

Delphi 6

实效编程百例

★求是科技 施新刚 编著

- ★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

实效编程百例

Delphi 6

实效编程百例

★求是科技 施新刚 编著

人民邮电出版社

图书在版编目 (C I P) 数据

Delphi 6 实效编程百例/施新刚编著. —北京: 人民邮电出版社, 2002.8
ISBN 7-115-10431-X

I. D... II. 施... III. DELPHI 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 050018 号

实效编程百例

Delphi 6 实效编程百例

-
- ◆ 编 著 求是科技 施新刚
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 010-67180876
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 21.75
字数: 635 千字 2002 年 8 月第 1 版
印数: 1-6 000 册 2002 年 8 月北京第 1 次印刷

ISBN 7-115-10431-X/TP · 2967

定价: 35.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内容提要

本书通过 130 个精选的例子讲解了利用 Delphi 进行应用程序开发的多个方面，其内容涵盖了界面外观、多媒体控制与图像处理、时间控制、操作系统、程序控制、磁盘文件、数据库、网络与通信、鼠标和键盘、数学算法和程序发布等方面。

本书内容突出了实用性，85%以上的实例模仿较常见的优秀软件的相关功能，其余实例是为帮助读者理解重点、难懂概念所做，并力求通过每章的例子重点讲述如何利用 Delphi 组件和 API 函数等来实现特定的功能。本书的另一个特点在于给出了实用性很强的“方案实例”而不仅仅是“功能实例”，其内容多为典型或通用的功能模块的解决方案，包括界面设计、操作流程以及代码控制等内容。

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



165.10.17



本书实例的分类按照完成功能来划分,包括界面外观设计、多媒体控制与图像处理、时间控制、操作系统、程序控制、磁盘文件、数据库、网络与通信、鼠标和键盘、数学算法与程序发布等内容,以使读者对计算机应用的各个方面有所了解。

对于数据库编程,精确掌握编程语言的语法概念并不意味着可以编写功能强大的应用程序,编程水平的提高需要依靠实际应用中经验的积累。本书通过 130 个新颖别致、风格各异的应用实例,详细讲解了如何利用 Delphi 的强大功能以及 API 函数开发应用程序。

每个实例的讲解分为 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。

本书内容突出了实用性,85%以上的实例模仿较常见的优秀软件的相关功能,其余实例是为帮助读者理解重点、难懂概念所做,并力求通过每章的例子重点讲述如何利用 Delphi 组件和 API 函数等来实现特定的功能。本书的另一个特点在于给出了不少实用性很强的“方案实例”,其内容多为典型或通用功能模块的解决方案,包括界面设计、操作流程以及代码控制等内容。

本书主要由施新刚、张增强等编写。此外刘冰玉、杨柯岚、王东、杨珏、赵昊彤、程凡、程卫峰、范桂山、赵微微、宋征、吴频、严庆子、文华、袁珂、岳进、钟明、黄成昆、王远、吴晓超、肖永顺、钱力鹏、马小来、李松、王琴芳、王国红等也参与了本书的资料搜集和写作工作。以上人员对本书的顺利完成付出了辛勤的汗水和心血,在此一并表示感谢。

由于时间、水平限制,缺点和不足之处在所难免,敬请读者批评指正。

编者

目 录

第 1 章 界面外观	1
实例 1 显示倾斜文字	2
实例 2 文字颜色渐变	4
实例 3 透明图片	5
实例 4 颜色渐变的进度条	7
实例 5 透明窗体	8
实例 6 多边形窗体	9
实例 7 可变背景窗体	10
第 2 章 界面操作	13
实例 8 调色板程序	14
实例 9 颜色下拉框	15
实例 10 窗体背景色渐进效果	17
实例 11 动态对话框	20
实例 12 窗体分割	22
实例 13 缩放控件	23
实例 14 动画按钮	25
实例 15 动态缩小的图形	26
实例 16 放大图像	29
实例 17 浏览大图片	31
实例 18 抓取屏幕	33
实例 19 资源管理器	38
实例 20 智能查找列表框中的内容	40
实例 21 带动态提示的文本编辑器	43
实例 22 修改光标和程序的图标	45
实例 23 持续响应鼠标单击	48
实例 24 始终保持在最前的窗体	51
实例 25 限制鼠标位置	53

实例 26 标题栏按钮	54
实例 27 拖放列表框中选中的对象	57
实例 28 创建多文档窗体	60

第 3 章 多媒体控制

实例 29 检测声卡是否存在	63
实例 30 视频播放器	63
实例 31 CD 播放器	66
实例 32 变换图像	68
实例 33 OpenGL 编程	70
实例 34 RealPlayer 音频播放器	73

第 4 章 时间控制

实例 35 文字逐个出现，模仿打字	78
实例 36 简单动画的实现	79
实例 37 嵌入式电子钟	80
实例 38 以日历形式进行时间信息的获取与控制	83
实例 39 毫秒级的控制	84
实例 40 读写系统时间	88
实例 41 同步网络时间（从时间服务器上获得）	90
实例 42 通过多个 Timer 控件执行多任务	92

第 5 章 操作系统

实例 43 修改计算机名称	99
实例 44 编写屏保程序	100
实例 45 设置墙纸和屏保程序	103
实例 46 动态调整屏幕分辨率	105
实例 47 获取系统硬件信息	108
实例 48 编辑注册表信息	110
实例 49 锁住计算机	113
实例 50 重新启动和关闭计算机	114
实例 51 获取 Windows 版本号和运行模式	116
实例 52 枚举可用字体	118

实例 53 获取打印机信息	119
实例 54 获取打印机队列的状态信息	121
实例 55 文本和图像的打印	123
实例 56 获得打印页边距	124
实例 57 利用 API 直接打印	126
实例 58 改变默认的打印机	132
第 6 章 程序控制	134
实例 59 系统托盘程序	135
实例 60 隐藏任务栏	138
实例 61 防止屏幕切换窗口	139
实例 62 枚举系统正在运行的程序	141
实例 63 查看 IE 浏览器是否运行	143
实例 64 调用 Windows 程序	143
实例 65 禁止运行同一程序多个实例	147
实例 66 修改系统菜单	148
实例 67 禁止窗体右上角各按钮	151
实例 68 多线程应用程序	152
实例 69 使用剪切板复制图片	155
实例 70 利用内存映射实现数据交换	158
实例 71 取得下拉条的信息	160
第 7 章 磁盘文件	163
实例 72 获取操作系统的序列号	164
实例 73 获取驱动器容量	165
实例 74 查看驱动器类型	167
实例 75 递归法遍历磁盘目录	168
实例 76 获取文件属性	170
实例 77 将文件删除到回收站中	171
实例 78 清空回收站	174
实例 79 启动控制面板中的内容	175
实例 80 创建和删除文件夹	177

实例 81 读写 ini 文件	180
实例 82 弹出和关闭光驱	183
实例 83 更改文件的扩展名	184

第 8 章 数据库

实例 84 处理数据库中日期型字段的显示与输入	187
实例 85 存取图像字段	190
实例 86 枚举 Access 数据库中自建表	193
实例 87 SQL 语句中使用通配符	194
实例 88 Delphi 数据集过滤技巧	196
实例 89 在数据库中存取 Word 文档	199
实例 90 通过注册表在程序中增减和修改数据源	202
实例 91 SQL 语句嵌套	205
实例 92 动态创建数据库对象	208
实例 93 代码控制连接 ODBC	211
实例 94 快速连接 SQLServer	214
实例 95 快速报表与报表预览	217
实例 96 动态改变 DBGrid 组件的颜色	219
实例 97 小写金额转换成中文大写	221

第 9 章 网络与通信

实例 98 获取网卡地址	224
实例 99 获得主机名和 IP 地址	226
实例 100 判断网址是否存在	228
实例 101 监测局域网内计算机	230
实例 102 Ping 测试	232
实例 103 设置代理服务器	235
实例 104 自定义网页浏览器	237
实例 105 修改 IE 的主页与标题	243
实例 106 收发送电子邮件	246
实例 107 FTP 上传下载	257
实例 108 网络聊天——WinSock-TCP	265

实例 109 广播信息——WinSock-UDP	269
实例 110 电话线实现远端通信	272
实例 111 串口采集外设数据	281
实例 112 用 API 实现串口通信	283
实例 113 测试网络速度	286
第 10 章 数学算法	290
实例 114 进制转换	291
实例 115 随机选号	293
实例 116 多点曲线光滑拟和	297
实例 117 统计中英文字符数	300
第 11 章 鼠标和键盘	302
实例 118 代码控制光标	303
实例 119 模拟鼠标的单双击	304
实例 120 模拟键盘输入	305
实例 121 限定鼠标区域	311
实例 122 截获鼠标移开事件	312
实例 123 截取键盘信息	314
实例 124 判断 PrintScreen 键是否按下	318
实例 125 时隐时现的鼠标	319
实例 126 鼠标拖动无标题栏窗口	320
第 12 章 程序发布	323
实例 127 产生程序序列号	324
实例 128 在 IE 工具栏中加入快捷图标	329
实例 129 设置程序为自动被执行	333
实例 130 限定程序的使用时限	334

第 1 章 界面外观



-  显示倾斜文字
-  文字颜色渐变
-  透明图片
-  颜色渐变进度条
-  透明窗体
-  多边形的窗体
-  可变背景窗体

实例 1 显示倾斜文字

实例目标

本例可以将给定的文字以要求的角度进行旋转，同时还可以确定显示倾斜文字所基于的原点。运行结果如图 1-1 所示。

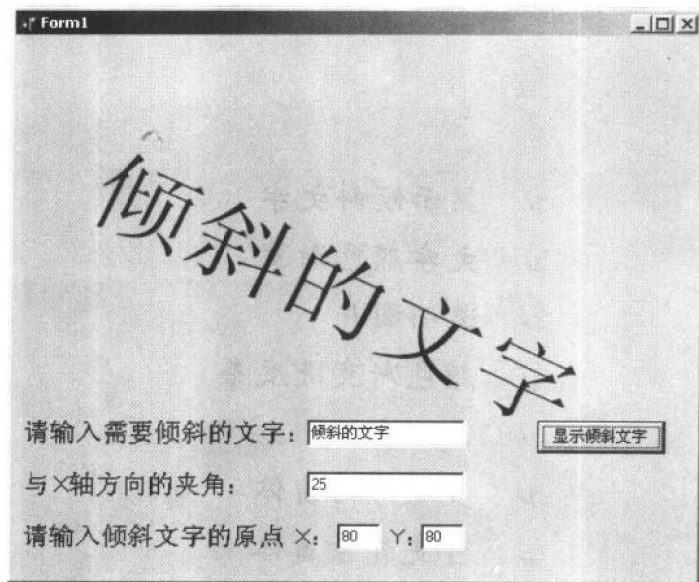


图 1-1 倾斜的文字

实现方法

新建一个窗体，在窗体中添加 4 个 Edit 控件，分别用于输入倾斜文字的内容、文字倾斜的角度以及倾斜文字所基于的原点坐标。再添加 4 个 Label 控件进行对应说明。Edit 控件里输入的内容都是字符串，要使用 StrToInt() 函数将其转成整数数值。

首先创建新的逻辑字体 LogicFont，接着设置逻辑字体 LogicFont 的一些属性，如高度、宽度、笔划粗细等。然后使用 lfEscapement 和 lfOrientation 属性设置倾斜角度。它们的单位为十分之一度，即如果想让倾斜角设为 45° 则应该设置成 450，这是实现倾斜文字的关键。因为输出的文字是有白底的，所以需要使用 SetBkMode() 函数将白底去掉。

然后用 GetDC() 函数取得当前窗口的句柄，用 SelectObject() 函数将当前窗口的字体用逻辑字体替换掉，再用 TextOut() 函数按照给定数据实现文字倾斜。

最后一定要用 ReleaseDC() 释放设备接口。

程序代码

本例很简单，下面直接给出完整代码：

```
unit Unit1;

interface

uses
```

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

type

```
TForm1 = class(TForm)
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit2: TEdit;
    Button1: TButton;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Edit4: TEdit;
    procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
```

var

```
Form1: TForm1;
```

implementation

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

var

```
    LogicFont : TagLogFontA;
    //逻辑字体
    TempFont, PrevFont : HFONT;
    //字体句柄
    TempDC : HDC;
    //图形设备句柄
```

begin

```
    LogicFont.lfHeight := 70;
    //设置字高
    LogicFont.lfWidth := 40;
    //设置字宽
    LogicFont.lfWeight := 10;
    //设置字体笔划粗细程度
    LogicFont.lfUnderline := 0;
    //设置下划线为没有
    LogicFont.lfStrikeOut := 0;
    //设置没有删除线
    LogicFont.lfItalic := 0;
    //设置不为斜体
    LogicFont.lfCharSet := GB2312_CHARSET;
    //设置字符集
    LogicFont.lfEscapement := -StrToInt(Edit2.Text)*10;
    LogicFont.lfOrientation := -StrToInt(Edit2.Text)*10;
    //设置顺时针方向与 X 轴的夹角
    LogicFont.lfFaceName := '宋体';
    //设置字体名称
    TempFont := CreateFontIndirect(LogicFont);
    //创建逻辑字体
    TempDC := GetDC(Handle);
    //取得窗口的设备句柄
    PrevFont := SelectObject(TempDC, TempFont);
```

```
//取出窗口设备的当前字体，并替换为新字体
SetTextColor(TempDC, clBlue);
//设置设备窗口的文字色彩
SetBkMode(TempDC, TRANSPARENT);
//去掉文字白底
TextOut(TempDC, StrToInt(Edit3.Text), StrToInt(Edit4.Text), PChar(Edit1.Text), Length(Edit1.Text));
//输出文字
SelectObject(TempDC, PrevFont);
//恢复原有的字体
DeleteObject(TempFont);
//删除逻辑字体
ReleaseDC(Handle, TempDC);
//释放设备接口
end;

end.
```

实例 2 文字颜色渐变

实例目标

本例将利用代码控制标签里的文字颜色 RGB()函数逐渐地改变文字颜色。

实现方法

程序响应 Timer 控件的 OnTimer 事件，其中调用函数 RGB()来有规律地改变文字颜色。可以设置 3 个全局变量 r、g、b，分别控制红绿蓝 3 种颜色分量，每一次时钟事件都使得 3 种分量值加 1，再用 RGB 函数获得新的颜色值，进而改变标签的文字颜色。

本例只是颜色渐变的方法之一，读者可以通过设置 r、g、b 的各种不同的初始值和增量，就能够实现多种类型的颜色渐变。

程序代码

本例比较简单，创建一个新项目，在窗体中添加一个 Timer 控件，将 Interval 属性设置成 100。添加一个 Label 控件，设置 Caption 属性值为“文字颜色渐进”。

设置 3 个全局变量 r、g、b 来控制颜色分量，编写 Timer 控件的时钟事件代码，每隔 0.1s 改变一次标签的文字颜色。

完整代码如下：

```
unit Unit1;

interface

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

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Label1: TLabel;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  end;
```



```
public
{ Public declarations }
end;

var
  Form1: TForm1;

implementation
var
  r,g,b:integer;

{$R *.dfm}

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if (r<=255) and (g<=255) and (b<=255) then
  begin
    Label1.Font.Color:=RGB(r,g,b);
    //改变文字颜色
    r:=r+1;
    g:=g+1;
    b:=b+1;
  end
  else
  begin
    r:=0;
    g:=0;
    b:=0;
  end;
end;
end;
```

实例 3 透明图片

实例目标

本例将实现透明的图片效果。用户事先指定某种颜色为透明色，则在窗体中显示图片时，被指定为透明色的内容将不予以显示，而显露出图片下层的内容，其效果如图 1-2 所示。

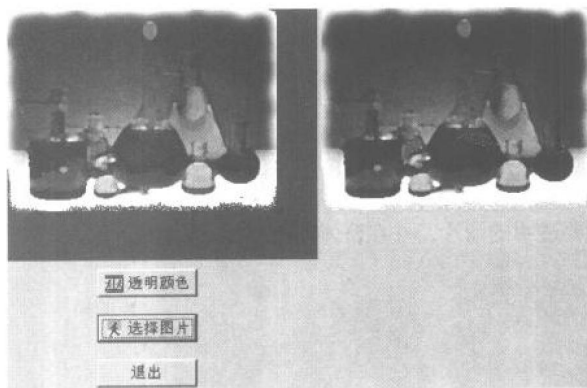


图 1-2 透明图片

在图 1-2 中,左边的图片是未做透明处理的情况,而右边的图片事先指定了红色作为透明色后进行透明显示的结果。

实现方法

本例利用 Canvas 属性对图片进行透明。首先将位图文件的 Transparent 属性设置为 True (即把透明打开),然后将 TransParentColor 属性设置成要透明的颜色,最后利用 Canvas.Draw 对透明过后的图片进行重绘,就实现了透明图片的显示。

程序代码

1. 新建一个 Delphi 项目。
2. 在窗体中添加一个 SpeedButton 控件、一个 BitBtn 控件、一个 Button 控件,分别对应图 1-3 左下方的 3 个按钮控件,设置它们的 Caption 属性分别为“透明颜色”、“选择图片”和“退出”,分别用于打开调色板、打开文件选择对话框和结束程序运行。
3. 在窗体中添加一个 OpenFileDialog 控件和 ColorDialog 控件,分别用于打开文件选择对话框、调色板。并将窗体的 BorderStyle 属性设置为 bsNone。添加控件后的窗体如图 1-3 所示。

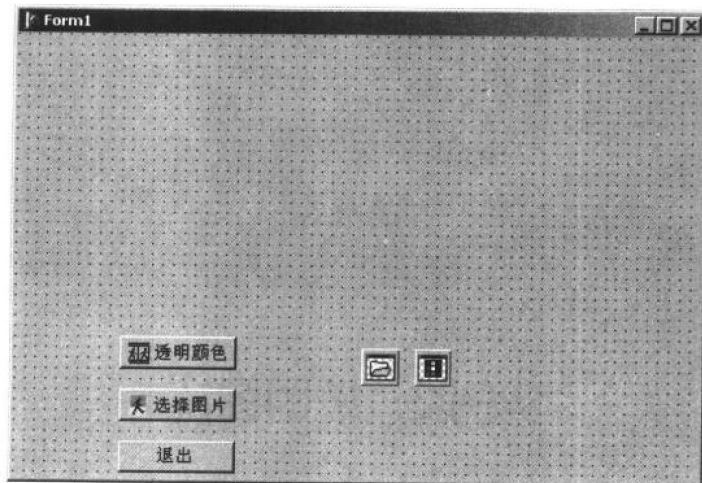


图 1-3 添加控件后的窗体

4. 添加一个全程变量 Trancolor, 用来保存需要透明的颜色。
5. 编写代码, 选择透明颜色。

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    ColorDialog1.Color:=TranColor;
    If ColorDialog1.Execute then
        TranColor:=ColorDialog1.Color;
    //选择要透明的颜色
end;
```

6. 编写代码, 选择要透明的图片, 并在窗体的两侧显示出来。

```
procedure TForm1.BitBtn1Click(Sender: TObject);
var
    Bitmap:TBitmap;
begin
    Bitmap:=TBitmap.Create;
    If OpenFileDialog1.Execute then
        begin
            try
```

```
with Bitmap do begin
  LoadFromFile(OpenDialog1.FileName);
  //选择要透明的图片
  Form1.Canvas.Draw(0,0,Bitmap);
  //显示未透明的图片
  Transparent:=True;
  TransParentColor:=TranColor;
  //设定透明色
  Form1.Canvas.Draw(280,0,Bitmap);
  //显示已透明图片
end;
finally
  Bitmap.Free;
end;
end;
end;
```

7. 编写代码, 设置退出按钮。

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Form1.Close;
end;
```

实例 4 颜色渐变的进度条

实例目标

本例实现颜色渐变的进度条。如图 1-4 所示, 按下“开始”按钮后进度条开始增加, 进度条完成不同的百分比时显示不同的颜色。

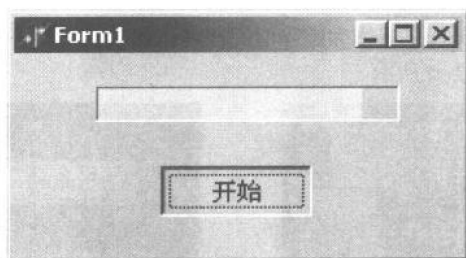


图 1-4 颜色渐变的进度条

实现方法

利用 CLX 中 ProgressBar 控件的 FillColor 属性改变进度条的颜色, 通过判断 ProgressBar 控件的 Step 属性来控制颜色的变化。

程序代码

新建一个 CLX 程序, 在窗体中添加 ProgressBar 控件和 Button 控件。要注意的是在 VCL 中 ProgressBar 控件是没有 FillColor 属性的, 窗体设计如图 1-4 所示。

响应“开始”按钮的单击事件, 进度条将随着进度的不同而改变颜色。代码如下:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  i:integer;
```