

IV. A Delphi grafikája

1. Alapok

GDI (Graphics Device Interface)	<i>képernyő, nyomtató, rajzgép egységes kezelése</i>
felismeri az aktuális periféria tulajdonságait	<i>koordináta-rendszerek dimenziók színek nincs animációt, egész szám koordináták, kétdimenziós</i>
leíró a fizikai eszközzel való kapcsolatra	<i>rajzeszközök egy halmaza, attribútumokat,</i>
háromféle grafika	<i>tervezésekor grafikus vezérlőket grafikus alprogramrendszer</i>
Form, Printer, Image	Canvas tulajdonsága

2 Vezérlők

Szövegek megjelenítése (TLabel)

<i>Elhelyezkedés</i>	BoudsRect	Left	Top
	Height	Width	AutoSize
	Align	Alignment	
<i>Szöveg</i>	Caption		
	Font	Show	Color
		Accelerator	
	WordWrap	Transparent	

```
procedure SetTextBuf(Buffer: PChar);  
function GetTextBuf(Buffer: PChar; BufSize: Integer): Integer;  
function GetTextLen: Integer;
```

Invalidate, Repaint, Refresh

Ábrák - TShape

<i>Elhelyezkedés</i>	BoudsRect	Left	Top
	Height	Width	Align
<i>Rajzolás</i>	Brush	TBrush	
	Pen	TPen	
	Shape	<i>TShapeType</i> = (stRectangle, stSquare, stRoundRect, stRoundSquare, stEllipse, stCircle);	

Képek - TImage

<i>Elhelyezkedés</i>	BoudsRect	Left	Top
	Height	Width	Align
	AutoSize	Center	
<i>Megjelenítés</i>	Canvas	Picture	Stretch

3. A Canvas objektum alapjellemzői

toll
ecset
font
bittérkép

Színek

```
type
    TColor=-(COLOR_ENDCOLORS + 1)..$2FFFFFFF; //pl. ($00bbggrr).

const
    clBackground=TColor(-COLOR_BACKGROUND - 1);
    clMenu=TColor(-COLOR_MENU - 1);
    clWindowFrame=TColor(-COLOR_WINDOWFRAME - 1);
    clBlack   = TColor($000000);
    clRed     = TColor($0000FF);
    clGreen  = TColor($008000);
    clBlue   = TColor($FF0000);
    clDkGray = TColor($808080);
    clWhite  = TColor($FFFFFF);

    clSajat :=RGB(10,12,100);
```

Objektumok Color tulajdonsága

Írásmódok

Canvas.PenMode

TPenMode = (... *pmBlack*, *pmWhite*, *pmCopy*, *pmXor*...)

Canvas.CopyMode

TCopyMode : longint

```
cmBlackness = BLACKNESS;      {minden fekete}
cmPatCopy   = PATCOPY;       {az ecsettel festünk}
cmSrcAnd    = SRCAND;
cmSrcCopy   = SRCCOPY;
```

Kivágás

Canvas.ClipRect

Trect

Koordináta-rendszerek

Az eszközkapcsolat leíró

Canvas.Handle

THandle

4. Pixelgrafika

Canvas

```
property Pixels[X, Y: Integer]: TColor;
```

Példa

```
for i:=10 to 100 do
  for j:=10 to 100 do
    Canvas.Pixel[i,j]:=clRed;
```

5. Rajzolás tollakkal

Canvas Pen

```
property Pen: TPen;
```

TPen - Color, Mode, Style, Width.

Color : TColor

Mode : PenMode

Style : TPenStyle;

TPenStyle=(psSolid, psDash, psDot, psDashDot, psDashDotDot, psClear,
psInsideFrame);

Width

Metódusok

SetColor, GetColor, SetStyle, GetStyle, SetWidth, GetWidth, SetMode

Canvas

```
property PenPos : TPoint;
```

Rajzolási metódusok

```
procedure MoveTo(X, Y: Integer);
procedure LineTo(X, Y: Integer);
procedure Polyline(const Points:array of TPoint);
procedure Arc(X1, Y1, X2, Y2,X3, Y3, X4, Y4: Integer);
```

Példa

```
procedure TForm1.Paint(Sender: TObject);
var i,j:integer;
const szin:array [1..15] of TColor =
  (clBlack,clMaroon,clGreen,
   clOlive,clNavy,clPurple,
   clTeal,clGray,clSilver,
   clRed,clLime,clBlue,
   clFuchsia,clAqua,clBlack);
  tipus: array [1..5] of TPenStyle =
  (psSolid, psDash, psDot, psDashDot,
   psDashDotDot);
begin
  for i:=1 to 5 do
    begin
      Canvas.Pen.Style:=tipus[i];
      for j:=1 to 15 do
        begin
          Canvas.Pen.Color:=szin[j];
          Canvas.Moveto(10, (50*i+3*j));
          Canvas.Lineto(100, (50*i+3*j));
        end;
      end;
    end;
end;
```

6 Festés

Canvas

Brush-TBrush

Color - TColor

Style TBrushStyle

```
type TBrushStyle = (bsSolid, bsClear, bsHorizontal, bsVertical, bsFDiagonal,  
bsBDiagonal, bsCross, bsDiagCross);
```

Bitmap - TBitmap

GetColor, SetColor, GetStyle, SetStyle, GetBitmap, SetBitmap

Festő metódusok

```
procedure Rectangle(X1, Y1, X2, Y2: Integer);  
procedure DrawFocusRect(const Rect: TRect);  
procedure RoundRect(X1, Y1, X2, Y2, X3, Y3: Integer);  
procedure Polygon(const Points: array of TPoint);  
procedure Ellipse(X1, Y1, X2, Y2: Integer);  
procedure Pie(X1, Y1, X2, Y2, X3, Y3, X4, Y4: Integer);  
procedure Chord(X1, Y1, X2, Y2, X3, Y3, X4, Y4: Integer);  
procedure FloodFill(X, Y: Integer; Color: TColor;  
FillStyle: TFillStyle);
```

Példa

```
procedure TForm1.Paint(Sender: TObject);  
var i,j:integer;  
const szin:array [1..15] of TColor =  
      (clBlack,clMaroon,clGreen,  
      clOlive,clNavy,clPurple,  
      clTeal,clGray,clSilver,  
      clRed,clLime,clBlue,  
      clFuchsia,clAqua,clBlack);  
      tipus: array [1..8] of TBrushStyle =  
      (bsSolid,bsClear,bsBDiagonal,  
      bsFDiagonal,bsCross,bsDiagCross,  
      bsHorizontal,bsVertical);  
begin  
  Canvas.Pen.Style:=psSolid;  
  Canvas.Pen.Color:=clBlack;  
  for i:=1 to 8 do  
    begin  
      Canvas.Brush.Style:=tipus[i];  
      Canvas.Brush.Color:=szin[i];  
      Canvas.Rectangle(110,10+30*(i-1),200,  
        10+30*(i-1)+28);  
    end;  
end;  
end;
```

7 Írás

Canvas.Font - TFont

Color - TColor

Height

Size

Name

TFontName = string[LF_FACESIZE - 1];

TFontStyles = set of TFontStyle;

Style

TFontStyle=(fsBold, fsItalic, fsUnderline, fsStrikeOut);

Pitch

TFontPitch = (fpDefault, fpVariable, fpFixed);

Példa

```
FontDialog1.Font:=Canvas.Font;
if FontDialog1.Execute then
begin
  Canvas.Font:=FontDialog1.Font;
end;
```

Metódusok

```
procedure TextOut(X, Y: Integer; const Text: string);
function TextHeight(const Text: string): Integer;
function TextWidth(const Text: string): Integer;
procedure TextRect(Rect: TRect; X, Y: Integer; const Text: string);
```

Példák

```
procedure TForm1.Paint(Sender: TObject);
var i :integer;
const
fonttip : array [1..4] of TFontStyle =
  (fsBold,fsItalic,fsUnderline,fsStrikeout);
fontnev : array [1..3] of string[LF_FACESIZE-1] =
  ('Times New Roman', 'Arial',
  'Courier New');
begin
for i:=1 to 5 do
begin
  Canvas.Font.Height:=i*18;
  if (i<>1) then Canvas.Font.Style:=[fonttip[i-1]];
  Canvas.Font.Name:=fontnev[i mod 3];
  Canvas.TextOut(210,10+45*(i-1), 'Canvas');
end;
end;

procedure TForm1.MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState;
  X, Y: Integer);
begin
  FontDialog1.Font:=Canvas.Font;
  if FontDialog1.Execute then
  begin
    Canvas.Font:=FontDialog1.Font;
    Canvas.TextOut(X,Y, 'FontDialog');
  end;
end;
```

8. Bitképek

Bitképek - *TBitmap*

API *hBitmap*, *hPalette*

Tulajdonságok

```
property Canvas: TCanvas;  
property Handle: HBITMAP;  
property Palette: HPALETTE;
```

Metódusok

```
function GetHeight: Integer;  
function GetWidth: Integer;  
procedure SetWidth(Value: Integer);  
procedure SetHeight(Value: Integer);
```

Grafikus objektumok lemezállományban

```
Bitmap.LoadFromFile('c:\bitmap.bmp');  
Bitmap.SaveToFile('c:\bitmap.bmp');
```

Bitképek festése

```
procedure Draw(X, Y: Integer; Graphic: TGraphic);  
procedure StretchDraw(const Rect: TRect; Graphic: TGraphic);
```

Példák

```
procedure TForm1.Paint(Sender: TObject);  
var BitMap : TBitmap;  
begin  
  Bitmap := TBitmap.Create;  
  Bitmap.LoadFromFile('c:\bitmap.bmp');  
  Form1.Canvas.Draw((Form1.Width div 2) -  
                    (Bitmap.Width div 2),  
                    (Form1.Height div 2) -  
                    (Bitmap.Height div 2), Bitmap);  
end;  
  
procedure TForm1.Paint(Sender: TObject);  
var BitMap : TBitmap;  
begin  
  OpenFileDialog.Filter:='BitMap | *.bmp';  
  if OpenFileDialog.Execute then  
  begin  
    Bitmap := TBitmap.Create;  
    Bitmap.LoadFromFile(OpenDialog1.FileName);  
    Form1.Canvas.Draw(X,Y, Bitmap);  
  end;  
end;
```

9. Egyéb rajzolósi lehetőségek

A Handle tulajdonság

Leképezési módok

Leképezési mód	logikai egység	x-teng. iránya	y-teng. iránya
MM_TEXT (*)	képpont	jobbra	lefelé
MM_LOMETRIC	0.1 mm	jobbra	felfelé
MM_HIMETRIC	0.01 mm	jobbra	felfelé
MM_LOENGLISH	0.1 inch	jobbra	felfelé
MM_HIENGLISH	0.01 inch	jobbra	felfelé
MM_TWIPS	1/1440 inch	jobbra	felfelé
MM_ISOTROPIC	(x=y)*	választható	választható
MM_ANISOTROPIC	(x<>y)*	választható	választható

```
function SetMapMode(DC HDC; MapMode: Integer) : Integer;  
function GetMapMode(DC:HDC) : Integer;
```

Példa

```
procedure TForm1.Paint(Sender: TObject);  
  var i,j:integer;  
  const szin:array [1..15] of TColor =  
    (clBlack,clMaroon,clGreen,  
     clOlive,clNavy,clPurple,  
     clTeal,clGray,clSilver,  
     clRed,clLime,clBlue,  
     clFuchsia,clAqua,clBlack);  
  tipus: array [1..5] of TPenStyle =  
    (psSolid, psDash, psDot, psDashDot,  
     psDashDotDot);  
  mm:integer=0;  
begin  
  mm:=SetMapMode(Canvas.Handle,MM_LOMETRIC);  
  for i:=1 to 5 do  
    begin  
      Canvas.Pen.Style:=tipus[i];  
      for j:=1 to 15 do  
        begin  
          Canvas.Pen.Color:=szin[j];  
          Canvas.Moveto(10,-(50*i+3*j));  
          Canvas.Lineto(100,-(50*i+3*j));  
        end;  
      end;  
    SetMapMode(Canvas.Handle,mm);  
  end;
```

10. Nyomtatás

uses Printers;

Printer

TObject leszármazottja

Tulajdonságok

Canvas

Fonts (*GetFont*)

Handle (*GetHandle*)

PageHeight (*GetPageHeight*)

PageWidth (*GetPageWidth*)

Nomtató metódusok

BeginDoc

NewPage

EndDoc

```
with Printer do
begin
  BeginDoc;
  {Rajzolás a Canvas objektumra, például:
  Canvas.MoveTo(10,10);
  Canvas.LineTo(100,100)
  }
  EndDoc;
end;
```

Egyéb információk

Nyomtató sorok

CreatePQ

DeletePQ

OpenJob

DeleteJob

Nyomtatók adatai

ExtDeviceMode

type

```
TExtDevMode = function(Wnd: HWND; Driver: THandle;
  DevModeOutput: PDevMode; DeviceName, Port: PChar;
  DevModeInput: PDevMode; Profile: PChar; Mode: Word): Word;
```

const

```
  dm_Orientation      = $00000001;
  dm_PaperSize        = $00000002;
```

DeviceCapabilities

type

```
TDevCaps=function(DeviceName, Port: PChar; index: Word;
  OutPut: PChar; DevMode: PDevMode): Longint;
```

const

```
  dc_TrueType        = 15;
  dc_Orientation      = 17;
```

API

```
function ResetDC(aHdc: HDC; DevMode: PDevMode): HDC;
```