

计算机高级编程技术



第八章 建立用户界面





第8章 建立用户界面

在Delphi编程设计界面的过程中，要经常使用到组件。通过可视组件，可以在窗体中方便地构建与用户交互的界面。另外一些不可视的组件可以为程序增加许多强大的功能。在Delphi集成开发环境的组件栏中，按照功能分类罗列了大量的组件，熟悉它们可以提高开发程序的进度。

- 8.1 常用的组件
- 8.2 常用组件的用法





8.1 常用的组件

- 8.1.1 组件的一般操作
- 8.1.2 文本组件
- 8.1.3 特殊输入组件
- 8.1.4 按钮及其分类组件
- 8.1.5 列表组件
- 8.1.6 分组组件
- 8.1.7 信息反馈组件
- 8.1.8 图形显示组件





8.1.1 组件的一般操作

在窗体中使用组件主要包括：向窗体中添加组件；删除窗体中的组件；调整组件的大小和位置；设置可视组件的TabOrder；组件的锁定和解锁；设置窗体中组件的属性；创建组件的事件处理过程。

□ 1. 向窗体中添加组件

- ① 在组件栏中需要添加的组件上单击鼠标左键，鼠标形状发生改变。然后在窗体中再单击一下鼠标左键，就将组件添加到窗体中了。





8.1.1 组件的一般操作

- ② 在组件栏中需要添加的组件上单击鼠标左键，鼠标形状发生改变。然后在窗体中某个位置按下鼠标左键并拖动，松开鼠标后，组件就以之前拖动的大小出现在窗体中。
- ③ 在组件栏中双击组件，则被双击的组件会自动添加到当前默认的窗口或容器组件中。
- ④ 在窗体中选择一个组件或多个，通过菜单命令“Edit”→“Copy”复制到剪贴板中，然后通过菜单命令“Edit”→“Paste”将剪贴板中的组件粘贴到窗体中。





8.1.1 组件的一般操作

❑ 2. 删除窗体中的组件

- 如果要删除窗体中的组件，可以先选中然后〈Delete〉键就可以了。也可以通过主菜单中的“Edit”→“Cut”命令实现。
- 此外，也可以一次选中多个组件进行删除操作。选定多个组件的方法有：
 - ① 按下〈Shift〉键的同时，利用鼠标左键单击各个组件。
 - ② 在窗体上没有组件的地方按下鼠标左键并拖动，将会出现一个虚框。松开鼠标左键后，该虚框“触及”到的组件就被选中了。





8.1.1 组件的一般操作

□ 3. 调整组件的大小和位置

改变组件在窗体中的大小和位置可以采用下面的方法：

① 鼠标操作

利用鼠标左键单击组件，被选中的组件四周出现8个正方形黑点。在组件上按下鼠标左键并拖动，可以方便地改变组件的位置。如果将鼠标移动到某个黑点上，鼠标形状将变成上下拖动的形状，此时可以按下鼠标左键进行拖动，改变组件的大小。





8.1.1 组件的一般操作

□ 3. 调整组件的大小和位置

改变组件在窗体中的大小和位置可以采用下面的方法:

② 键盘操作

利用键盘的上〈↑〉、下〈↓〉、左〈←〉、右〈→〉移动键可以在窗体中选择组件。选中某个组件后，通过〈Ctrl〉+〈↑〉、〈Ctrl〉+〈↓〉、〈Ctrl〉+〈←〉、〈Ctrl〉+〈→〉可以改变组件的位置，通过〈Shift〉+〈↑〉、〈Shift〉+〈↓〉、〈Shift〉+〈←〉、〈Shift〉+〈→〉可以改变组件的大小。





8.1.1 组件的一般操作

□ 4. 设置可视组件的TabOrder

在Windows的对话框中可以通过按下〈Tab〉键，依次定位到各个具有输入焦点的可视组件。可以通过设置组件的TabOrder属性来确定这种定位的顺序。

□ 5. 组件的锁定与解锁

一般在窗体中将组件的大小和位置安排好后，可以将组件锁定，这样就不会因为误操作而改变已经设置好的大小和位置。





8.1.1 组件的一般操作

□ 6. 设置窗体中组件的属性

在窗体中选中单个组件后，在对象编辑器中就可以对其属性进行设置了。如果选中了多个组件，则可以统一地进行属性设置。

□ 7. 创建组件的事件处理过程

在窗体中选中某个组件后，在对象编辑器的Events属性页中就可以创建相应的事件处理过程。双击某个事件右边的下拉式组合框，则自动创建该组件的事件处理过程，并将光标定位到代码编辑器中。





8.1.2 文本组件

在应用程序的界面上，往往需要用户输入一些文字，或者以一定的格式将文本显示给用户，这时可通过文本组件来实现。

组件名	功能
Edit	Edit组件可以显示、编辑单独的一行文本
Memo	Memo组件可以显示、编辑多行文本
MaskEdit	MaskEdit组件可以显示、编辑具有指定格式的文本，如6位数字的邮政编码、固定格式的电话号码等
RichEdit	RichEdit组件可以显示、编辑具有丰富文本格式(Rich Text Format)的文件，如.rtf文件。





8.1.2 文本组件

□ 1. Edit组件

Edit组件是常用组件之一，在使用Edit组件的时候应注意：

- ① **AutoSelect**属性可以设置当Edit组件获得输入焦点时，是否自动选中所有的文本。
- ② **PasswordChar**属性可将用户输入的所有字符以同一个字符显示。
- ③ **Edit**组件最常用的事件（也是默认事件）为**OnChange**，也就是当Edit组件中的文本发生改变的时候，该事件的处理过程执行。





8.1.2 文本组件

□ 2. Memo组件

对一些比较长的文本内容可以利用Memo组件来显示、编辑。在使用Memo组件的过程中应注意：

- ① 可以通过Text属性来存取文本内容。
- ② 可以通过Modified属性来确定文本内容是否发生了改变。
- ③ Memo组件没有从TCustomEdit类继承AutoSelect属性，所以不可以使用该属性。

□ 3. MaskEdit

可以通过EditMask属性来设置输入的格式。





8.1.2 文本组件

□ 4. RichEdit

在使用RichEdit组件的时候，有下面的注意点：

① 如果PlainText属性设置为True，则在RichEdit组件中显示RTF文件的时候，将格式标记以普通文本的方式显示出来。如果PlainText属性设置为False，则根据RTF文件中的格式设置来显示。通常将PlainText属性设置为False。

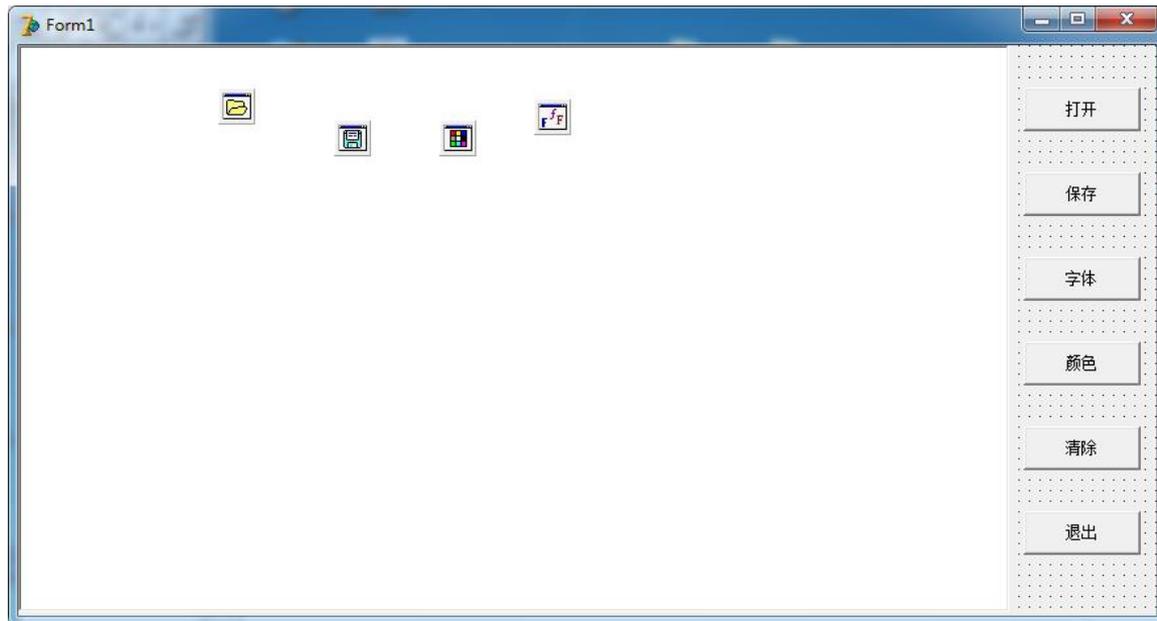
② 通过方法Lines.LoadFromFile和Lines.SaveToFile可以方便地存取RTF文件。





8.1.2 文本组件

实例8-1：记事本





8.1.2 文本组件

控件设置

控件名	设置	备注
Memo	Lines:	显示文本
OpenDialog	Filter: 文本文件 *.txt	打开文件
SaveDialog	Filter: 文本文件 *.txt	保存文件
ColorDialog		设置字体颜色
FontDialog		设置字体
button		





8.1.2 文本组件

主要代码1

```
procedure TForm1.Button1Click(Sender: TObject); //打开
begin
  OpenFileDialog1.Execute;
  memo1.Lines.LoadFromFile(OpenDialog1.FileName);
end;
procedure TForm1.Button2Click(Sender: TObject); //保存
begin
  SaveDialog1.Execute;
  memo1.Lines.SaveToFile(SaveDialog1.FileName);
end;
```





8.1.2 文本组件

主要代码2

```
procedure TForm1.Button3Click(Sender: TObject); //字体
begin
  FontDialog1.Execute;
  Memo1.Font:=FontDialog1.Font;
end;
procedure TForm1.Button4Click(Sender: TObject); //颜色
begin
  ColorDialog1.Execute;
  Memo1.Color:=ColorDialog1.Color;
end;
```





8.1.2 文本组件

主要代码3

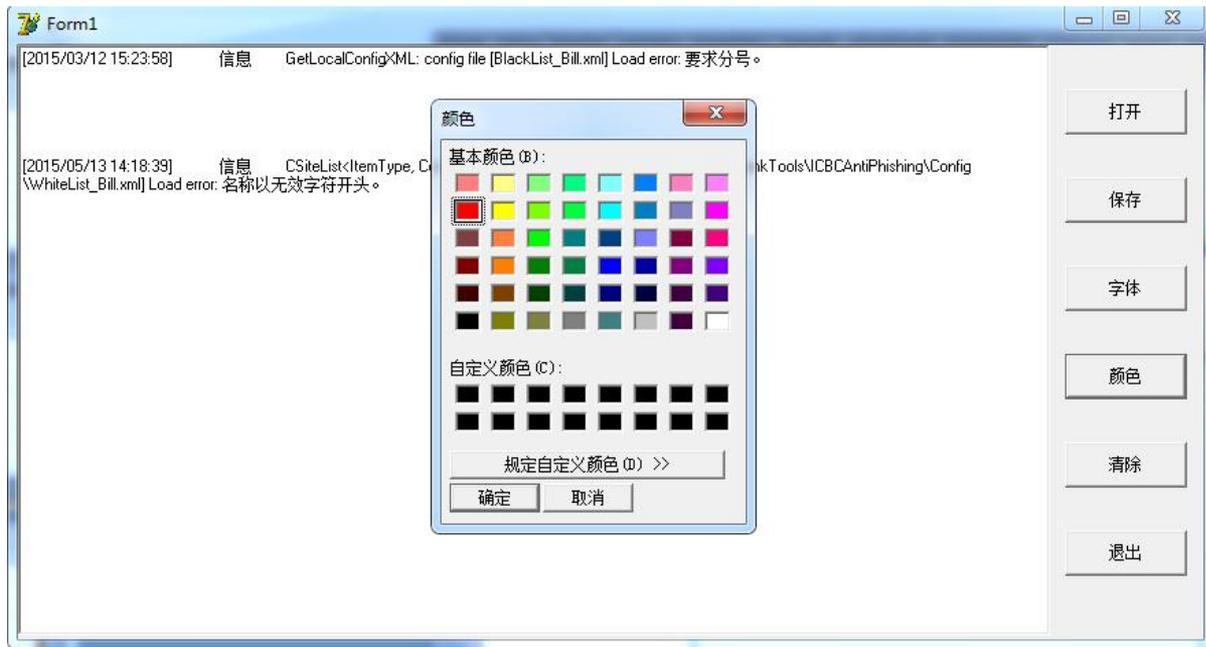
```
procedure TForm1.Button5Click(Sender: TObject); //清除  
begin  
    Memo1.Clear;  
end;  
procedure TForm1.Button6Click(Sender: TObject); //退出  
begin  
    application.Terminate;  
end;
```





8.1.2 文本组件

运行结果





8.1.3 特殊输入组件

□ 1. ScrollBar

- ① Kind属性可以设置滚动条是水平的还是垂直的。
- ② Max、Min属性用来确定滚动条表示数值的范围，数据类型为32位有符号整数类型（Integer）。
- ③ LargeChange、SmallChange可以设置滚动条数值改变的大小。
- ④ Position属性表示了滚动条当前的位置。





8.1.3 特殊输入组件

□ 2. TrackBar

- ① **Frequency**属性表示每隔几个数值显示一个刻度。
- ② **LineSize**属性与**PageSize**属性对应，表示比较小的变化时位置的改变。
- ③ **PageSize**属性表示比较大的变化时位置的改变。
- ④ **SliderVisible**属性用来设置是否显示滑动按钮。
- ⑤ **ThumbLength**属性表示滑动按钮的宽度(或高度)。
- ⑥ **TickMask**属性用来指定标尺显示的位置。





8.1.3 特殊输入组件

□ 3. UpDown

- ① **Associate**属性指定与**UpDown**组件关联的组件，可以通过下拉的组合框进行选择。
- ② **Increment**属性指定通过**UpDown**组件而每次增加或减少的步长。
- ③ **Orientation**属性确定组件的方向。
- ④ **Thousand**属性可以设置当数值超过**999**时，是否在千位后显示一个逗号。
- ⑤ **Wrap**属性可以设置相关联的组件中的数值是否循环变化。如在最大值的时候，再按上箭头，则相关联的组件中数值变为最小值。





8.1.3 特殊输入组件

□ 4. HotKey

HotKey组件可以在设计阶段指定热键，通过设置HotKey属性设定热键，也可以在运行期间改变。





8.1.4 按钮及其分类组件

□ 1. Button

- ① 如果某个按钮的**Default**属性设置为**True**，则在该视图中按下回车键〈**Enter**〉后，将执行该按钮的**OnClick**事件的处理过程。
- ② 通常在一个窗口中只设置一个默认按钮。





8.1.4 按钮及其分类组件

□ 2. BitBtn

- ① 可以自己制做位图按钮上的图形，也可以采用Delphi指定的默认图形。
- ② NumGlyphs属性指明该位图按钮使用图形的个数，可以设置为1~4之间的一个整数。
- ③ 在位图按钮具有多个图形的时候，位图按钮通常显示的是第1个图形；当按钮被按下后，显示第3个图形；当按钮失效后，显示第2个图形。

□ 3. SpeedButton

在加速按钮上可以同时显示图形和文字，用法与位图按钮很相似。





8.1.4 按钮及其分类组件

□ 4. CheckBox

- ① 可以通过**Alignment**属性设置复选按钮上文字的位置。
- ② 可以在设计阶段就设置好复选按钮的状态。通过**Checked**属性可以将复选按钮设置为“选中”和“未选中”状态，通过**State**属性可以将复选按钮设置为“选中”、“未选中”或“部分选中”。





8.1.4 按钮及其分类组件

□ 5. RadioButton

通常在使用单选按钮时，总是将其进行分组。在同一组中，只能同时选中一个按钮，其余按钮自动取消选中。在实现单选按钮的分组时，可以有两种方法：

- ① 利用分组框组件（**GroupBox**）和单选按钮组件（**RadioButton**）实现。
- ② 利用单选按钮分组框（**RadioGroup**）实现。可以通过**ItemIndex**属性来确定选中哪一个单选按钮。





8.1.5 列表组件

□ 1. ListBox

- ① 可以在设计阶段改变Items属性来预先设置列表框的内容，同时也可以可以在运行期间通过代码改变列表框的内容。
- ② 如果将ListBox组件的MultiSelect属性设置为True，就可一次选多个条目。
- ③ 如果将ListBox组件Sorted属性设置为True，可以自动将列表框中字符串排序。





8.1.5 列表组件

□ 2. ComboBox

- ① 组合框有五种类型，可以通过**Style**属性来设置。
- ② 通过属性**DropDownCount**可以改变列表框显示条目的个数。当列表框中的条目大于**DropDownCount**时，将会在列表框右面自动出现一个滚动条。





8.1.5 列表组件

□ 3. TreeView

- ① 树的生成可以在设计阶段进行，也就是改变**Items**属性。
- ② 在程序运行期间，可以动态地改变树的内容。可以通过**TreeView**组件**Items**属性的**Add**、**AddChild**、**Delete**函数等实现。
- ③ 一般在树上的每个条目前会有一个小图标。有时当该条目被选中时，会显示另一个小图标。这些小图标由**ImageList**组件进行统一管理。
- ④ 如果将**RightClick**属性设置为**True**，则在树的某个条目上单击鼠标右键，也可以将该条目选中。





8.1.5 列表组件

□ 4. ListView

- ① 在ListView中有一些图标，添加图标的方法与TreeView组件一样。
- ② 通过对ListView组件的Columns属性进行设置，可以确定将要显示的列数，并且可以为每一列指定文字的对齐方式、显示的宽度、图标索引号等。





8.1.5 列表组件

□ 5. ImageList

ImageList组件主要用来对窗体中的图标进行管理，统一分配索引号，然后提供给TreeView组件、ListView组件等。

通常应将Masked属性设置为True，这样，图标中的背景框就不显示出来。

□ 6. CheckListBox

可以通过CheckListBox组件的Items属性，在设计阶段添加、修改列表内容。





8.1.5 列表组件

□ 7. DateTimePicker

DateTimePicker组件可以实现日期或者事件的快速输入。

① 如果Kind属性设置为dtkDate，则输入的是日期；如果Kind属性设置为dtkTime，则输入的是时间。

② 在输入的是日期的情况下，修改DateFormat属性可以设置日期的格式，有两种格式：dtLong（长日期型）和dtShort（短日期型）。





8.1.6 分组组件

在GroupBox、Panel、ScrollBar、TabControl和PageControl等组件中可以放置其他组件，所以是容器组件。在使用容器组件的时候应注意：

- ① 应先添加容器组件，然后选中该容器组件，再向其中添加其他组件。
- ② 如果将容器组件移动、删除，则其中的组件一起被移动、删除。
- ③ 容器组件中的其他组件只能在该容器组件中显示，超过容器组件边框的部分将不显示出来。
- ④ 在窗体上设置Tab键顺序的时，容器组件和它包含的组件是作为一个组件来参加排序。容器组件中的组件可独立于容器外的组件进行Tab键顺序排列。





8.1.6 分组组件

□ 1. GroupBox

如果要将几个组件放在一个GroupBox组件中，必须先在窗体中放置这个GroupBox组件，然后选中该GroupBox组件，再添加其他组件。

□ 2. RadioGroup

通过RadioGroup组件可以快速生成一组单选按钮。通过Items属性，可以预先设定单选按钮。Columns属性可以设置分组框中的列数，各个按钮之间的距离会自动设置成等间距的。





8.1.6 分组组件

□ 3. Panel

在窗体中显示图形的时候，一般先添加一个**Panel**组件，然后在上面再添加**Image**组件。这样在图形显示的时候，可以以**Panel**组件的边框做边界。

□ 4. ScrollBox

如果**ScrollBox**组件中的其他组件放置的位置超出了**ScrollBox**组件的边界，则会自动出现水平方向的或垂直方向的滚动条，通过滚动条，可以看到其他组件超出边界的地方。





8.1.6 分组组件

□ 5. TabControl

- ① 通过修改**Tabs**属性，可以手工地添加标签的个数。
- ② 在**Images**属性中与**ImageList**组件关联，可以为每个标签设置一个图标。
- ③ 改变**Styles**属性，可以将标签的显示方式设置为标签方式、按钮方式、平板按钮方式。





8.1.6 分组组件

□ 6. PageControl

- ① 在PageControl组件上单击鼠标右键，在弹出式菜单中选中“New Page”命令，将会增加一个新的页面。
- ② 对PageControl组件，利用鼠标左键单击标签与单击页面位置选中的对象是不同的。
- ③ 在PageControl组件的不同页面中，可以加入不同的组件。





8.1.6 分组组件

□ 6. PageControl

④ 在选中整个组件的情况下，改变ActivePage属性，可以改变当前的激活页。另外还可以改变Images属性，使之与某个ImageList组件相关联，为每个页指定一个图标。

⑤ 在选中具体页面的情况下，可以改变PageIndex属性，也就是标签排列的顺序会发生改变。另外可以改变ImageIndex属性，指定相关ImageList组件中的图标索引值。





8.1.6 分组组件

□ 7. HeaderControl

① 改变Align属性，可以指定HeaderControl组件放置的位置。默认值为alTop，组件停靠在窗体的上部，宽度充满整个窗口。将Align属性设置为alNone，则只能改变HeaderControl组件的位置与大小了。

② 选中HeaderControl组件并单击鼠标右键，在弹出式菜单中选择“Sections Editor...”命令，可以对HeaderControl组件的列进行编辑，指定每列的文字、宽度等。





8.1.7 信息反馈组件

□ 1. Label

通常利用标签组件在窗体上显示静态文本。此外，将标签组件与其他没有标题的组件相关联后，可以通过键盘上的〈Tab〉键使这些组件获得输入焦点。通过设置Caption属性，可以改变Label组件显示的内容。

□ 2. StatusBar

状态条通常放置在窗体的底部，在上面动态地显示一些信息。





8.1.7 信息反馈组件

□ 3. ProgressBar

ProgressBar具有Max和Min属性，用来表示进度条所表示的最大值和最小值。通常，Max设置为100，Min设置为0，

Position属性的初始值设置为0。经过以上设置后，就可将整个工作分成了100份，每完成1份，就将Position属性的数值加1。

□ 4. StaticText

StaticText组件的用法类似于Label组件。StaticText组件上可以显示只读文本，该组件具有一个Widnows句柄。





8.1.8 图形显示组件

□ 1. Image

① 通过**Picture**属性，可以在设计阶段预先装入一个图形文件，可以为**bmp**、**jpg**、**jpeg**、**wmf**、**emf**和**ico**等。

② 在装入大型压缩图片的时候，可以将**IncrementalDisplay**属性设置为**True**。

□ 2. Bevel

通过改变**Bevel**组件的**Shape**属性和**Style**属性，可以设置**Bevel**组件的形状和风格。





8.1.8 图形显示组件

□ 3. Shape

- ① 展开Shape组件的Brush属性，可以对画刷的颜色、类型进行设置。
- ② 展开Shape组件的Pen属性，可以对画笔的颜色、模式、类型和宽度进行设置。
- ③ 可以改变Shape组件的Shape属性，生成不同形状的图形，主要有矩形、正方形、圆形、椭圆形、圆角矩形等。





8.1.8 图形显示组件

□ 4. PictureBox

在PictureBox中做图类似于Image组件，但不可从图形文件中装入图形。

□ 5. Animate

Animate组件可以播放一些动画.avi文件，但这些动画文件必须是无声的。将Active属性设置为True，则可以播放动画。

□ 6. Splitter

分隔条控件可以将一个窗体分成两部分，并可以动态地改变这两部分的大小。





8.2 常用组件的用法

实例8-2

The screenshot shows a Windows form titled "Form1" with the following components:

- 姓名(N):** Text box containing "张兵一".
- 学历(E):** Dropdown menu set to "大学".
- 性别(S):** Radio buttons for "男" (selected) and "女".
- 职称(T):** Radio buttons for "讲师" (selected), "副教授", and "教授".
- 备注(R):** Checkboxes for "中国科学院院士" (checked), "中国工程院院士" (unchecked), and "863首席科学家" (checked).
- 简历(M):** A text area displaying the generated resume: "姓名:张兵一", "学历:大学", "性别:男", "职称:讲师", "该同志为中国科学院院士", "该同志为863首席科学家".
- Buttons:** "刷新简历(R)" and "Close".

题目：本例实现一个教师简历的信息输入。一个教师的属性有姓名、学历、性别、职称等，还有一些备注。当上述这些属性有改动后，“刷新简历(R)”按钮就被激活，按下后，就在简历一项中生成该教师的一个大致简历。按下“关闭”按钮后，如果教师属性有改动但简历没有刷新，则提示是否刷新，根据回答做出响应后退出程序。





8.2 常用组件的用法

实例8-2

在实现例程的过程中应注意：

- 一开始，“刷新简历”按钮为失效状态。当教师任一个属性改动后，该按钮激活。
- 所有组件的字体属性都与窗体的字体属性一致。
- 所有组件都可以通过热键激活，通过键盘进行操作。
- 不同事件使用同一个事件处理过程。
- 体现单选按钮的两种使用方法：一种是GroupBox组件与RadioButton组件的组合（性别属性）；另一种是RadioGroup的使用（职称属性）。
- 关闭按钮按下后，要检测简历是否更新。





8.2 常用组件的用法

实例8-2

控件设置1

组件名称	组件标题	其他属性
Label1	姓名(&N)	FocusControl:Edit1
Edit1		将Text属性清空
Label2	学历(&E)	FocusControl:ComboBox1
ComboBox1		Items: 中专<回车>大专<回车>大学; Style:csDropDownList
GroupBox1	性别(&S)	Checked:True
RadioButton1	男	
RadioButton2	女	
RadioGroup1	职称(&T)	Items:讲师<回车>副教授<回车>教授; ItemIndex:2
GroupBox2	备注(&B)	





8.2 常用组件的用法

实例8-2

控件设置2

组件名称	组件标题	其他属性
CheckBox 1	中国科学院院 士	
CheckBox 2	中国工程院院 士	
CheckBox 3	863首席科学 家	
Label3	简历(&M)	FocusControl:Memo1
Memo1		将Lines属性中的内容清空
Button1	刷新简历(&R)	Default:True
BitBtn1	关闭(&C)	Kind:bkCustom; ModalResult:mrNone; NumGlyphs:1; Glyph:装入图形





8.2 常用组件的用法

实例8-2

主要代码1

```
var
  Form1:TForm1;
  Dirty:Boolean;           // 标志属性内容是否修改
procedure TForm1.FormCreate(Sender:TObject);
begin
  Button1.Enabled:=False; // 将“刷新”按钮失效
  Dirty:=False; // 标志属性内容没有修改
end;
procedure TForm1.Button1Click(Sender:TObject);
begin
  Memo1.Clear; // 清空Memo1组件，便于重新生成简历内容
  if Edit1.Text<>"then// 如果已经输入姓名，则添加到简历中
    Memo1.Text:='姓名:'+Edit1.Text;
```





8.2 常用组件的用法

实例8-2

主要代码2

```
if ComboBox1.Text<>"then// 如果在组合框中已选择学历，则添加到简历中
    Memo1.Text:=Memo1.Text+#13#10'学历:'+ComboBox1.Text;
if RadioButton1.Checked=True then// 将性别添加到简历中
    Memo1.Text:=Memo1.Text+#13#10'性别:男'
else
    Memo1.Text:=Memo1.Text+#13#10'性别:女';
if RadioGroup1.ItemIndex=0 then// 将职称添加到简历中
    Memo1.Text:=Memo1.Text+#13#10'职称:讲师'
else if RadioGroup1.ItemIndex=1 then
    Memo1.Text:=Memo1.Text+#13#10'职称:副教授'
else
    Memo1.Text:=Memo1.Text+#13#10'职称:教授';
```





8.2 常用组件的用法

实例8-2

主要代码3

```
if CheckBox1.Checked=True then// 将备注添加到简历中
    Memo1.Text:=Memo1.Text+#13#10'该同志为中国工程院院士';
if CheckBox3.Checked=True then
    Memo1.Text:=Memo1.Text+#13#10'该同志为863首席科学家';
Dirty:=False; // 标志简历内容已经刷新
Button1.Enabled:=False; // 将“刷新”按钮失效
end;
procedure TForm1.BitBtn1Click(Sender:TObject);
begin
    if Dirty=True then
        begin
            if MessageDlg('简历内容已经修改，是否刷新？',
                mtInformation,[mbYes,mbNo],0)=mrYes then
```





8.2 常用组件的用法

实例8-2

主要代码4

```
begin
    Button1Click(BitBtn1); // 刷新简历内容
    ShowMessage('你选择的是刷新。');
end
else
    ShowMessage('你选择的是不刷新。');
end;
Close; // 关闭程序
end;
procedure TForm1.Edit1Change(Sender:TObject);
begin
    Button1.Enabled:=True; // 让"刷新"按钮有效
    Dirty:=True; // 标志属性内容已经修改
end;
```





8.2 常用组件的用法

实例8-3

本例主要实现了动态图像的装入，并且可以向图像上添加说明文字的功能。

在实现例程的过程中应注意：

① 一开始，“应用”按钮为失效状态。当翻页步长或显示文字发生改变后，该按钮被激活。

② 事先应寻找一个位图文件，最好在Image组件中添加的滚动条组件才能派上用场。





8.2 常用组件的用法

实例8-3

控件设置

组件名称	组件标题	其他属性
Panel		
Image1	Picture: None	
ScrollBar1		Kind:sbHorizontal; LargeChange: 5; SmallChange: 1
GroupBox1	设置(&S)	
Label1	翻页步长(&T)	FocusControl: Edit1
Edit1		ReadOnly: True; Text:5
UpDown1		Associate: Edit1; Increment: 1; Max: 20; Min:0; Position: 5
Label2	显示文字(&E)	FocusControl: Edit2
Edit2		清空Text
Button1	应用(&A)	Default: True
Label3		Alignment: taRightJustify; Transparent:True





8.2 常用组件的用法

实例8-3

主要代码1

```
procedure TForm1.FormCreate(Sender:TObject);
begin
    // 动态装入图片
    Image1.Picture.LoadFromFile('MyPicture.bmp');
    // 设置滚动条的最大值
    ScrollBar1.Max:=Image1.Picture.Width-Image1.Width;
    // 将Image1组件的宽度等同于图片的宽度
    Image1.Width:=Image1.Picture.Width;
    Button1.Enabled:=False; // 将“应用”按钮失效
end;
procedure TForm1.ScrollBar1Change(Sender:TObject);
begin
```





8.2 常用组件的用法

实例8-3

主要代码2

```
// 改变图片的左边界位置，让图片滚动显示
Image1.Left:=-ScrollBar1.Position;
end;
procedure TForm1.Edit1Change(Sender:TObject);
begin
  Button1.Enabled:=True; // 让“应用”按钮有效
end;
procedure TForm1.Button1Click(Sender:TObject);
begin
  Label3.Caption:=Edit2.Text; // 更新图片下方显示的文字
```





8.2 常用组件的用法

实例8-3

主要代码3

```
// 更新滚动条翻页时的步长
ScrollBar1.LargeChange:=StrToInt(Edit1.Text);
Button1.Enabled:=False; // 将“应用”按钮失效
end;
procedure TForm1.FormShow(Sender:TObject);
begin
// 在最初视图显示时将"应用"按钮失效
Button1.Enabled:=False;
end;
```

