

80例上手 Delphi 7编程

一本不可多得的程序员进阶完全手册

程旭松 许哲平 万 重 编著

- 编程高手为你精心打造的80个精彩实例
- 双色印刷使你轻松把握编程技巧
- 赏心悦目的版式设计使你阅读更轻松
- 大量可以重复利用的实例代码
- 一个实例教给你一种新的编程思路
- 学习程序规划设计的最优方案
- API函数的多层次应用与掌握
- 常用控件的学习与应用

超值精美光盘

- 全书实例完整源代码——“拿来就用”的现成资源
- 全书实例可执行程序——本程序执行效果先睹为快
- 电脑报·编程乐园精华——众多编程爱好者经验分享
- Delphi控件——让Delphi编程更轻松
- 编程素材——丰富的图标、音效等素材为Delphi 7编程锦上添花
- 6 Delphi爱好者精彩范例软件——经典实例让编程更上一层楼

80例上手 Delphi 7编程

程旭松 许哲平 万 重 编著

江苏工业学院图书馆
藏书章



山东电子音像出版社出版

内容提要

学编程的人都知道，无论哪一种程序语言，都是学会容易学精难！有没有一种好的方法能够使人快速上手呢？本手册就是这样一个桥梁，能够带你快速掌握“程序员修炼之道”！

本手册精心选用生动有趣、涉及面广的 Delphi 7 程序语言实用案例，分为界面设计、系统设置、硬件访问与控制、控件应用、文件操作、图形图像多媒体、数据库应用、网络与通讯、趣味游戏 9 个章节，对编程思路和实例代码进行了深入浅出的详细解析，帮助你通过实例代码掌握编程思想与方法，迅速提高自己的程序设计水平。光盘中更提供了全部实例的源代码，使你可以“拿来就用”！

精彩光盘

- ★ 全书实例完整源代码——“拿来就用”的现成资源
- ★ 全书实例可执行程序——程序执行效果先睹为快
- ★ 电脑报·编程乐园精华——众多编程爱好者经验分享
- ★ Delphi 控件——让 Delphi 编程更轻松
- ★ 编程素材——丰富的图标、音效等素材为 Delphi 7 编程锦上添花
- ★ Delphi 爱好者精彩范例软件——经典实例让编程更上一层楼

书 名：80 例上手 Delphi 7 编程
编 著：程旭松 许哲平 万 重
责任编辑：李 萍
执行编辑：范晓霞
封面设计：邓玉萍
组版编辑：李品娟
监 制：谢均建
出版单位：山东电子音像出版社
地 址：济南市胜利大街 39 号
邮政编码：250001
电 话：(0531)2060055-7616
发 行：山东电子音像出版社
经 销：各地新华书店、报刊亭
C D 生产：北京中联鸿远光盘科技发展有限公司
文本印刷：重庆升光电力印务有限公司
开本规格：787mm × 1092mm 1/16 18.25 印张 600 千字
版 本 号：ISBN 7-900398-58-9
版 次：2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷
定 价：28.00 元(1CD+手册)



★全书实例完整源代码及可执行程序

全书 80 个实例完整源代码及可执行程序

光盘根目录

★电脑报·编程乐园精华

VB 实例与技巧 (181 篇)

Delphi 实例与技巧 (89 篇)

VC、C++ Builder、Turbo C 实例与技巧 (50 篇)

PowerBuilder 实例与技巧 (25 篇)

VFP 实例与技巧 (31 篇)

编程乐园精华\vb

编程乐园精华\delphi

编程乐园精华\VC

编程乐园精华\PowerBuilder

编程乐园精华\VFP

★Delphi 控件

TW3DCanvas (显示立体文字及图形)

BUPack (最大的 VCL 控件包之一)

coolcontrol (特殊界面效果控件)

CoolFormD5 (不规则窗体控件)

CPort (串口 VCL 组件)

DelphiX (DirectX 编程控件)

DevExpress (OFFICE XP 界面控件)

disksn (读硬盘序列号控件)

eltreelite (代替 TreeView 等的 VCL 控件)

flatstyle (Flat 样式控件)

FolderTree (改良版 TTreeView 控件)

Fxtool (图形/LABEL 的花絮控件)

gif (显示 Gif 动画的控件)

GifForm (生成有换肤功能的程序)

GIFImageD6 (显示图形的 GIF 控件)

graphicex (支持非常多的图像格式的控件包)

Jandraw (改良版的 TImage 控件)

msocket (用于广播的 SOCKET 类)

PicShow (提供 122 种不同的图形显示特效)

SakEmail (强大的电子邮件的 VCL 控件)

textfade (文字浮现的特效控件)

trayicon (将程序加入 Windows 桌面整理列的控件)

vcl_filefind (查找文件的控件)

vcl_rplib275_d6 (RX Library 控件包)

xpmenu (提供像 XP 那样的菜单)

Delphi 控件\3DCANVAS.exe

Delphi 控件\BUPack.exe

Delphi 控件\coolcontrol.exe

Delphi 控件\CoolFormD5.exe

Delphi 控件\CPort.exe

Delphi 控件\DelphiX.exe

Delphi 控件\DevExpress.exe

Delphi 控件\disksn.exe

Delphi 控件\eltreelite.exe

Delphi 控件\flatstyle.exe

Delphi 控件\FolderTree.exe

Delphi 控件\Fxtool.exe

Delphi 控件\gif.exe

Delphi 控件\GifForm.exe

Delphi 控件\GIFImageD6.exe

Delphi 控件\graphicex.exe

Delphi 控件\Jandraw.exe

Delphi 控件\msocket.exe

Delphi 控件\PicShow.exe

Delphi 控件\SakEmail.exe

Delphi 控件\textfade.exe

Delphi 控件\trayicon.exe

Delphi 控件\vcl_filefind.exe

Delphi 控件\vcl_rplib275_d6.exe

Delphi 控件\xpmenu.exe

★编程素材

图标

素材\图标\电脑
素材\图标\动物
素材\图标\人物
素材\图标\植物
素材\图标\自然
素材\图标\爱心
素材\图标\箭头
素材\图标\交通
素材\图标\节日
素材\图标\食品
素材\图标\手势
素材\图标\书写
素材\图标\物品

音效

素材\音效\动物
素材\音效\交通
素材\音效\军事
素材\音效\生活
素材\音效\娱乐
素材\音效\自然

★ Delphi 爱好者精彩范例软件

[illegible]

目 录

第1章 界面设计

本章为你准备了7个编程实例，内容包括制作屏保程序、给Memo组件加框、制作相框效果的窗体、制作桌面小精灵以及自做漂亮的VCD播放器等等。这些实例的制作效果都非常突出，简单易学，更重要的是相信你能从中学到不少编程技巧。

- 1** 实例1 制作屏幕保护程序
- 4** 实例2 给Memo组件的显示区域加边框
- 6** 实例3 制作相框效果的窗体
- 9** 实例4 实现带阴影效果的画框
- 12** 实例5 制作几何形状的窗体
- 16** 实例6 制作桌面小精灵
- 19** 实例7 制作一个漂亮的VCD播放器

第2章 系统设置

本章准备了10个实例供大家学习，内容涉及如何通过程序获取器信息、文件日期信息、获取Windows系统中相关目录的位置，实现目录的复制、删除和移动，更换壁纸和启动关闭屏保，隐藏任务栏和桌面图标，格式化磁盘等等。你可以根据自己对Delphi掌握的程度，有选择性的学习。

- 23** 实例1 驱动器信息轻松获取
- 26** 实例2 获取文件日期信息
- 29** 实例3 获取Windows、System和Temp目录
- 31** 实例4 打造文件管理器
- 34** 实例5 如何实现目录的复制、删除和移动
- 37** 实例6 更换壁纸和启动、关闭屏保
- 40** 实例7 让你的任务栏和桌面图标都“藏”起来
- 43** 实例8 重启或关闭计算机就这么简单
- 46** 实例9 制作实用的托盘程序
- 50** 实例10 在编程中实现磁盘格式化

第3章 硬件访问与控制

在本章的10个实例中，你可以学到硬件访问与控制各个方面的编程技巧。从运用各种技术获取系统的硬件信息，到对这些硬件进行一些属性设定，涉及到基本硬件操作的诸多方面，例如：从注册表中读取各类文件信息，读取内存状态信息，获取CPU信息，获取和设置显示器模式等等。

- 53** 实例1 从注册表中读取各类硬件信息
- 57** 实例2 通过API函数读取电源电池信息
- 59** 实例3 动态监视内存状态巧实现
- 63** 实例4 获取CPU相关信息
- 67** 实例5 获取和设置显示器模式
- 70** 实例6 用Delphi实现硬盘和光驱的操作
- 74** 实例7 获取显卡和声音媒体设备信息
- 78** 实例8 键盘的监测、控制和模拟
- 81** 实例9 打造超强鼠标
- 85** 实例10 两招搞定端口信息获取

第4章 控件应用

Delphi之所以受到广大程序员的青睐，其中有一个重要的原因就是它拥有丰富多彩、功能强大的第三方控件。本章主要讲述通过TShellTreeView、TShellComboBox、TShellListView三个控件实现类似Windows系统资源管理器功能，利用文件分割器控件division.zip打造自己的文件分割器等方面的内容。

- 92** 实例1 “我的资源管理器”自己做
- 94** 实例2 打造自己的文件分割器
- 97** 实例3 制作带背景图的标签
- 99** 实例4 编写一个滚动提示控件
- 103** 实例5 在控件中添加消息处理过程
- 105** 实例6 自制邮件发送客户端
- 108** 实例7 将系统中的COM接口包装为控件
- 109** 实例8 实现LISTBOX控件折行显示的妙招

第5章 文件操作

统计中英文字数、播放Flash、在Delphi中操作Excel、删除或移动正在操作的文件、实现文件拖放功能等等，都是我们在用机过程中时常面对的问题，本章就通过Delphi编程技术，实现相关功能。在这里，你不仅仅可以学到基本的编程技巧，还提供了许多宝贵的经验和大家分享。

- 112** 实例 1 实现中英文字数统计超简单
- 114** 实例 2 将 SWF 文件转换成 EXE 文件
- 117** 实例 3 将 EXE 文件转换成 SWF 文件你也能行
- 119** 实例 4 在 Delphi 中操作 Excel
- 121** 实例 5 灵活使用 INI 文件编程
- 124** 实例 6 删除或移动正在使用的文件
- 127** 实例 7 实现应用程序的文件拖放功能
- 129** 实例 8 从木马程序看文件捆绑技术

第 6 章 图形图像多媒体

图像的效果处理对于许多行业，例如设计、建筑、GIS 等等来说都是非常重要的，例如艺术字效果、水彩效果、失真效果、彩色电视机雪花效果、浮雕效果、渐变色效果等等，让人看起来非常美观，本章主要就向你讲述通过 Delphi 编程技术实现特殊效果的内容。

- 134** 实例 1 在 Delphi 中实现艺术字效果
- 138** 实例 2 用点像素法实现图像特效效果
- 141** 实例 3 利用 ScanLine 属性实现特殊图像效果
- 145** 实例 4 用 FillRect 实现的浮雕和编织带效果
- 147** 实例 5 制作具有“换肤”功能的个性相册
- 151** 实例 6 通过编程实现 Photoshop 图层效果
- 157** 实例 7 渐变色和动画矩形效果教你做
- 160** 实例 8 制作动画个性屏保
- 164** 实例 9 在 IntraWeb 中显示图片数据
- 168** 实例 10 Mapobjects 实现世界地图系统

第 7 章 数据库应用

本章同样为大家准备了 10 个实例，从运用原始技术动态创建表格，到高级的 IntraWeb 技术、Rave 数据库报表以及决策图的实现，包括了数据库开发过程中的一些主要问题，我们都为你进行了详细的讲解和分析，让你在学习之后真正做到举一反三。

- 173** 实例 1 用 Session 操纵 BDE
- 176** 实例 2 基于 SQL 语句的短消息管理系统
- 180** 实例 3 文本和数据记录的转换
- 183** 实例 4 手动创建数据表格

- 186** 实例 5 设计主从关系表格
- 188** 实例 6 制作决策图效果
- 192** 实例 7 实现多媒体数据库
- 196** 实例 8 在 Rave 中显示数据表格中的内容
- 198** 实例 9 利用 IntraWeb 技术实现数据操作
- 201** 实例 10 利用 ADO 控件从 SQL Server 2000 中读取数据

第 8 章 网络与通讯

互联网的普及给我们的生活带了越来越多的方便和惊喜，通过网络，我们可以轻而易举地实现软件升级、信息发送、软件下载续传、网络文件传输等功能，其实这些看似“不一般”的技术，通过 Delphi 编程技术都可以实现，不信的话就翻开本章内容看看吧！

- 205** 实例 1 制作升级精灵程序
- 212** 实例 2 断点续传软件巧制作
- 219** 实例 3 实现邮件群发功能
- 223** 实例 4 编写 P2P 文件传输程序
- 228** 实例 5 用 Delphi 实现网络信使服务
- 231** 实例 6 轻松实现网址连接状态检测
- 233** 实例 7 从互联网校对系统时间

第 9 章 趣味游戏

“好的游戏 = 优秀的游戏设计思想 + 精美的游戏素材 + 可靠的程序代码”，这是做出一个好游戏的基本规则。本章就从思想、素材、程序这三个方面向你讲述 Delphi 游戏编程的技术。从游戏入门基础知识，到运用 OpenGL 技术开发游戏，基本上涉及到游戏开发过程中的一些关键问题和技术。

- 236** 实例 1 制作键盘打字练习软件
- 240** 实例 2 制作流行的拼图游戏
- 247** 实例 3 乒乓球游戏制作
- 252** 实例 4 实现小狗过桥游戏
- 255** 实例 5 打地鼠游戏（界面实现）
- 260** 实例 6 打地鼠游戏（主体部分）
- 265** 实例 7 扫雷游戏（界面实现）
- 270** 实例 8 扫雷游戏（主体部分）
- 275** 实例 9 用 OpenGL 实现旋转星球
- 279** 实例 10 用 OpenGL 实现星空闪电图

第1章

界面设计

本章要点：

本章为你准备了7个编程实例，内容包括制作屏保程序、给Memo组件加框、制作相框效果的窗体、制作桌面小精灵以及自做漂亮的VCD播放器等等。这些实例的制作效果都非常突出，简单易学，更重要的是相信你能从中学到不少编程技巧。

实例 1

制作屏幕保护程序

【程序说明】

用户在使用电脑的时候，经常会暂时离开电脑去处理其他事情，一般情况下可以使用加密屏幕保护程序来防止其他无关人员使用电脑。那么，怎样在程序中实现屏幕保护程序呢？本例将为你讲述这个问题。程序运行效果如图所示：



【编程思路】

让窗体覆盖整个屏幕，隐去鼠标的显示，就完成了—个屏幕保护程序的雏形了，再在鼠标的 OnMouseMove 和 OnClick 事件上加上事件恢复对鼠标的显示和启动密码验证对话框，在填入

正确的密码后关闭整个程序，一个完整的屏保程序就完成了。Delphi 中的函数 `ShowCursor` 是专门用来控制鼠标的隐藏和显示，该函数定义如下：

```
int ShowCursor(  
    BOOL bShow // 鼠标显示参数  
);
```

当参数 `bShow` 取值为 `False` 时，鼠标自动隐藏；当参数 `bShow` 取值为 `True` 时，鼠标自动显示。

【编程步骤】

(1) 新建一个应用程序，将窗体的 `FormStyle` 属性和 `WindowState` 属性分别设置为 `fsStay OnTop` 和 `wsMaximized`，将 `Color` 属性设置为 `clBlack`。

(2) 在窗体上加入 1 个 `Image` 组件和 1 个 `Timer` 组件，将 `Image` 组件的 `Align` 属性和 `Center` 属性分别设置为 `alClient` 和 `True`；将 `Timer` 组件的 `Interval` 属性设置为 3000。

(3) 新建一个窗体，在窗体上加入 1 个 `Edit` 组件和 2 个 `Button` 组件，将 `Edit` 组件的 `Text` 属性设置为空，将 2 个 `Button` 组件的 `Caption` 属性分别设置为“确定”和“取消”。

【代码解析】

主窗体代码：

```
unit MainUnit;  
  
interface  
  
uses  
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
    ExtCtrls, JPEG;  
  
type  
    TMainForm = class(TForm)  
        Image1: TImage;  
        Timer1: TTimer;  
        procedure Timer1Timer(Sender: TObject);  
        procedure FormCreate(Sender: TObject);  
        procedure Image1Click(Sender: TObject);  
        procedure FormKeyPress(Sender: TObject; var Key: Char);  
    private  
        { Private declarations }  
    public  
        { Public declarations }  
    end;  
  
var  
    MainForm: TMainForm;  
    AppPath: String;  
    i: Integer;  
    const sCount= 3; // 设置屏保图片总数
```



```
implementation

uses PWFormUnit;

{$R *.DFM}

procedure TMainForm.Timer1Timer(Sender: TObject); // 设置屏保图片循环
begin
    if i < sCount then
    begin
        i := i + 1;
        Image1.Picture.LoadFromFile(AppPath+ IntToStr(i)+ '.jpg'); // 读取屏保图片
    end
    else
        i := 0;
    end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
    i := 0;
    AppPath := ExtractFilePath(ParamStr(0));
    Image1.Picture.LoadFromFile(AppPath+ '3.jpg');
    ShowCursor(False); // 隐藏鼠标
end;

procedure TMainForm.Image1Click(Sender: TObject);
begin
    PWForm.Show;
    ShowCurSor(True); // 显示鼠标
end;

procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char);
begin
    PWForm.Show;
    ShowCurSor(True);
end;

procedure TMainForm.FormKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if (Key=VK_F4) and ([ssAlt]=shift) then // 屏蔽 Alt+F4 键让程序不能被关闭
        Key:=0;
    end;

end.
```

【程序小结】

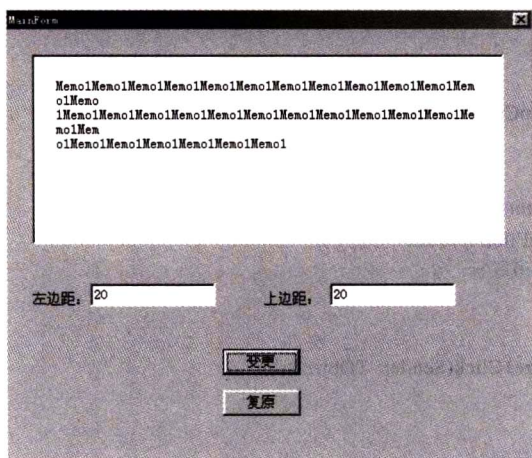
本例主要是依靠控制鼠标隐藏和显示的函数 **ShowCursor** 来实现屏幕保护功能，该函数只是众多鼠标控制函数中的一个，你甚至还可以通过其他鼠标函数定位鼠标，或者指定鼠标的移动范围，有兴趣的朋友可以深入研究一下。

实例 2

给 Memo 组件的显示区域加边框

【程序说明】

使用过 Delphi 的 Memo 组件的人都知道, 在 Memo 组件的 Lines 属性中加入的文字都是从左边界开始显示的, 在文字太多的时候就显得不太美观。如何给 Memo 组件设置一个像网页那样的边界呢? 本例将为你解决这个问题。程序运行效果如图所示:



【编程思路】

其实要实现这一功能非常简单, 只要通过 API 函数 `SendMessage` 向系统发送消息 `EM_GETRECT` 得到组件的矩形边界, 修改后再通过 `SendMessage` 向系统发送消息 `EM_SETRECT` 设置组件的矩形边界就可以了。消息 `EM_GETRECT` 和 `EM_SETRECT` 的定义分别如下:

```
EM_GETRECT
WParam= 0 // 必须为 0
lParam= (LPARAM)(LPRECT)lprc; // 指向 RECT 结构体的指针

EM_SETRECT
wParam= 0; // 必须为 0
lParam= (LPARAM)(LPRECT)lprc // 指向 RECT 结构体的指针
```

【编程步骤】

(1) 新建一个应用程序, 在窗体上添加 1 个 Memo 组件、2 个 Label 组件、2 个 Edit 组件和 2 个 Button 组件。

(2) 如效果图所示在 Memo 组件的 Lines 属性中添加字符, 将 2 个 Label 组件的 Caption 属性分别设置为“左边距”和“上边距”, 将 2 个 Button 组件的 Caption 属性分别设置为“变更”

和“复原”，将 2 个 Edit 组件的 Text 属性设置为空。

【代码解析】

```
unit MainUnit;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TMainForm = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  MainForm: TMainForm;
  Rgt, Btm: Integer;
implementation

{$R *.DFM}

procedure TMainForm.Button1Click(Sender: TObject);  改变矩形边界
var
  Rect: TRect;
begin
  SendMessage(Memo1.Handle, EM_GETRECT, 0, LongInt(@Rect));  // 获取 Memo1 的矩形边界
  Rgt:= Rect.Right;  // 保存原始矩形边界宽度
  Btm:= Rect.Bottom;  // 保存原始矩形边界高度
  if Edit1.Text<> " then
    Rect.Left:= StrToInt(Edit1.Text)  // 更改 Rect 边界值
  else ShowMessage(' 请填入左边距 ');
  if Edit2.Text<>" then
    Rect.Top:= StrToInt(Edit2.Text)  // 更改 Rect 边界值
  else ShowMessage(' 请填入上边距 ');
  Rect.Top:= StrToInt(Edit2.Text);
  Rect.Right:= Rect.Right- StrToInt(Edit1.Text)+ 1;
  Rect.Bottom:= Rect.Bottom- StrToInt(Edit2.Text)+ 1;
```



```

    SendMessage(Memo1.Handle, EM_SETRECT, 0, LongInt(@Rect)); // 重绘 Memo1 的矩形边界
    Memo1.Refresh;
end;

procedure TMainForm.Button2Click(Sender: TObject); // 复原原始矩形边界
var
    Rect: TRect;
begin
    Rect.Left:= 1;
    Rect.Top:= 1;
    Rect.Right:= Rgt;
    Rect.Bottom:= Btm;
    SendMessage(Memo1.Handle, EM_SETRECT, 0, LongInt(@Rect));
    Memo1.Refresh;
end;

end.

```

【程序小结】

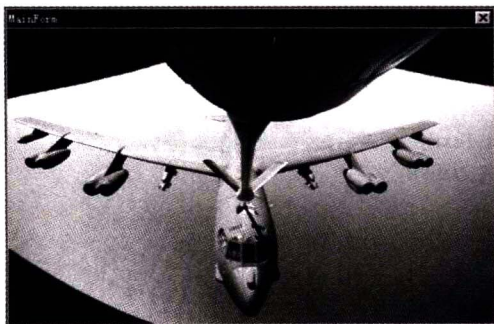
给 Memo 组件的显示区域加边框的原理，主要是通过 API 函数 SendMessage 向系统发送消息得到和设置 Memo 组件的矩形边界来实现，该方法作用于 RichEdit 组件同样合适；还原 Memo 组件显示区域的原理也非常简单，只要在更改 Memo 组件矩形边界前将原始的矩形边界保存下来就可以恢复了。

实例 3

制作相框效果的窗体

【程序说明】

本例介绍如何给窗体加上一个边框使窗体做出相框效果。程序运行效果如图所示：



【编程思路】

要实现窗体的相框效果，必须用到一些与画图有关的 API 函数。

(1) API 函数 **GetWindowDC** 可以获取整个窗体设备上下文，包括标题栏、菜单以及滚动条等。MSDN 中该函数声明如下：

```
HDC GetWindowDC(  
    HWND hWnd // 窗口句柄  
);
```

(2) API 函数 **ReleaseDC** 用于释放设备上下文。MSDN 中该函数的声明如下：

```
Int ReleaseDC(  
    HWND hWnd, // 窗口句柄  
    HDC hDC // 设备上下文句柄  
);
```

(3) API 函数 **CreatePen** 用于创建指定样式、指定宽度和指定颜色的画笔。MSDN 中该函数的声明如下：

```
HPEN CreatePen(  
    Int fnPenStyle, // 画笔样式  
    Int nWidth, // 画笔宽度  
    COLORREF crColor // 画笔颜色  
);
```

通过将参数 **fnPenStyle** 设置为不同的值，可以画实线、虚线和点划线等。

(4) API 函数 **DeleteObject** 用于删除逻辑画笔、画刷、字体、位图、区域及调色板等，并且释放与这些对象关联的系统资源。MSDN 中该函数声明如下：

```
BOOL DeleteObject(  
    HGDIOBJ hObject // 图像对象句柄  
);
```

(5) API 函数 **SelectObject** 用于将对象（画笔、字体等）选入设备上下文。MSDN 中该函数的声明如下：

```
HGDIOBJ SelectObject(  
    HDC hdc, // 设备上下文句柄  
    HGDIOBJ hgdiobj // 对象句柄  
);
```

(6) API 函数 **Rectangle** 用于实现窗体的边框。MSDN 中该函数声明如下：

```
BOOL Rectangle(  
    HDC hdc, // 设备上下文句柄  
    Int nLeftRect, // 矩形左上角坐标 x  
    Int nTopRect, // 矩形左上角坐标 y  
    Int nRightRect, // 矩形右下角坐标 x  
    Int nBottomRect // 矩形右下角坐标 y  
);
```

【编程步骤】

- (1) 新建一个程序，在窗体上添加 1 个 Image 组件。
- (2) 将 Image 组件的 Align 属性设置为 alClient。

【代码解析】

```
unit MainUnit;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, JPEG;

type
  TMainForm = class(TForm)
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
  private
    procedure WMNCPaint(var Msg: TWMNCPaint); message WM_NCPAINT;
    { Private declarations }
  public
    { Public declarations }
  end;

var
  MainForm: TMainForm;

implementation

{$R *.DFM}
procedure TMainForm.WMNCPaint(var Msg: TWMNCPaint);
var
  DC: hDC;
  Pen: hPen;
  OldPen: hPen;
  OldBrush: hBrush;
begin
  inherited; // 获取窗口文本上下文
  DC:= GetWindowDC(Handle);
  Msg.Result:= 1;
  Pen:= CreatePen(PS_SOLID, 3, RGB(225, 1, 1)); // 创建宽度为 3 的实现画笔
  OldPen:= SelectObject(DC, Pen); // 将新创建的画笔选入窗体的设备上下文
  OldBrush:= SelectObject(DC, GetStockObject(NULL_BRUSH)); // 将系统库存的空画刷选入窗体的设备上下文
  Rectangle(DC, 0, 0, MainForm.Width, MainForm.Height); // 绘制镜框
  SelectObject(DC, OldBrush);
  SelectObject(DC, OldPen);
  DeleteObject(Pen); // 删除画笔
  ReleaseDC(Handle, Canvas.Handle); // 释放设备上下文

end;
```