end;

procedure TPhone.setprice(dPrice: double);
begin
  fprice := dPrice;

end;

function TPhone.calculateAge: integer;
var
  NumDays: integer;
begin
  NumDays := round(now - fPurchaseDate);
  result := NumDays div 365;

end;

function TPhone.tostring;
begin
  result := 'Brand : ' + fBrand + #13 + 'Model: ' + fModel + #13 + 'Owner: ' +
    fOwner + #13 + 'Purchase Date: ' + formatdatetime('dddddd', fPurchaseDate)
    + #13 + 'Price: ' + floattostrf(fprice, ffcurrency, 10, 2)
    + #13 + 'Age: ' + inttostr(calculateAge());
end;

end.
```

1.2   USING THE PHONE_U CLASS

```pascal
unit Unit1;

interface

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

type
  TForm1 = class(TForm)
    pnlHeading: TPanel;
    GroupBox1: TGroupBox;
    lblModel: TLabel;
    cmbModel: TComboBox;
    lblBrand: TLabel;
    edtBrand: TEdit;
    lblOwner: TLabel;
    edtOwner: TEdit;
    lblDate: TLabel;
    edtPrice: TEdit;
    lblPrice: TLabel;
    redOutput: TRichEdit;
    btnAddPhone: TButton;
    edtDate: TEdit;
    btnOverloadConstructor: TButton;
    procedure btnAddPhoneClick(Sender: TObject);
    procedure btnOverloadConstructorClick(Sender: TObject);
    // procedure cmbModelChange(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  objPhone: Tphone; // TPhone "instance" declared

implementation

{$R *.dfm}

procedure TForm1.btnAddPhoneClick(Sender: TObject);
var
  sBrand, smodel, sowner: string;
  dtDate: tDAteTime;
  rPrice: real;
begin

  smodel := cmbModel.Text;
  sBrand := edtBrand.Text;
  sowner := edtOwner.Text;
  dtDate := strtodate(edtDate.Text);
  rPrice := strtofloat(edtPrice.Text);

  // calls the TPhone constructor with all the details
  objPhone := TPhone.create(sBrand, smodel, sowner, dtDate, rPrice);
  redOutput.Lines.Add(objPhone.tostring());
```

```
end;

procedure TForm1.btnOverloadConstructorClick(Sender: TObject);
var
  phone: Tphone; // creates another instance of TPhone, different from objPhone
begin
  // calls the overloaded version of the constructor with less details
  // ie: only two fields.
  phone := Tphone.create('Nokia', '3310');
  redOutput.Lines.Add(Phone.ostring());
end;

end.
```

Sodacom Phone Store

Phone Details

Model

Brand

Owner

Purchase Date

Price

Add Phone

Overload Constrcutor

Brand: Nokia
Model: 3310