

时尚百例丛书



Delphi 6.0

时尚编程百例

TP312DE
908

100%
时尚百例丛书

Delphi 6.0

时尚编程百例

网冠科技 编著

光盘包含本书素材、效果文件



机械工业出版社

Delphi 是面向对象的可视化编程语言，它是目前面向对象和控件开发性能最好的工具之一。Delphi 6.0 是其最新版本。

本书通过 100 个实例，讲解 Delphi 6.0 的开发和应用方法。全书共分 7 篇，包括 Delphi 6.0 新功能、控件应用、文件操作、硬件和系统、图形图像、多媒体、网络和数据库等方面的知识。

本书既可作为不同层次培训班的教材，也可以作为 Delphi 爱好者的命令参考手册使用。

图书在版编目 (CIP) 数据

Delphi 6.0 时尚编程百例 / 网冠科技编著.

—北京: 机械工业出版社, 2001.10

(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: 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 10 月第 2 版·第 1 次印刷

787mm × 1092mm $\frac{1}{16}$ · 20.25 印张 · 498 千字

14001-20000 册

定价: 37.00 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。

为了让读者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 and 事业中去。

机械工业出版社



前 言

《Delphi 6.0 时尚编程百例》是“时尚百例丛书”中的一本。

在计算机软件开发领域中流传着这样的说法：“真正的程序员用 VC，聪明的程序员用 Delphi”。这足以体现 Delphi 的巨大魅力。Delphi 是美国 Borland 公司推出的可视化编程产品，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。2001 年 6 月份，Borland 公司发布了 Delphi 6.0。

目前有众多的 Windows 下的开发工具，而 Delphi 的开发环境是目前面向对象和控件开发性能最好的工具之一。它使用 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的、可重复利用的、完整的、面向对象的程序语言，拥有当今世界上最快的编辑器，以及最为领先的数据库技术，功能强大。对于广大的程序开发人员来讲，使用 Delphi 开发应用软件，无疑会大大提高编程效率。

本书主要以实例为主，全书精心制作了 100 个实例。全书从 7 个方面进行讲解：第一篇为 Delphi 6.0 新功能；第二篇为控件应用；第三篇为文件操作；第四篇为硬件和系统；第五篇为图形图像；第六篇为多媒体；第七篇为网络和数据库。本书重点在 Delphi 6.0 新功能、控件的使用、图形图像、多媒体、网络和数据库几个方面。

全书的 100 个实例均在 Delphi 6.0 下调试通过，用户只需将光盘中的工程文件复制到用户的计算机中，并将文件的只读属性取消，在 Delphi 中打开工程文件即可进行编译和调试。建议用户阅读了每个实例的“实例说明”和“编程思路”部分之后，首先思考实例中要实现的重点（如果能自己尝试编制则最佳），再去运行光盘中相应的程序，并与自己的想法进行对照，这样的学习效果要好得多。

编程是实践性很强的技术过程，只有多动手才能体会开发工具的精华。鉴于本书丰富的实例，读者完全可以在参考书中实例的基础上自己多做些练习，使所学知识融会贯通。本书由杨聪、胡韬、李景彬等编著，由于时间和水平的限制，书中缺点和错误之处，希望读者批评指正。



网冠科技

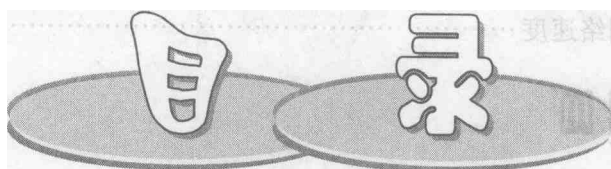
本书光盘含配套素材（使用方法请见光盘中“光盘使用说明书”），技术支持请点击网冠科技站点 <http://netking.163.com>。E-mail: netking_@yeah.net。

網易 NETEASE
WWW.163.COM

是网易

公司的标志。





出版说明 前 言

第一篇 Delphi 6.0 新功能

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	带阴影的窗体	16
实例 6	素描程序	19
实例 7	透明窗体	22
实例 8	动态工具栏	24
实例 9	休眠的青蛙	28
实例 10	获得计算机硬件信息	31
实例 11	统计程序	34
实例 12	渐变窗体	37
实例 13	浮动工具栏	40
实例 14	无标题栏的窗体	43
实例 15	查看其他计算机的共享资源	46
实例 16	根据程序数据创建图表	50
实例 17	信使服务	53
实例 18	Ping 操作	58
实例 19	超链接	63
实例 20	创建简单的数据库	66
实例 21	使用 Database Form Wizard 创建数据库	69
实例 22	实现网上邻居的文件复制	72
实例 23	得到 MAC 地址	75
实例 24	获得收藏夹的网址并且实现链接	78
实例 25	获取指定计算机的共享资源	82
实例 26	新邮件到达的通知	86
实例 27	编写需要服务器验证的邮件发送程序	90
实例 28	Jpeg 格式的图像转换为 Bmp 格式	94



实例 29	远程时间查询	97
实例 30	测试网络速度	100

第二篇 控件应用

实例 31	TRadioButton 控件	104
实例 32	组合框组件 TComboBox	106
实例 33	PaintBox 控件	108
实例 34	TProgressBar 控件	110
实例 35	列表控件	112
实例 36	TListBox 控件	114
实例 37	拖动列表项	117
实例 38	拖动树型节点	120
实例 39	TRichEdit 控件	123
实例 40	TTimer 控件	126
实例 41	TStringList 控件	129
实例 42	动态菜单	133
实例 43	自定义颜色对话框	138
实例 44	可移动的分割窗体	141
实例 45	格式化文本	142
实例 46	TStringGrid 控件	144
实例 47	动态生成控件	147
实例 48	图像浏览器	148

第三篇 文件操作

实例 49	统计文本的中英文个数	153
实例 50	文件阅读器	155
实例 51	查找替换工具	158
实例 52	生成到应用程序的快捷方式	162

第四篇 硬件和系统

实例 53	获得 Windows 的用户名和序列号	166
实例 54	操作系统类型和版本	168
实例 55	取得登录 Windows 用户密码	170
实例 56	锁定鼠标	173
实例 57	在 Windows 桌面显示内容	175
实例 58	限制程序执行个数	178
实例 59	隐藏桌面图标	180

实例 60	隐藏任务栏	182
实例 61	获得临时目录	184
实例 62	驱动器容量	185
实例 63	获得内存使用状态	187
实例 64	动态改变显示器分辨率	189
实例 65	打印预览程序	191

第五篇 图形图像

实例 66	得到某一点的色彩值	198
实例 67	使用 JPG 文件	201
实例 68	Form 背景贴图	203
实例 69	将 ico 文件转化为位图	205
实例 70	倾斜的文字	208
实例 71	位图显示 (1)	211
实例 72	位图显示 (2)	214
实例 73	抓取屏幕	216
实例 74	处理位图	222
实例 75	将彩色位图转化为黑白图	227
实例 76	图形的显示	230
实例 77	绘图程序 (1)	232
实例 78	绘图程序 (2)	237
实例 79	图形的局部放大	242

第六篇 多媒体

实例 80	TMediaPlayer 控件	248
实例 81	动画按钮	252
实例 82	使用 Animate 控件	254
实例 83	转动的地球仪 (1)	257
实例 84	转动的地球仪 (2)	259
实例 85	动态改变光标和鼠标	263
实例 86	使用 wav 文件	265
实例 87	使用资源文件	267
实例 88	立体迷宫	273
实例 89	火炮的例子 (1)	277
实例 90	火炮的例子 (2)	279

第七篇 网络和数据库

实例 91	获得本地的 IP	282
实例 92	网络聊天工具	285
实例 93	FTP 实现	290
实例 94	静态网页	295
实例 95	动态网页	298
实例 96	DataBase Form Wizard 建立程序	300
实例 97	手动创建数据库	305
实例 98	使用 SQL	307
实例 99	远端服务器	309
实例 100	同远端服务器交互	312

第一篇

Delphi 6.0 新功能

本篇总览

Delphi 6.0 新增加了网络和数据库功能，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。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的可重复利用的、完整的、面向对象的程序语言(Object-Oriented Language)，以及当今世界上最快的编辑器、最为领先的数据库技术。对于广大的程序开发人员来讲，使用 Delphi 开发应用软件，无疑会大大地提高编程效率，而且随着应用的深入，将会发现编程不再是枯燥无味的工作——Delphi 的每一个设计细节，都将带来一份欣喜。

本篇通过实例讲解了 Delphi 6.0 的新功能和特性。

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。

在本例中,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,如图 1-1 所示。

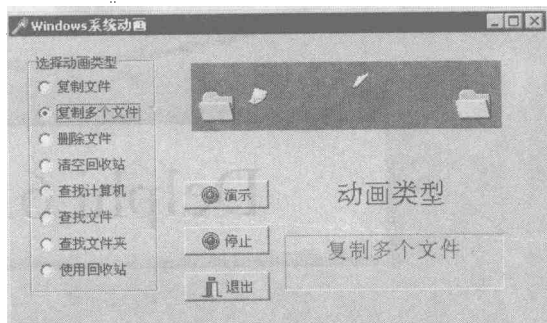


图 1-1 效果图

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:play(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,调用了一个 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 6.0,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。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 `Lable` 控件和一个 `StaticText` 控件。图 1-2 为添加控件完毕后的窗体效果。

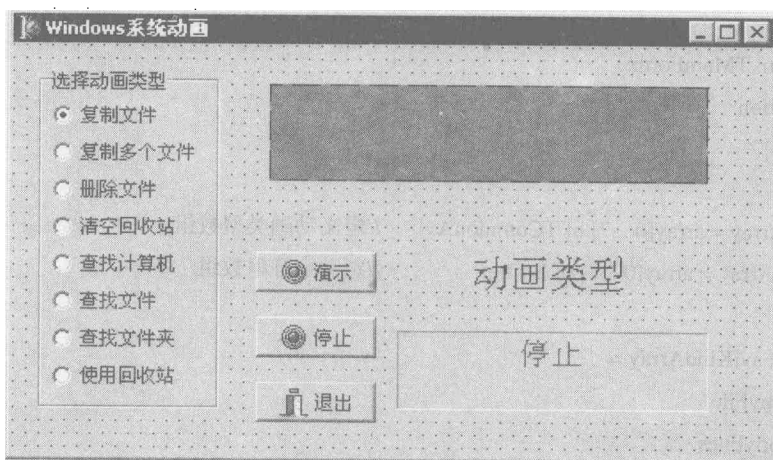


图 1-2 添加控件完毕后的窗体效果

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的状态和其对应的字符的 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，如图 2-1 所示，还可以使用 Shift 键进行切换。ASCII 码是“美国信息交换标准代码”的简称，是目前世界最为流行的字符信息编码方案。

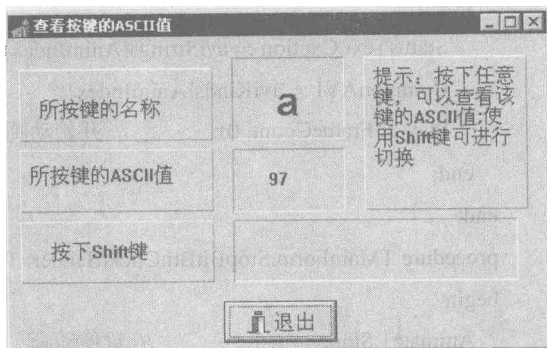


图 2-1 效果图

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 6.0，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。
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。

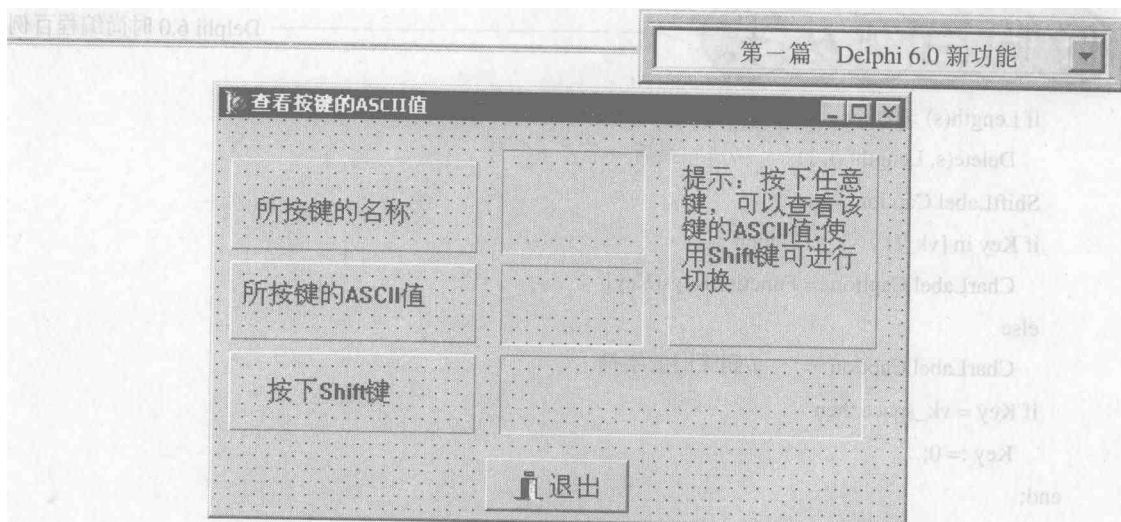


图 2-2 添加完控件后的窗体

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.
```