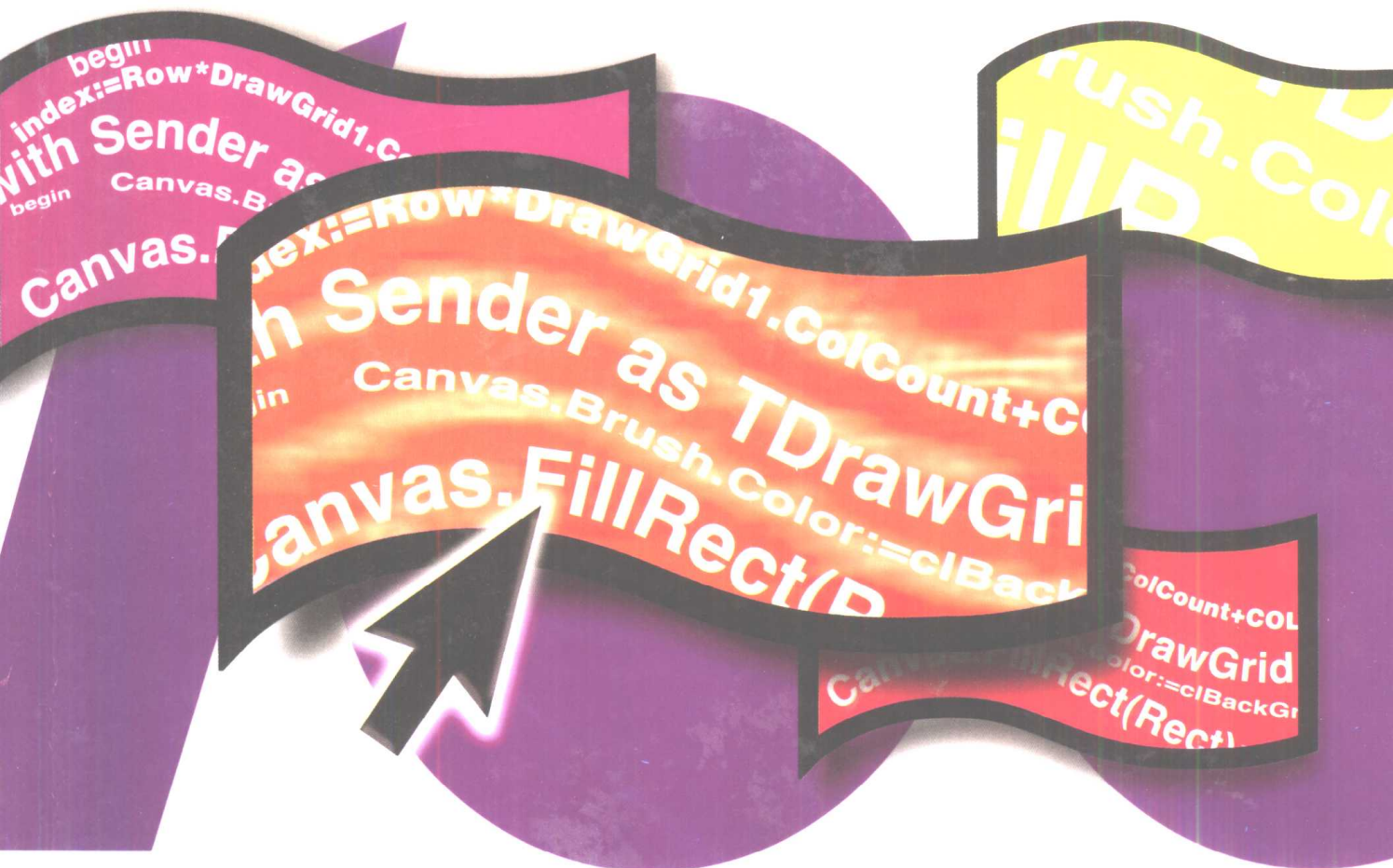


时尚百例丛书



Delphi 5.0 控件 时尚编程百例



●网冠科技 编著

100

时尚百例丛书

Delphi 5.0控件 时尚编程百例

网冠科技 编著

光盘包含书中素材、效果文件



989240



机械工业出版社

JS68 / 45 03

Delphi 5.0 是由 Inprise 公司推出的面向对象的可视化软件开发工具。
本书从控件的角度讲解 Delphi 5.0 的强大功能。

本书通过 100 个实例讲解 Delphi 5.0 在控件方面的强大功能。全书共分 5 篇：第一篇讲解 Delphi 5.0 的基本控件及其常见用法；第二篇讲解窗体的基本用法和高级处理技巧；第三篇讲解多媒体的用法；第四篇讲解图形图像的处理技巧；第五篇讲解有关数据库的操作。

本书既可以作为计算机专业不同层次培训班的教材,也可以作为命令参考手册供查询使用。

图书在版编目 (CIP) 数据

Delphi 5.0 控件时尚编程百例 / 网冠科技编著.

-北京: 机械工业出版社, 2001.4

(时尚百例丛书)

ISBN 7-111-08855-7

I. D... II. 网... III. Delphi 语言-程序设计, IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 14764 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 王琼先

责任印制: 郭景龙

北京京丰印刷厂印刷·新华书店北京发行所发行

2001 年 7 月第 1 版·第 2 次印刷

787mm×1092mm¹/₁₆·21.25 印张·527 千字

6 001—11 000 册

定价: 38.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换
本社购书热线电话 (010) 68993821、68326677-2527

时尚百例丛书

追求时尚 追求完美

出版说明

随着 21 世纪的到来,人们更深切地感受到了计算机在生活和工作中的作用越来越重要,越来越多的职业需要具有计算机的应用技能。掌握计算机是职业的需要,更是事业发展的需要。

目前计算机技术不但广泛地应用在办公自动化中,它还全面渗透到各行各业。如果要从事平面设计的相关行业,就应该学会平面设计软件,如 Photoshop、CorelDRAW、FreeHand 等;如果要从事三维设计的相关行业,就应该学会三维设计软件,如 3DS MAX、Maya、Poser 等;如果要从事多媒体设计的相关行业,就应该学会多媒体制作软件,如 Authorware、Director、Premiere 等;如果要从事与网络相关的行业,就应该学会 Flash、Dreamweaver、Fireworks、ASP、PHP、JavaScript 等;如果要从事建筑产品、工业产品设计的相关行业,就应该学会 AutoCAD、3DS VIZ、Protel 等;如果要从事软件开发的相关行业,就应该学会 VB、VC、VFP、Delphi、PowerBuilder 等编程。

所有与计算机相关的职业都要求工作者有很强的计算机操作技能,做到运用自如,熟练而且深入地掌握软件的应用。而要做到这一点,必须从软件的各个方面入手,通过实例演练的方式训练自己,而且要反复练习,做到举一反三。

为了让大家能深入而且熟练地掌握相关软件的应用方法,机械工业出版社特别为广大读者推出了这套时尚百例丛书。本丛书对每一个应用软件精心制作了 100 个实例,其宗旨就是让读者全方位掌握软件的应用,为广大读者提供一条快速掌握计算机应用技能的捷径。

本丛书采用新颖的版式,将知识和实例紧密结合,通过对各种实例的详细讲解,使读者不必事先学习各种软件,而从实例的制作过程中体会到每个软件每项功能的使用方法,并自己做出各种实例效果,这样既节省了大量时间,同时也使读者有身临其境的感觉,并可以反复演练,将所学知识运用到职业工作中去。

书山有路勤为径。愿广大读者能通过本丛书的学习掌握计算机技能,并应用到自己的工作和事业中去。

机械工业出版社



前言

《Delphi 5.0 控件时尚编程百例》是“时尚百例丛书”中的一本。

Delphi 5.0 是由 Inprise 公司推出的 Delphi 的最新版本。Delphi 5.0 继承了以前版本的优点，并且增添了许多新的功能，比如在 ADO、数据模块设计、IBX、MIDAS、CORBA、程序调试、VCL、窗体框架和桌面定义等方面都增强了许多，可以大大提高软件系统的设计、管理和开发的速度。它集中了 Visual C++ 和 Visual Basic 两者的优点：学习过程中很容易上手，而且功能也非常强大，尤其在界面设计、数据库编程和网络编程等方面更具独到之处，是其他软件开发工具无法比拟的。

本书按照不同的控件类型分成了若干个部分进行叙述，每部分都有一定的主题，并且由浅入深地编排了 100 个精炼实用的例子，每个例子都具有较强的针对性，突出讲解了某一控件的具体功能。在书中很少有大段的代码和高深的术语出现，重点放在了实例的创建思路与过程上，便于读者掌握各个实例中包含的知识点。书中语言通俗易懂，读者可以很轻松地完成本书所有内容的学习。

Delphi 是一种功能强大的工具，但由于含有大量不同的单元，因此编程环境相比起其他编程软件而言相当复杂。本书将帮助读者以一种更为简洁、迅速的方式掌握 Delphi 编程技术。

本书内容广泛，涉及到 Windows 编程的各个方面，根据各个控件所实现的功能不同，本书可分为 5 篇，第一篇讲解 Delphi 5.0 的基本控件及其常见用法，读者可以对一些基本控件的常见和高级用法有所了解，并学习文件管理操作和程序界面制作；第二篇讲解窗体的基本用法和高级处理技巧，对 Delphi 5.0 最常见也是最重要的控件有更清晰的认识；第三篇讲解多媒体的用法；第四篇讲解图形图像的处理技巧，学习和了解 Windows 图形编程的高级处理技巧；第五篇讲解有关数据库的操作，内容丰富，包含的实例功能强大。

本书主要针对中高级的 Delphi 用户使用，也非常适合于初学者学习。由于作者水平有限，书中难免存在错误和疏漏之处，恳请读者批评指正。



网冠科技

本书光盘含配套素材，技术支持请点击网冠科技站点 <http://netking.163.com>。E-mail:

netking_@163.com.

网易 NETEASE
WWW.163.COM

是网易公司的标志。



出版说明

前言

第一篇 Delphi 5.0 基本控件

实例 1	弹性小球	2
实例 2	菜单加工具栏的文字编辑器	5
实例 3	自动关机程序	8
实例 4	猜数字游戏	11
实例 5	你的彩票中奖了吗	14
实例 6	欢迎使用笔记本	18
实例 7	电子钟数字控件	21
实例 8	简单的文字编辑程序	24
实例 9	深入了解你的计算机信息	27
实例 10	多页面程序设计	30
实例 11	制作一个简便的颜色调节程序	33
实例 12	利用 Animate 控件制作动画	36
实例 13	简单的字处理软件	39
实例 14	实现查找功能的列表框	42
实例 15	列表框的灵活运用	45
实例 16	在程序运行的时候修改列表框	48
实例 17	利用递归查找所有目录	51
实例 18	简易日历	54
实例 19	把汉字转换为拼音的小程序	57
实例 20	ListView 和 CheckBox 控件的使用	60
实例 21	将阿拉伯数字转换为汉字	63
实例 22	文件查询程序	66
实例 23	文件数目和大小的统计程序	69
实例 24	利用拼音完成查找功能	72
实例 25	五子棋大战	75
实例 26	自制的资源管理器	78
实例 27	简历生成器	81

实例 28	按键计算器	85
实例 29	多功能秒表	89
实例 30	多文档应用程序	93
实例 31	计算器	98

第二篇 窗体控件

实例 32	俄罗斯方块游戏	103
实例 33	以色彩渐进层效果显示的窗体控件	106
实例 34	制作窗体的渐进效果	109
实例 35	制作透明窗体	112
实例 36	椭圆窗体中的渐近线	115
实例 37	制作多边形窗体	118
实例 38	改变桌面的工具	121
实例 39	带提示状态条的文本编辑器	124
实例 40	可变窗体	128
实例 41	特殊窗体	132
实例 42	单击客户区移动窗体	136
实例 43	魔幻窗体	139

第三篇 多媒体和动画

实例 44	自制的媒体播放器	143
实例 45	媒体播放器	146
实例 46	随机图形动画	149
实例 47	屏保程序	153
实例 48	图形动画	157
实例 49	CD 播放器	161
实例 50	地球自转	164
实例 51	阴影动态文字	167
实例 52	月亮、地球、太阳	172

第四篇 图形图像

实例 53	自制的图片浏览器	177
实例 54	流光异彩的广告牌	180
实例 55	图像的特殊效果	183
实例 56	旋转图形和文本的程序	186
实例 57	绘制立体文本	189



实例 58	将彩色图像转换为灰度图像 (一)	192
实例 59	将彩色图像转换为灰度图像 (二)	195
实例 60	笔刷	198
实例 61	数学曲线	203
实例 62	多边形绘制程序	207
实例 63	图形绘制程序 (一)	210
实例 64	图形绘制程序 (二)	213
实例 65	图像浏览器	218
实例 66	动态图像浏览器	222
实例 67	文本图案	227
实例 68	渐变图案	230
实例 69	绘制直线和椭圆	234
实例 70	图像格式转换器	237
实例 71	图片浏览器	241

第五篇 网络与数据库

实例 72	事件备忘录	245
实例 73	利用 ADO 动态浏览数据	248
实例 74	从已建数据源中查询数据	251
实例 75	根据数据动态生成图表	254
实例 76	将文本添加到数据表	257
实例 77	数据浏览树	260
实例 78	数据库事务的应用	263
实例 79	ADO 复杂数据库编程	266
实例 80	DBGrid 控件的高级使用	269
实例 81	简易浏览器	272
实例 82	文件传输工具 (一)	275
实例 83	文件传输工具 (二)	278
实例 84	聊天工具	281
实例 85	网址检索	284
实例 86	消息发送工具	287
实例 87	邮件发送工具	290
实例 88	邮件阅读工具	293
实例 89	Web 浏览计数器	296
实例 90	C/S 系统编程	299
实例 91	用户密码验证	302
实例 92	数据库窗口引导	306
实例 93	数据表的维护	309

实例 94	数据表查询运算	312
实例 95	信息输出	315
实例 96	成绩统计	318
实例 97	教务选课	321
实例 98	数据表浏览	324
实例 99	数据库监视器	327
实例 100	事务处理	330



第一篇

Delphi 基本控件

本篇总览

本篇主要介绍 Delphi 5.0 中, 用于开发程序界面的几个基本控件及和文件处理有关的控件使用。这些控件包括命令按钮、选择按钮、文本框、Memo 控件、文件处理控件等。在这些控件内部封装了大量的编程代码, 包含大量的方法和属性。通过它们, 读者可以轻易开发出常见的 Windows 界面以及进行常见的文件处理。

本篇中, 读者可以了解 Delphi 5.0 中控件的基本概念, 并能够运用一些有关基本控件处理的中高级知识来开发应用程序。同时, 读者也可以对面向对象的编程机制有一个大概的了解。



实例 1 弹性小球

实例说明

在这个例子中, 小球会来回弹跳, 当它撞着区域的边缘时, 会沿相反的方向行进, 并且在行进的过程中, 小球的半径会不断变大, 而且利用窗口右下角的 **TrackBar** 控件, 你可以随意调整小球来回运行的速度。熟悉这个例子之后, 还可以进一步扩展它, 使小球沿着不同的角度在窗口上作弹性碰撞。

本程序主要应用 **Timer** 组件及其 **OnTimer()** 事件, 如图 1-1 所示为效果图。

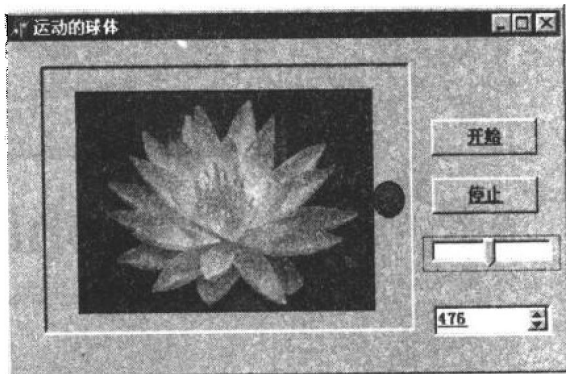


图 1-1 效果图

编程思路

在 Delphi 中, 有一个 **Timer** 组件, 在运行程序的时候, 你看不见它, 也不能直接对它进行操作。Timer 组件每隔一段固定的时间就会自动触发 **OnTimer()** 事件, 本例正是利用这个功能, 在每隔一段时间触发的 **OnTimer()** 事件中, 小球沿着给定的轨道前进一段距离, 当小球到达指定区域的边缘时, 再令它沿相反的方向前进。利用控件 **TButton** 可以激活 **Timer** 组件, 触发 **OnTimer()** 事件, 利用 **TrackBar** 控件可以控制 **OnTimer()** 事件触发的时间间隔, 可设为 1~1000ms, 可以随意控制小球行进的速度。

创作步骤

1. 启动 Delphi 5.0, 打开一个新的标准工程。如果 Delphi 已经运行, 那么请在 **File** 菜单中单击 **New** 菜单项, 打开一个新的标准工程。在面板控件中加入 **Shape** 控件, 并且将 **Shape** 控件的 **Brush** 中的 **Color** 属性设置为红色, **Shape** 属性设置为 **stCircle**, 向 **Form** 中加入另外两个 **TButton** 控件, 将其中一个控件的名称设置为“开始”, 这个控件用来控制小球运动的开始和激活 **Timer** 控件, 将另外一个控件的名称设置为“停止”, 控制小球运动的结束。加入控件 **TrackBar**, 控制小球行进的速度。利用 **TrackBar** 中的 **Min**、**Max** 和 **Position** 选项, 可以设置小球运行速度的最大值、最小值和当前值, 设置 **Min** 值为 1, **Max** 值为 1000, 值越小, 激发 **OnTimer()** 事件的时间间隔越小, 小球也就运行得越快。加入控件后的窗体如图 1-2 所示。

2. 为了实现小球来回移动的功能, 还需要设置一个全局变量, 即整数 **I**, **I** 值的初始值大小设为 1。其值可变, 当 **I=1** 时, 小球向左运动, 撞到面板的左壁时, **I** 值变为 2, 此时小球向右运动, 撞到面板右壁时, **I** 值又变为 1。这样通过 **I** 值的变化, 就造成了小球的来回移动。读者需要在 **TForm1.Timer1Timer()** 中添加如下代码:

```
timer1.interval:=trackbar1.position;
```

```
//设置 Timer 组件的触发时间间隔
```

```
if i=1 then
```

```
shape1.left:=shape1.left-10;
```

//使小球的位置发生变化

```
shape1.width:=(shape1.width+1)mod 70;
```

//使小球所在的矩形宽度发生变化

```
shape1.height:=(shape1.height+1)mod 70;
```

//使小球所在的矩形高度发生变化

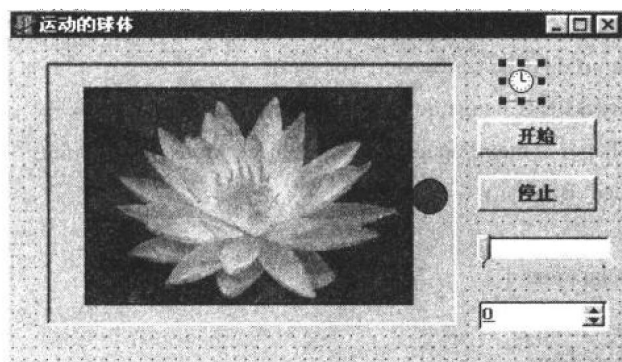


图 1-2 插入控件后的窗体效果

3. 为了使小球运行的速度发生变化, 还必须在 TForm1.Button1Click() 中加入如下代码:

```
timer1.interval:=trackbar1.position;
```

//将 Position 的值赋给 Timer1 的 Interval 属性

```
timer1.Enabled:=true;
```

//设置 Timer1 的值为真

4. 右下方的“SpinEdit”控件同样是用来控制小球的运行速度的, 因此必须随着 TrackBar 值的变化而变化, 代码如下:

```
trackbar1.position:=spinedit1.Value;
```

写入这句代码之后, 当你单击 SpinEdit 控件的上下按键时, 就会使 Timer 控件的时间间隔发生变化, 同时改变小球的运行速度。

5. 编写 TForm1.TrackBar1Change() 事件的处理代码, 具体代码如下:

```
spinedit1.value:=trackbar1.Position;
```

//将 TrackBar 值的变化通知 SpinEdit 控件

本程序源代码如下: (只列出一部分, 其他的详见光盘)

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
```

```
ExtCtrls, ComCtrls, StdCtrls, jpeg, Spin;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Panel1: TPanel;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
TrackBar1: TTrackBar;
```

```
Timer1: TTimer;
```

```
Shape1: TShape;
```

```
Image1: TImage;
SpinEdit1: TSpinEdit;
procedure Timer1Timer(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure SpinEdit1Change(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure Shape1ContextPopup(Sender: TObject; MousePos: TPoint;
    var Handled: Boolean);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
    i: integer;
implementation
{$R *.DFM}
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    timer1.interval:=trackbar1.position;
    if i=1 then
    begin
        if shape1.left>0 then
        begin
            shape1.left:=shape1.left - 10;
            shape1.width:=(shape1.width+1)mod 70;
            shape1.height:=(shape1.height+1)mod 70;
        end
        else
            i:=2;
        end;
    if i=2 then
    begin
        if shape1.left<(panel1.width - shape1.width - 5) then
        begin
```

实例2 菜单加工具栏的文字编辑器

实例说明

菜单和工具栏是 Windows 最大的两个特色，在接触 Windows 时，无论是新用户，还是 Windows 的老用户，对这两个工具都不会陌生吧。本例效果就是利用 Delphi 5.0 自带的菜单编辑器和工具栏设计和制作的。能完成多种文字编辑功能，而且界面美观，所列出的文件编辑功能十分全面。

本程序主要应用了菜单编辑器和 Memo 控件的有关属性。效果图如图 2-1 所示。

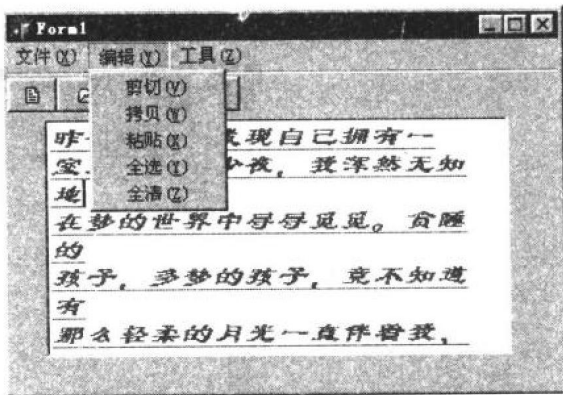


图 2-1 效果图

编程思路

在 Delphi 中，菜单编辑器可能是最常用的一个控件了，因为基本上每一个涉及到 Windows 的程序，其软件的设置都离不开菜单的制作，利用菜单编辑器可以得到一系列标准的 Windows 控件。而利用 ToolBar 和 SpeedButton 则可以制作出精美的工具栏，工具栏的作用，菜单上都会列出来，工具栏上的工具只是为了方便操作而设，而几个通用对话框也是设计这个程序必不可少的一部分，例如打开对话框、保存对话框以及字体对话框等。

创作步骤

1. 启动 Delphi 5.0，打开一个新的标准工程。如果 Delphi 已经运行，那么请在 File 菜单中单击 New 命令，打开一个新的标准工程。利用菜单编辑器，在窗体上添加“文件”、“编辑”、“工具”3 个主菜单，然后在每个主菜单里添加次一级菜单，相信读者朋友都有这方面的经验，所以这里就略去。接着添加一个 ToolBar 工具栏，向上面依次添加 4 个 SpeedButton 按钮。这里有必要介绍一下 SpeedButton 的用法，它可以在按钮上添加图形，还可以设置凸现和凹下的状态。然后将 SpeedButton 的属性中的 Flat 属性设置为 True，GroupIndex 的属性设置为 0，这样按钮就会呈透明状，并且具有一种平面效果。当用鼠标单击按钮时，它被压下去，释放后又会上弹起来，与常见的工具栏控件没有什么区别。添加控件后的窗体如图 2-2 所示。

2. 在编写菜单的处理事件时，最关键的是编写好 Click() 事件，例如下面这段代码：

```
if opendirlog1.Execute then
```

```
Memo1.Lines.LoadFromFile(opendirlog1.filename);
```





```
end;           //编写打开菜单的事件处理程序
if savedialog1.Execute then
Memo1.Lines.savetofile(savedialog1.filename);
end;           //编写保存菜单的事件处理程序
```

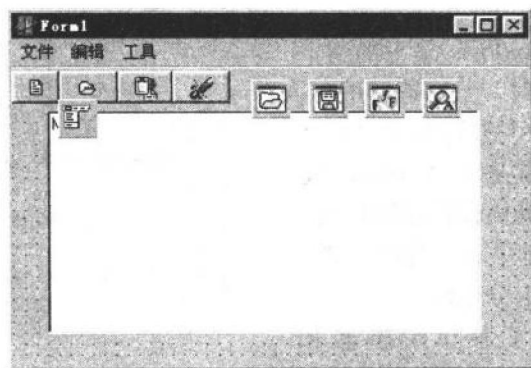


图 2-2 添加控件后的窗体

其他一些与对话框有关的菜单操作用法与上面所述类似。

3. 在编写工具栏上 SpeedButton 按钮的 OnClick()事件处理时,要注意对剪贴板的操作。如果要使控件产生 Cut 操作,利用代码: Memo1.CutToClipboard; 如果要使控件产生复制操作,利用代码: Memo1.CopyToClipboard; 如果要使控件产生粘贴操作,利用代码: Memo1.PasteFromClipboard 来完成。这些工具栏上按钮的操作同样适用于菜单上类似的用法。

4. 关于查找对话框的 TForm1.FindDialog1Find()的事件的代码编写是其中较为复杂的一段。首先需要定义两个变量: ftext 和 loc, ftext 表示所要查找的字符串, loc 代表 pos(ftext, Memo1.Lines.Text)函数的返回值。pos(ftext, Memo1.Lines.Text)函数是执行查找的函数,当所要查询的范围包括预先定义的字符串时, loc 值就返回字符串所在位置,否则为 0。接着完成查找后的字符串定位功能。

```
ftext:=finddialog1.FindText;           //将需要查找的文本赋给变量 ftext
loc:=pos(ftext, Memo1.Lines.Text);      //利用 pos 函数查找文本所在的第一个位置
if loc<>0 then                          //如果文本中包含所要查询的文本
begin
Memo1.SetFocus;                        //将控件 Memo1 作为当前对象
Memo1.selstart:=loc - 1;               //为字符串定位
Memo1.sellength:=length(ftext);        //使与查询条件相同的字符串高亮显示
而要使查找对话框可用,就需要在查找菜单中打开查找对话框,利用代码:
```

```
begin
finddialog1.Execute;                  //运行查找对话框
end;
```

本程序源代码如下: (只列出一部分,其他详见光盘)

```
procedure TForm1.N4Click(Sender: TObject);
begin
if opendialog1.Execute then
```



```
Memo1.Lines.LoadFromFile(opendialog1.filename);
end;
procedure TForm1.N6Click(Sender: TObject);
begin
if savedialog1.Execute then
Memo1.Lines.savetofile(savedialog1.filename);
end;
procedure TForm1.N7Click(Sender: TObject);
begin
close;
end;
procedure TForm1.N8Click(Sender: TObject);
begin
Memo1.CutToClipboard;
end;
procedure TForm1.N9Click(Sender: TObject);
begin
Memo1.copytoclipboard;
end;
procedure TForm1.N10Click(Sender: TObject);
begin
Memo1.PasteFromClipboard;
end;
procedure TForm1.N11Click(Sender: TObject);
begin
Memo1.SelectAll;
end;
procedure TForm1.N12Click(Sender: TObject);
begin
Memo1.Clear;
end;
procedure TForm1.N13Click(Sender: TObject);
begin
if fontdialog1.Execute then
Memo1.Font:=fontdialog1.Font;
end;
procedure TForm1.N14Click(Sender: TObject);
begin
finddialog1.Execute;
```

实例 3 自动关机程序

实例说明

注册表一直是大家非常感兴趣而又常常琢磨不透的东西。Delphi 5.0 对注册表提供了大量的操作函数，利用它们可以方便、快捷地对注册表进行操作：包括对机器的管理、权限设置的修改和文件查找。本例是用于自动关机的程序，如图 3-1 所示，在窗口右方的文本编辑框中输入所设定的关机时间，程序将实现相应时间内的关机。单击“加入注册表”按钮，把这一消息添加到注册表内。

本程序主要利用了注册表的有关函数，包括自动关机 and 把信息添加到注册表内。

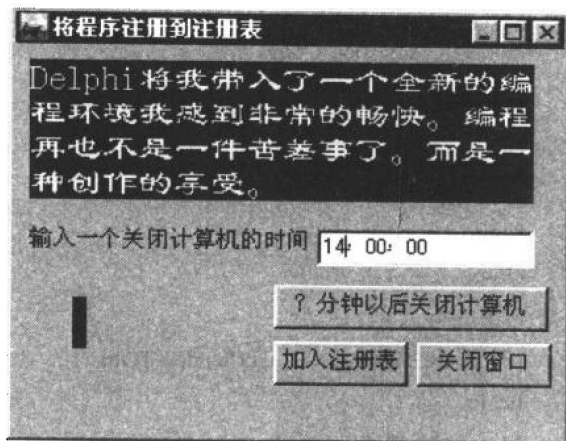


图 3-1 效果图

编程思路

本程序的难点是注册表的使用，包括如何在规定的时间内关闭计算机，如何把这一信息加入到注册表。而这些都是需要利用 Delphi 中有关注册表的函数来完成。本例先需要设置程序的外观，然后再在相对应的按钮的 OnClick() 事件中加入相应的代码，Delphi 5.0 提供了用于关闭 Windows 和用于添加到注册表的函数操作，这在源代码中可以看到。

创作步骤

1. 启动 Delphi 5.0，打开一个新的标准工程。如果 Delphi 已经运行，那么请在 File 菜单中单击 New 命令，打开一个新的标准工程。向上面加入三个 TButton 按钮、若干 Edit 控件和一个 Label 控件，并且在最大的 Label 控件中输入如图 3-1 所示的一段话，得到的添加控件后的效果如图 3-2 所示。

2. 接着编写 TForm1.RegistryWrite() 事件的处理，在这个程序中最重要代码是：writestring('registry','e:\mywebsite\002\registry.exe'); 其作用就是把关机的时间写入到注册表中，让 Windows 定时处理，这样就省去了每次都必须打开该程序的烦恼，下面详细介绍这段代码：

```
reg:TRegistry;                                //定义一个变量为 TRegistry，这是注册表函数专用的。
begin
reg:=TRegistry.Create;                        //使这个变量成为可用的
with reg do                                  //利用这个变量编辑注册表
```