Epreuve écrite – CORRIGE MODELE

Examen de fin d'études secondaires 2015

Section: B

Branche: Informatique

Numéro d'ordre du candidat

CORRIGE MODELE

```
unit UMain;
//------------------------------------------------------------------------------
interface                                                               // 2 p.

uses
  SysUtils, Graphics, Forms, Grids, Classes, Controls, StdCtrls, ExtCtrls;

type
  TfrmMain = class(TForm)
    lbCoords: TListBox;
    sgSegments: TStringGrid;     imgDessin: TImage;
    btnNouveau: TButton;         btnAjout: TButton;
    btnTransfert: TButton;       btnDessin: TButton;
    Label1: TLabel;              Label2: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure btnNouveauClick(Sender: TObject);
    procedure imgDessinMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure btnAjoutClick(Sender: TObject);
    procedure btnTransfertClick(Sender: TObject);
    procedure btnDessinClick(Sender: TObject);
  end;

var
  frmMain: TfrmMain;

implementation
{$R *.dfm}
//------------------------------------------------------------------------------
procedure TfrmMain.FormCreate(Sender: TObject);                         // 2 p.
begin
  randomize;
  imgDessin.Canvas.Pen.Color := clBlack;
  imgDessin.Canvas.Brush.Color := clWhite;
  imgDessin.Canvas.Brush.Style := bsSolid;
  imgDessin.Canvas.Rectangle(0,0,imgDessin.Width,imgDessin.Height);
  sgSegments.Cells[0,0] := 'x1';
  sgSegments.Cells[1,0] := 'y1';
  sgSegments.Cells[2,0] := 'x2';
  sgSegments.Cells[3,0] := 'y2';
  sgSegments.Cells[4,0] := 'dist.';
end;
//------------------------------------------------------------------------------
procedure TfrmMain.btnNouveauClick(Sender: TObject);                    // 3 p.
var I : integer;
begin
  imgDessin.Canvas.Pen.Color := clBlack;
  imgDessin.Canvas.Brush.Color := clWhite;
  imgDessin.Canvas.Brush.Style := bsSolid;
  imgDessin.Canvas.Rectangle(0,0,imgDessin.Width,imgDessin.Height);
  lbCoords.Items.Clear;
  sgSegments.RowCount := 2;
  for I := 0 to 4 do
    sgSegments.Cells[I,1] := '';
end;
//------------------------------------------------------------------------------
procedure TfrmMain.imgDessinMouseDown(Sender: TObject;                  // 3 p.
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  if (Button = mbLeft) then begin
    lbCoords.Items.Append(inttostr(X)+';'+inttostr(Y));
    imgDessin.Canvas.Pen.Color := clBlack;
    imgDessin.Canvas.MoveTo(X-5,Y);
    imgDessin.Canvas.LineTo(X+6,Y);
    imgDessin.Canvas.MoveTo(X,Y-5);
    imgDessin.Canvas.LineTo(X,Y+6);
  end;
end;
//------------------------------------------------------------------------------
```

```
//----------------------------------------------------------------------------
procedure TfrmMain.btnAjoutClick(Sender: TObject);                     // 5 p.
var X, Y : integer;
begin
  X := random(imgDessin.Width-10)  + 5;
  Y := random(imgDessin.Height-10) + 5;
  lbCoords.Items.Append(inttostr(X)+';'+inttostr(Y));
  imgDessin.Canvas.Pen.Color := clBlack;
  imgDessin.Canvas.Brush.Style := bsClear;
  imgDessin.Canvas.MoveTo(X-5,Y);
  imgDessin.Canvas.LineTo(X+6,Y);
  imgDessin.Canvas.MoveTo(X,Y-5);
  imgDessin.Canvas.LineTo(X,Y+6);
  imgDessin.Canvas.Ellipse(x-4,Y-4,x+5,y+5);
end;
//----------------------------------------------------------------------------
procedure TfrmMain.btnTransfertClick(Sender: TObject);                 // 7 p.
var X1, Y1, X2, Y2, I, P, T : integer;
    S : string;
begin
  if (lbCoords.Items.Count < 2)
  then sgSegments.RowCount := 2
  else sgSegments.RowCount := lbCoords.Items.Count div 2+1;
  for I := 0 to (lbCoords.Items.Count div 2) - 1 do begin
    S := lbCoords.Items[2*I];
    P := pos(';',S);
    X1 := strtoint(copy(S, 1, P-1));
    Y1 := strtoint(copy(S, P+1, length(S)));  // length(S) ou 3 ou 4 ou 5 est OK
    S := lbCoords.Items[2*I+1];
    P := pos(';',S);
    X2 := strtoint(copy(S, 1, P-1));
    Y2 := strtoint(copy(S, P+1, length(S)));  // length(S) ou 3 ou 4 ou 5 est OK
    T := round(sqrt(sqr(X2-x1)+sqr(Y2-Y1)));
    sgSegments.Cells[0,I+1] := inttostr(X1);
    sgSegments.Cells[1,I+1] := inttostr(Y1);
    sgSegments.Cells[2,I+1] := inttostr(X2);
    sgSegments.Cells[3,I+1] := inttostr(Y2);
    sgSegments.Cells[4,I+1] := inttostr(T);
  end;
end;
//----------------------------------------------------------------------------
procedure TfrmMain.btnDessinClick(Sender: TObject);                    // 8 p.
var X1, Y1, X2, Y2, I, T, TMIN, TMAX : integer;
    COULEUR : TColor;
begin
  if(sgSegments.Cells[4,1] <> '') then begin
    // recherche TMIN et TMAX
    TMIN  := strtoint(sgSegments.Cells[4,1]);
    TMAX := TMIN;
    for I := 1 to sgSegments.RowCount-1 do begin
      T  := strtoint(sgSegments.Cells[4,I]);
      if T > TMAX then TMAX := T;
      if T < TMIN then TMIN := T;
    end;
    // dessin des segments en couleur appropriée
    for I := 1 to sgSegments.RowCount-1 do begin
      X1 := strtoint(sgSegments.Cells[0,I]);
      Y1 := strtoint(sgSegments.Cells[1,I]);
      X2 := strtoint(sgSegments.Cells[2,I]);
      Y2 := strtoint(sgSegments.Cells[3,I]);
      T  := strtoint(sgSegments.Cells[4,I]);
      if T >= TMAX
      then COULEUR := clLime
      else if T <= TMIN
           then COULEUR := clRed
           else COULEUR := clBlack;
      imgDessin.Canvas.Pen.Color := COULEUR;
      imgDessin.Canvas.MoveTo(X1,Y1);
      imgDessin.Canvas.LineTo(X2,Y2);
    end;
  end;
end;
//----------------------------------------------------------------------------
end.
```