

网冠科技 编著

Delphi 7.0

时尚编程百例

第3版



百例丛书



机械工业出版社
CHINA MACHINE PRESS



100

时尚百例丛书

Delphi 7.0

时尚编程百例

第3版

网冠科技 编著

光盘包含本书素材、效果文件



本书附光盘 1 张



燕山大学图书馆藏书



TP311.56/245

机械工业出版社



0663333

37

Delphi 7.0 是 Inprise 公司（原 Borland 公司）推出的可视化、面向对象的高效率快速应用程序开发工具，是一个优秀的前端开发平台。

本书融合 Delphi 7.0 开发应用程序的理论和实践，全面深入地讲解了利用 Delphi 7.0 开发应用程序的常用方法和技巧。书中包含 100 个实例，全面讲解了使用 Delphi 7.0 进行程序设计的开发方法和技巧。内容包括：Delphi 7.0 的基本应用、控件的使用技巧、文件操作技巧、系统及对象操作、图形图像及多媒体处理、网络和数据库编程技巧等内容。本书内容丰富，包含的实例功能强大。

本书既可作为不同层次计算机培训班的教材，也可以作为命令参考手册使用。

图书在版编目（CIP）数据

Delphi 7.0 时尚编程百例 / 网冠科技编著. -3 版

-北京：机械工业出版社，2003.1

（时尚百例丛书）

ISBN 7-111-08564-7

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（2000）第 76583 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：孙 业

责任印制：付方敏

北京铭成印刷有限公司印刷·新华书店北京发行所发行

2003 年 1 月第 3 版·第 1 次印刷

787mm × 1092mm 1/32 · 21 印张·521 千字

25 001-30 000 册

定价：37.00 元（含 ICD）

凡购本书，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

封面无防伪标均为盗版

时尚百例丛书

追求时尚 追求完美

出版说明

随着计算机迅速应用于人们工作和生活的各个方面,越来越多的职业需要具有计算机应用技能的人。

如何结合自己的实际工作,选择要学习的软件技术?下面提供了各具体行业的工作人员需学习和掌握的应用软件及相关技术,供读者参考:

- 平面设计及相关行业——Photoshop、CorelDRAW、Illustrator、Pagemaker、FreeHand、PhotoImpact 等软件;
- 三维及相关行业——3DS MAX、Maya、SoftImage XSI、Poser、Lightwave 等软件;
- 多媒体设计及相关行业——Authorware、Director、Premiere、Combustion、After Effects、Cult 3D、Web3D 等软件;
- 网络应用及相关行业——Flash、Dreamweaver、Fireworks、FrontPage、ASP、ASP.NET、HTML、PHP、JavaScript、VBScript 等软件及组网建网技术;
- 建筑及装潢设计行业——AutoCAD、3DS MAX、3DS VIZ、Lightscape 等软件;
- 现代工业产品及相关行业——Alias、Pro/E、Solidworks、UG、I-Deas、Rhino、Protel 等软件;
- 软件开发及相关行业——VB.NET、VC.NET、VB、VC、VFP、Delphi、PowerBuilder、C/C++、C++ Builder、JBuilder 等编程软件;
- 办公及应用行业——Windows 9x\xp、Office、WPS Office 等软件、及硬件故障排除和网络等技术。

所有与计算机相关的职业,都要求其工作人员有很强的计算机操作技能,熟练地掌握各种相关软件的应用。要做到这一点,必须在掌握软件的基本操作方法的前提下,通过实例演练的方法训练自己,只有通过反复练习,才能做到举一反三,在工作实践中灵活高效地应用。

为了让读者迅速地熟练掌握各种软件的应用方法和技巧,我们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”,对每一个常用软件都精心制作了100个实例,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,内容通俗易懂,将软件知识和实例紧密结合。通过对各种实例的详细讲解和操作实践,即使是事先没有学习过这种软件的读者,也能从实例的制作过程中体会到这种软件的各项功能的使用方法,并能自己制作出各种实例的效果。这样既节省了读者的大量时间,又能使读者在反复实践的同时,提高学习兴趣,并将学到的知识和技能迅速应用到实际工作中去。

机械工业出版社

0751561



前言

《Delphi 7.0 时尚编程百例》是“时尚百例丛书”中的一本。

Delphi 7.0 是 Inprise 公司新一代面向对象、可视化应用程序的开发环境，运行于 Windows 95/98/2000/NT/XP 操作系统上。Delphi 7.0 的基础语言是 Pascal，它继承了 Pascal 语言严谨的优点，如代码结构清晰、可读性好和代码执行效率高等。使用 Delphi 7.0 可以编写 Win 32 控制台应用程序及具有图形用户界面 (GUI) 的 Win 32 应用程序。Delphi 7.0 的集成开发环境不仅提供了一整套用于设计、编写、测试、调试和发布应用程序的工具软件，同时还包含了大量的控件、模板和 Wizards。它与传统的 Windows 应用程序的开发方法相比，具有快速和真正可视化的特点。更加难能可贵的是，Delphi 7.0 非常适用于多媒体、动画、网络 and 数据库等应用程序的开发。

本书以实例的形式向读者解析如何使用 Delphi 7.0。全书制作了 100 个实例，满足不同层次读者开发各种应用程序的需要。每个实例都具有较强的针对性，突出讲解了 Delphi 7.0 的开发思想、技术和方法。

Delphi 是一种功能强大的工具，由于含有大量不同的单元，因此与其他编程环境相比相对复杂一些。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，根据不同的编程方向，本书分为 6 篇进行讲解。第一篇讲解 Delphi 7.0 的基本应用；第二篇讲述控件的使用技巧；第三篇讲解对文件的操作技巧；第四篇讲解系统和对象的操作，其中包含多种系统控制技巧；第五篇讲解图形图像及多媒体方面的处理技巧；第六篇讲解网络和数据库的技巧，全书内容丰富，包含的实例功能强大。

本书主要供各级 Delphi 用户及编程爱好者阅读。由于作者水平有限，书中难免存在错误和疏漏，恳请读者批评指正。



网冠科技

本书光盘含配套素材（使用方法请见光盘中“光盘使用说明书”），技术支持请点击网冠科技站点 <http://netking.163.com>。E-mail: www1@publicb.bta.net.cn。

目 录

出版说明 前 言

第一篇 Delphi 7.0 基 本 应 用

实例 1	Windows 系统动画	2
实例 2	查看按键的 ASCII 值	6
实例 3	喷射屏幕的制作	9
实例 4	不同的图像显示方法	12
实例 5	梦幻图形	16
实例 6	素描程序	18
实例 7	透明窗体	21
实例 8	动态工具栏	23
实例 9	休眠的青蛙	27
实例 10	获得计算机硬件信息	30
实例 11	统计程序	33
实例 12	渐变窗体	36
实例 13	浮动工具栏	39
实例 14	无标题栏的窗体	42
实例 15	查看其他计算机的共享资源	45
实例 16	根据程序数据创建图表	49
实例 17	信使服务	52
实例 18	Ping 操作	57
实例 19	超链接	62
实例 20	显示数据库中的多媒体信息	65
实例 21	使用 Database Form Wizard 创建数据库	68
实例 22	实现网上邻居的文件复制	71
实例 23	得到 MAC 地址	74
实例 24	获得收藏夹的网址并且实现链接	77
实例 25	获取指定计算机的共享资源	81
实例 26	新邮件到达的通知	85
实例 27	编写需要服务器验证的邮件发送程序	89
实例 28	Jpeg 格式的图像转换为 Bmp 格式的图像文件	93

实例 29	远程时间查询	96
实例 30	测试网络速度	99

第二篇 控件应用

实例 31	TRadioButton 控件	103
实例 32	组合框组件 TComboBox	105
实例 33	PaintBox 控件	107
实例 34	TProgressBar 控件	109
实例 35	列表控件	111
实例 36	TListBox 控件	113
实例 37	拖动列表项	116
实例 38	拖动树型节点	119
实例 39	TRichEdit 控件	122
实例 40	TTimer 控件	125
实例 41	TStringList 控件	128
实例 42	动态菜单	132
实例 43	自定义颜色对话框	137
实例 44	可移动的分割窗体	140
实例 45	使用 ListView 控件	141
实例 46	响应 ListBox 控件的滚动条事件	145
实例 47	将指定文件的路径拖动到 Memo 控件中	149
实例 48	大小写转换	151

第三篇 文件操作

实例 49	统计文本的中英文个数	155
实例 50	文件阅读器	157
实例 51	查找、替换工具	160
实例 52	生成应用程序的快捷方式	164
实例 53	OLE 自动化	167
实例 54	多文档程序中实现菜单合并	171
实例 55	显示时间	177
实例 56	文件关联操作	180
实例 57	取得指定文件夹的信息	188

第四篇 系统和对象

实例 58	限制程序执行个数	194
-------	----------------	-----

实例 59	隐藏桌面图标	196
实例 60	隐藏任务栏	198
实例 61	获得临时目录	200
实例 62	驱动器容量	201
实例 63	获得内存使用状态	203
实例 64	动态改变显示器分辨率	205
实例 65	打印预览程序	207
实例 66	系统托盘程序	213
实例 67	创建对象	218
实例 68	使用对象	222
实例 69	检测声卡	226
实例 70	编写 Hook 程序	227
实例 71	测试 Hook 程序	231
实例 72	使用 Application 对象	233
实例 73	测试 CPU 速度	235
实例 74	树型窗口	238
实例 75	映射网络驱动器	240

第五篇 图形图像及多媒体

实例 76	图形的显示	243
实例 77	反射效果	245
实例 78	使用材质填充曲面	249
实例 79	制作六面体	252
实例 80	TMediaPlayer 控件	258
实例 81	动画按钮	262
实例 82	使用 Animate 控件	264
实例 83	转动的地球仪 (1)	267
实例 84	转动的地球仪 (2)	269
实例 85	动态改变光标和鼠标	273
实例 86	使用 wav 文件	275
实例 87	使用资源文件	277
实例 88	立体迷宫	283
实例 89	火炮的例子 (1)	287
实例 90	火炮的例子 (2)	289

第六篇 网络和数据库

实例 91	SQL 编辑器	292
-------	---------------	-----

实例 92	使用缓冲池技术	301
实例 93	网页浏览器界面设计	305
实例 94	网页浏览器代码编写	308
实例 95	聊天室	311
实例 96	IP 地址输入框	315
实例 97	手动创建数据库	318
实例 98	使用 SQL	320
实例 99	远端服务器	322
实例 100	同远端服务器交互	325

第一篇

Delphi 7.0 基本应用

本篇总览

Delphi 是全新的可视化编程环境，它提供了一种方便、快捷的 Windows 应用程序开发工具。目前 Delphi 正在为提供一套完整的企业级网络开发平台努力。Delphi 7.0 更全面地支持 XML 语言，提供了更多的网络开发控件，同时 Delphi 7.0 还支持 Microsoft 公司的 .net 技术。这样程序员就能够随心所欲地开发各种应用程序。

本篇讲述了 Delphi 7.0 的一些新增功能，以及读者编程时经常要用到的各种技术。

实例 1 Windows 系统动画

实例说明

在 Windows 操作系统中执行一些操作时,例如复制、删除文件,经常可以看到一些动画,这往往给用户产生良好的视觉效果。利用 Delphi 为程序添加动画,是一件很容易的事。Win32 控件板中的 Animate 控件可以播放任意的 AVI 剪辑文件,从而实现在窗口中播放动画的效果。

如图 1-1 所示,在左侧的单选按钮组中选择一种动画类型,单击“演示”按钮就可实现动画的播放,窗体的右下角提供了动画类型的文字提示,单击“停止”按钮,动画即停止播放。

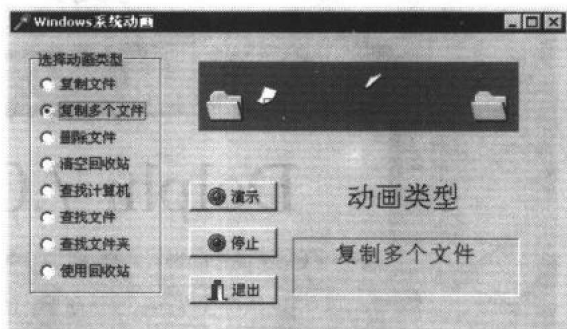


图 1-1 效果图

编程思路

Animate 控件有众多的属性,CommonAVI 属性是较为重要的一个,利用它可以直接调用标准的 WindowsAVI 剪辑文件。CommonAVI 属性有如下参数可选: aviNone、aviCopyFile、aviCopyFiles、aviDeleteFile、aviEmptyRecycle、aviFindComputer、aviFindFile、aviFindFolder、aviRecycleFile。本程序的核心在于,如何用 Animate 控件进行选择、播放和停止动画。为了播放动画,可以通过 Onclick 事件调用 Animate 控件的 Play 方法: paly (1, FrameCount, 0); 而相应地调用 Stop 方法则可停止动画的播放。为了显示当前播放的动画名称,调用了 name 属性为 StatusText 的 StaticText 对象,再定义常数数组 aviStrings (存储有需要实现的动画类型的名称),将 aviStrings 数组中的值赋给 StatusText 即可。

创作步骤

1. 启动 Delphi 7.0, 执行 File→New→Application 命令打开一个新的标准工程。将窗体的尺寸设置为 280×480, 在窗体上加入 RadioGroup 控件, 并将其放置在窗体的左侧。选中 RadioGroup 控件, 在 Object Inspector (对象管理器) 中, 将 Caption 设为“选择动画类型”; 单击 Items 右侧的按钮, 在弹出的 String List Editor 对话框中输入下列文本:

复制文件
复制多个文件
删除文件

清空回收站

查找计算机

查找文件

查找文件夹

使用回收站

2. 加入 Animate 控件, 调整其大小, 并放置在窗体的右上方, 将 Color 属性设为 clOlive.

3. 加入三个 BitBtn 按钮, 分别将 Caption 属性设为“演示”、“停止”、“退出”; 再单击 Glyph 属性右侧的按钮, 并分别为它们加载合适的、具有功能提示的图片。

4. 在窗体的右下方分别加入一个 Label 控件和一个 StaticText 控件。图 1-2 为添加控件完毕后的窗体效果。

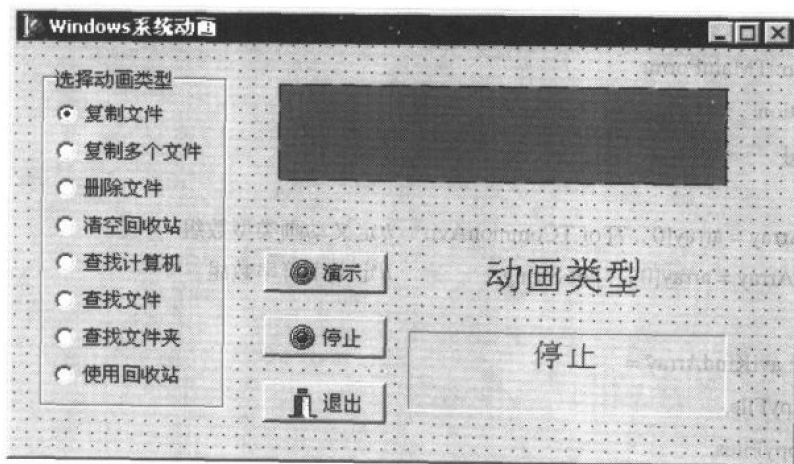


图 1-2 添加控件完毕后的窗体效果

5. 为程序添加事件处理过程代码, 这里重点说明控制动画播放的 Play 方法。Play (1, FrameCount, 0) 方法共有三个参数, 第一个和第二个参数用来指定被播放动画的帧的编号范围, 1 表示第一帧; 最后一个参数则是用来指定重复播放动画的次数, 如果将其指定为 0, 可以使动画一直播放下去, 直到 Stop 方法被调用为止。

本程序完整的代码如下:

```
unit Main;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Buttons, ExtCtrls, ComCtrls;
type
  TMainForm = class(TForm)
    Animate1: TAnimate;
    RadioGroup1: TRadioGroup;
    GoBitBtn: TBitBtn;
    StopBitBtn: TBitBtn;
```



```
    BitBtn1: TBitBtn;
    StatusText: TStaticText;
    Label1: TLabel;
    procedure GoBitBtnClick(Sender: TObject);
    procedure StopBitBtnClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    MainForm: TMainForm;
implementation
{$R *.DFM}
type
    aviKindArray = array[0 .. 7] of TCommonAvi; //定义动画类型数组
    aviStringArray = array[0 .. 7] of String;    //定义字符串数组
const
    aviKinds: aviKindArray =
        (aviCopyFile,
         aviCopyFiles,
         aviDeleteFile,
         aviEmptyRecycle,
         aviFindComputer,
         aviFindFile,
         aviFindFolder,
         aviRecycleFile); //将动画类型赋值于 aviKindArray 数组
    aviStrings: aviStringArray =
        ('复制文件',
         '复制多个文件',
         '删除文件',
         '清空回收站',
         '查找计算机',
         '查找文件',
         '查找文件夹',
         '使用回收站'); //将动画类型的中文名称赋值于 aviKindArray 数组
procedure TMainForm.GoBitBtnClick(Sender: TObject);
var
    AnimIndex: Integer; // 动画类型索引
```

```
begin
  AnimIndex := RadioGroup1.ItemIndex;
  with Animate1 do
  begin
    StatusText.Caption := aviStrings[AnimIndex];
    CommonAVI := aviKinds[AnimIndex];
    Play(1, FrameCount, 0);    // 开始动画
  end;
end;

procedure TMainForm.StopBitBtnClick(Sender: TObject);
begin
  Animate1.Stop;              // 暂停动画
  StatusText.Caption := '停止';
end;
end.
```

实例 2 查看按键的 ASCII 值

实例说明

在 Delphi 中, 可以构造一个过程来处理 OnKeyDown、OnKeyPress、OnKeyUp 等事件或者这些事件的任意组合, 使得应用程序能够接收键盘的输入。

这里编制了一个可以查看按键状态和对应字符的 ASCII 值的程序。按下键盘上的任意按键, 将在窗体上显示出所按键的名称及其 ASCII 值, 还可以使用 Shift 键进行上档切换, 如图 2-1 所示。ASCII 码是“美国信息交换标准代码”的简称, 是目前世界上最为流行的字符信息编码方案。

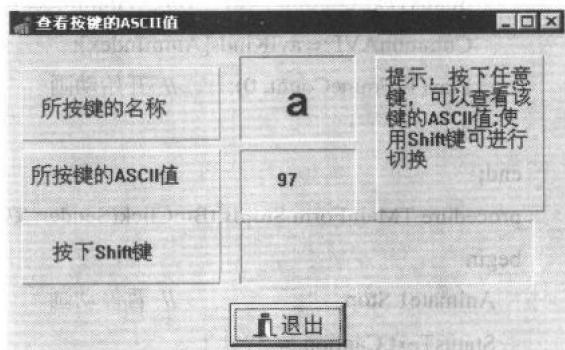


图 2-1 效果图

编程思路

下面先简要介绍键盘响应事件的功能:

OnKeyDown —— 按下任意键, 包括功能键和特殊键, 例如 Shift、Alt 和 Ctrl 键等, 均将触发该事件。

OnKeyPress —— 按下 ASCII 字符键和控制键将触发该事件。

OnKeyUp —— 释放一个按下的键将触发该事件。

这些事件触发时, 均将接收一个名为 Key 的参数, 代表一个按键。在 OnKeyDown 和 OnKeyUp 事件中, Key 代表 Windows 下的一个虚键代码, 而在 OnKeyPress 事件中 Key 代表一个 ASCII 字符对应的虚键代码。虚键代码一般是以 vk_ 开头, 如 vk_ctrl 是 Ctrl 键的虚键代码。

创作步骤

1. 启动 Delphi 7.0, 执行 File→New→Application 命令打开一个新的标准工程。
2. 为窗体加入 7 个 Bevel 控件, 将 3 个 Bevel 控件放置在窗体的左侧, 并将 Style 属性设置为 bsRaised; 将另外 3 个 Bevel 控件分别于前面 3 个 Bevel 对齐, 再调节它们的大小, 并将 Style 属性设置为 bsLowered; 将 1 个 Bevel 控件放置于窗体的右上侧, 调整为正方形形状, Style 属性设置为 bsRaised。
3. 将 7 个 Label 控件分别放置于 7 个 Bevel 控件上, 再为窗体添加一个 BitBtn 按钮, 并为 BitBtn 按钮加载一幅具有功能提示作用的图片。
4. 添加完控件后的窗体效果如图 2-2 所示。

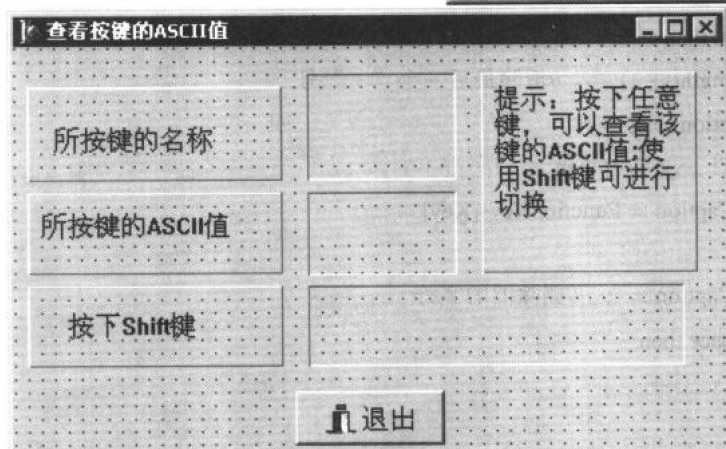


图 2-2 添加完控件后的窗体

5. 为程序添加事件处理过程代码。在 Main 单元的实现部分中，首先定义了常量 Ctrl_A 和 Ctrl_Z，它们分别代表 Ctrl+A 和 Ctrl+Z 键的 ASCII 值：

```
const
  ctrl_A = 1;           //Ctrl+A 组合键的 ASCII 值
  ctrl_Z = 26;          //Ctrl+Z 组合键的 ASCII 值
```

程序中用这些值定义了控制键的范围。另外还利用了一个常量数组，可以自动初始化括号中的值：

```
FunctionKeys: array [vk_f1 .. vk_f12] of string[3] =
  ('F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12');
```

6. 以下是该程序的声明部分的完整代码，其余部分可以参考光盘：

implementation

{ \$R *.DFM }

const

```
  ctrl_A = 1;           //Ctrl+A 组合键的 ASCII 值
  ctrl_Z = 26;          //Ctrl+Z 组合键的 ASCII 值
```

```
  FunctionKeys: array [vk_f1 .. vk_f12] of string[3] =
    ('F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12');
```

procedure TMainForm.FormKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

var

s: string;

begin

ValueLabel.Caption := IntToStr(Key); //将 Key 的值转换为字符型

s := '';

//显示组合键状态

if ssShift in Shift then s := s + 'Shift+';

if ssAlt in Shift then s := s + 'Alt+';

if ssCtrl in Shift then s := s + 'Ctrl+';

```

if Length(s) > 0 then
    Delete(s, Length(s), 1);    //清除最后一个“+”号
ShiftLabel.Caption := s;
if Key in [vk_f1 .. vk_f12] then
    CharLabel.Caption := FunctionKeys[Key]
else
    CharLabel.Caption := "";    //删除旧的字符
if Key = vk_space then
    Key := 0;
end;
procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char);
begin
    if Ord(Key) in [ctrl_A .. ctrl_Z] then
        CharLabel.Caption := '^' + Chr(Ord(Key) + Ord('A') - 1)
    else
        CharLabel.Caption := Key;
        ValueLabel.Caption := IntToStr(Ord(Key));
end;
procedure TMainForm.FormKeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    //释放按键后清空各个显示框
    CharLabel.Caption := "";
    ValueLabel.Caption := "";
    ShiftLabel.Caption := "";
end;
end.

```