

```

unit ViaSoft.Controller.MaquinaDinheiro;

interface

uses
  ViaSoft.Controller.Interfaces, System.Classes;

type

  TMaquinaDinheiro = class(TInterfacedObject, IMaquina)
  private
    function ValorCedula(Posicao : Integer): Double;
  public
    Lista : TList;
    function MontarTroco(Troco: Double): TList;
    procedure MostrarTroco;
    constructor Create;
    destructor destroy; override;
  end;

  TCedula = (ceNota100, ceNota50, ceNota20, ceNota10, ceNota5, ceNota2, ceMoeda100, ceMoeda50,
ceMoeda25, ceMoeda10, ceMoeda5, ceMoeda1);

  TTroco = class
  private
    FTipo: TCedula;
    FQuantidade: Integer;
  public
    constructor Create(Tipo: TCedula; Quantidade: Integer) ;
    function GetTipo: TCedula;
    function GetQuantidade: Integer;
    procedure SetQuantidade(Quantidade: Integer);
  end;

implementation

uses
  System.SysUtils, FMX.Dialogs, System.TypeInfo;

{ TMaquinaDinheiro }

constructor TMaquinaDinheiro.Create;
begin

end;

destructor TMaquinaDinheiro.destroy;
begin

inherited;
end;

function TMaquinaDinheiro.ValorCedula(Posicao : Integer):Double;
begin
  case TCedula(Posicao) of
    ceNota100:
      Result := 100;

```

```

ceNota50:
    Result := 50;
ceNota20:
    Result := 20;
ceNota10:
    Result := 10;
ceNota5:
    Result := 5;
ceNota2:
    Result := 2;
ceMoeda100:
    Result := 1;
ceMoeda50:
    Result := 0.50;
ceMoeda25:
    Result := 0.25;
ceMoeda10:
    Result := 0.10;
ceMoeda5:
    Result := 0.05;
ceMoeda1:
    Result := 0.01;
end;
end;

function TMaquinaDinheiro.MontarTroco(Troco: Double): TList;
var Valor : Double;
    I, Qtde : Integer;
    ValidaInt : String;
begin
    Lista := TList.Create;
    try
        Troco := StrToFloat(FormatFloat('#0.00',Troco));
        for I := 0 to 11 do
            begin
                Valor := ValorCedula(I);
                ValidaInt := FloatToStr(Troco/Valor);

                if Length(ValidaInt) > 1 then
                    Qtde := Trunc(Troco/Valor)
                else
                    Qtde := StrToInt(ValidaInt);

                if Qtde > 0 then
                    begin
                        Troco := StrToFloat(FormatFloat('#0.00',Troco - (Valor * Qtde)));
                        Lista.Add(TTroco.Create(TCedula(I),Qtde)) ;
                    end;
                end;
            end;
        end;
        Result := Lista;

        MostrarTroco;
    finally
        FreeAndNil(Lista);
    end;
end;
end;

```

```

procedure TMaquinaDinheiro.MostrarTroco;
var I : Integer;
    Valor : Double ;
    Show : TStringList;
    Cedula, Tipo, Qtde : String;
begin
    Show := TStringList.Create;
    for I := 0 to Lista.Count-1 do
        begin
            Qtde := IntToStr(TTroco(Lista[I]).GetQuantidade);
            Cedula := GetEnumName(TypeInfo(TCedula),Integer(TTroco(Lista[I]).GetTipo));
            Valor := ValorCedula(Integer(TTroco(Lista[I]).GetTipo));

            if Cedula.Contains('Nota') then Tipo := 'Nota' else Tipo := 'Moeda';

            Show.Add(Qtde +' '+Tipo+' de - '+Valor.ToString+' - TTroco('+Cedula+', '+Qtde+')');

        end;
        ShowMessage(Show.Text);
        FreeandNil(Show);
    end;

{ TTroco }

constructor TTroco.Create(Tipo: TCedula; Quantidade: Integer);
begin
    Self.FTipo := Tipo;
    SetQuantidade(Quantidade);
end;

function TTroco.GetQuantidade: Integer;
begin
    Result := FQuantidade;
end;

function TTroco.GetTipo: TCedula;
begin
    Result := FTipo;
end;

procedure TTroco.SetQuantidade(Quantidade: Integer);
begin
    FQuantidade := Quantidade;
end;

end.

```