第 1 章　Delphi 的基础知识

Delphi 是由 Inprise 公司（前 Borland 公司）推出的可视化编程环境，它提供了一种方便、快捷的 Windows 应用程序开发工具。Delphi 使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了可重复利用的完整的面向对象程序语言（Object-Oriented Language）、当今世界上最快的编辑器、最为领先的数据库技术。程序开发人员使用 Delphi 开发应用软件，无疑会大大提高编程效率。

1.1　Delphi 简介

Delphi 到 2005 年已经历了 9 代产品的发展历程，每一代产品都是伴随 Windows 操作平台的升级而升级。

本书采用 Delphi 6 作为平台，讲述 Delphi 程序开发的基本知识。

1.2　Delphi 可视化编程的基本概念

一些早期的具有 OOP 性能的程序语言如 C++，Pascal，Smalltalk 等，虽然具有面向对象的特征，但不能轻松地画出可视化对象，与用户交互能力较差，仍然要编写大量的代码。

Delphi 使用“可视化”的编程方法。程序员不必自己建立对象，利用 Delphi 所提供的可视“控件”，只要在提供的程序框架中加入完成功能的代码，如选择命令、移动鼠标等，不必考虑按精确次序执行的每个步骤。在这种机制下，不必编写一个大型的程序，只要建立一个由若干微小程序组成的应用程序，这些微小程序可以由用户启动的事件来激发。这样就可以快速创建强大的应用程序而无需涉及不必要的细节。

简单地说，“可视化编程”就是使用 Delphi 的 Object Pascal 语言，利用它所提供的可视“控件”来创建“对象”。这是一种编程方法的新概念。

1.2.1　对象的属性、事件和方法

对象（Object）在现实生活中是很常见的，如：一个人是一个对象，一部汽车是一个对象。如果将一部汽车拆开来看便有“发动机、方向盘、转向轴、车轮…”，每一个又都是一个对象，即汽车对象是由多个“子”对象组成的。在可视化编程中，常见的对象有：窗体、编辑框、列表框等。

从可视化编程的角度来看，对象是一个具有属性（数据）和行为方式（方法）的实体。简单地说，属性用于描述对象，方法让对象做一些动作，而对象动作时常会引起事件。一个对象建立以后，其操作就通过与该对象有关的属性、事件和方法来描述。

在可视化编程中，Delphi 的窗体与控件都是程序被操作的对象，这些对象都有其自己的属性和方法。

1．对象的属性

属性（Property）是对象的一项描述内容，用以描述对象的一个特性，不同的对象有不同的属性，而每个对象都由若干属性来描述。在可视化编程中，常见的属性有标题（Caption）、名称（Name）、字体（Font）、是否可见（Visible）等。通过修改或设置某些属性便能有效地控制对象的外观和操作。

属性值的设置或修改可以通过对象观察器（Object Inspector）中的属性窗口来进行，也可以通过编程的方法在程序运行的时候来改变对象的属性。

在程序中设置属性的一般格式是：

〈对象名〉.〈属性名〉:=〈属性值〉;

例如，设置标签对象 Label1 的标题为“欢迎使用 Delphi”的命令是：

```
Label1.Caption := '欢迎使用 Delphi';
```

2．对象的事件

所谓事件（Event），是由 Delphi 预先定义好的、能够被对象识别的动作，如单击（OnClick）事件、双击（OnDblClick）事件、移动鼠标（OnMouseMove）事件等，不同的对象能识别的事件也不相同。

对象的事件是固定的，用户不能建立新的事件。为此，Delphi 提供了丰富的内部事件，这些事件足以应付 Windows 中的绝大部分操作需要。

事件过程（Event Procudure）是程序员为处理特定事件而编写的一段程序。当事件由用户触发（如 OnClick）或由系统触发（如 OnActivate）时，对象就会对该事件作出响应（Respond）。响应某个事件后所执行的程序代码就是事件过程。一个对象可以识别一个或多个事件，因此可以使用一个或多个事件过程对用户或系统的事件作出响应。虽然一个对象可以拥有多个事件过程，但在程序中要使用多少事件过程，则由程序员根据程序的具体要求来确定。对于必须响应的事件需编写该事件的事件过程，而不必理会的事件则不需要编写事件过程，只要交给 Delphi 的默认处理程序即可，例如命令按钮的 OnClick 事件是最重要的事件，而 OnMouseUp 事件则可有可无，全视设计人员的需要。

3．对象的方法

方法（Method）是与对象相关联的过程与函数的统称，即在对象中说明的并且用户可以调用的公共函数和过程，由 Delphi 系统内部定义，不用编写代码。方法用于完成某种特定的功能而不能响应某一事件，如对象移动（Move）、画线（Line）、显示（Show）等。每个方法可以完成某个功能，但其实现步骤和细节用户既看不到也不能修改，用户能做的工作就是按照约定直接使用（调用）它们。

方法也被“封装”在对象之中，不同的对象具有不同的内部方法。Delphi 提供了大量的方法供不同的对象调用。选中对象，按〈F1〉键激活帮助，选择 Methods，就能看到可以调用的方法列表。

1.2.2 控件

控件是建立程序界面的基本元素，是可视化编程的基础。控件的使用，充分体现了当今

流行的面向对象的程序设计思想。

1．类与控件

Delphi 的控件由可视化组件库——Visual Component Library（VCL）提供。这个组件库是一个类库，其中定义了众多的类。VCL 中的有些类被可视化地安排在组件板（参见 1.3.4）上，称之为组件（components）；有些则是不可见的，如 Tstream 类。VCL 中所有的类都是从 TObject 类继承而来，TObject 类称为基类。组件是从 TComponent 类继承而来，从 TComponent 类中又派生出可视化组件与非可视化组件等子类。可视化组件或称控件是从 TControl 类派生的，如 TButton、TEdit 等；非可视化组件是控件以外的所有组件，即从 TComponent 类中派生但不是从 TControl 类派生的组件。在设计时，非可视化组件以图标的形式出现在窗体上，如 TOpenDialog（运行时可见）、TDataSource（运行时不可见）。

VCL 中主要的类及它们之间的层次关系如图 1-1 所示。

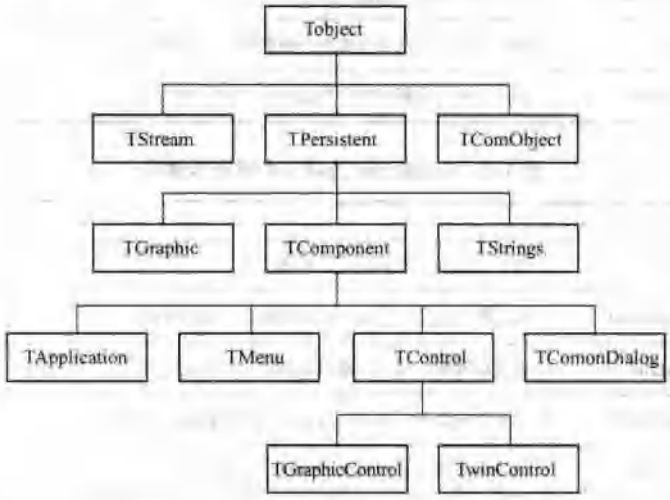

图 1-1　VCL 中主要的类之间的层次关系

为简单起见，本书将组件板中的组件（可视化组件与非可视化组件）都称为控件。

2．常用控件介绍

Delphi 6 的组件板含有 27 个选项卡，总共包括 350 多个控件。表 1-1～表 1-3 给出前 3 个选项卡中最常用控件的简要说明。

表 1-1　Standard 选项卡中常用控件简介

组件图标	组件名称	说　明
	MainMenu	用于设计和创建主菜单以及下拉式菜单，非可视控件
	PopupMenu	用于设计和创建弹出式菜单，可以在窗体或控件上使用，非可视控件
A	Label	作为标签，用来显示不可变文本，可视控件
ab|	Edit	用于设计供用户输入单行文本的编辑框，可视控件
	Memo	用于设计供用户输入多行文本的备注框，可视控件

（续）

组件图标	组件名称	说　明
	Button	按钮控件，可视控件
	CheckBox	复选框控件，为用户提供选项，可视控件
	RadioButton	单选按钮，为用户提供多选一选择，可视控件
	ListBox	列表框，可视控件
	ComboBox	组合框，提供输入功能的列表框，可视控件
	ScrollBar	用于设计滚动条，可视控件
	GroupBox	组框，用于控件的分组，可视控件
	RadioGroup	用于设计单选按钮组，可视控件
	Panel	用于创建可以包含其他控件的面板，可视控件
	ActionList	用于收集应用程序对用户的回应，非可视控件

表 1-2　Additional 选项卡中常用控件简介

组件图标	组件名称	说　明
	BitBtn	用于创建带位图的按钮，可视控件
	SpeedButton	与面板一起用于创建工具栏的按钮，可视控件
	MaskEdit	格式编辑框，用于控制数据的输入，可视控件
	StringGrid	用于简化显示字符串和与对象有关的数据，可视控件
	DrawGrid	用于以表格形式控制显示信息，可视控件
	Image	用于显示图标及图形对象，可视控件
	Shape	用于显示正方形、长方形、圆、椭圆等几何图形，可视控件
	Bevel	显示立体矩形，可视控件
	ScrollBox	可以滚动的显示框，可视控件
	CheckListBox	具有复选框的列表框，可视控件
	Splitter	用于在程序中创建用户可以改变大小的区域，可视控件
	StaticText	静态文本框，与 Label 相似，可视控件
	ControlBar	创建可以移动的工具栏，可视控件
	LabeledEdit	创建一个具有标签的编辑框，可视控件
	ColorBox	创建一个彩色枚举常量的下拉列表框，可视控件
	Chart	创建图标与图形，可视控件

表 1-3 Win32 选项卡中常用控件简介

组件图标	组件名称	说明
	TabControl	供用户选择的多标签控件，可视控件
	PageControl	选项卡控件，设计时通过单击鼠标右键，在弹出菜单中选择“New Page”创建新选项卡页面，可视控件
	ImageList	为其他控件添加图像，非可视控件
	RichEdit	设计可使用多种颜色、字体以及具有文本查找替换功能的编辑框，可视控件
	TrackBar	滑动控件，可视控件
	ProgressBar	进度栏控件，可视控件
	UpDown	数字控制滚动条，可视控件
	Hotkey	为应用程序添加热键，可视控件
	Animate	播放一系列 AVI 电影位图，可视控件
	DateTimePicker	下拉式日历控件，可视控件
	MonthCalendar	月历控件，可视控件
	TreeView	提供树形查看方法，可视控件
	ListView	以列的形式、大图标、小图标的形式查看数据，可视控件
	HeaderControl	创建多个可移动大小的标头，可视控件
	StatusBar	设计状态栏，可视控件
	ToolBar	设计工具栏，可视控件
	CoolBar	设计可改变大小，可移动、可窗口化的工具栏，可视控件
	PageScroller	控制其他多控件的滚动，可视控件
	ComboBoxEx	创建一个图形组合框，可视控件

1.3 Delphi 可视化编程的环境

Delphi 的 IDE（Integrated Development Environment，集成开发环境）是进行设计、运行和测试等可视化编程的理想环境。

1.3.1 进入 Delphi 环境

启动 Windows，并从“开始”菜单选择“Borland Delphi 6”→“Delphi 6”，以启动 Delphi。首次加载 Delphi 6，屏幕上会出现如图 1-2 所示的 5 个窗口。

（1）标题为“Delphi 6-Project2”的 Delphi 主窗口。Delphi 的主窗口位于屏幕的上端，包括主菜单、工具栏和组件板。

（2）对象 Tree view（Object Tree view）。

图 1-2　Delphi 6 的集成开发环境

（3）对象观察器（Object Inspector）。

（4）标题为“Form1”的窗体设计器。

（5）标题为“Unit1.pas”的代码编辑器，刚启动时这一窗口的大部分被窗体设计器所掩盖。按〈F12〉键，可以在窗体设计器与代码窗口之间进行切换。

1.3.2　Delphi 的主菜单

Delphi 6 的主菜单包括 File、Edit、Search、View、Project、Run、Component、Database、Tools、Windows 和 Help 等 11 个下拉菜单，其中包含了 Delphi 6 编程的所有命令与功能。单击菜单栏中的菜单名，即可打开下拉菜单。在下拉菜单中显示了各种功能选项，包含执行该项功能的热键和快捷键。表 1-4 给出主菜单的功能简介。

表 1-4　主菜单简介

菜 单 名	功　能
File	提供工程和各个单元的文件操作命令
Edit	提供编辑代码和控件的命令，如复制、粘贴、删除等
Search	提供查找、替换代码等功能
View	用于打开集成环境中的各种工具窗口
Project	用于工程的管理及设置等
Run	提供运行、调试程序的各种命令，如设置断点，单步执行等
Component	用于建立、安装和设置控件（组件）
Database	提供开发数据库的各种工具
Tools	用于开发环境的设置并提供辅助开发的工具
Windows	用于在打开的各个窗口之间转移焦点
Help	提供全面的帮助信息

1.3.3 Delphi 的工具栏

Delphi 6 在默认的 IDE 中配置了 5 个工具条：Standard、View、Debug、Help 和 Desktop。工具条中的按钮是菜单功能的快捷方式，各种图标直观地表示了它能执行的动作。

表 1-5 给出了 5 个工具条中工具按钮的名称与功能简介。

表 1-5 工具按钮简介

工具条名称	按钮图标	按钮名称	功能
Standard		New	执行 File→New 菜单命令
		Open	左侧为 File→Open 菜单命令，右侧列出最近完成的工程和单元
		Save	执行 File→Save 菜单命令
		Save All	执行 Save All 菜单命令
		Open Project	执行 File→Open project 菜单命令
		Add File to project	执行 Project→Add to Project 菜单命令
		Remove file from project	执行 Project→Remove From Project 菜单命令
View		View Unit	执行 View→Units 菜单命令
		View Form	执行 View→Forms 菜单命令
		Toggle Form/Unit	执行 View→Toggle Form/Unit 菜单命令
		New Form	执行 File→New Form 菜单命令
Debug		Run	左侧为 Run→Run 菜单命令，右侧列出最近完成的工程和单元
		Pause	执行 Run→Program Pause 菜单命令
		Trace into	执行 Run→Trace into 菜单命令
		Step over	执行 Run→Step over 菜单命令
Help		Help Contents	执行 Help→Delphi Help 菜单命令
Desktop		Desktop Speedsetting	用来选择一个保存过的桌面
		Save current Desktop	激活 Save Desktop 对话框，以保存当前桌面
		Set debug Desktop	设置当前桌面为调试桌面，调试时将自动显示

1.3.4 Delphi 的组件板

包含 350 多个控件的组件板是 Delphi 6 可视化编程的核心部件。它由 27 个选项卡组成，每张选项卡中包含若干图形按钮，这些图形按钮都代表相应的控件（组件）。编程时可以方便地选择需要的控件并将它放到窗体中去。

1．组件板的组成

Delphi 6 的组件板如图 1-3 所示。组件板是一个选项卡风格的工具栏，各种控件按功能组织在不同的选项卡中。选项卡标签反映其功能的分类，如 Standard 卡中包含的是标准的常用控件、Dialogs 卡中包含了常用的对话框控件等。

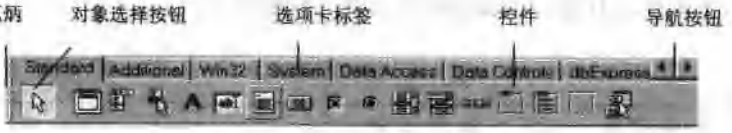

图 1-3　Delphi 6 的组件板

组件板左端的抓柄用来拖动组件板，可根据用户需要来定制组件板在 IDE 中的位置。

单击导航按钮可以左右滚动组件板中的选项卡标签，以便找到所需要的选项卡。

在每张选项卡中都有一个“对象选择按钮”，当在组件板上选择了控件后，对象选择按钮将弹起，此时单击它将取消所选择的控件。当连续添加同一个控件时，该按钮特别有用，单击它可以取消连续添加控件的操作。

2．组件板中控件的使用

将组件板上控件添加到窗体中去的方法有如下 3 种：

- 单击组件板上的所需控件的按钮，然后在窗体适当位置拖动鼠标画出控件，即可将控件添加到窗体的指定位置。
- 双击组件板上的所需控件的按钮，即可将控件添加到窗体的中心位置。
- 按下〈Shift〉键不放，单击组件板上的所需控件的按钮，所选控件出现蓝色边框，同时对象选择按钮将弹起。可以在窗体适当位置拖动鼠标连续画出该类控件。系统将按照添加的顺序为每个控件确定默认名称。用鼠标单击对象选择按钮，即可取消该类控件的连续添加操作。

组件板中的控件分为可视控件与非可视控件两种。在设计时可以通过设置可视控件的属性来改变其外观，如 Tbutton、Tlabel、Tedit 等都是可视控件。非可视控件则在设计时见不到它的外观，只在窗体上用一个图标表示添加了这种控件，如 TsaveDialog、TopenDialog、Ttimer 等都属于非可视控件。虽然在设计时不能见到非可视控件的外观，但是仍然可以通过对象观察器来设置其属性。

1.3.5　对象观察器

对象观察器（Object Inspector）是进行可视化编程时使用最为频繁的工具之一。设置窗体和控件的属性、切换设计对象以及为窗体、控件选择或添加事件处理过程等操作都在其中进行。

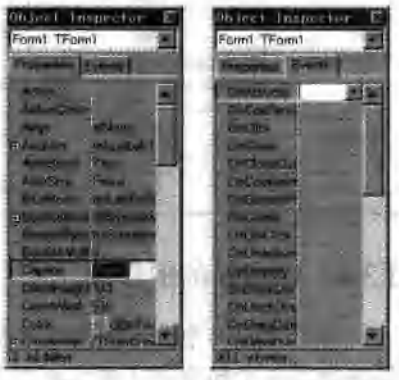

图 1-4　Delphi 6 的对象观察器

1．对象观察器的组成

对象观察器由“对象”列表框、“Properties”（属性）选项卡和“Events”（事件）选项卡组成，如图 1-4 所示。

（1）“对象”列表框：位于对象观察器上部的下拉

列表框，其中显示了窗体上所有对象的名称和类型，包含窗体本身。

可以通过对象列表框在窗体中的各个控件之间切换，也可以快速地回到窗体本身。当窗体中含有较多的对象时，这是切换对象尤其是回到窗体的最快捷途径。

对象列表框中当前对象的变化将影响属性选项卡与事件选项卡中所显示的属性和事件列表。

（2）“Properties”（属性）选项卡：显示窗体中当前被选择对象的属性信息，并允许改变对象的属性。其中左边一栏是属性名，右边一栏是属性值。在设计时，对窗体及其中控件的属性设置主要在此进行。

首次启动时，对象观察器窗口显示的是当前窗体 Form1 的属性。对象观察器根据对象属性的多少，决定是否有滚动显示。移动滚动条，可以查看当前对象的全部属性。

（3）“Events”（事件）选项卡：列出当前对象可以响应的事件信息。其中，左边一栏是事件名，右边一栏是响应事件的事件过程名，如果事件过程名是空白，说明还未定义相应的事件过程。

2．属性选项卡的使用

在建立新对象（窗体或控件）之初，属性选项卡的属性值一栏给出属性的默认值。根据属性类型的不同，采用不同的方法为属性赋值。

（1）属性为数值或字符串类型：只需在属性值一栏中直接输入数值或字符串。输入字符串时不必带引号。

（2）属性为布尔类型：该类属性的取值只有两种：True 与 False。选择该属性，右边属性值一栏出现下拉列表框。单击下箭头，在列表中选择一项即可。

用鼠标双击属性值栏，可以在两个值之间切换。

（3）属性为枚举类型：该类属性的取值有若干种，Delphi 为该类属性提供了一个下拉列表框，设置属性时，只需在下拉列表框中选择一个列表项。

布尔类型可以看作是枚举类型的特例，它们的设置方法是类似的。

（4）属性为集合类型：该类属性的取值为集合。集合类型的取值用方括号[ ]表示，方括号中列出集合中的元素，每个元素之间用逗号隔开。

属性名前面一般都有一个“+”号，单击“+”号（“+”号变“-”号），属性名下列出所有可能的元素名，通过属性值栏可以逐个选择集合中的元素，来构造一个属性值（集合）。

（5）属性为对象类型：该类属性的取值为对象。一个对象一般都有自己的属性、事件和方法。因此属性名前面也有一个“+”号，右边一栏提示为对象类型（如 TForm），并有一个“...”按钮。

单击“+”号（“+”号变“-”号），属性名下列出所有子属性名，按照前面介绍的方法分别对这些子属性设置属性值即可。

为了方便设置对象类型的属性，Delphi 为许多控件的该类属性提供了对话框，如 Font 属性、Items 属性、Picture 属性、Glyph 属性等。单击属性值栏的“...”按钮，即可打开该属性设置对话框。

3．事件选项卡的使用

事件选项卡的事件栏中列出了当前对象（窗体或控件）的所有预置的事件，如 OnActivate、OnClick、OnKeyPress 等。在进行编程时，要从这些预置的事件中选择一个或多

个事件，添加到窗体对应的单元中，并给这些事件的处理过程添加处理代码。下面是为按钮对象 Button1 编写 OnClick 事件过程的步骤：

（1）在窗体中选中按钮 Button1，或在对象列表框中选择按钮 Button1。

（2）在事件选项卡中选择 OnClick 事件，用鼠标双击右边一栏，为 Button1 添加 OnClick 事件过程。

Delphi 自动打开代码编辑器窗口，在窗体 Form1 的单元文件 Unit1 的 Interface（接口）部分插入该事件过程的声明：

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

同时在 Implementation（实现）部分插入该事件过程的代码框架：

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

end;
```

并将光标停留在过程体的首行处（begin 与 end 之间），等待键入过程代码。

（3）键入过程代码，如：

```
Showmessage('欢迎使用 Delphi!');
```

（4）为按钮控件 Button1 编写 OnClick 事件过程的工作完成。

按〈F9〉键运行程序，在出现的程序界面中单击 Button1 按钮，将出现图 1-5 所示的信息框。

图 1-5　事件过程执行结果

1.3.6　对象 TreeView

对象 TreeView 位于对象观察器的上部。对象 TreeView 以树状表的形式显示窗体中可视化或非可视化控件之间的逻辑关系。

对象 TreeView 与对象观察器、窗体设计器同步，即在此三个工具的任何一个中改变焦点时，另外两个工具中的焦点也会随之改变。

如果对象 TreeView 被关闭，则按〈Alt+Shift+F11〉组合键或在“View”菜单中选择“Object TreeView”选项，可以打开对象 TreeView，如图 1-6 所示。

图 1-6　对象 TreeView

1.3.7　窗体设计器

在 Delphi 中，窗体设计器是开展大部分设计的工作区域，设计用户界面直接在窗体设计器中进行，运行结果和设计样板完全一致。当部件被放到窗体上时，Delphi 会自动生成大部分的用户界面代码，所需做的只是在它生成的框架中加入完成所需功能的程序段而已。

首次启动 Delphi 6 时显示的是窗体 Form1。有两种方法可以调整窗体的大小：

（1）将鼠标指向窗体的边界处，光标变成双向的箭头，按下鼠标左键，并拖动鼠标即可改变窗体的宽度或高度。

（2）在对象观察器上的属性选项卡中修改窗体的高度（Height）属性和宽度（Width）属性，也可改变窗体的大小。

可以把各种控件放在窗体中，通过移动位置、改变尺寸等操作随心所欲地安排它们，以此来开发应用程序的用户界面（见图 1-7）。可以把窗体想像成一个可以放置其他控件的容器。窗体上有栅格（Grids），供放置控件时对齐位置用，在程序运行时 Grids 是不可见的。

一个真正的应用程序可能有不止一个窗口，可以选用不同的窗体进行设计。其他窗体可以是对话框（Dialog Box）、数据录入框等。

图 1-7　Delphi 6 的窗体设计器

1.3.8　代码编辑器

代码编辑器是程序代码的输入和编辑工具，尽管可视化编程技术的运用大大减轻了程序员编写代码的工作量，但并不能完全取代原始的代码编写工作。代码的编写仍然是整个程序设计的核心，一个程序的好坏将部分地取决于代码的编写。

代码编辑器是 Delphi 提供的一个功能强大、使用方便的代码编写工具，它能提示和帮助程序员完成代码的编写。

1．代码编辑器的组成

代码编辑器是一个选项卡风格的文本编辑器。通过选项卡标签可以选择要编辑的文件，当程序中含有不止一个窗口时，会有几个单元的源程序出现在代码编辑器中，如图 1-8 所示。

代码编辑器的标题条中显示了当前正在编辑的单元文件名。要查看某一特定程序的源代码，只需用鼠标单击写有该单元文件名的标签，就可以对该单元文件进行编辑了。

当单元文件较多的时候，选项卡右上方的导航按钮可以左右滚动选项卡标签，以便找到所需要的单元文件。

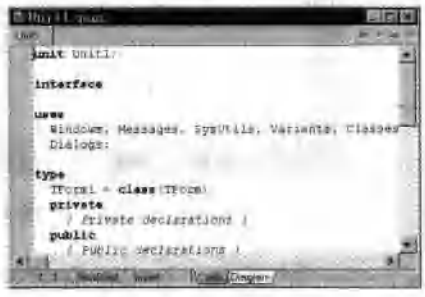

图 1-8　Delphi 6 的代码编辑器

代码窗口一开始处于窗体设计器之下。按〈F12〉键可以在代码窗口与窗体设计器之间切换。在“对象观察器”的事件选项卡中双击事件名右边的事件过程栏，可以打开“代码编辑器”，闪烁的光标将定位于事件过程中。如果是首次进入该事件过程，Delphi 会自动生成大部分的过程框架。

2．Delphi 的代码洞察

Delphi 6 的代码洞察（Code Insight）技术包括如下的代码信息显示工具。

（1）代码完善（Code Completion）。在编写代码时，只要输入已创建对象的名称和句点“.”，稍作停顿，系统将自动弹出一个提示列表框，列出该对象的所有属性和方法，如图 1-9 所示。选择所需的属性或方法，代码编辑器会自动将其插入代码行中。

（2）参数提示（Code Parameters）。在编写代码时，只要输入已创建对象的方法（内部过程或函数）名或已经声明的子程序（过程或函数）名并输入一个左括号，稍作停顿，系统

将自动弹出一个参数提示框，提示各参数的类型，如图 1-10 所示。

图 1-9　Delphi 6 的代码完善

图 1-10　Delphi 6 的代码参数提示

（3）代码模板（Code Templates）。用于提供语句模板以帮助代码的编写。例如，输入“Case”，按组合键〈Ctrl+J〉，系统自动弹出有关“Case”的代码模板供程序员选用，如图 1-11 左所示。

选中一种模板即可添加到代码中，既节省了代码输入的时间，又避免了输入时的语法错误，如图 1-11 右所示。

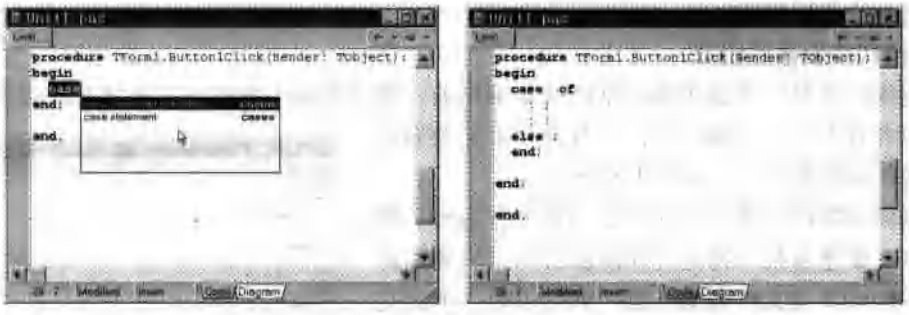

图 1-11　Delphi 6 的代码模板

（4）符号洞察（Tooltip Symbol Insight）。在编写代码时，将鼠标置于一个标识符（类型名、变量名、函数名、过程名）上，稍作停顿，系统将自动弹出一个黄色提示框。提示框中显示该标识符的类型（变量以 var 表示，函数以 func 表示）、所属的单元名、声明的位置（行号）等信息，如图 1-12 所示。

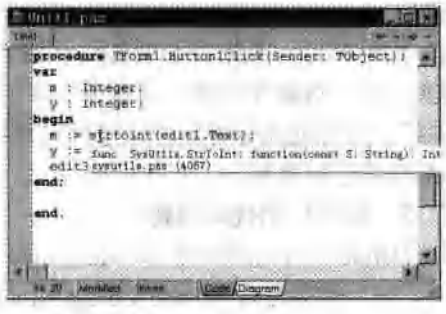

图 1-12　Delphi 6 的符号洞察

1.4　Delphi 可视化编程的步骤

“可视化编程”与传统的编程方法不同，不再需要编写大量代码去描述界面元素的外观与位置，而是采用面向对象、事件驱动的方法，利用 Delphi 所提供的可视“控件”，在系统提供的程序框架中加入完成功能的代码，其余的都交给 Delphi 去做。因此，Delphi 可视化编程的一般

步骤为：

（1）设计界面。利用控件在窗体上创建各种对象。

（2）设置属性。设置窗体和控件等对象的属性。

（3）编写代码。在 Delphi 所提供的程序框架中加入完成功能的代码。

当然，也可以在创建对象的同时，一边设置对象的属性，一边编写事件的过程代码。

1.4.1 创建一个工程

在 Delphi 中开发的每个应用程序都被称为工程，Delphi 编程首先从建立一个工程开始。新建一个工程有如下两种方法：

（1）启动 Delphi 后，系统将自动生成一个默认的工程 Project1。

（2）在“File”菜单中选择“New Application”选项，系统将创建一个新的工程。新工程的名称依次为 Project2、Project3…。

新创建的工程中包含一个默认的主窗体 Form1 和相应的单元 Unit1。根据工程设计的需要，还可以添加多个窗体，单击 View 工具条中的“New Form”按钮，或者在“File”菜单中选择“New Form”选项，都可以在当前的工程中添加一个新窗体。添加的窗体名称依次为 Form2、Form3…。

1.4.2 添加控件

设计工程直接面对的是窗体，因此主要工作就是在“窗体设计器”中完成窗体的设计。在窗体中可以添加各种对象，进行窗体的界面设计。向窗体中添加对象的方法为：

（1）单击组件板中的控件图标。

（2）在窗体的适当位置按下鼠标左键并拖动鼠标，即可画出相应的对象。

如图 1-13 所示，在窗体 Form1 上绘出了程序所需的对象，依次分别为标签 Label1 和按钮 Button1、Button2，同类型的对象序号依次自动增加。

图 1-13 增加一个标签和两个按钮

1.4.3 设置属性

对象属性的设置是在对象观察器中属性选项卡中进行的，其操作方法如下：

（1）首先设置窗体 Form1 的属性。单击窗体的空白区域（不要单击任何控件），确认选中的是窗体，可从“对象”下拉列表框中查看。

在属性名一栏中找到标题属性 Caption，将其值改为“第一个例子”，如图 1-14 所示。

当然，窗体的其他属性也可根据程序的需要进行设置。如窗体的名称属性 Name、运行时窗体的背景颜色、边框风格、窗体的大小以及最大、最小化的状态等。

（2）设置其他对象的属性。单击窗体上的对象，确认选中该对象，然后根据需要逐一设置对象的各属性。

选中标签“Label1”，标签的四周出现 8 个黑色小方块（表示选中）。修改其标题（Caption）属性为：欢迎使用 Delphi 6；用鼠标单击颜色（Color）属性右边的箭头按钮，从弹出的调色

板窗口中选择“黄色（clYellow）”（见图 1-15 左）；用鼠标单击字体（Font）属性右边的“...”按钮，从弹出的字体对话框中设置相应的字体、字体样式和字体大小等（见图 1-15 右）。

图 1-14　设置窗体 Form1 的属性

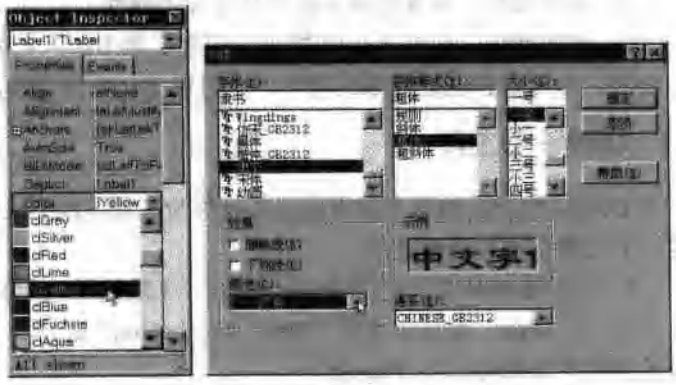

图 1-15　调色板与字体对话框

将两个按钮的标题分别设置为“时间”和“关闭”。所有对象的属性设置参见表 1-6。

表 1-6　属性设置

对　象	属　性	属 性 值	说　明
Form1	Caption	第一个例子	窗体的标题
Label1	Caption	欢迎使用 Delphi 6	标签的标题
	Color	（黄色）	背景色
	Font.Color	clRed（红色）	字体颜色
	Font.Name	隶书	字体名
	Font.Size	26	字体大小
	Font.Style	[fsBold]	粗体字
Button1	Caption	时间	按钮的标题
Button2	Caption	关闭	按钮的标题

属性设置后的窗体如图 1-16 所示。

图 1-16　属性设置后的窗体

1.4.4 编写代码

在对象观察器中选择“事件”选项卡，并确认对象列表框中的当前对象是 Button1。在事件选项卡中选择 OnClick 事件，用鼠标双击右边一栏，为 Button1 添加 OnClick 事件过程。

Delphi 自动生成该事件过程的代码框架，同时打开代码编辑器窗口，将光标定位在代码框架中过程体的首行处，等待输入过程代码（见图 1-17 左）。

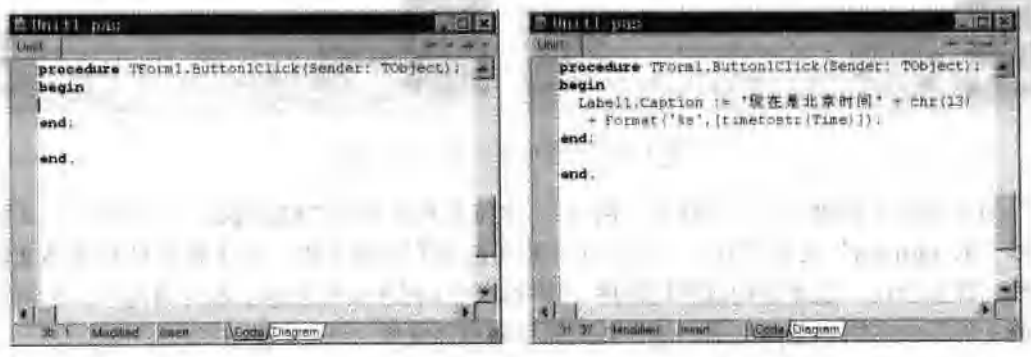

图 1-17　输入事件过程代码

在 begin 与 end 之间输入过程代码：

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Label1.Caption := '现在是北京时间' + chr(13) + Format('%s',[timetostr(Time)]);
end;
```

如图 1-17 右所示，其中粗体部分是系统自动生成的框架，不必重复输入。

用同样的方法，输入按钮 Button2 的单击（OnClick）事件过程代码如下：

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  close;
end;
```

1.4.5 保存工程

设计好的应用程序在运行之前最好先保存起来，即以文件的方式保存到磁盘上，以免因意外而丢失。一般是先将程序写入磁盘，然后再调试程序；当然，也可先对程序进行调试和运行，再写入磁盘。

保存文件的方法有两种：

- 单击菜单“File”→“Save All”，如图 1-18 所示。
- 单击“Standard”工具条上的“Save All”按钮。

图 1-18　“File”菜单

如果是从未保存过的新建工程，Delphi 将依次打开“Save Unit1 As”对话框与“Save Project1

As”对话框，分别保存单元文件与工程文件，如图 1-19 所示。

图 1-19　保存单元文件与工程文件

为上面的工程建立一个文件夹“例 1”，并将单元文件以“Main.pas”为名保存，工程文件以“Myproj.dpr”为名保存。由于一个工程可能含有多种文件，如工程文件和窗体文件，这些文件集合在一起才能构成应用程序。因此，建议程序员在保存工程时将同一工程所有类型的文件存放在同一文件夹中，便于修改和管理程序文件。

如果当前的工程文件在磁盘上已经保存过，系统将直接保存文件，而不会弹出上述对话框。

说明：也可以分别保存单元文件与工程文件，或将当前的单元文件或工程文件以另外的文件名保存在磁盘上。其操作见表 1-7。

表 1-7　保存文件的菜单命令

菜 单 命 令	说　明
“File” → “Save”	保存单元文件，如果是首次保存，则同“File” → “Save As”
“File” → “Save As”	打开“Save Unit As”对话框，将当前单元文件更名保存
“File” → “Save Project As”	打开“Save Project As”对话框，将当前工程文件更名保存

1.4.6　运行工程

单击“Debug”工具条上的“Run”按钮或按〈F9〉键，系统将开始编译、连接、运行该工程。若发现错误，编译器将返回代码编辑器，并给出提示；若无错误，系统将生成可执行文件 myProj.exe，并执行它。程序执行的结果如图 1-20 左图所示。用鼠标单击“时间”按钮，窗体显示见图 1-20 右。

图 1-20　运行工程

单击窗体上的“关闭”按钮可关闭该窗口结束运行，返回“窗体设计器”窗口。

1.4.7 关闭工程和关闭 Delphi

选择菜单“File”→“Close All”，Delphi 将关闭和工程有关的所有文件。在关闭工程及每一个文件之前，Delphi 会检查是否在最后一次修改后做过保存。如果没有保存，将弹出“Confirm”对话框，询问是否保存（见图 1-21）。如果单击“Yes”按钮，则保存该文件（首次保存还会询问文件名），然后继续关闭工程中的其他文件。如果单击“No”按钮，则不保存该文件，继续关闭工程中的其他文件。如果单击“Cancel”按钮，则取消此次关闭工程文件的操作，返回设计状态。

图 1-21 Confirm 对话框

关闭 Delphi 只需单击主窗口右上角的“关闭”按钮，或是选择菜单“File”→“Exit”即可。在关闭 Delphi 之前，系统将先关闭当前的工程。在关闭工程的过程中，如果单击“Cancel”按钮，则取消此次关闭 Delphi 的操作，返回设计状态。

1.4.8 修改工程

修改工程包括修改对象的属性和代码，也可以添加新的对象和代码，直到满足工程设计的需要为止。

在 Delphi 中，选择菜单“File”→“Open Project”，重新打开上述工程 myProj，并作如下修改：

（1）修改按钮的 Caption 属性：将两个按钮的标题分别设置为“时间(&T)”和“关闭(&E)”，其中“&T”使得字母 T 下方显示下划线，并且成为一个热键：当从键盘上键入〈Alt+T〉时，相当于用鼠标单击该按钮。

（2）修改按钮 Button1 的 Default 属性改为：True。此时，按钮 Button1 被加上黑边，表示该按钮是窗体中的默认按钮。当程序运行时，按下回车键，相当于用鼠标单击默认按钮。

按〈F9〉键，重新运行程序，可以验证上述改变。

1.4.9 Delphi 程序的基本组成

下面介绍 Delphi 为上例中的 myProj 工程创建的文件。

打开 myProj 工程所在的目录“例 1”，可以看到 7～8 个文件，其中与程序设计紧密相关的文件只有 3 类：工程文件、单元文件、窗体文件。

1. 工程文件

工程文件的扩展名为：.dpr，是一个 Pascal 语言源程序，它是整个应用程序的主程序。该程序执行两项功能：一是说明工程中的单元模块；二是启动应用程序。工程文件由 Delphi 自动生成，一般情况下不要修改这个文件。

选择菜单“Project”→“View Source”，可以在代码编辑器中打开工程文件 myproj.dpr 进行查看和修改：

```
program myProj;                    // 保留字 program 表明这是一个工程文件，
```