Delphi 数据库开发基础与应用

高 英 石文华 编 著

人 民 邮 电 出 版 社

图书在版编目（**CIP**）数据

Delphi 数据库开发基础与应用/高英，石文华编著．—北京：人民邮电出版社，2006.1
ISBN 7-115-13761-7

Ⅰ．D…　Ⅱ．①高…②石…　Ⅲ．软件工具－程序设计　Ⅳ．TP311.56

中国版本图书馆 CIP 数据核字（2005）第 159345 号

内 容 提 要

本书循序渐进地介绍了 Delphi 数据库系统的开发方法。书中首先介绍 Delphi 语法和基本组件的使用，然后介绍 Access 和 SQL Server 数据库管理系统，并着重介绍 SQL 语句和数据库访问控制组件等 Delphi 数据库开发的基础知识，最后以两个信息管理系统为实例详细介绍了使用 Delphi 开发数据库应用程序的全过程。

本书各章节既相对独立又紧密联系，内容由浅入深，讲解详细，重点突出，配有大量鲜活实用的例程，适合初级读者学习 Delphi 数据库应用程序设计，也可作为各类培训班的培训教材。

Delphi 数据库开发基础与应用

◆ 编　　著　高　英　石文华
责任编辑　刘　浩

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：19.75
字数：479 千字　　2006 年 1 月第 1 版
印数：1 – 6 000 册　　2006 年 1 月北京第 1 次印刷

ISBN 7-115-13761-7/TP・4881

定价：29.00 元

读者服务热线：（010）67132692　印装质量热线：（010）67129223



前　言

Delphi 是 Inprise（原 Borland）公司推出的应用程序开发工具，具有功能强大、运行速度快、易于学习和使用以及开发效率高等特点。它是可视化应用编程开发环境、可重用性面向对象编程语言、快速编译器和数据库的完美结合。Delphi 具备了作为优秀的编程开发软件的一切条件：与 VC++相比，具有更加简单易懂的 IDE 开发环境；与 VB 相比，代码更加规范，增加了程序的可移植性，且开发效率更高。

为了满足广大读者，尤其是编程爱好者的愿望和需求——迅速掌握 Windows 环境下的 Delphi 编程，特别是数据库应用开发的方法，我们组织编写了这本书。本书主要面对 Delphi 的初级用户，因此书中会有一些必要的理论介绍，特别是面向对象的编程方法。但作者认为，优秀的编程指导书籍应该是在实例中体现编程方法和技巧，通过引导读者亲自动手编程来理解整个编程体系，因此书中采用了大量鲜活实用的例程，紧密结合各部分的知识进行示范，在开发中激发读者的兴趣，从而迅速加入 Delphi 中、高级用户的行列。

作者在写作时以 Delphi 数据库开发应用为准则，针对 Delphi 整个开发体系和特点进行组织，制定了全书的布局，涵盖了 Delphi 数据库开发所有的重要方面，为开发人员提供了完整的知识架构。本书各章节都对应 Delphi 数据库开发应用的某一方面，具有相对的独立性，同时又紧密联系，秉承着另一准则——由浅入深。这两条准则便是这本书的经纬，希望读者在阅读本书之前要理解这一点，以更有效地使用本书。

本书配套代码可以从 www.ucbook.com 处下载，其中包括企业库存管理系统和人事管理系统的全部源代码。

本书由高英、石文华主笔，参加编写的还有张巍、张晓冬等同志。限于作者水平，书中不足和纰漏之处在所难免，恳请广大读者批评指正（责任编辑电子函件：book_better@sina.com），同时，也欢迎业内优秀作者联系我们。

编者

2006 年 1 月



目　录

第 1 章　Delphi 程序设计基础 ……1
1.1　Delphi 开发环境……1
1.2　Delphi 语言基础……15
第 2 章　面向对象设计基础……45
2.1　程序设计中的面向对象概念……45
2.2　Delphi 体系结构的关键类……61
2.3　异常处理 ……71
第 3 章　Delphi 常用组件 ……77
3.1　常用组件介绍……77
3.2　常用组件用法实例……100
第 4 章　Access 数据库管理 ……106
4.1　Access 数据库的基本操作 ……106
4.2　表管理 ……111
4.3　查询管理 ……120
4.4　维护数据库安全……131
第 5 章　SQL Server 数据库管理……134
5.1　常用工具 ……134
5.2　数据库管理 ……141
5.3　表管理 ……149
5.4　视图管理 ……155
5.5　数据库备份 ……159
第 6 章　常用 SQL 语句……161
6.1　INSERT 语句……161
6.2　UPDATE 语句 ……163
6.3　DELETE 语句 ……164
6.4　SELECT 语句……165
6.5　数据库对象管理语句……173
第 7 章　Delphi 数据库编程 ……181
7.1　Delphi 中数据库开发常用组件……181

7.2 Delphi 中 ADO 技术的使用 ……193
7.3 Delphi 中使用 SQL ……198
7.4 程序中的数据处理……203
第 8 章 企业库存管理系统（Access） ……208
8.1 系统设计 ……208
8.2 数据库设计 ……210
8.3 设计工程框架……216
8.4 系统主界面设计……226
8.5 登录模块设计……227
8.6 客户管理模块设计……230
8.7 仓库管理模块设计……235
8.8 货物管理模块设计……237
8.9 库存操作管理模块设计……242
8.10 库存警示管理模块设计……256
8.11 统计查询管理模块设计……261
8.12 用户管理模块设计……265
第 9 章 人事管理系统（SQL Server） ……270
9.1 系统设计 ……270
9.2 数据库设计 ……271
9.3 设计工程框架……275
9.4 系统登录 ……283
9.5 系统主界面设计……286
9.6 部门管理模块设计……295
9.7 工作简历管理模块设计……298
9.8 人员考勤管理模块设计……301
9.9 奖惩培训模块设计……304

第1章 Delphi 程序设计基础

本章介绍 Delphi 开发环境和编程语言基础，其中编程语言基础着重介绍了 Delphi 程序设计中常用的语法结构和语句等基本概念。如果读者在 Delphi 编程方面已经有了一定经验，可以有选择地阅读感兴趣的章节。

1.1 Delphi 开发环境

Delphi 提供了一个高度可视化的集成开发环境，也被称作 IDE（Integrated Development Enviroment）。可视化开发环境通常分为三个组成部分，即代码编辑器、调试器和窗体设计器，这三部分是同步工作的。当开发人员在窗体设计器中进行编辑时，Delphi 在后台自动为正在窗体中操作的组件生成代码。开发人员还可以自己在代码编辑器中加入代码来定义应用程序的行为，同时还可以在同一个代码编辑器中通过设置断点和监控点等来调试程序。

1.1.1 集成开发环境

启动 Delphi 后，系统弹出 Delphi 的集成开发环境，主要由主窗口、对象观察器、窗体设计器、代码编辑器、代码浏览器和对象浏览器几部分组成，如图 1.1 所示。

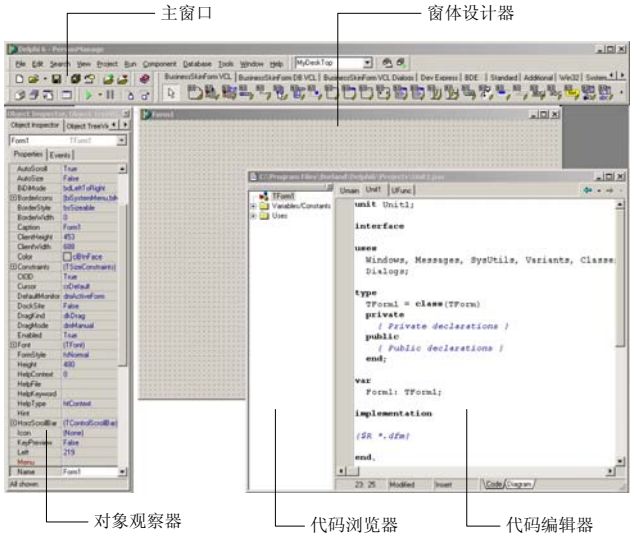


图 1.1 Delphi 集成开发环境

1．主窗口

主窗口是 Delphi 集成开发环境的控制中心，由菜单栏、工具栏及组件选项板三部分构成，如图 1.2 所示。

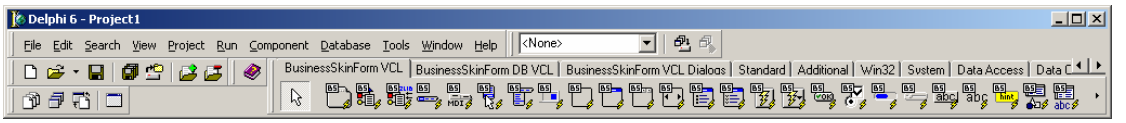


图 1.2　Delphi 主窗口

（1）系统菜单

提供 Delphi 集成开发环境中开发应用程序所需要的各种功能，表 1.1～1.8 所示详细介绍了各常用的菜单项的主要功能和操作说明。

表 1.1　　File 菜单

菜　单　项	操 作 说 明
New	用于打开常用的项目或文件等，如图 1.3 所示。如果选择其他新的项目，可以选择 New->Other 命令，系统将弹出“New Items”对话框供选择，如图 1.4 所示
Open	用于打开需要的项目、单元文件或文本文件等。Delphi 中所支持的主要文件类型包括：项目组文件（.bpg）、项目文件（.dpr/.bpg）、包文件（.dpk）、单元文件（.pas）、窗体文件（.dfm）、Pascal 项目文件（.pas）
Save	用于保存当前已打开的文件，文件名称不变
Save As	用于将当前打开的文件保存为另一个名称或存入另一个目录中
Save Project As	用于将当前打开项目文件另存为一个新文件
Save All	用于保存所有已打开的文件
Close	用于关闭当前活动窗口和与之对应的所有文件
Close All	用于关闭所有已打开的文件

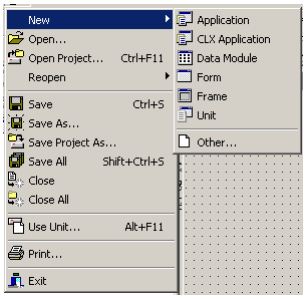


图 1.3　File->New 菜单项

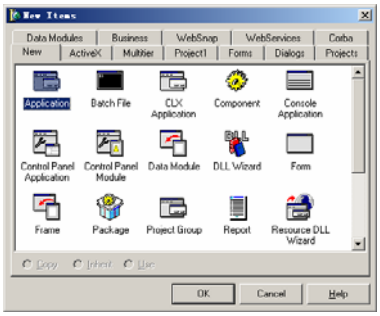


图 1.4　新项目选择对话框

表 1.2　　Edit 菜单

菜　单　项	操 作 说 明
Align	用于调整组件的位置，选择 Edit->Align 命令，系统将弹出“Alignment”对话框，如图 1.5 所示。对话框中属性描述如下： ● No Change　不改变组件的位置 ● Left sides　将各组件的左边界对齐 ● Centers　将各组件沿中线（水平或垂直方向）对齐 ● Right sides　将各组件的右边界对齐 ● Tops　将各组件沿顶端对齐 ● Bottom　将各组件沿底端对齐 ● Space equally　使各组件间距相等 ● Center in window　使组件沿窗体窗口中央对齐

续表

菜　单　项	操　作　说　明
Size	选择 Edit->Size 命令，系统将弹出“Size”大小设置对话框，如图 1.6 所示。对话框中属性描述如下： ● No change　不改变组件的尺寸 ● Shinft to smallest　使所有选中组件的尺寸等于其中尺寸最小的组件 ● Grow to largest　使所有选中组件的尺寸等于其中尺寸最大的组件 ● Width　指定选中组件的宽度 ● Height　指定选中组件的高度

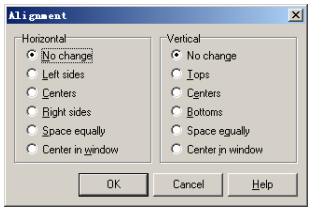


图 1.5　“Alignment”对话框

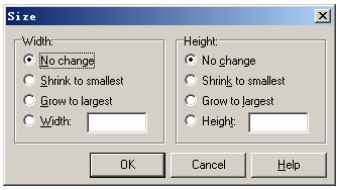


图 1.6　“Size”对话框

表 1.3　　View 菜单

菜　单　项	操　作　说　明
Unit	用于列出项目中所有的单元文件
Form	用于列出项目中所有的窗体文件
Project Manager	用于显示项目管理器窗口，其功能和使用将在后面详细介绍
Object Inspector	用于显示对象查看窗口，其功能和使用将在后面详细介绍
Browser	用于显示对象浏览窗口，其功能和使用将在后面详细介绍

表 1.4　　Project 菜单

菜　单　项	操　作　说　明
Import Type Library	选择 Project->Import Type Library 命令，系统将弹出“Import Type Library”对话框，如图 1.7 所示，对话框中显示了用户系统中已注册的类型库文件
Complie	用于编译当前工程中最近一次编译连接后修改的所有文件
Options	用于设置项目中的窗体、应用、编译和连接选项，并可以管理项目的目录。选择 Project->Options 命令，系统弹出“Projiect Options”对话框，如图 1.8 所示，对话框中属性描述如下： ● Forms　在 Forms 标签中可以对当前工程的窗体进行设置，如可以在 Main form 项中设置主窗体。如果希望程序一运行就创建某个窗体，则可以将该窗体的名称设置在 Auto-create forms 项中。否则，将其移动到右边的 Available form 项中 ● Application　在 Application 标签中可以指定工程的名称、帮助文件名和应用程序的图标等 ● Compiler　在 Compiler 标签中可以对编译过程进行设置。建议进行更严格的编译，如将 Runtimes errors 项中的所有选项都选中 ● Linker　在 Linker 标签中可以对链接过程进行设置。例如，如果选中“EXE and DLL options/Generate console application”项，则生成的应用程序为控制台程序 ● Directories/Conditionals　Directories/Conditionals 标签可以对工程在编译、链接过程中的目录和条件等进行设置 ● Version Info　在 Version Info 标签中可以控制工程的版本信息 ● Packages　在 Packages 标签中可以新增、删除和编辑组件库的设置信息

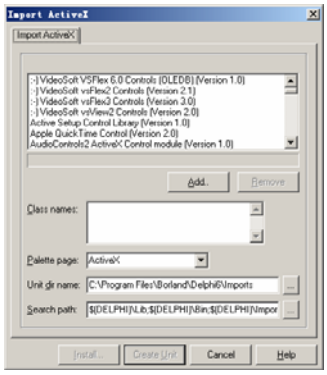


图 1.7 “Import Type Library”对话框

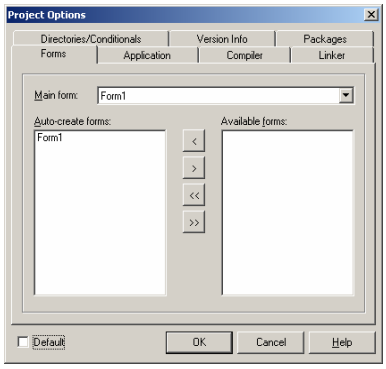


图 1.8 “Project option”对话框

表 1.5　　Run 菜单

菜　单　项	操 作 说 明
Run	用于提供编译和调试应用程序
Step Over	用于单步调试
Program Reset	用于结束执行中的程序，并释放其所占的内存空间

表 1.6　　Component 菜单

菜　单　项	操 作 说 明
New Component	用于为一个组件创建一个单元文件
Install Compoment	可以在一个存在的包或新的包中创建文件

表 1.7　　Database 菜单

菜　单　项	操 作 说 明
Explore	用于打开“SQL Explore”，创建、查看或编辑数据和 BDE 别名，如图 1.9 所示

表 1.8　　Tool 菜单

菜　单　项	操 作 说 明
Enviroment Options…	用于设置版面、资源库的路径和组件面板的显示内容，如图 1.10 所示

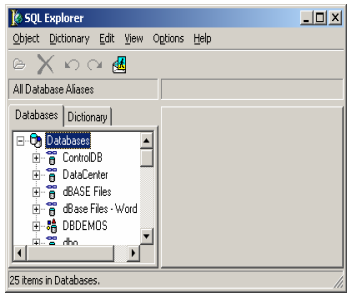


图 1.9 SQL Explorer

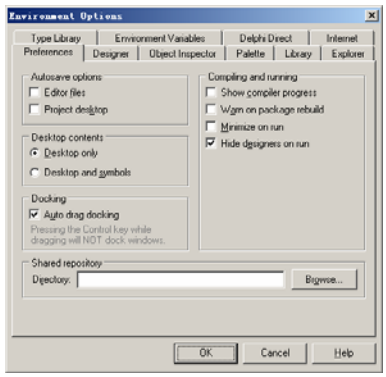


图 1.10 “Enviroment Options”对话框

（2）快捷工具

为了提高效率提供一些快速执行某些常用菜单命令的快捷键。表现形式是一些带图标的

按钮，这些按钮上的图标直观地显示出该按钮所实现的功能，如图 1.11 所示。



图 1.11　快捷键工具

（3）组件面板

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。所有的组件根据其功能划分为多个类，每一种类别的组件放置在特定的一个组件面板页上，组件分为可视化组件和不可视化组件，如图 1.12 所示。



图 1.12　组件面板窗口

若组件在设计阶段窗口中所生成的组件和在执行阶段所看到组件的外观是一致的，则将此类组件称为“可视化组件”，例如 Standard、Win32、Additional 和 System 等选项页。

某些组件在设计阶段会以该组件图标显示在窗体上，可以改变该组件属性和加上事件处理程序，但无法改变对象的大小，此类组件在执行阶段会隐藏在屏幕后执行，在窗口上无法看到该组件的图标，例如 Dailog 和 Servers 等选项页。

如果对系统提供的默认组件面板不够满意，可以选择菜单 Component 中的 configure Palette 命令，系统将弹出“Palette Properties”对话框，如图 1.13 所示，如果加入新的选项卡，可以单击“Add”按钮，在弹出的对话框中输入新选项卡的名称，如图 1.14 所示，单击“OK”按钮，新建的选项卡就被加入到 Page 列表中。如果隐藏组件，可以单击“Hide”按钮。如果要改变选项卡的位置，可以通过选择“Move Up”或“Move Down”按钮实现。

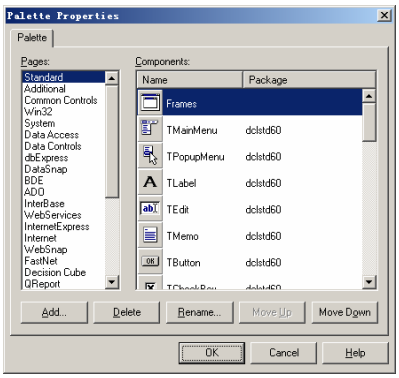


图 1.13　Palette Properties

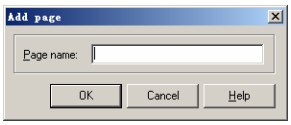


图 1.14　Add Page

2．对象观察器

对象观察器是 Delphi 面向对象可视化编程的基本工具，如图 1.15 所示。利用对象观察器，可以修改窗体或组件的属性，或者使它们能够响应不同的事件。对象观察器窗口分为属性（Properties）和事件（Events）两页。属性是一些数据，如高度、颜色和字体等，它们决定了组件在屏幕上的外观。事件则是一种消息处理机制，它能够捕捉某种情况的发生并作出反应，像鼠标单击和窗口重画就是两种典型的事件。切换时只需在窗口上部单击所需页的标签即可。在窗体设计器中选中哪个组件，对象观察器中就显示这个组件的属性和事件。

3．窗体设计器

窗体设计器主要用于设计程序软件的用户界面，如图 1.16 所示。在默认情况下，每次启

动 Delphi 时，系统会自动创建一个新的项目并显示一个空白的窗体。程序设计开始，从组件面板上选择一个组件并把它放到新创建的窗体上，可以利用鼠标调整组件在窗体设计器上的位置和大小，也可以利用对象观察器和代码编辑器来控制组件的外观和行为。

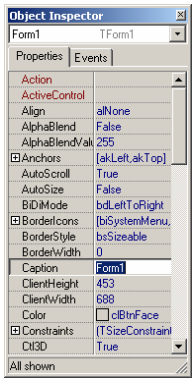


图 1.15　对象观察器

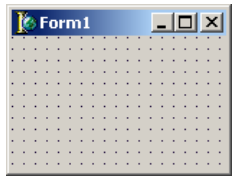


图 1.16　窗体设计器

窗体设计时，最好给新创建的窗体起一个具有一定意义的名字，这样不仅可以增加程序的可读性，而且便于日后的编译和调试。

4．代码编辑器

在 Delphi 中，代码编辑工作是在代码编辑窗口完成的，如图 1.17 所示。代码编辑器是一个高性能的编辑器，它提供了许多强大的功能。当创建一个新的窗体后，代码编辑器中就会自动生成一个相应的新单元，单元的内容对应于该窗体的源代码文件（.pas）。同时系统会自动将这个新单元按照产生的顺序，依次添加到代码编辑器顶部的标签中。

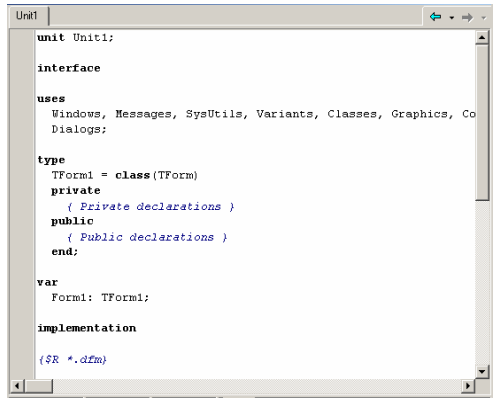


图 1.17　代码编辑窗口

在编写代码时，Delphi 还提供了对象成员自动感知功能，如图 1.18 所示。它可以自动显示与类有关的变量和成员函数，当输入“.”符号后，系统会自动完成变量和成员函数的检索。

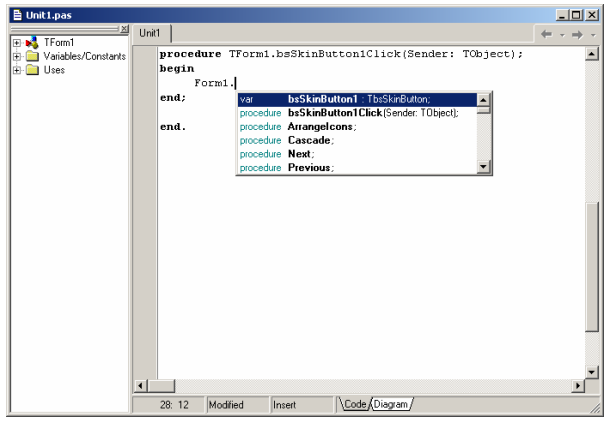


图 1.18　对象自动感知功能

在代码编辑窗口输入函数或过程名的左括号“(”后，系统将自动显示函数或过程的参数名和类型，如图 1.19 所示。

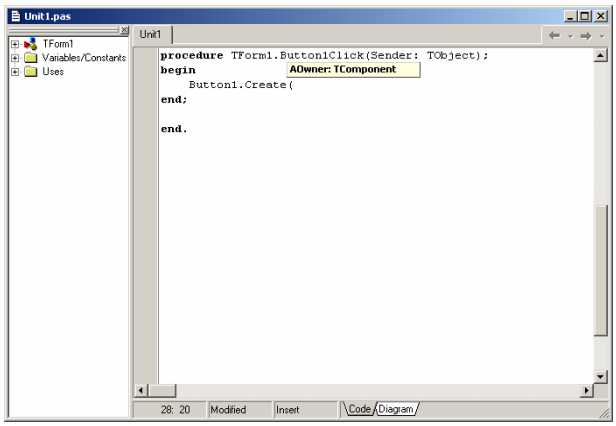


图 1.19　函数或过程参数名和类型的自动显示

如果想同时看到多个代码编辑器窗口，可以通过选择菜单 View 中的 New Edit Window 命令来实现。在代码编辑器窗口中单击右键，随之弹出的右键快捷菜单中，有许多与代码编辑有关的命令，介绍如下。

（1）New Editor Window 命令

该命令用于启动一个新的编辑窗口，通过窗口间代码的参考和对照，使代码编写更加直观。

（2）Complete class at cursor 命令

在新定义的类结构内部，选择该命令，系统将自动完成类定义的其他代码生成。举例说明如下：

```
type
  Person = class(TObject)
  public
    constructor Create(na:string);
    function Name:string;
  private
    pname:string;
  end;
```

调用命令后，代码编辑器中的代码发生改变：

```
type
  Person = class(TObject)
  public
    constructor Create(na:string);
    function Name:string;
  private
    pname:string;    <私有变量定义>
  end;
constructor Person.Create(na: string);
begin
```

```
    <事件代码>
    end;
    function Person.Name: string;
    begin
    <函数代码>
    end;
```

（3）Toggle Bookmarks 命令和 Goto Bookmarks 命令

在代码编辑器中使用 Toggle Bookmarks 命令可以定义或取消书签，使用 Goto Bookmarks 命令可以转到某个定义好的书签处。

（4）Properties 命令

该命令用于设置代码编辑器的属性，包括属性、代码格式、代码字体以及语法颜色等。选中 Properties 命令，系统将弹出“Editor Properties”对话框，如图 1.20 所示。通过这些属性的设置可以使代码编辑器更符合开发人员的编辑习惯。

5．代码浏览器

代码浏览器以一种树状视图的方式显示了列在代码编辑器中的单元文件，如图 1.21 所示。它显示了类、变量、使用的程序单元和事件处理过程，通过它可快速浏览和定位在代码编辑器窗口的内容。在代码浏览器中右键单击一个节点即可以看到该节点的可用选项。也可以通过修改主菜单下的“Environment Options”对话框中的 Explorer 页来控制代码浏览器的操作，如排序和过滤等。

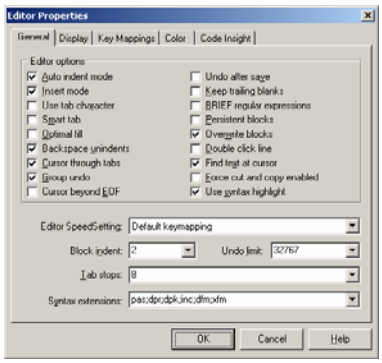


图 1.20　“Editor Properties”对话框

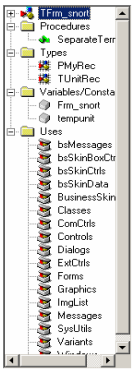


图 1.21　代码浏览窗口

6．对象浏览器

对象浏览器以一种树状视图的形式显示了窗体中的所有组件和对象之间的关系，如父子关系、属性和组件的关系以及属性和属性之间的关系。可以通过选择主菜单 View 中的 Object TreeView 命令，打开对象浏览器窗口，如图 1.22 所示。

对象浏览器对应于代码浏览器和窗体设计器。因此，当对其中任意工具的单元做了修改，其他两种工具的单元也会做相应的修改。

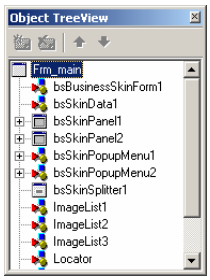


图 1.22　对象浏览窗口

1.1.2　Delphi 工程管理

选择菜单 View 中的 Project Manager 命令，系统将弹出项目管理器窗口，如图 1.23 所示。

在Delphi中项目是构成应用程序或动态连接库的所有文件（窗体、单元、资源等）的集合。项目管理器显示了与项目相关的窗体和单元文件。项目管理器除了负责监视项目状态以外，还起到管理项目组的作用。

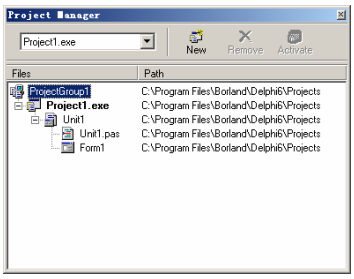


图 1.23　项目管理器窗口

1．增加和删除文件

要向项目组中添加项目，可以在项目管理器窗口中单击鼠标右键选择 Add New Project 或 Add Exit Project 命令，完成建立一个新项目或将一个已存在的项目添加到项目组中。

要删除项目组中的某个项目，选中目标项目后，单击鼠标右键选择 Remove File 命令完成操作。

要向项目中添加单元，选中目标项目后，单击鼠标右键选择 Add 命令，完成向目标项目添加单元操作。

要删除项目中的一个单元，选中目标单元后，单击鼠标右键选择 Remove From Project 命令，完成删除项目中一个单元操作。

2．项目编译

选择需要编译的项目后，单击鼠标右键，选择 Build 命令可实现编译并连接该项目中的所有文件，选择 Compilecom 命令则只编译而不连接该项目。

选择 Build Sooner 命令使目标项目的编译顺序提前一个项目，而选择 Build Later 命令将使目标项目的编译顺序向后移动一个项目。

3．调试

当程序编译、链接通过以后就可以运行，运行结果如果只是在一些地方与当初的预想不同，可以利用集成调试器来解决问题，找出可能存在逻辑错误的地方。通常，可以在可能存在问题的代码处设置一些“断点”，程序执行到断点处就自动暂停，并转到代码编辑器中设置了断点的代码处。此时，可以通过“单步执行”将语句一条一条地执行下去。在执行的过程中，同时可以查看一些变量的数值，也可以查看堆栈、CPU、汇编代码和当前运行的线程等情况，然后找出代码错误的地方并加以改正。

（1）设置/去除断点

断点的设置可以在程序调试前进行，也可以在程序调试期间进行。设置/去除断点最简单的方法是在代码编辑器中，利用鼠标点击代码行左边的灰色竖条处。

（2）单步执行

当程序运行到断点处暂停后，可以按 F8 键或通过主菜单 Run 中的 Step Over 命令单步执行下去。如果遇到过程或函数调用，程序将一次执行完成。

按下 F7 键或选择主菜单 Run 中的 Trace Into 命令，可以在遇到过程或函数调用的时候，进入过程或函数的内部。

此外，选择主菜单 Run 中的 Run to Cursor 命令可以使程序执行到当前光标所在的行后暂停，这在程序调试的过程中也是非常有用的。

（3）中止运行

在程序调试期间，可能会由于代码的问题，程序在运行时停在了某个地方而不继续下去。这时可以选择主菜单 Run 中的 Program Reset 命令来终止程序的运行。

1.1.3 程序文件

程序文件包括工程文件、单元文件、窗体文件和资源文件等。每个 Delphi 项目工程都由若干个相关的文件构成。一些文件是在编辑时创建的，如窗体的定义文件。另外一些文件则是在编译时生成。为了有效地管理 Delphi 项目工程，下面将介绍项目中各种文件的用途和结构。

1．项目文件（.dpr）

项目文件是 Delphi 在编辑时自动创建的，文件后缀名为.dpr。项目编译执行后，会生成.exe 或.dll 文件，前者为应用程序，后者为动态链接库。通过选择菜单 Project 中的 View Source 命令，可以查看项目文件的源代码。系统缺省生成的项目文件源代码一般如下：

```
program Project1;    //项目类型和名称
uses    //引用单元
  Forms,    //窗体类
  Unit1 in 'Unit1.pas' {Form1},    //Unit1 单元说明
{$R *.res}    //引用资源文件
begin    //程序执行部分
  Application.Initialize;    //初始化
  Application.CreateForm(TForm1, Form1);    //创建窗体
  Application.Run;    //运行，即显示窗体 Form1
end.    //执行部分结束
```

其中 uses 语句告诉编译器当前项目中使用了哪些单元，如果包含其他自动创建的窗体，将以同样的形式列在 uses 语句中。

uses 语句可以放在接口部分（Interface 之后），也可放在实现部分（Implementation 之后），但是 uses 语句必须出现在它所有子程序、数据类型或变量被使用之前。

{$R *.res}用于引用项目的资源文件。告诉编译器去连接一个资源文件，且资源文件名与项目文件相同，但扩展名为.res。项目的资源文件中包含了程序图标和版本信息。

begin...end 之间的语句是应用程序要执行的主代码块。

项目文件是主窗体以及其他自动创建的窗体实例化的地方。一般不需要编辑项目文件，除非要执行程序初始化例程、显示启动画面或执行其他必须在程序启动时运行的例程。

2．单元文件（.pas）

使用单元可以提高代码在各项目之间的结构化和重用性。在创建窗体、数据模块、组件或者程序模块时，会产生相应的单元文件。单元文件编译后，将生成一个二进制的文件，扩展名为.dcu。下面是一个窗体的单元文件源代码：

```
//单元开始
unit Unit1;    //单元标题
//界面部分开始
interface

uses    //引用的单元文件
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, bsSkinCtrls;
```

```
Type    //系统自动添加的类型声明
  TForm1 = class(TForm)
    Edit1: TEdit;    //文本编辑框对象
    Button1: TButton;    //按钮对象
    procedure Button1Click(Sender: TObject);    //Button1 OnClick 事件处理过程
  private
    { Private declarations }      //该单元私有变量和方法声明
  public
    { Public declarations }       //该单元公用变量和方法声明
  end;

var    //变量声明
  Form1: TForm1;
//界面部分结束

//实现部分开始
Implementation
{$R *.dfm}    //指明了对应的窗体文件名称与单元文件名称相同

procedure TForm1.Button1Click(Sender: TObject);
  begin    //程序开始
    Edit1.Text:='单元文件';     //赋值给文本编辑框
  end;    //程序结束
//实现部分结束

end.
//单元结束
```

单元文件中包含有：单元标题、接口部分和实现部分。单元中可以包含事件处理程序、过程及相关的函数。单元通常由以下 5 大部分组成。

（1）单元标题

用于设置单元的名称，该名称在 uses 子句中引用。但名称必须惟一，即设定单元名字不能被重复使用。

已设定的单元标题不应随意进行修改，否则会出现编译错误。

（2）界面部分

位于标题头后，以关键字 interface 开始，implementation 前的部分。界面部分说明公共的常量、类型、变量、过程及函数等。对过程和函数，只列出说明部分，过程体或函数体在实现部分。

通常 uses 子句中常引用一些标准单元，如 Windows 单元、SysUtils 单元、Classes 单元、Graphics 单元、Forms 单元等。

（3）实现部分

从关键字 implementation 开始直到本单元末尾或初始化部分之前。主要用于定义在界面

部分说明的全部公共过程体和函数体。也可说明局部的常量、类型、变量、过程或函数。

其中的{$R *.dfm}是编译指令，指示编译器在编译时要连接窗体。

（4）初始化部分

为可选的部分，以关键字 initialization 开始，位于实现部分之后，主要是对本单元中的一些数据进行必要的初始化，如给变量赋初值、为实现部分分配资源等。

（5）结束部分

为可选的部分，若单元具有初始化部分就必须具有结束部分。以关键字 finalization 开始，直到单元结束，主要为结束单元的语句。在初始化部分获得的任何资源（内存、文件等）在结束部分都要释放。结束部分的执行顺序与初始化部分的执行顺序是正好相反的。

3．窗体文件（.dfm）

窗体文件存储了窗体的二进制信息，文件后缀名为.dfm。当创建一个窗体时，Delphi 会同时创建一个单元文件和一个窗体文件。窗体文件中保存了窗体上所有组件的非缺省属性值。用 Delphi 的代码编辑器打开后缀为.dfm 的文件，系统以文本的形式显示文件，可以修改窗体文件中的内容。选择右键菜单中的 View As Text 命令，可以看到该窗体文件的文本形式。

4．资源文件（.res）

Delphi 的每个项目文件都有一个资源文件，扩展名为.res 的资源文件是 Delphi 自动创建的，包括应用程序的图标、应用程序版本信息及其他信息。

5．使用帮助系统

Delphi 提供了强大的帮助功能。

下面列举了使用帮助系统的一些小知识，如果读者自己也总结了一些经验，可以对下面的各项进一步完善。

① 当遇见问题时，可以尝试按下 F1 键，Delphi 会根据当前的焦点而启动相应的帮助。Delphi 的联机帮助具有智能判断的功能。如果当前位置在对象编辑器中的某个属性中时，按下 F1 键后则会自动显示该属性的帮助内容；如果当前显示的是 Delphi 系统的某个对话框，按下 F1 键后则会显示该对话框的有关帮助内容；如果弹出了某个菜单，这时按下 F1 键，则会显示该菜单命令的有关帮助内容。

② 如果需要浏览系统的一些帮助内容，比如说对象 Pascal 语言，可以通过选择菜单 Help 中的 Delphi Help 命令打开“帮助主题：Delphi Help”对话框，在目录标签中展开相关的条目，进行系统学习。

③ 在使用 Delphi 编程的过程中，常常要使用到 Windows API 函数，然而通过选择菜单 Help 中的 Delphi Help 打开的“帮助主题：Delphi Help”对话框中，在“索引”标签中可能查不到有关的帮助信息，比如 Windows API 函数 MessageBeep 的有关帮助信息。一种解决的办法就是在代码编辑器中输入 Windows API 函数的名称，然后将光标定位到该名称中，接着按下 F1 键。另一种办法就是直接打开帮助文件 Win32.hlp，在其中的“索引”标签中查找。可以在 Delphi 安装光盘上找到帮助文件 Win32.hlp。

④ 在帮助文件打开的情况下，如果要浏览相关内容，可以按下工具栏中的“〈〈”或“〉〉”按钮，通常这样是在同一个主题中进行浏览。有些帮助文件中有 Topic group，通过它可以新打开一个主题列表窗口，这样可以非常方便地浏览一个完整主题的帮助内容。此外，通过一些 See else 热链接，也可以查看相关的帮助内容。