



求是科技

实效编程百例

Delphi

实效编程百例

★求是科技 施新刚 等编著

第二版

★学习编程技巧

★积累编程经验

★剖析功能模块

★突出应用实效



源代码光盘
CD-ROM



人民邮电出版社
POSTS & TELECOM PRESS

实效编程 6 例

Delphi

实 效 编 程 百 例

★求是科技 施新刚 等编著

(第二版)



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

Delphi 实效编程百例 (第二版) / 施新刚编著. — 北京: 人民邮电出版社, 2004.8
ISBN 7-115-12429-9

I. D... II. 施... III. 软件工具—程序设计 IV. TP311.56
中国版本图书馆 CIP 数据核字 (2004) 第 071424 号

内容提要

本书在第一版的基础上进行了修订, 通过 133 个精选的例子从多个方面讲述了如何利用 Delphi 开发应用程序, 涵盖了窗体控制、界面操作、多媒体控制、操作系统、程序控制、磁盘文件、数据库、网络与通信、算法、鼠标和键盘等内容。

本书的绝大多数实例重点突出了实用性, 多为模仿优秀软件的相关功能; 另一部分实例侧重帮助读者理解 Delphi 的重点以及难懂的概念, 在这部分实例中都是用最简单的代码说明最关键的问题。

在此次修订中, 纠正了原稿中程序和文档的问题, 删除那些实效性已经不是很强的案例, 根据当前开发特点, 新增更多实用的案例。并且考虑当前 Delphi 7 已经成为主流开发工具, 我们将全部程序都移到了 Delphi 7 版本。

本书适用于已经初步掌握 Delphi 编程概念、方法的读者阅读, 本书可以帮助读者迅速掌握实际应用中的各种经验、技巧。

实效编程百例

Delphi 实效编程百例 (第二版)

◆ 编 著 求是科技 施新刚 等
责任编辑 张立科

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 19.25

字数: 466 千字

2004 年 8 月第 2 版

印数: 9 001-15 000 册

2004 年 8 月北京第 3 次印刷

ISBN7-115-12429-9/TP·4075

定价: 32.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

第二版序

在近几年的计算机图书市场上，编程小实例（通常称为百例）类的图书已经很普遍，而且为很多读者所接受。经调查发现，绝大多数百例类的图书，主要面向了最初级读者，其内容在实用程度上相对较差，经常有很多纯粹娱乐性或很偏门的小技巧性的案例。

我们通过网络调研、召开读者座谈会等途径，进行调研，结果表明很多已经具备了一定开发能力的人（且以实际开发人员居多），他们也非常喜欢和需要这种百例类的图书，只是多数图书内容在实用性方面满足不了他们的需要。

为此，我们于 2002 年初策划，并于当年 6 月份前后出版了这套实效百例丛书的第一批，一套 4 本，主要针对当时最主流的开发工具（Visual Basic 6.0、Visual C++ 6.0、Visual FoxPro 6.0 以及 Delphi 6.0），书中非常强调实例的实用功效。

在内容安排上，本套丛书按照程序的完成功能进行案例分类，尽量将相关的内容归纳在一章中。方便读者在有实际需求的时候，能快速找到解决方法（参见书中相关实例）。

在每个实例的讲解过程中，基本都分为 3 个步骤：

实例目的——讲解本例的功能所在，指出本例要到达的目的和效果，让读者做到心中有数。

实现方法——讲解技术原理/设计思路，给出技术原理的合理解释、规范的算法和流程描述，便于读者阅读代码、学习程序设计方法。

程序代码——给出具体的实现过程，包括界面设计、编写代码和注释，读者可参照实现。

自出版以来，本套丛书中取得了不错的销售效果，在当前众多百例类图书中处于佼佼者的地位。承蒙广大读者的厚爱，给我们反馈了很多非常宝贵的意见和建议，除了指出图书中存在的一些问题、部分案例在功能存有不足之外，还针对当前软件开发的特点，提出了不少好的建议和期望。为此，我们决定将第一期的 4 本图书进行深入修订、改版：

首先，我们安排了富有开发经验的开发人员，对图书中所有案例程序进行了严格测试，找出问题，并仔细审阅图书文档，针对错误和不足之处，加以完善，让读者阅读起来更为轻松、顺畅。

其次，我们充分考虑读者建议，结合最新的技术手段以及当前软件开发特点，对原书中各案例进行了筛选，去掉哪些相对失去了实效性的案例。

最后，我们又进行了优秀案例的增补工作，引进了很多实效性好、技术性强、适用于当前实际开发的案例。

本书由求是科技策划，在本书写作过程中得到了郭瑞军、卞志强、郭琦、马伟勤、林陈雷、罗颂、刘敬、王作远、窦亮、陈艳华、陈林、张增强、张文毅、彭涛、王志伟、吴星、华剑锋、吴明辉、曹广鑫、常明、刘萌、黄超、孙炳文、童行行、张益贞、闫海荣、马方魁、罗建光、贾博、张双民、李廷文的帮助，在此表示感谢。

尽快我们进行了全面修订和升级，但由于水平有限，可能还会存于一定的问题和不足，恳请广大读者提出宝贵意见和建议。作者的网站 <http://www.cs-book.com>，欢迎访问。

编者

2004 年 6 月

目 录

第 1 章 窗体控制 1

- 实例 1 不用代码实现异形窗口 2
- 实例 2 制作透明的窗口 3
- 实例 3 让每个窗体都执行相同的 OnCreate 事件 4
- 实例 4 始终置前的窗体 6
- 实例 5 磁性窗口 7
- 实例 6 截获单击窗体最小化按钮事件 8
- 实例 7 使窗体永远最小化 9
- 实例 8 窗体背景色渐进效果 10
- 实例 9 控制 MDI 子窗体大小 12
- 实例 10 窗体分割 14
- 实例 11 显示倾斜文字 16
- 实例 12 文字颜色渐变 18
- 实例 13 透明图片 19
- 实例 14 颜色渐变的进度条 21
- 实例 15 可变背景窗体 22

第 2 章 界面操作 25

- 实例 16 动态对话框 26
- 实例 17 在状态条中插入可视控件 28
- 实例 18 动态缩小的图形 29
- 实例 19 放大图像 31
- 实例 20 浏览大图片 33
- 实例 21 抓取屏幕 35
- 实例 22 资源管理器 41
- 实例 23 智能查找列表框中的内容 42
- 实例 24 带动态提示的文本编辑器 45
- 实例 25 修改光标和程序的图标 47

- 实例 26 持续响应鼠标单击 51

- 实例 27 标题栏按钮 53

- 实例 28 拖放列表框中选中的对象 55

- 实例 29 创建多文档窗体 58

- 实例 30 嵌入式电子钟 59

- 实例 31 颜色下拉框 61

第 3 章 多媒体控制 64

- 实例 32 检测声卡是否存在 65

- 实例 33 视频播放器 65

- 实例 34 CD 播放器 68

- 实例 35 变换图像 70

- 实例 36 RealPlayer 音频播放器 72

- 实例 37 全屏播放电影 74

- 实例 38 播放 Flash 动画 76

- 实例 39 简单动画的实现 77

第 4 章 操作系统 79

- 实例 40 获得密码编辑框中的内容 80

- 实例 41 修改计算机名称 81

- 实例 42 设置墙纸和屏保程序 82

- 实例 43 动态调整屏幕分辨率 84

- 实例 44 获取系统硬件信息 87

- 实例 45 将文件打开的历史记录保存到注册表 88

- 实例 46 锁住计算机 92

- 实例 47 重新启动和关闭计算机 93

- 实例 48 获取 Windows 版本号和运行模式 95

- 实例 49 枚举可用字体 97

- 实例 50 获取打印机信息 98

实例 51	获取打印机队列的状态信息	99
实例 52	文本和图像的打印	101
实例 53	获得打印页边距	103
实例 54	利用 API 直接打印	104
实例 55	改变默认的打印机	110

第 5 章 程序控制

实例 56	产生程序序列号	113
实例 57	毫秒级的控制	118
实例 58	系统托盘程序	121
实例 59	隐藏任务栏	125
实例 60	枚举系统正在运行的程序	126
实例 61	查看 IE 浏览器是否运行	128
实例 62	调用 Windows 程序	129
实例 63	禁止运行同一程序或多个实例	132
实例 64	修改系统菜单	133
实例 65	禁用窗体右上角各按钮	136
实例 66	响应标题的双击事件	137
实例 67	使用剪切板复制图片	138
实例 68	利用内存映射实现数据交换	139
实例 69	取得下拉条的信息	142
实例 70	通过多个 Timer 控件执行多任务	144
实例 71	多线程应用程序	150

第 6 章 磁盘文件

实例 72	打开 CHM 文件	154
实例 73	获取操作系统的序列号	155
实例 74	获取驱动器容量	156
实例 75	查看驱动器类型	157
实例 76	递归法遍历磁盘目录	159
实例 77	获取文件属性	161
实例 78	将文件删除到回收站中	162
实例 79	清空回收站	164
实例 80	启动控制面板中的内容	166
实例 81	将声音集成在 EXE 文件中	168
实例 82	获取同文件扩展名相关联的应用程序	170
实例 83	读写 INI 文件	172

实例 84	弹出和关闭光驱	175
实例 85	更改文件的扩展名	176
实例 86	记录文件的读写	177
实例 87	从文件中存取组件	180

第 7 章 数据库

实例 88	处理数据库中日期型字段的显示与输入	184
实例 89	存取图像字段	187
实例 90	枚举 Access 数据库中自建表	191
实例 91	SQL 语句中使用通配符	192
实例 92	Delphi 数据集过滤技巧	193
实例 93	设置 ODBC 数据源	196
实例 94	SQL 语句嵌套	199
实例 95	动态创建数据库对象	201
实例 96	快速连接 SQLServer	204
实例 97	快速报表与报表预览	206
实例 98	动态改变 DBGrid 组件的颜色	208
实例 99	优化/恢复 FoxPro 数据库记录	210

第 8 章 网络与通信

实例 100	同步网络时间	216
实例 101	检测计算机是否联网	218
实例 102	映射网络驱动器	219
实例 103	检测 Modem 当前状态	220
实例 104	获取网卡地址	221
实例 105	获得主机名和 IP 地址	223
实例 106	判断网址是否存在	225
实例 107	监测局域网内计算机	226
实例 108	Ping 测试	228
实例 109	设置代理服务器	231
实例 110	自定义网页浏览器	233
实例 111	修改 IE 的主页与标题	239
实例 112	网络聊天——TCP	241
实例 113	广播信息——UDP	245
实例 114	电话线实现远端通信	249
实例 115	串口采集外设数据	258
实例 116	用 API 实现串口通信	259

实例 117 在 IE 工具栏中加入快捷图标	262	实例 125 模拟鼠标的单双击	283
第 9 章 算法	266	实例 126 模拟键盘输入	284
实例 118 进制转换	267	实例 127 限定鼠标区域	290
实例 119 随机选号	269	实例 128 截获鼠标移开事件	291
实例 120 多点曲线光滑拟和	272	实例 129 注册全局热键	292
实例 121 统计中英文字符数	276	实例 130 判断 PrintScreen 键是否按下	294
实例 122 小写金额转换成中文大写	277	实例 131 时隐时现的鼠标	295
实例 123 获取汉字拼音字头	279	实例 132 鼠标拖动无标题栏窗口	296
第 10 章 鼠标和键盘	281	实例 133 检测 Shift、Alt 和 Ctrl 键是否 按下	297
实例 124 代码控制光标	282		



第 1 章 窗体控制



- ❏ 不用代码实现异形窗口
- ❏ 制作透明的窗口
- ❏ 让每个窗体都执行相同的 OnCreate 事件
- ❏ 始终置前的窗体
- ❏ 磁性窗口
- ❏ 截获单击窗体最小化按钮事件
- ❏ 使窗体永远最小化
- ❏ 窗体背景色渐进效果
- ❏ 控制 MDI 子窗体大小
- ❏ 窗体分割
- ❏ 显示倾斜文字
- ❏ 文字颜色渐变
- ❏ 透明图片
- ❏ 颜色渐变的进度条
- ❏ 可变背景窗体

实例 1 不用代码实现异形窗口

实例目标

在论坛上经常看到有人谈论异形窗口的问题，但都不能得到比较好的解决办法：不是代码太长而不易记住，就是实现的异形窗口的效果不太好。本例实现一个异形窗口，不需要写一句代码就能轻松实现梦寐以求的异形窗口，效果如图 1-1 所示，标志的轮廓即为该异形窗口的边缘。

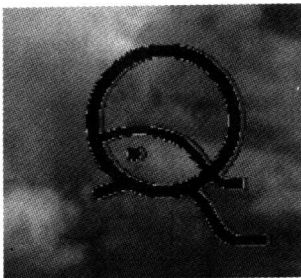


图 1-1 异形窗口

实现方法

窗体中通过一个 Image 控件加载一幅图片，期望图片中的图案成为异形窗体的轮廓。将窗体的 Transparent Color 属性设为 True, Transparent Color Value 属性设为图片底色，运行程序即可达到预期效果。

程序代码

- (1) 新建一个 Delphi 项目。在窗体上放置一个 Image 组件，并调入一个图片。
- (2) Form1 和 Image1 的主要属性设置如下：

```
object Form1: TForm1
  AutoSize = True
  BorderStyle = bsNone
  TransparentColor = True
  TransparentColorValue = clBlue
  object Image1: TImage
    AutoSize = True
  end
end
```

- (3) 因为窗体运行时没有标题栏，所以为 Image1 的 OnDBClick 编写代码。

```
procedure TForm1.Image1DBlClick(Sender: TObject);
begin
  Close;
end;
```

- (4) 编译运行就能看到异形窗口了。

注意：将窗体的 TransparentColorValue 设置成 clBlue，因为图像的底色为 clBlue，其他的图片请根据具体情况设置 TransparentColorValue 的取值。

说明：此实例仅能应用于 Windows 2000 或以上版本。Windows 9X 中不会出现如图 1-1 所示效果，只能显示原图片。

实例 2 制作透明的窗口

实例目标

在以前的 Delphi 版本中，制作透明窗口是比较复杂的，需要调用很多 API 函数，编写大量的代码。现在只需要两行代码就能制作出如图 1-2 所示的透明窗口。

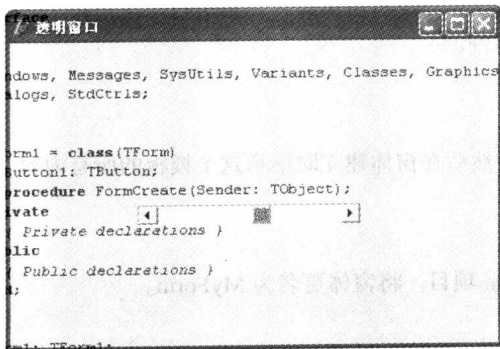


图 1-2 透明窗口

实现方法

利用 Delphi 7 中 Form 的 AlphaBlend 属性可以设置窗体是否透明，设置成 True 即可将整个窗体包括窗体里的控件设置为透明。AlphaBlendValue 属性则确定窗体的透明程度，设置成 255 就是不透明，设置成 0 则表示完全透明。

程序代码

- (1) 创建一个新的 Delphi 项目。在窗体上添加一个 TScrollBar 组件，其主要属性设置如下：

```
object ScrollBar1: TScrollBar
  Left = 112
  Top = 144
  Width = 193
  Height = 16
  Max = 255
  PageSize = 0
  Position = 125
  TabOrder = 0
  OnChange = ScrollBar1Change
end
```

- (2) 为窗体的 OnCreate 事件和 ScrollBar1 的 OnChange 事件添加如下代码。

```
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
  Form1.AlphaBlendValue:=ScrollBar1.Position;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Form1.AlphaBlend:=True;
  Form1.AlphaBlendValue:=ScrollBar1.Position;
end;
```

- (3) 编译运行。拖动窗体上的滚动条，窗体的透明度会随之更改。

实例 3 让每个窗体都执行相同的 OnCreate 事件

实例目标

根据代码的共用性，可以让每个窗体都执行相同的 OnCreate 事件。实际编程过程中，经常会遇到许多窗体都共同调用同一段代码的情况，最典型的应用就是软件的加密检验代码。这些代码是完全相同的，为了加大盗版商的破解难度，当启动所有窗体时，都要进行一次检验。本例就是向读者介绍如何让每个窗体都执行相同的 onCreate 事件。

实现方法

首先创建一个窗体模板，然后在窗体建立时继承这个模板的所有内容。

程序代码

(1) 创建一个新的 Delphi 项目。将窗体更名为 MyForm。

(2) 编写代码如下：

```
unit CommonForm;

interface

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

type
  TMyForm = class(TForm)
  protected
    constructor Create(Aowner: TComponent); override;
  end;

var
  MyForm: TMyForm;

implementation

{$R *.dfm}

constructor TMyForm.Create(Aowner: TComponent);
var
  i: Integer;
begin
  inherited Create(AOwner);
  with self do
  begin
    Caption := Caption + ' [未注册]';
    for i := 0 to ComponentCount - 1 do
    begin
      if (Components[i] is TButton) then
        (Components[i] as TButton).Cursor := crHandPoint;
    end;
  end;
end;

end.
```

(3) 添加一个窗体 Form1。在窗体上添加一个 Button，并设置其 Caption 属性为“显示 Form2”。编

写代码如下:

```
unit Unit1;

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

type
  TForm1 = class(TMyForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Unit2;

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  Form2.Show;
  Form1.Hide;
end;

end.
```

(4) 再添加一个 Form2。在窗体上添加两个 Button，并设置其 Caption 属性分别为“显示 Form1”、“退出”。代码如下:

```
unit Unit2;

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

type
  TForm2 = class(TMyForm)
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
```

```
implementation

uses Unit1;
{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
    Form2.Hide;
    Form1.Show;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
    Application.Terminate;
end;

procedure TForm2.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    Application.Terminate;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
    inherited;
end;
end.
```

(5) 编译运行，会看到一个窗体，标题上有“未注册”字样。单击【显示 Form2】按钮，在 Form2 窗体的标题栏也有“未注册”字样。表明这两个窗体都执行了相同的 OnCreate 事件。

实例 4 始终置前的窗体

实例目标

将一个窗体始终保存在最前，不被其他窗体覆盖，与一些流行的下载软件一样，总有一个小窗口在最前。运行本实例，效果如图 1-3 所示。

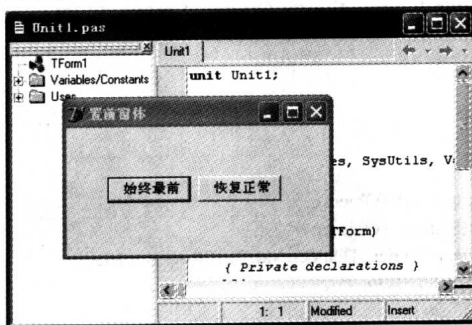


图 1-3 被设置为始终置前的窗体

实现方法

简单的方法可以将 Form 的 FormStyle 属性设置为 fsStayOnTop。利用 Windows API 函数 SetWindowPos

同样能够将窗体置前。SetWindowPos 函数的原型如下：

```

Bool SetWindowPos(HWND hWnd, //handle of window
  HWND hWndInsertAfter, //placement-order handle
  Int X, //horizontal position
  Int Y, //vertical position
  Int cx, //width
  Int cy, //height
  UINT uFlags //window-positioning flags);

```

程序代码

(1) 新建一个 Delphi 项目。在 Form1 上放置两个 Button 按钮，它们的主要属性设置为：

```

object Button1: TButton
  Caption = 始终最前
  TabOrder = 0
  OnClick = Button1Click
end
object Button2: TButton
  Caption = 恢复正常
  TabOrder = 1
  OnClick = Button2Click
end

```

(2) 为两个 Button 编写 OnClick 事件代码。

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  SetWindowPos(Form1.handle, HWND_TOPMOST, Form1.Left, Form1.Top, Form1.Width, Form1.Height, 0);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  SetWindowPos(Form1.handle, HWND_NOTOPMOST, Form1.Left, Form1.Top, Form1.Width, Form1.Height, 0);
end;

```

实例 5 磁性窗口

实例目标

类似于 Winamp 一样具有磁性化的效果，在窗口离屏幕边缘比较近的时候，窗口如同磁石一样，自动贴到屏幕边缘。

实现方法

利用窗口移动时产生的消息 WM_MOVE，编写一个过程，实现窗口位置的变化。

程序代码

(1) 创建一个新的 Delphi 项目。
(2) 编写程序代码如下：

```

unit unitFrmMain;

interface

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

```

```

type
  TForm1 = class(TForm)
  private
    { Private declarations }
    procedure WMMOVE(var Msg: TMessage); message WM_MOVE;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

{ TForm1 }

procedure TForm1.WMMOVE(var Msg: TMessage);
begin
  inherited;
  if Form1.Left < 20 then
    Form1.Left:=0;
  if Form1.Top < 20 then
    Form1.Top:=0;
  if (Screen.Width - Form1.Left -Form1.Width)<20 then
    Form1.Left :=Screen.Width -Form1.Width ;
  if (Screen.Height - Form1.Top -Form1.Height)<20 then
    Form1.Top :=Screen.Height -Form1.Height;
end;

end.

```

实例 6 截获单击窗体最小化按钮事件

实例目标

在窗体右上角上单击最小化按钮时，将此消息截获，并取消执行最小化事件，如图 1-4 所示。



图 1-4 截获单击窗体最小化按钮事件

实现方法

利用单击窗体上的最小化按钮时产生的消息 WM_SYSCOMMAND，编写一个过程，显示截获信息。

程序代码

(1) 新建一个 Delphi 项目。

(2) 编写程序代码如下:

```
unit Unit1;

interface

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

type
  TForm1 = class(TForm)
    procedure wmsyscommand(var msg:twmsyscommand);message wm_syscommand;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.wmsyscommand(var msg:twmsyscommand);
begin
  if (msg.CmdType=sc_minimize) or (msg.cmdtype=sc_restore) then
  begin
    messagebox(0,'已经捕获了最小化按钮事件!','信息',48);
    msg.Result:=0
  end
  else
    inherited;
end;

end.
```

实例 7 使窗体永远最小化

实例目标

程序启动时, 将程序窗口最小化到任务栏, 并且只能处于最小化状态, 使窗体的内容不可见。

实现方法

首先要将 WindowState 属性设置为 wsMinimized, 使程序在启动时就进行最小化状态。然后使用 Msg 对象控制窗体的 WMQueryOpen 事件, 始终使窗体处于最小化状态。

程序代码

(1) 创建一个新的 Delphi 项目。

(2) 设置窗体的 WindowState 属性为 wsMinimized。

(3) 在单元文件的 Private 部分添加私有过程：

```
private
PROCEDURE WMQueryOpen(VAR Msg : TWMQueryOpen); message WM_QUERYOPEN;
```

(4) 编写私有过程，其代码如下：

```
PROCEDURE TForm1.WMQueryOpen(VAR Msg : TWMQueryOpen);
begin
    Msg.Result := 0;
end;
```

实例 8 窗体背景色渐进效果

实例目标

这里制作了窗体背景色渐进效果，当窗体初始化时，它就呈现出一种色彩渐进的效果。如图 1-5 所示，用户可以通过调节 SpinEdit 控件的上下按钮来使它的渐进色彩发生变化，如果单击【开始】按钮，这个窗体的背景就会发生动态改变，颜色在不断地改变，渐进效果也在不断变化，而且可以调节它的变化速度。

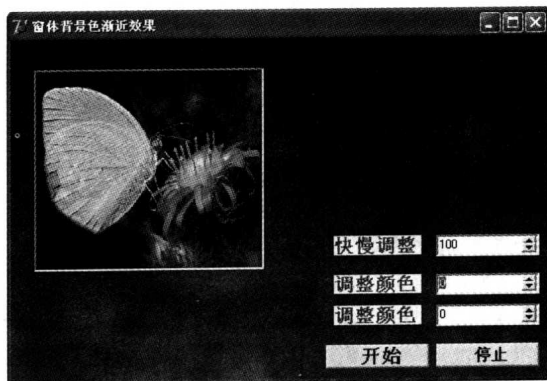


图 1-5 背景渐进效果

实现方法

本程序利用循环函数来实现色彩的渐进效果，另外需要通过控制 SpinEdit 控件的值来控制窗体的色彩变化。而在实现动态效果时，需要对控件 Timer 进行设置，每经过一段设定的时间，触发一次 Timer 控件的 OnTimer() 事件，利用它完成窗体背景色彩的动态改变。

程序代码

(1) 创建一个新的 Delphi 项目。在新建的窗体上添加一个 Panel 控件，一个 Image 控件，3 个 Label 控件，3 个 SpinEdit 控件以及两个 Button 按钮，其中 Image 控件放到 Panel 控件中，并为 Image 选择一个图片，如图 1-6 所示。