“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

编程宝典
2002
6

北京希望电子出版社 总策划
张龙卿 编 写

Delphi 6.0 数据库深入编程技术

作者精心设计14个完整范例深入描述
Delphi 6 开发数据库的方法和技巧，
重点突出，实用性强

中国科学出版集团
北京希望电子出版社

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

编程宝典
2002
6

北京希望电子出版社 总策划
张龙卿 编 写

Delphi 6.0 数据库深入编程技术

作者精心设计14个完整范例深入描述
Delphi 6 开发数据库的方法和技巧，
重点突出，实用性强

中国科学出版集团
北京希望电子出版社

内 容 简 介

这是一本面向初、中级读者的自学指导书，作者通过精心设计近 30 个（完整范例 14 个）范例深入介绍 Delphi 6 开发数据库的方法和技巧。全书由 14 章构成，主要内容包括 Delphi 6 基础；开发数据库应用程序，开发 BDE 数据库应用程序；开发 ADO 数据库应用程序；开发 InterBase 数据库应用程序；dbExpress 开发跨平台数据库程序；使用 Database Desktop 和 Datapump；数据库综合实例；打印报表；Decision Cube 面板与数据分析；界面设计常用功能；处理应用程序的异常；建立跨平台应用程序；跨平台应用程序实例，

本书内容新、范例丰富、重点突出、实用性强，不但是从事用 Delphi 6 进行数据库开发的广大初、中级编程员的自学指导书，对高级程序设计人员也有重要的参考价值。

本 CD 为配套书。

系 列 盘 书：“十五”国家重点电子出版物规划项目　计算机知识普及和软件开发系列
编程宝典 2002（6）
盘 书 名：Delphi 6 数据库深入编程技术
总 策 划：北京希望电子出版社
文本著作者：张龙卿
CD 制 作 者：希望多媒体开发中心
CD 测 试 者：希望多媒体测试部
责 任 编 辑：但明天
出版、发行者：北京希望电子出版社
地 址：北京中关村大街 26 号，100080
网址: www.bhp.com.cn　E-mail: lwm@hope.com.cn
电话: 010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309
（图书发行和技术支持）
010-62613322-215（门市）　010-62547735（编辑部）
经 销：各地新华书店、软件连锁店

排 版：希望图书输出中心　全卫
CD 生 产 者：北京中新联光盘有限责任公司
文本印刷者：北京广益印刷有限公司
开本 / 规格：787 毫米×1092 毫米　16 开本　20.75印张　470 千字
版次 / 印次：2002 年 1 月第 1 版　2002 年 1 月第 1 次印刷
本 版 号：ISBN 7-900088-08-3
印 数：1-5000
定 价：35.00 元（本版 CD）

前　言

Delphi 6可以直接开发同时运行在Windows和Linux平台上的应用程序。前段时间Borland公司（即Inprise公司）发布的Kylix应用程序，实际上是Delphi6的Linux版本，只能用于开发运行在Linux上的应用程序。对于习惯于使用Windows开发环境的许多开发人员来说，如果要在Linux下使用Kylix开发应用程序，可能还需要很长时间的熟悉过程。而使用Delphi6开发跨平台的应用程序却没有任何障碍，熟悉Delphi的用户仍可以像以前那样使用它在Windows下开发应用程序。现在，Delphi6可以建立两种类型的应用程序：一种是使用VCL组件库开发的应用程序，它只能运行于Windows平台上，这主要是保证与以前版本的兼容；另一种是使用CLX库开发的应用程序，这种应用程序既可以运行在Windows平台上，也可以运行于Linux平台上。使用CLX与VCL开发应用程序的方法完全相同，它们具有的大部分组件的名称、属性、方法及事件等也非常类似，通过本书的学习，可以很快掌握它们的异同点，这样就可以迅速地使用CLX组件开发出跨平台的应用程序。这些应用程序如果要移植到Linux平台上运行，只需要使用Kylix重新编译即可，几乎不需要修改任何代码及界面。

本书首先讲解了使用Delphi6开发应用程序的基础知识，以便使初学者迅速入门。然后对各种数据库组件的属性、方法、事件，数据库应用程序的体系结构及开发不同数据库应用程序的方法进行了详细讲解，另外，还重点讲述了CLX及VCL库的异同点，同时说明了如何使用CLX库开发跨平台的应用程序。还讲解了如何进行数据图表分析、建立快速报表、处理异常及如何使用设计界面常用组件等内容。

本书的一个突出特点是，通过大量实例说明问题，同时每个实例都提供一些实用的技巧。对一些组件的使用方法及开发应用程序中遇到的许多难题，都进行了透彻的剖析，以便读者在学习过程中少走弯路，提高学习和工作效率。

如果要学习使用Delphi6开发数据库应用程序和跨平台应用程序，本书可能是最合适的选择，它不但可以作为入门教程，也可以作为参考书。

本书由银河科技文化公司编写，另外，北京的欧洋、张劲、刘凯平、黄木平和深圳的梅红威参与了部分章节的编写工作，并提供了许多参考资料，在此一并表示感谢。

编　者

目　录

第 1 章　Delphi 6 基础

大家知道，Delphi 在开发数据库及设计应用程序界面方面有着不同寻常的优势：开发简单、设计方便、容易上手、帮助完善。只要对编程略有基础，则使用 Delphi 开发一般的应用程序界面及数据库应用程序都易如反掌，所以它越来越受到程序员的青睐。业界人员都认为：执着的程序员使用 C++语言，聪明的程序员使用 Delphi。使用 Delphi 编程，往往可以使程序员的工作事半功倍。

Delphi 6 在 Delphi 5 基础上又进行了很大的改进，它几乎可以实现 C++能够实现的所有功能，即不但设计应用程序界面方便快捷，在开发底层的应用程序方面也可以与 C++语言媲美。为了使大家迅速掌握 Delphi 6 的新增功能，下面首先介绍一下 Delphi 6 的新特点。

1.1　Delphi 6 的新特点

1．BizSnap 轻松创建基于 SOAP/XML 的 Web 服务，简化电子商务的集成。Delphi 6 无缝集成了基于 SOAP 的 Web 服务和 XML 数据交换技术，简化新一代电子商务的开发，是当前唯一在 Internet 上集成 Web 服务、B2B、B2C 和 P2P 的快速开发工具。

2．WebSnap 是基于组件的 Web 应用开发平台，支持 Apache、Netscape 和微软 IIS 等主流 Web 应用服务器。WebSnap 将 Delphi 应用及以当前流行的 HTML 开发环境——如 Dreamweaver、Frontpage、VBScript 和 JavaScript 所开发的网站无缝集成在一起。

3．DataSnap 创建高性能、提供 Web 服务组件，使客户端应用能够快捷地与 Internet 主流数据库连接。DataSnap 支持 Oracle、MS-SQL Server、Informix、IBM DB2、SyBase 和 InterBase 等主流数据库。它遵循业界标准的 SOAP/XML HTTP 协议，使客户端无需数据驱动和复杂配置，即可直接与其相连接。DataSnap 还支持 DCOM、CORBA、TCP/IP。

4．构建单一代码的 Windows/Linux 应用。Delphi 6 与 Kylix 兼容。使用 Kylix，可在 Linux 平台上重新编译基于 Windows 平台的 CLX 应用；而利用 Delphi 6，即可在 Windows 上重新编译基于 CLX 组件的 Linux 应用。Delphi 6 包含 BaseCLX、VisualCLX、DataCLX 和 NetCLX 四个组件。

5．Delphi 6 与 AppServer 集成。Delphi 6 通过最新 SIDL 与 AppServer 连接。它为 AppServer 应用开发出高性能、具有丰富 GUI 环境的客户端应用，通过 Internet 将 AppServer 的 EJB 功能作为遵循业界标准的 SOAP/XML Web 服务发布到全球。

6．Borland VisiBroker for Delphi——支持客户端与服务器端的完整开发。Delphi 为 AppServer 和 VisiBroker 应用开发出高性能、具有丰富 GUI 环境的客户端和 Web 浏览器。在复杂的 IT 环境下，Delphi 6 所建立的 VisiBroker CORBA 对象能与任意 CORBA 客户端或对象进行互操作。

7．支持最新 Windows 2000/XP 和 Office 2000。通过 ActionBands、ActionManagers 和 Shell Controls 创建的用户界面支持微软最新平台。使用户无需在繁琐的用户界面上耗费精力，便能轻松定制属于自己的 GUI 应用。

1.2　Delphi 6 的开发界面

打开 Delphi 应用程序，用鼠标单击系统菜单 File|New Application，则自动生成第一个项目文件，缺省文件名为 Project1.dpr，缺省表单为 Form1，代码编辑器中单元文件缺省名字为 Unit1.pas。这时的界面如图 1-1 所示。

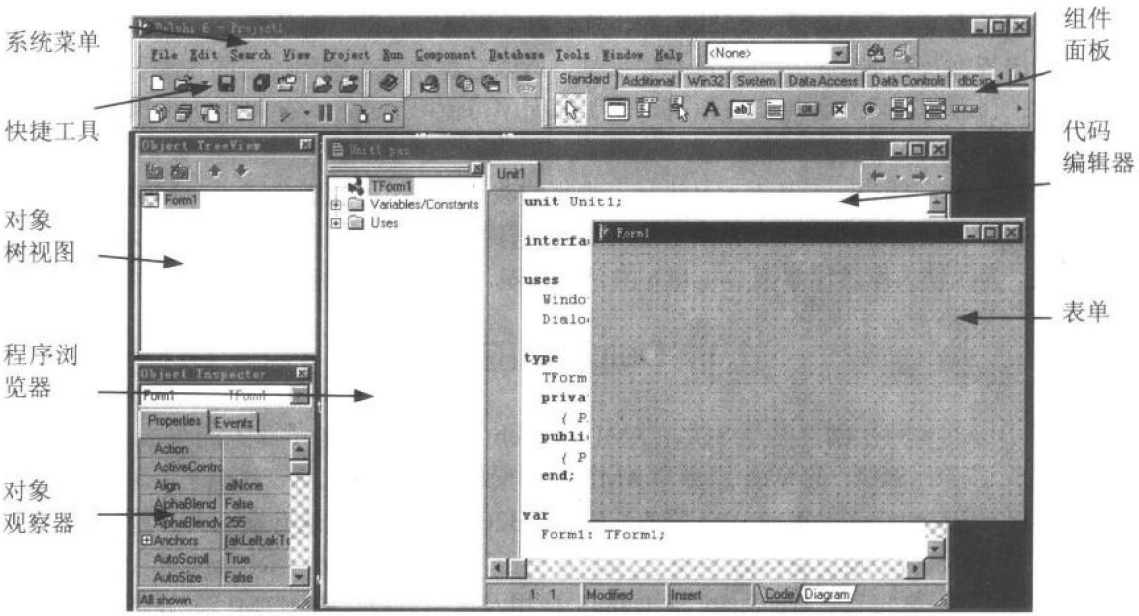

图 1-1　Delphi 6 开发界面

从图 1-1 中可以看出，Delphi 6 的集成开发环境（IDE）包括如下几部分：系统菜单（Menu）、快捷工具（Speed Bar）、组件面板（Component Palette）、对象树视图（Object TreeView）、程序浏览器（Code Explorer）、对象观察器（Object Inspector）、代码编辑器（Code Editor）和表单（Form），其中对象树视图是 Delphi 6 中新增加的功能。

下面分别说明各部分的作用。

1．系统菜单（Menu）　系统菜单提供了 Delphi 6 集成开发环境中开发应用程序所需要的各种功能。

2．快捷工具（Speed Bar）　快捷工具是部分系统菜单的另一种表现形式，通过这些工具可以快速打开一项菜单所提供的功能。其中，与 Internet 有关的三个快捷工具是新增加的，要显示或关闭一些快捷工具，可以通过在该面板上单击鼠标右键，此时会弹出一个菜单，如图 1-2 所示，然后根据需要选择或取消选择某个菜单，即可打开或关闭相应类别的快捷工具。

图 1-2　快捷工具弹出菜单

要区分某个菜单控制哪些快捷工具，可以反复打开再关闭每一个菜单，然后观察快捷工具面板的变化，即可知道其控制的快捷工具。

3．组件面板（Component Palette）　使用 Delphi 开发应用程序时，大部分界面的设计

及功能的实现都是通过组件面板上的各个组件实现的。这些组件以类的形式出现，所以如果要使用某个组件，则相当于使用了该组件的实例。所有的组件根据其功能划分为多个类表，每一种类别的组件放置在特定的一个组件面板页上。Delphi 6 中定义了这些组件的两种标准格式：一种格式为可视化组件类型库 VCL（Visual Conponent Library），这种格式的组件用来开发运行在 Windows 操作系统上的应用程序，Delphi 5 及以前的版本中的所有组件都属于该格式；另一种格式是跨平台的组件库 CLX（Borland Component Library for Cross Platform），这是 Delphi 6 新增加的功能，用于建立可以同时运行在 Windows 和 Linux 操作系统上的应用程序。Linux 上对应 Delphi 6 的 Kylix 也使用该 CLX 组件库开发跨平台的应用程序。注意，该组件库是 Borland 产品中特有的功能。

由于使用 VCL 建立 Windows 应用程序与使用 CLX 建立跨平台的应用程序使用了不同的组件库，所以对应的组件面板也不同。

（1）VCL 组件面板　共包括 27 个组件面板，这些面板的作用如下：

- Standard（标准）　该组件面板包括一些 Windows 标准的控制项或选择菜单。
- Additional（附加）　该组件面板包括一些常用的控制项。
- Win32　该组件面板包括一些 Win 32 中常用的控制项。
- System（系统）　该组件面板包括一些与系统有关的控制项。
- Data Access（数据访问）　该组件面板包括一些存取数据库的不可见组件。
- Data Controls（数据控制）　该组件面板包括一些设计数据库应用程序界面使用的可见数据库组件。
- dbExpress　该组件面板包括一些建立跨平台数据库应用程序的组件。
- DataSnap　该组件面板包括一些用于建立多层数据库的组件。
- BDE　该组件面板包括一些使用 BDE（Borland 数据库引擎）开发数据库应用程序时使用的数据库控制项。
- ADO　该组件面板包括一些使用 ADO 开发数据库应用程序时使用的数据库控制项。
- InterBase　该组件面板包括一些开发数据库应用程序时使用的数据库控制项。
- WebServices　该组件面板包括一些通过 SOAP 开发 Web 服务的组件。
- InternetExpress　该组件面板包括一些用于建立以 Web 浏览器为基础的瘦客户机应用程序组件。
- Internet　该组件面板包括一些协助开发和管理 Internet 应用程序的组件。
- WebSnap　该组件面板包括一些可以快速建立 Web 应用程序的组件。
- FastNet　该组件面板包括一些用于开发及管理网络应用程序的组件。
- Decision Cube（决策立体图表）　该组件面板包括一些分析和显示数据的组件。
- QReport（报表）　该组件面板包括一些建立打印报表所需要的各种组件。
- Dialogs（对话框）　该组件面板包括一些建立各种对话框的组件。
- Win 3.1　该组件面板包括一些兼容 Windows 较早版本的组件。
- Samples（例子）　该组件面板包括一些例子组件。
- ActiveX　该组件面板包括一些实现 ActiveX 功能的组件。
- COM+　该组件面板包括一些用于开发兼容微软 COM+组件的控件。

- Indy Clients（Indv 客户） 该组件面板包括一些建立 Indv 客户的组件。
- Indy Servers（Indv 服务器） 该组件面板包括一些建立 Indv 服务器的组件。
- Indy Misc 该组件面板包括各种实现 Indy 服务的组件。
- Servers（服务器） 该组件面板包括一些与微软的 Office 套件相关的组件。

（2）CLX 组件面板 共包括 14 个组件面板，其中大部分面板的名字与 VCL 组件面板的名字相同，其包括的组件也实现类似的功能，只有 Common Controls 组件面板是 CLX 特有的，它包括与 VCL 中的 Win32 组件面板相似的组件，也实现类似的功能。

图 1-3 对象树视图

4．对象树视图（Object TreeView） 对象树视图是 Delphi 6 中新增加的功能，如图 1-3 所示，它以树状图表显示放置在一个表单、数据模块或框架中的所有可见或不可见组件。对象树视图显示组件之间的逻辑关系，比如同属关系、父子关系和属性联系。一个表单与放置在该表单上的按钮之间就是父子关系；一个属性集与它的 FieldDefs 属性就有联系。有一些联系是隐含的，比如数据集组件与它的属性。可以通过将一个组件拖拉到另一个组件的顶部来建立其他的关系，只要它们之间有建立联系的可能性即可。

比如，在表单中放置了一个面板 Panel 组件 Panel1 和一个标签 Label 组件 Label1，则可以在对象树视图中将 Label1 拖拉到 Panel1 的下面，这样 Panel1 和 Label1 之间就形成了一种父子关系，同时也将表单 Label1 放置到了 Panel1 上面。这样在表单中拖拉 Panel1 时，Label1 也跟随移动。

在树状图中有一些节点以黑白颜色的图标显示，则这些节点代表了隐含的组件，例如，一个 Dataset（数据集）有一个缺省的 session（会话）与它关联，应用程序运行时会产生该会话。

如果一个节点使用黄色圆内的一个问号标注，则表示该项还没有完全定义或存在问题，比如，一个没有设置 Dataset 属性的数据源就会显示该图标。

如果在一个项上双击鼠标，则可以打开代码编辑器，可以为事件处理器编写代码。

如果在对象树视图的一个项上单击鼠标右键，则会显示一个上下文菜单，该菜单是在表单、数据模块和框架中相同组件上面单击鼠标打开菜单的子集。

如果要在代码编辑器的 Diagram 页面建立一个图表，则可以从 Object TreeView 中拖拉一些元素到 Diagram 页面中。

可以通过 View|Object TreeView 打开 Object TreeView 窗口。该窗口可以与对象观察器面板合并，合并的方法是将对象树视图拖拉到对象观察器面板的中间位置，如果在对象观察器的中间出现一个虚线框，则应释放鼠标，此时两个窗口就放置在一起，如图 1-3 所示。当然，该窗口也可以与代码编辑器等窗口放置到一起，只要将其拖拉到合适的位置即可。

5．代码浏览器（Code Explorer） 代码管理器以树状的结构管理程序的代码，它一般与代码编辑器放置在一起，使用它可以很容易地在单元文件中进行导航。可以通过

View|Code Explorer 菜单打开该窗口，其显示如图 1-4 中右边窗口所示。

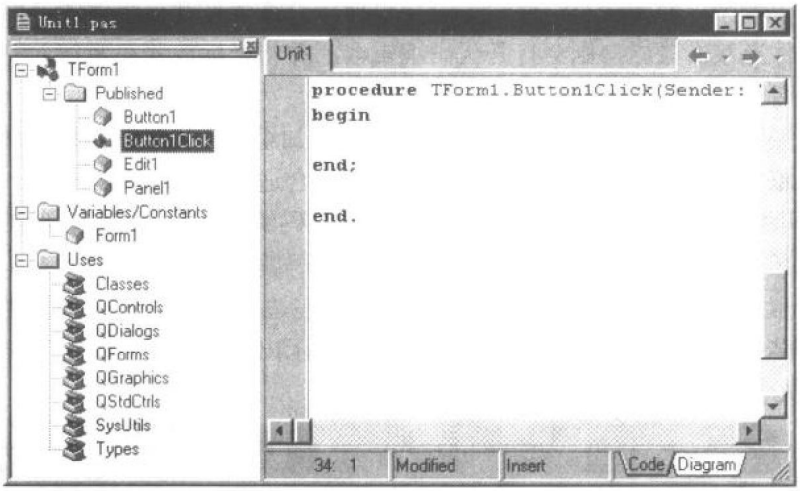

图 1-4　代码浏览器（右边）

代码浏览器窗口包含的树状图显示了在应用程序的一个单元文件中定义的所有类型、类、属性、方法、全局变量和全局程序，它也会显示在 uses 语句中包括的其他单元名称的列表，可以展开或收缩树的节点。

当选择一个新的单元文件时，代码浏览器中也会显示该单元文件中的内容。代码浏览器一次只能显示一个单元文件的内容。

如果在代码浏览器的一个项上面双击鼠标，则会在代码编辑器中导航到其对应的代码位置。

要调整代码浏览器的设置，可以选择菜单 Tools|Environment Options|Explorer。

6．对象观察器（Object Inspector）　它用来监视或设置表单中组件的属性或事件。可以按 F11 键或选择 View|Object Inspector 菜单打开它。当在表单、数据模块或框架中选择一个组件时，则对象观察器中会显示出该组件的属性。其显示如图 1-5 所示。

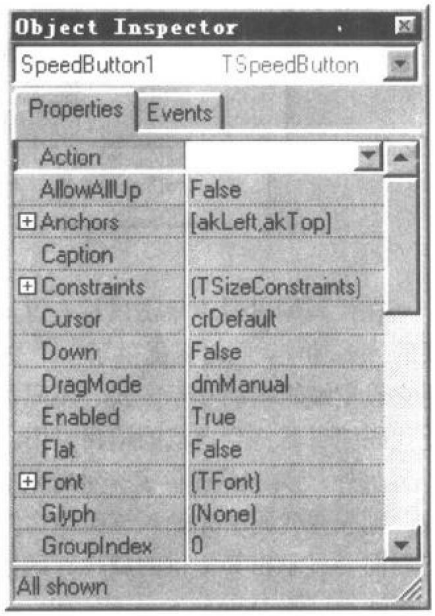

图 1-5　对象观察器

对象观察器包括三部分：对象选择器、属性标签页和事件标签页，它们的作用如下：

- 对象选择器　它位于面板的顶部，可以从下拉框中选择任何一个组件。
- 属性标签页　该标签页主要在应用程序的设计阶段设置各个组件的一些属性。属性用来描述组件的一些特征。Delphi 6 中对一些经常需要设置的或重要的属性使用褐色标注。
- 事件标签页　事件通常指一个对象的反应行为，例如，用户在按钮上单击鼠标，则会触发一个 OnClick 事件。如果要建立一个窗口，则会出发 OnCreate 事件。一

个组件执行的动作都是通过触发某个事件实现的，对应每个事件的代码相当于一个子程序。要为一个组件设置触发事件，只需要打开该组件的事件面板，在对应组件上双击鼠标，则会打开代码编辑器，并将光标放置在新加入的事件中，然后编写相应的实现代码即可。

Delphi 应用程序的建立步骤是，先在表单中添加组件，再设置组件的属性，并通过事件面板加入必要的事件，然后在代码编辑器中添加合适的执行代码。

7. 代码编辑器（Code Editor）　顾名思义，代码编辑器用于编辑程序的代码，它具有比较完整的编辑功能，提供了多种编辑功能键。代码编辑器的显示如图 1-6 中右侧的窗口所示，它一般与代码浏览器放置在一起（见图 1-6 左边窗口）。另外，在编译代码时如果遇到错误或警告等内容，则在代码编辑器的下面会显示出一个编译消息窗口（见图 1-6 下面的窗口），从中可以确定代码中引起错误的位置。

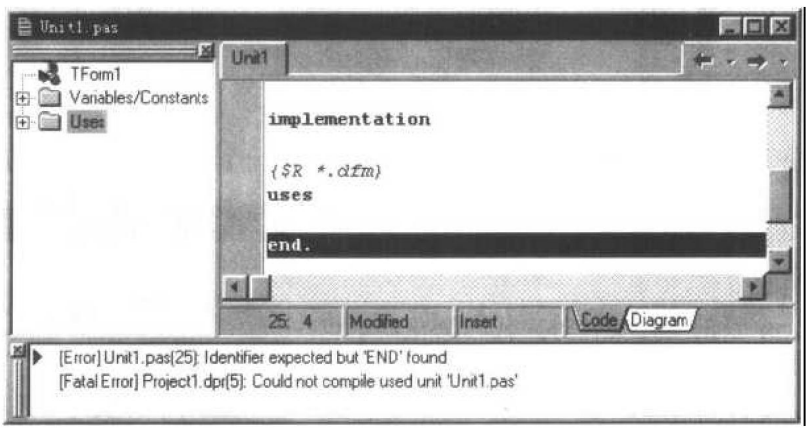

图 1-6　代码编辑器

在代码编辑器中单击鼠标右键，则会打开一个上下文菜单，该菜单提供了编辑代码需要的许多功能，同时还提供了一个 Properties 菜单用于设置代码编辑器的属性，比如可以使用它设置编辑器的显示方式、不同类型的代码使用的颜色等，通过这些属性的设置，可以使代码编辑器符合自己的编辑习惯。

另外，Delphi 提供了一种特别有用的功能，即如果要使用某些组件或对象的属性或函数，则应在代码中先输入该组件实例的名字，然后再输入一个小点，比如为 Form1.，此时它后面会显示出其提供的属性、函数或子对象的列表，同时不同类型的内容使用不同的颜色标注。这与 Delphi 5 是不同的。另外，可以按照子对象的作用范围或子对象名称的字母排列顺序对该列表框中对象进行排列。在列表中单击鼠标右键，并从弹出菜单中选择合适的菜单即可确定排列顺序。图 1-7 显示的是按照名称排列的列表。

通过子对象列表的功能，可以很容易地查找到当前对象的一些属性、函数及子对象等。

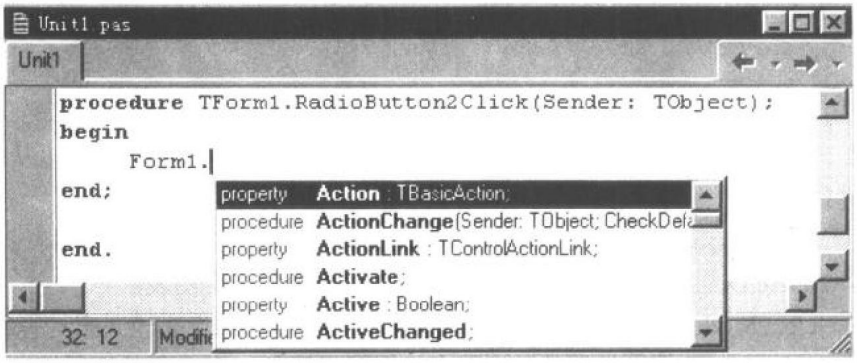

图 1-7　代码编辑器中显示的子对象列表框

Delphi 中的所有对象都是有归属关系的，父对象与子对象之间使用一个小点“.”表示其父子关系。比如，如果一个按钮 Button1 放置在表单 Form1 中，则在第二个表单 Form2 对应的单元文件 Unit2 中修改 Button1 的标题时，可以使用如下的代码：

```
Form1.Button1.Caption:="确定";
```

如果是在 Form1 的单元文件 Unit1 中修改 Button1 的标题，则可以省略 Form1。　可见，一个小点“.”建立的联系，决定了一个对象的归属。

8. 对象图表　在图 1-1 中介绍设计界面的时候，没有提到对象图表，因为它是代码编辑器中的一个页面（Diagram）。这是 Delphi 6 新增加的功能，所以单独讲解该部分可能会更有利。其显示如图 1-8 所示，该图表示在一个组合框中放置了两个单选按钮，故它们构成了父子关系。

如果没有将组件从 Object TreeView 中拖拉到 Diagram 页面，则它不会显示任何组件内容。通过拖拉，可以将多个组件和它们的子对象和属性放置到 Diagram 页面中。

- 如果选择多个组件直接拖拉到 Diagram 页面中，则这些组件会垂直排列。
- 如果在按住 Shift 的同时拖拉多个组件到 Diagram 页面中，则这些组件会水平排列。
- 要重新排列 Diagram 页面中的组件，则需要重新从 Object TreeView 中拖拉它们。

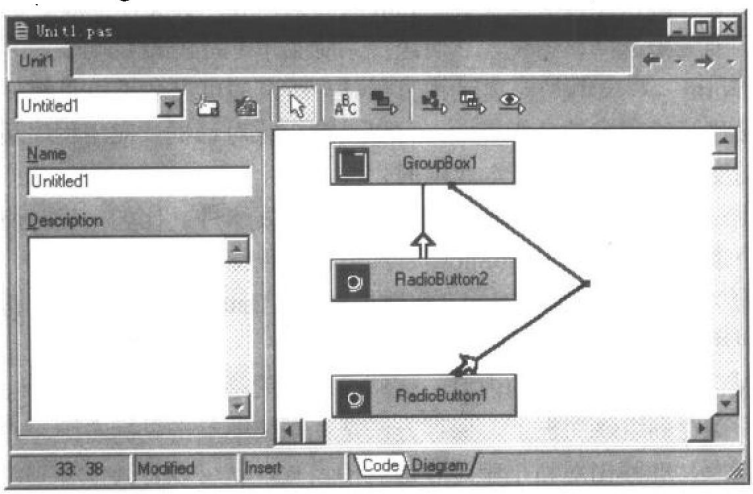

图 1-8　代码编辑器中的 Diagram 页面

Diagram 页面可以划分为三部分：上面一行提供了一些建立对象关联的工具及对图表

进行管理的工具；左下角部分是当前选择图表的名字及描述；右下角部分则显示了对象之间的联系。

（1）工具图标的作用　Diagram 页面顶部一排快捷工具图标的作用如下：

- Untitled1　该列表框中列出了已经建立的所有对象图表，可以从中选择一个进行编辑。
- New Diagram（新建图表）　单击该图标，可以建立一个新的图表。
- Delete Diagram（删除图表）　该图标用于删除一个图表。
- Select Mode（选择模式）　单击该按钮，可以进入选择模式来选择组件。
- Comment Block（注释块）　单击该图标会添加一个黄色的注释块，可以输入注释内容。
- Allude Connector（隐含连接器）　选择该图标后，可以在 Diagram 页面上的对象之间连接隐含的连接关系，使用一个单向箭头将两个对象连接起来，对整个应用程序没有任何影响。所以一般使用该连接器连接注释块。
- Properties Connector（属性连接器）　如果一个对象的属性值是另一个对象，则可以使用该图标建立两者的联系。比如将 DBGrid1 的 DataSource 属性设置为一个数据源对象 DataSource1，则将 DBGrid1 和 DataSource1 同时拖拉到 Diagram 页面时，会显示两者的关系。如果还没有建立关系，则可以通过 Properties Connector 建立连接。
- Master|Detail Connector（主从连接器）　用于建立主从表之间的关联。
- Lookup Connector（查找连接器）　建立查找字段与当前数据集的关联。

注意：Diagram 页面建立的关联也会作用到 Object TreeView 窗口中相应对象上面。

（2）对象之间的联系　通过以上的各个快捷图标可以建立对象之间的联系，这些关联使用一些带箭头的线段表示。可以在一条线段的任何位置按下鼠标左键并拖拉鼠标，则可以使线段出现拐角（如图 1-8 所示）。

在保存单元文件时，会自动保存 Diagram 页面的内容，其文件扩展名是.DDP。如果要打印一个图表，则应先选择该图表，然后选择系统菜单 File|Print 即可。

9．表单（Form）　表单是放置组件的容器，所有的可见和不可见组件都可以放置到上面。表单也是显示给用户的最终界面，可以是窗口或对话框。应用程序运行时，用户与应用程序进行的交互，都是通过表单实现的。

表单本身也是一个组件，可以设置它的属性及触发事件。比如，当在应用程序显示表单时要触发一个事件，可以通过 OnShow 事件编写实现特定功能的代码。

与表单非常类似的是框架（Frame）和数据模块（Data Module），它们都是存放其他组件的容器，在应用程序中都不会显示。

框架用于保存一些重复使用的内容，这些内容一般是多个控件的组合。可以将整个框架作为一个组件保存到组件面板上，保存方法是在框架中单击鼠标右键，并从选择菜单中选择 Add to Palette，在弹出对话框中进行适当的设置，就可以将其保存到组件面板中重复使用。注意，当原框架中的组件进行了改变时，保存在面板上的组件也会做相应的改变。在表单中使用框架建立组件时，原框架中各个组件的触发事件仍然起作用，但是在插入到表单的实例中却无法看到这些触发事件。比如，在框架中插入了一幅图像、一个按钮，然

后将其保存到组件面板中。以后在表单中插入该框架生成的组件时，则会同时插入框架中的图像或按钮，也会使用它们具有的触发事件。

数据模块主要用于保存不可见的数据库组件，比如 DataSource、SQLDataSet、Table 等组件，以便集中管理和设置。

1.3 Delphi 应用程序的组成

Delphi 开发的应用程序或动态链接库是以项目（Project）的形式出现的，一个项目是多个文件的集合。其中一些文件是设计项目时生成的；另一些文件则是在编译项目时形成的。一般情况下，你不必对每个文件进行处理，尽管可以直接编辑大多数文件的源代码和脚本，但使用 Delphi 的集成开发环境及可视化工具更方便，也更可靠。

在打开 Delphi 时，会自动建立一个项目文件，如果使用 File|Save Project As 保存整个项目文件，则可以看到该文件包括如下文件：项目文件（.dpr）、项目选项文件（.dof，Kylix 中为.kof）、配置文件（.cfg，Kylix 中为.conf）、资源文件（.res）、表单文件（.dfm，跨平台应用程序中为.xfm）、单元文件（.pas）。其中前 4 个文件都使用项目的名字，比如为 project1；后两个文件使用单元的名字，比如为 Unit1。

1．项目文件　每个项目都包括 Object Pascal 源代码，Delphi 将其编译为最终的应用程序或动态库。项目文件一般包含对使用的所有表单和单元的引用，在装载、保存或编译一个项目时，Delphi 查看项目文件就知道各类文件的作用。项目文件最终会生成.exe 或.dll 文件，前者是应用程序，后者是动态链接库。

注意：由于 Delphi 本身维护项目文件，所以一般不需要手工修改它。对项目文件的改变也可以通过 Project Manager（项目管理器）实现，可以由 View|Project Manager 打开它。

要查看项目文件的源代码，可以通过 Project|View Source 打开它，也可以由 View|Units 菜单或快捷按钮打开它。缺省生成的项目源文件代码如下：

```
program Project1;
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

该源代码各部分的作用如下：

- **program**　保留字 program 指出该项目是一个应用程序，其名字为 Project1；如果该项目是一个动态库，则会使用 library 保留字。
- **uses**　该语句告诉编译器当前项目中使用了哪些单元，并将其链接到项目上。在表单中添加新的组件时，与其有关的标准单元文件会自动添加到该语句中，所以一般不需要对其修改。如果要引用自己建立的单元，可以再使用一个 uses 语句，并将其放置在{$R *.res}上面，这样源代码显得比较清晰，这种情况一般出现在表单

单元文件或单独的单元文件中。

- Unit1 缺省生成的单元标识符，它对应缺省建立的表单 Form1，UNIT1.PAS 是包含该单元源代码的文件名，它与单元标识符必须一致，否则项目无法正确被编译。所以如果修改了单元标识符，则也应以新的名字保存单元文件。
- **in** 该保留字告诉编译器如何找到每个单元的源文件，注释{Form1}表示该单元文件对应表单的名字。
- {$R *.res} $R 是一条编译器指令，它告诉编译器应该将与项目文件同名的资源文件*.res（*表示与当前的项目文件同名）链接到项目中。项目的资源文件包括项目的图标、图像等内容。
- **begin...end** 该部分是当前项目的主要源代码块，其中 Application.Initialize 语句用于对应用程序初始化；Application.CreateForm(TForm1, Form1)语句则用于建立参数中指定的表单，如果项目中有多个表单，则也会有多条与其对应的这种语句；Application.Run 语句会运行整个应用程序。

2、单元文件 Delphi 的 Object Pascal 语言支持单独编译的代码模块，称为单元，即一个单独的源代码文件。使用单元可以提高代码在各项目之间的结构化和重用性。Delphi 项目中一般的单元文件都与一个表单对应，它包含了事件处理程序和表单的其他代码。但是单元文件也可以不与表单关联，即可以通过 File|New Unit 菜单建立单独存在的单元文件，这些单元文件可以被多个项目使用。例如，可以编写实现特定功能的过程、函数、动态库和组件，然后将其保存为一个单独的单元文件，然后将这些单元文件拷贝到保存其他项目的目录中，并使用 uses 语句引用这些单元文件，即可重用它们。

单元文件被编译时，会生成一个二进制的编译文件，其扩展名是.dcu，这些文件用于项目快速的编译和链接。

一个单元文件可以划分为接口和实现两部分。其中接口部分以 interface（接口）保留字开始，到 implementation（实现）关键字为止，该部分用于单元文件中使用的类或变量的声明等；implementation 关键字以后的代码是该单元文件中实现的功能，包括对各个组件事件的处理及实现不同功能的过程和函数等。

缺省生成的表单单元文件的源代码如下，后面都添加了注释：

```
unit Unit1;          //单元文件的名字
interface            //接口部分的开始
uses                 //引用的标准单元文件
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs;
type                 //Delphi自动添加的类型声明，这是表单特有的
  TForm1 = class(TForm)
  private
    { Private declarations }     //该单元私有变量的声明
  public
    { Public declarations }      //声明公用类型
  end;                           //结束类型声明
var                              //声明变量或类的实例
  Form1: TForm1;
implementation                   //程序代码实现功能部分的开始
```