计算机应用二次开发丛书

Delphi 数据库开发

清宏计算机工作室 编著

机械工业出版社
China Machine Press

计算机应用二次开发丛书

Delphi 数据库开发

清宏计算机工作室 编著

机 械 工 业 出 版 社

本书向读者介绍了可视化软件开发工具——Borland Delphi 4.0（以下简称 Delphi 4.0）在数据库开发方面的应用。

全书总共分为 10 章。第 1 到第 3 章是 Delphi 的基础入门，带领读者浏览 Delphi 的可视化开发环境、介绍 Delphi 的基本开发功能等，使读者熟悉 Delphi 的开发环境。第 4 章直截了当地向读者介绍 SQL 语言，给以后的数据库开发打下基础。第 5 到 9 章由浅入深地向读者介绍 Delphi 的数据库编程机制以及它所提供的数据库组件。最后一章向读者介绍了一个应用 Delphi 数据库编程的实例，帮助读者灵活运用在前面各章中学到的知识。

本书结合生动的例子详细地介绍了 Delphi 4.0 的数据库开发机制。全书图文并茂，既适合初学者用于入门学习，也适合中级读者作为 Delphi 数据库开发的进阶读物。

图书在版编目(CIP)数据

Delphi 数据库开发/清宏计算机工作室编著. —北京：机械工业出版社，2000. 1

(计算机应用二次开发丛书)

ISBN 7-111-07781-4

I. D… II. 清… III. Delphi 语言-数据库系统-软件开发 IV. TP312

中国版本图书馆 CIP 数据核字（1999）第 74731 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：边 萌 吴天培 封面设计：姚 毅

责任印制：路 琳

北京市密云县印刷厂印刷·新华书店北京发行所发行

2000 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16· 24 印张· 577 千字

0 001－5 000 册

定价：38.00 元

凡购本书，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

丛书前言

随着计算机技术的飞速发展，计算机应用及开发迅速得到推广，多种编程开发语言不断为人们掌握和使用。当前，有一大批从事计算机应用开发者对于某一种或某些开发工具的掌握已经达到了一定水准，但他们要进一步提高开发能力，就迫切地需要高层次的开发参考书。为了满足这些计算机开发人员的需求，进一步提高他们的技术水平，推动我国计算机事业的发展，我们特地组织编写了这套《计算机应用二次开发丛书》，适于中高级计算机开发人员学习。

本套丛书主要以常用的开发工具为基础，吸收成功的开发经验和技巧，结合开发原理，向读者介绍了如何有效地利用这些基本开发工具，开发实用性较强的应用程序。使读者不仅知道该怎样做，还知道为什么这样做，达到触类旁通、举一反三的目的。

本丛书重点明确、图文并茂、实用性强，每章除了讲解开发原理之外，还附有大量实例，这些实例绝大多数都是作者在开发工作中的实际成果或其中的一部分，因此，适用性非常强。

由于本丛书的读者定位比较高，适于已经具备一定开发能力的读者阅读。因此在使用这套书之前，要求读者对相应的开发工具已经初步了解。

在这套书中，我们首先推出了《C++ Builder 数据库开发》、《C++ Builder 多媒体开发》、《Delphi 数据库开发》和《AutoCAD 2000 二次开发》4 本书。此后，我们还会相继推出其他图书，以满足广大计算机开发人员的需求。

为了把这套丛书编写得更好，我们真诚地希望广大读者提出宝贵意见和建议。

编 者 的 话

Microsoft Windows 是基于图形界面的多任务、多窗口操作系统。自 1983 年 Windows 问世以来，日趋完善，Windows 标准已经不断被广大用户接受和认可，从而成为了计算机软件设计的标准。

早期的 Windows，没有很好地应用程序开发工具，那时的程序员形容开发一个 Windows 程序，就像经过一场可怕的梦魇。经过多年的实践和探索，很多程序员发现，随着面向对象技术的进步，开发一般的 Windows 应用程序后很多的工作是重复的，于是在很多大软件公司里形成了自己的开发系统，随着这些开发系统的不断进步和完善，在 20 世纪 90 年代中期，一些公司开始将这些称为“代码生成器”的工具推向市场。当用户第一次看到这些可视化开发工具时，心中都充满了一种期待已久的惊喜。软件开发已经成为了大多数计算机工作者力所能及的事情。

在众多的可视化开发工具中，当年的 Borland 公司推出的基于 Object PASCAL 的 Delphi 不愧是一朵奇葩。笔者从 Delphi 1.0 推出时就开始使用 Delphi 进行软件开发，对 Delphi 的功能机制和特点有较深入的了解。当年的 Delphi 以它的编译速度以及高效的生成代码，被称为当时的 VB 杀手。基于结构化的原码以及大量的原码公开，使得 Delphi 不仅为使用者提供了快捷的开发环境，而且也给了使用者发挥自己创造力的空间。这也是 Delphi 多年来一直受到广大软件开发者欢迎的原因。

Delphi 除了拥有其他可视化开发工具的优点外，还提供了很好的数据库支持。Delphi 配有 Database Engine，可通过 SQL Links、ODBC 访问多种数据库，而且还提供了强大的开发 Client/Server 模式的数据库应用程序的能力。在数据库前端应用程序的开发中，Delphi 使用了 Multi Session 和 Thread Safe 的数据库引擎、数据库过滤器(Filter)、Visual Query Build、查询引擎和最新的数据更新模式等，最大限度地为 Delphi 的用户提供了方便。

本书通过实例介绍 Delphi 4.0 的控件使用和数据库开发机制，将作者多年来应用 Delphi 开发数据库应用程序的经验融入其中，希望能够为读者在使用 Delphi 4.0 的过程中提供一点帮助。读者的收获就是我们的成功。

由于水平有限，书中难免存在错误与不足，欢迎广大读者批评指教。

编者

第 1 章　Delphi 快 速 入 门

本章介绍 Delphi 4.0 的一些基本概念和基本程序框架，读者将对 Delphi 有一个大体的了解。“Delphi”在希腊语中兼有神圣和智慧的意义，是一个城市的名字。由于 Delphi 采用了内置优化的高速编译器，使得编译速度快得难以相信。用 Microsoft Visual C++编写过程序的程序员都应该对编译速度的重要性有很深的记忆。Visual C++功能很强大，但调试程序时编译的时间过长，会使多数人感到厌倦，Delphi 4.0 在编译时几乎感觉不到等待。Delphi 的编译器采用了事件编译和选择链接技术，生成的可执行文件冗余更少，运行速度更快。另外，由于 Delphi 生成的可执行文件不需要动态连接库的支持，可以直接交付使用，这会使 Delphi 的程序员既拥有 VB 编程的方便快捷，又有用 VC 编程的那种专业的感觉。

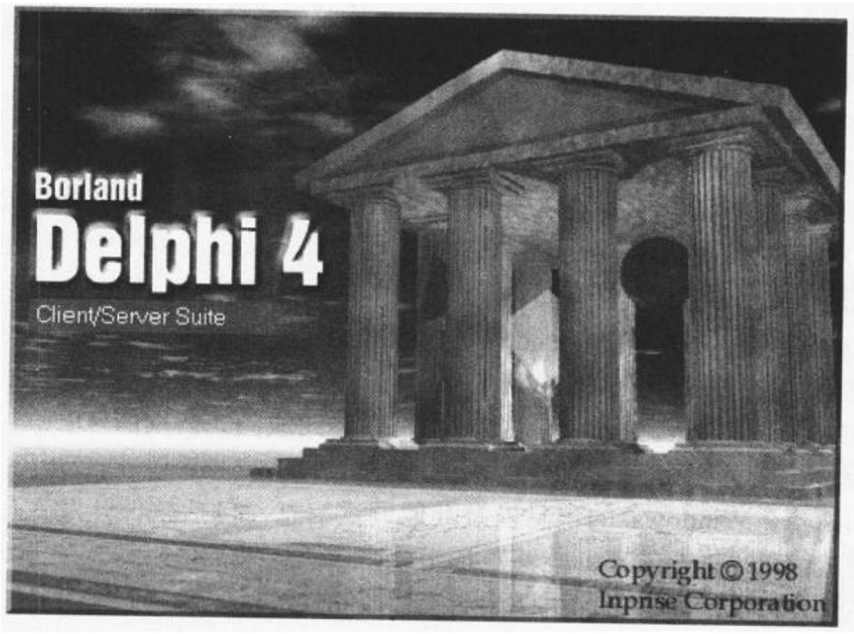

图 1-1　Delphi 的欢迎界面

1.1　Delphi 4.0 简介

有人说：“真正的程序员用 C，聪明的程序员用 Delphi”。图 1-1 所示为 Delphi 4.0 的欢迎界面。

Delphi 采用最先进的面向对象技术，并且继承了 Pascal 语言的所有精华，语法上基本不变，使学过 Pascal 或 C 的程序员很容易掌握。即使从没有接触过类似高级语言的读者，也很容易接受这种十分自然的语言。如果编写简单的应用程序，甚至可以不涉及语言，只需

填表格就可以完成。事实上，Delphi 的语言非常简单，编程者不需要在这方面浪费很大的精力。

Windows 是完全不同于 DOS 的具有图形界面的操作系统，图形用户界面（GUI）已经使微型计算机工业产生了全面的变革。现在用户所看到的不再是 DOS 用户们过去经常面对的“C:>”提示符，而是充满了图表以及使用鼠标与菜单项的桌面景象。Windows 应用程序一般都有一个前后一致的用户界面，这样用户就不必过多考虑除软件功能之外的操作问题了。经过多年来的不断改进，现在的图形用户界面与最早的图形用户界面已经有很大的不同了，它在美观简洁的基础上，不断赋予用户对软件更大的决定权，让用户能够真正成为软件的控制者，而不仅仅是软件的使用者。

GUI 对于程序员是难以忍受的，但对于计算机操作的初学者却是非常亲切的。于是软件用户都希望程序支持 Windows 图形用户界面，开发者就考虑怎样开发 Windows 应用程序。由于很长一段时间没有这种很好的很高效的开发工具，程序员只有用 Windows SDK 和 C 语言来开发，这使开发 Windows 应用软件比开发 DOS 应用软件困难得多。早期的 Windows 程序员形容开发一个 Windows 软件，就象一场噩梦。Delphi 的产生是一场革命，复杂的 Windows 应用程序开发时间被缩短到原来的零头大小，编译错误也将被及时指出，编程变成一项艺术和有意思的工作。

Delphi 4.0 是一个完全 32 位的 Windows 开发工具。是新一代面向对象的快速应用软件开发工具（RAD，Rapid Application Development），复杂的 Windows API（Application Programming Interface）函数都被封装成一个个组件（Component，可视编程的最小单元与工具），并通过对象继承（Object Inheritance）的方式把 Windows 程序的结构也封装起来，剩下的工作就是把组件添加到空白的窗体上，用艺术的眼光设计自己的程序。

倍受瞩目的 Delphi 4.0 是 Borland 更名为 Inprise 后推出的一个具有战略意义的产品。Delphi 4.0 增加了“模块窗口”（Modal Form）。模块窗口能够显示当前所有映射到应用程序地址空间的模块，包括应用程序自身、应用程序显式或隐式调用的 DLL 以及操作系统调用的 DLL，模块窗口可帮助您优化程序结构。此外，Delphi 4.0 还能自动记录在调试过程遇到的事件，如断点、Windows 的消息以及其他调试信息，包括软件开发人员指定要记录在调试器中的调试信息。Delphi 4.0 能够监视指针错误，如果某个指针试图非法访问内存的某个地址，程序就会暂时中断运行，由调试器接管控制权，就好像遇到断点一样。Code Insight 是一组代码自动化功能的总称，它是一个带有一定人工智能的功能，能够根据用户对代码的不完全描述，快速生成代码，从而减少语法错误并提高编程效率。从 Delphi 3.0 开始就已经有了 Code Insight 功能，但 Delphi 4.0 又作了很多改进。由于 Delphi 4.0 是一个完全面向对象的编程工具，编程过程中经常要声明和实现类（Class），而类不同于一般的数据类型，它具有特殊的语法，很多初学者往往对类的使用缺少经验，即使是 Delphi 的高手，也会对过分严谨和古板的 Object Pascal 语法感到枯燥乏味。“类自动完成”向导，可以快速生成有关类的代码。在设计期编写代码时，Delphi 4.0 在后台运行编译器，这样用户无须显式地编译代码就能看到每个符号实际是怎样存储的，能够及时地消除错误。在设计期编写代码时，经常要重复输入诸如 if...then...else 或 for...do 等基本 Object Pascal 语句结构，Delphi 4.0 把一些常用的代码结构预先做成模板，只要选择一个模板，Delphi 就会自动把该模板的代码插入

到代码编辑器中。代码模板也可以自定义，键入一个类名或对象名，再键入一个句号（小圆点），Delphi 就会自动弹出一个列表框，列出该类的所有属性、方法和事件，不用查阅帮助或手册，不用担心写错成员的名称。键入例程名或方法名，再键入左圆括号，Delphi 就自动显示该例程或方法的所有参数名称及其数据类型，供程序员参考，这样就能保证参数的个数、顺序和数据类型总是正确的，而且免去了程序员在编写代码时需要频繁查阅“帮助”的麻烦。

Delphi 4.0 完全支持 ActiveX，可以很方便地创建、注册、安装、发布和使用 ActiveX 组件、ActiveForm 和 OLE 自动化对象，这应当归功于“对象接口”技术和 DAX 技术。

Delphi 4.0 内建了对 COM（Component Object Modal）的支持，由于 COM 对象具有语言和平台无关性，用 Delphi 4.0 创建的对象可以与用 Visual Basic、Java、C＋＋及其他语言实现的对象交互。在多层 Client/Server 环境中，COM 对象可以封装商业规则，为分布式的客户提供服务。此外，Delphi 4.0 用“包”（Package）技术减少冗余的 VCL 框架代码，从而使应用程序更精巧，更适合于在 Internet/Intranet 上传输和发布。Delphi 4.0 还取消了原来的包编辑器，改用项目管理器对包项目统一进行管理。

Delphi 4.0 包含了一组 Internet 组件，可以很方便地实现 WinSock 编程，访问 HTTP、UDP、FTP、SMTP、POP3 和 NNTP 等服务。WebBridge 简化了对 NSAPI 和 ISAPI 的访问，WebModules 封装了 Web 服务器的应用逻辑，WebDispatcher 实现了 HTTP 请求消息和 HTTP 响应消息的动态调度。运用 ActiveForm 技术和 RemoteDataBroker 技术可以把分布式的 Client/Server 结构扩展到 Internet/Intranet 上。

Delphi 4.0 的开放体系结构支持多个数据库引擎，任何一个数据库引擎都能与 Delphi 的数据显示组件如 TDBGrid、TDBEdit 协调工作。

Delphi 4.0 还可以访问 Oracle 的 BLO（Binary Large Object）型字段，实际的数据存储在外部文件中。Delphi 4.0 支持 Oracle 8 的抽象数据类型（ADT），允许自己定义数据类型，自定义的数据类型可以基于真实的类型，也可以基于已定义的抽象类型。

1.2　Delphi 基本概念

1.2.1　Delphi 的基本形式

一个 Delphi 程序首先是应用程序框架，只要拥有应用程序的框架，用户设计的应用程序就可以按照用户设计的思想运行了。当然，如果您想丰富程序的功能，只需在应用程序中加入内容即可。

应用程序的框架是一个空白窗口，在空白窗口的背后，应用程序的框架正在等待用户的修改和丰富。由于您并未告诉它接收到用户输入后作何反应，窗口除了响应 Windows 的基本操作（移动、缩放等）外，它只是接受用户的输入，然后再忽略。Delphi 把 Windows 编程的回调、句柄处理等反复过程都放在一个不可见的 Romulam 覆盖物下面，这样您可以不为它们所困扰，轻松从容地对可视组件进行编程。

1.2.2　面向对象编程的概念

面向对象的程序（Object-Oriented Programming，简记为 OOP）的最根本的目的就是使程序员更好的理解和管理庞大而复杂的程序，它在结构化程序设计的基础上完成进一步的抽

象。这种在设计方法上更高层次的抽象正是为了适应目前软件开发的特点。如图 1-2 所示为 Delphi 4.0 的对象继承形式。

面向对象的程序设计是 Delphi 诞生的基础。OOP 立意于创建软件重用代码，具备更好地模拟现实世界环境的能力，这使它被公认为是自上而下编程的优胜者。它通过给程序中加入扩展语句，把函数“封装”进 Windows 编程所必需的“对象”中。面向对象的编程语言使得复杂的工作条理清晰、编写容易。说它是一场革命，不是对对象本身而言，而是对它们处理工作的能力而言。

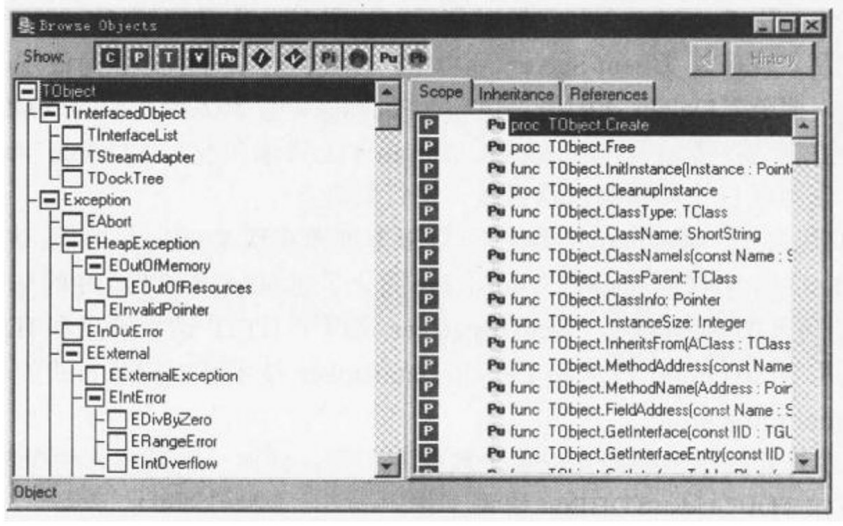

图 1-2　Delphi 4.0 的对象继承树

对象并不与传统程序设计和编程方法兼容，只是部分面向对象反而会使情形更糟。除非整个开发环境都是面向对象的，否则对象产生的好处还没有带来的麻烦多。而 Delphi 是完全面向对象的，这就使得 Delphi 成为一种触手可及的促进代码重用的开发工具，从而具有强大的吸引力。

1.3　第一个简单的 Delphi 程序

1.3.1　Hello World!

作为第一个 Delphi 程序，我们重在向读者介绍 Delphi 程序的概况，像所有的编程书籍都有一个“Hello World”工程一样，这个例子也是一个可视化编程方法的“Hello World”程序，产生一个标准的 Windows 风格的矩形主窗体（Form）。主窗体包含一个按钮（Button），单击（Click）这个按钮就关闭主窗口，同时也就关闭了整个应用程序。这个主窗口可以像任何一个标准的 Windows 窗口一样，由用户改变其大小和在屏幕中的位置。

1.3.2　添加组件并设置属性

这个例子中只包含一个按钮组件，选中 Delphi 主界面上组件面板上的“Standard”选项卡（鼠标单击），就会出现一些组件允许用户选择。选择其中的按钮组件（上面写有“OK”

字样的组件）。

Standard 页存放了所有的 GUI 标准组件，也是最常用的组件。我们向程序中添加的组件都从组件面板中选取。将鼠标移入 Form1 窗体，在我们要加入按钮（Button）组件的位置上单击鼠标左键，这样，一个按钮就被加入 Form1 窗体中了。再通过拖放按钮周围的手柄（或者称为标志点）来设定按钮的合适大小和位置，选中的组件四周都有改变大小的手柄，我们将鼠标移动到手柄上，按下左键，再拖动鼠标，就可以改变按钮的大小，可以看到一个动态显示当前拖动到的大小区域的矩形；松开鼠标左键，按钮的大小就变为设置的大小（矩形的大小）。

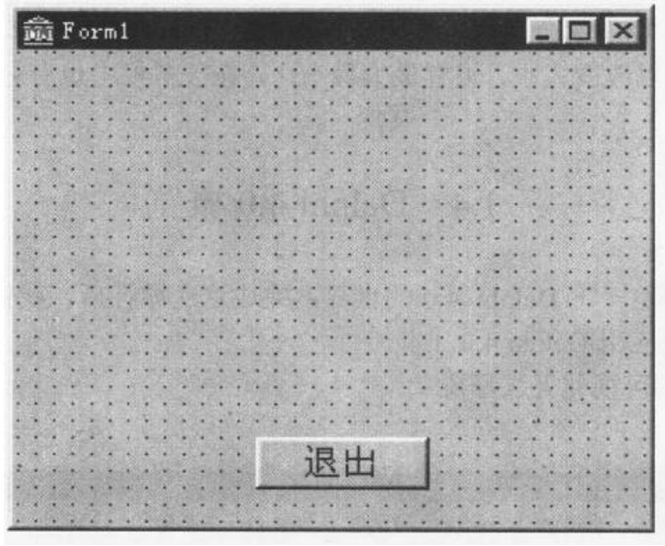

图 1-3　向窗体中添加组件并设置其属性

按钮组件上显示的文字是通过设定 Caption（标题）属性来确定的，选中 Button1 组件，在左边的 Object Inspector（对象观察器或称为“对象浏览器”、“属性编辑器”等）中设定 Caption 属性为“退出”，表明这个按钮的功能是退出应用程序。Object Inspector 用来显示当前选择的组件的各种属性以及对应于相应的组件的名称等。用户可以通过对象浏览器来改变组件的属性。Object Inspector 还有一页是有关事件响应的。事件响应将在后面编写程序代码时再做简单介绍。在 Object Inspector 窗口中找到 Font（字体）属性，点击属性右边的带有“…”的按钮（我们把这个按钮叫做“More”按钮或者省略号按钮），会弹出一个“字体设置”对话框，由于字体属性包含的元素非常多，是一个复杂的属性，所以 Delphi 就采取这种弹出对话框的方式来帮助用户设置属性，这里选择宋体，字体大小 12，如图 1-3 所示。

1.3.3　编写组件的事件响应

Delphi 编程的一个特殊概念就是有关事件响应的概念，这一概念在这里不便展开细说，读者在读本书的后边各章时就会对事件响应的编程方法有更深刻的认识，这里先有一个感性认识。我们现在来实现用户单击“退出”按钮就将整个程序退出，双击“退出”按钮，就会弹出一个代码编辑窗口，在代码编辑窗口中声明了一个事件的相应函数，

TForm1.Button1Click，在其中的 Begin 和 End 之间写入 Close 语句，这个语句是 Form1 组件的方法，意思是将窗体 Form1 关闭。

```
Procedure TForm1.Button1Clicked（Sender:Tobject）
Begin
  Close
End;
```

现在我们的第一个 Delphi 程序全部编写完毕，选择“Run”菜单下的“Run”命令，或直接按下快捷键 F9，程序就会运行。程序包括一个主窗口，这个主窗口已经具备了 Windows 窗口的所有特征，可以改变大小、设置最大最小化，在任务栏中也有它的名字。单击“退出”按钮关闭程序。

1.4 Delphi 4.0 概览

下面我们快速浏览一下 Delphi 4.0 的世界，完成这次旅行后，我们将开发完成自己的一个应用程序，当然是应用 Delphi 4.0 了。

1.4.1 进入 Delphi 的可视化开发环境

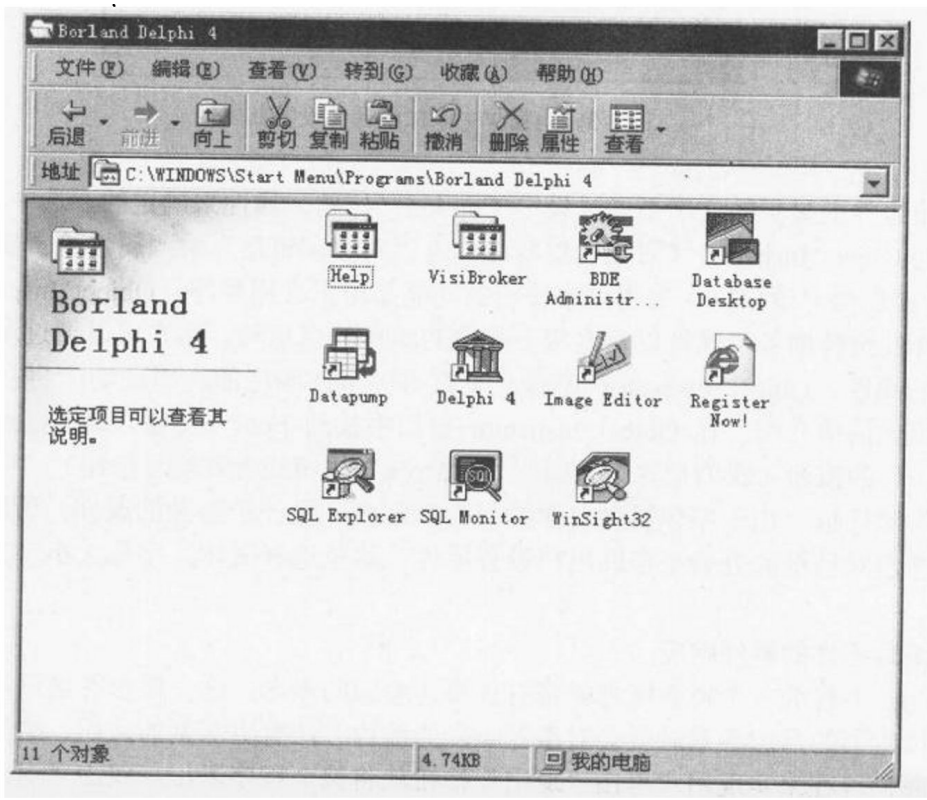

图 1-4 Delphi 4.0 的程序组

启动 Delphi 4.0，先启动 Windows98 并打开 Borland Delphi 4.0 程序组，如图 1-4 所示。双击“Delphi 4”程序项以启动 Delphi 应用程序，加载后会出现如图 1-5 所示的窗口。

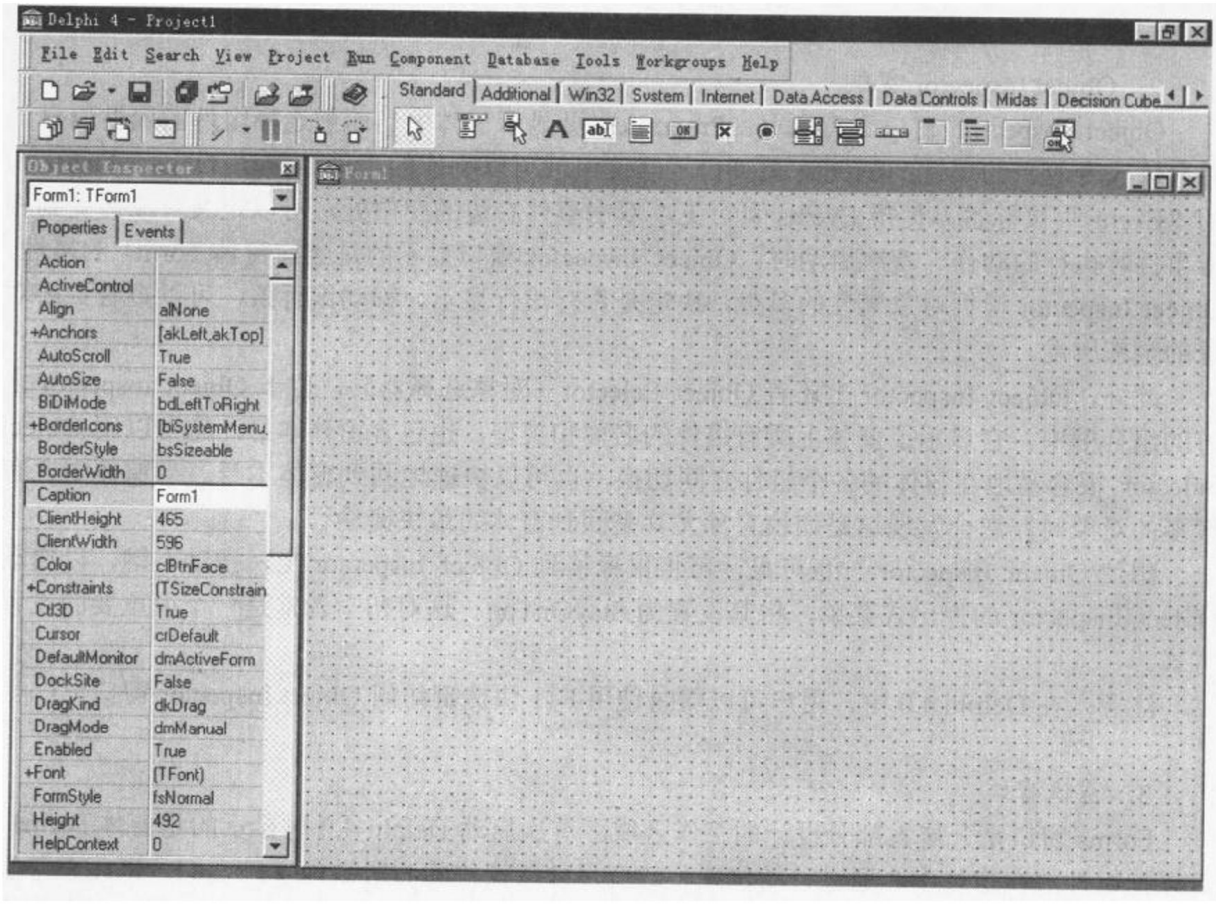

图 1-5　Delphi 的运行界面

首次加载 Delphi，屏幕上会出现四个窗口：

(1) 标题为“Delphi-Project1”的 Delphi 主窗口。

(2) Object Inspector 窗口。

(3) 标题为“Form1”的窗体（Form）窗口。

(4) 标题为“Unit1.PAS”的代码编辑窗口。刚启动时此窗口的大部分被“Form1”窗体所掩盖。将“Form1”窗体移开，或单击 Form1 窗体下方的状态行，可以使其全部可见。在“Form1”窗体的任意可见位置单击鼠标，可以恢复主窗体全部可见。

以下我们将对这四个窗口分别进行介绍。

1.4.2　Delphi 4.0 可视化编程环境介绍

1. 主窗口（MainForm）

Delphi 的主窗口位于屏幕的上端，包括 Menu（菜单）、Speedbar（加速条）和 Component Panel（组件选项板）。Menu 是下拉式主菜单。Speedbar 位于主窗口的左下端，由两排共 14

个加速按钮组成。这些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。Component Panel 由一行、若干页对象按钮所组成，利用它来选择需要的组件并将它放到窗体中去。

2. Object Inspector 窗口

Object Inspector 窗口含有两页：Properties 页显示窗体中当前被选择组件的属性信息，并允许改变对象的属性；Events 页列出了当前组件可以响应的事件。按动 Object Inspector 的“Events”页标签，使得 Events 页可见，双击事件右边的空白处，可以定义对象接受到相应事件时执行的动作。首次启动时，Object Inspector 窗口显示的是当前窗体 Form1 的属性。Object Inspector 根据对象属性的多少，决定是否有滚行显示。移动滚行条，可以查看当前对象的全部属性。

此外，Object Inspector 上还有 Object Selector（对象选择器），位于 Object Inspector 上方的组合框中。它显示了窗体上所有组件的名称和类型，也包含窗体本身。您可以用 Object Selector 很容易地在窗体的各个组件之间切换，也可以快速地回到窗体本身。当窗体中含有较多的对象时，您会发现这是切换对象尤其是回到窗体的最快捷途径。

想使 Object Inspector 一直可见，可将鼠标移到 Object Inspector 上，按动右键，以启动 Object Inspector 的弹出式菜单，将其设置为 StayOnTop。这对初学者常是一个很重要的设置方式。

注意：在 Delphi 4.0 中，用户可以随时使用 F11 快捷键访问 Object Inspector 窗口。

3. 窗体窗口

Forms 窗口是开展大部分设计的工作区域。首次启动 Delphi 4.0 时显示的是窗体 Form1。可以把组件放在窗体中，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。您可以把窗体想象成一个可以放置其他组件的容器。窗体上有栅格（Grids），供放置组件时对齐位置用，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，您可以选用不同的窗体进行设计。其他窗体可以是对话框（DialogBox）、数据录入框等。

4. 代码窗口

代码窗口一开始处于窗体窗口之下。因为在 Delphi 中，设计用户界面直接在窗体中进行，运行结果和设计模板完全一致。

当组件被放到窗体上时，Delphi 会自动生成大部分的用户界面代码。您所应做的只是在它为您生成的框架中加入完成所需功能的程序段而已。点动 Form1 的状态行使代码窗口可见。图 1-6 显示了空窗体 Form1 的代码窗口。

这个窗口中是代码编辑器，可以在其中书写 Delphi 应用程序的源代码。当程序中含有不止一个窗口时，会有几个库单元的源程序出现在代码编辑器中。

代码编辑器的标题条中显示了当前正在编辑的库单元文件名。要查看某一特定程序的源代码，只需用鼠标点动写有该库单元文件名的页标签，就可以对该库单元进行编辑了。

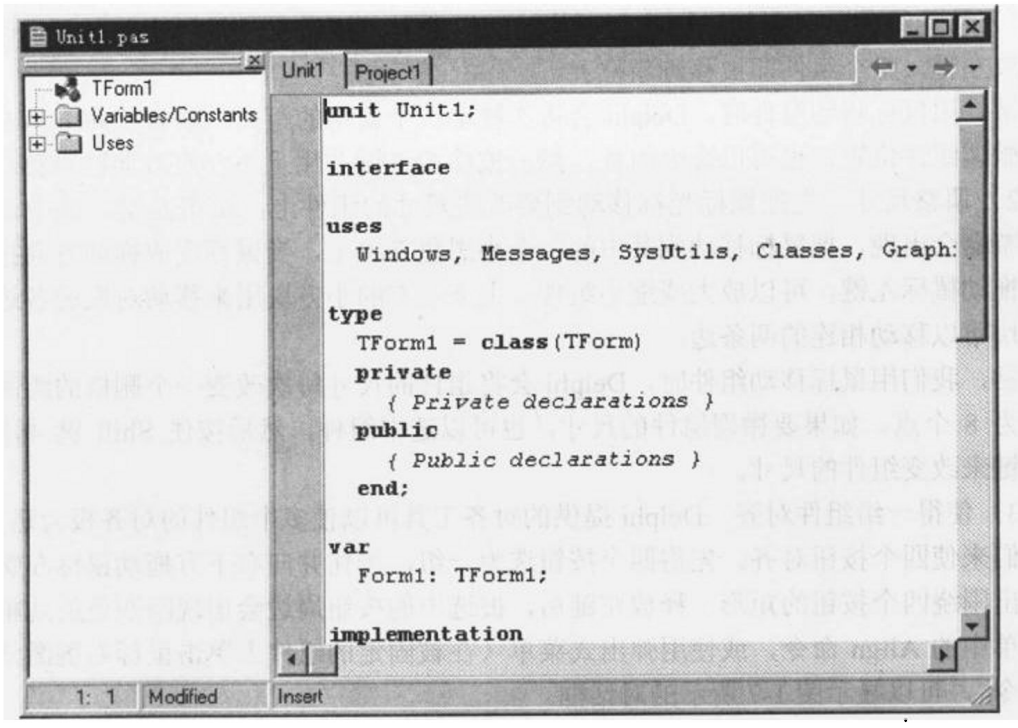

图 1-6　Delphi 开发环境的代码编辑器

1.4.3　设计简单的用户界面

编一个程序要做的第一件工作是确定最终用户将看见什么，换句话说，首先应该考虑的是应用程序的接口。我们将设计一个简单的程序：在屏幕上开一个窗口，窗口中有一个图框，用三个按钮来改变图框的形状；再用一个图标按钮来进行图框的颜色设置，通过颜色编辑对话框来选择变成哪一种颜色。

1. 选取组件加入到窗体中

Delphi 为用户提供了丰富的组件库，分为可视组件和不可视组件。组件是完成各项功能的封装零件，从某种程度上说，Delphi 编程也就是组件组装、配置的过程。各个不同功能的组件分别排列在 Component Panel 的各页上。

移动鼠标到 Component Panel 上，每当鼠标在某个组件的按钮上停留时，会弹出提示框提示该组件的名称。在要选择的组件上单击左键，则该组件按钮被按下，表示组件已被选择。然后，将鼠标移动到窗体上，按下左键，该组件被放到窗体中。组件的轮廓线上会显现八个用于调整尺寸的黑色小方块（Sizing Handles）。它除了供用户调整尺寸使用之外，还可以表示该对象处于当前编辑状态。此时，按“Delete”键可以将该组件删除。

按照这种方法加入所有需要的组件，一个 Shape 组件、三个 Button 组件以及一个 BitBtn 组件。这三类组件中，除了 Button 组件在 Standard 页外，Shape 组件和 BitBtn 组件都在 Addition 页上。

2. 组件的调整与对齐

组件在窗体中的排列必须美观整齐，需要进行位置、大小和显示字体的调整。Delphi 提供的对齐工具和窗口栅格为这些调整提供了方便。

（1）移动组件　只需把鼠标落到想移动的组件上，按住左键并移动光标，到合适的位置再释放左键，整个组件即被移到新位置。

通常，用鼠标移动组件时，Delphi 会每次移动一个栅格的距离，缺省状态下为 8 个点。如果要微调组件位置，也可以选中组件，然后按住 Ctrl 键并用上下左右方向键来移动组件。

（2）调整尺寸　先把鼠标光标移动到要改变尺寸的组件上，单击左键，选中该组件，尺寸调整器会出现，把鼠标移动到其中的一个小黑色方块上，当鼠标变成拖动方向指示时，按下并拖动鼠标左键，可以放大或缩小组件。上下左右的小方块用来移动对应的各边，四个角的方块可以移动相连的两条边。

通常，我们用鼠标移动组件时，Delphi 会将组件的尺寸每次改变一个删格的距离，缺省状态下为 8 个点。如果要微调组件的尺寸，也可以选中组件，然后按住 Shift 键并用上下左右方向键来改变组件的尺寸。

（3）使得一组组件对齐　Delphi 提供的对齐工具可以使多个组件的对齐极为迅速方便。下面我们来使四个按钮对齐。先将四个按钮选为一组，按住并向右下方拖动鼠标左键，在窗体上画出围绕四个按钮的矩形，释放左键后，被选中的按钮周边会出现暗灰色的边框。选用 Edit 菜单中的 Align 命令，或使用弹出式菜单（在被固定的组件上单击鼠标右键激活）中的相同命令 ，可以显示图 1-7 所示的对话框。

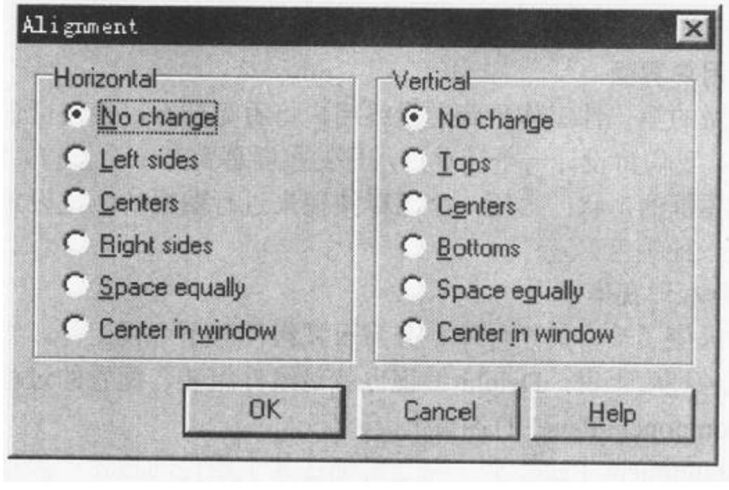

图 1-7　Alignment 组件对齐对话框

要使按钮沿左边对齐并使它们在垂直方向上均匀分布，先在 Alignment 对话框的“Horizontal”栏内选择“Left sides”，在“Vertical”栏内选择“Space equally”，单击 OK 按钮，Delphi 就会自动将它们按照上面所介绍的方式对齐。然后，您可以将它们四个作为一组来移动。在四个按钮以外的窗体上按动鼠标左键，就释放了组中的组件，使它们成为分离的组件。

利用对齐面板来对齐组件也是很方便的。首先要将要对齐的组件选成一组，选择 View 菜单中的 Alignment Palette 命令显示对齐模板，如图 1-8 所示。按照所示的方式选择即可达到对齐的目的。

调整“Shape”组件的大小，使之与右边的按钮组相匹配。再改变窗体的大小，按住并拖动窗体右下方使之刚好包容窗体上的全部组件。这样，您的用户界面就会比较美观。

（4）锁定组件　通常，当在窗体上布置好了组件之后，需要把窗体上的组件锁定。在锁定组件时，不能改变组件的大小和位置，只能通过属性编辑器来改变组件的属性和事件响应程序。这样可以防止由于错误的鼠标动作而改变组件的布局。

选择主菜单上的 Edit 菜单中的 Lock Controls 命令，使得组件被锁定（不能移动、也不能改变大小）。解锁只需再次选择此项即可。

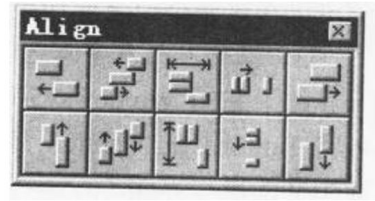

图 1-8　Delphi 的组件对齐模板

3. 保存所做的工作

从主菜单上选择 File 菜单中的 Save Project As...命令，Delphi 会显示标题为“Save Unit1 As”的文件保存对话框，Delphi 4.0 允许用户更改存储路径，您可以在下拉式列表框中选择。最好将您的文件保存在自己的目录中。在编辑框中键入“demoform.pas”以保存库单元文件；然后显示标题为“Save Project As”的另一个文件保存对话框。键入“sample.dpr”，Delphi 保存这两个文件并返回窗体窗口。不要把库单元和项目存成一样的文件名，Delphi 要求两者不同。

第一次保存后，以后可以随时通过 Speed Bar 中的“Save All”和“Save file”来保存项目文件和库单元文件。一般来讲，当确认文件的改变后，要同时存储这两个文件。

1.4.4　改变对象的属性

上述的项目虽能够运行，但它对您的按动按钮的操作是没有什么反应的，而且，所有的组件上还写着我们不需要的字样。单击窗口的关闭按钮结束运行，回到设计界面。下面，我们将仔细讲述如何在 Object Inspector 中改变组件的各种属性。

1. 用 Properties 页改变组件的属性值

首先要改变各种组件的标题。先给窗口命名为“Demo”。按动 Object Inspector 上端的 Object Selector 的题条或者其右端的下拉标志，找到 Form1 项，并点动左键，窗体被选中。当然，用户也可以在窗体上单击（注意不要击在组件上），来选中窗体。在 Object Inspector 的 Properties 页中，找到 Caption 属性并用左键选中，将其右端的 Form1 改为 Demo，同时，您会发现窗体的标题已经相应地做了改变。

用鼠标点中窗体中的 Shape 组件，Object Inspector 列出了它的属性。选中 Shape 属性，您会发现右端出现了下拉标志。点动此标志，可以查看对象的 Shape 属性可选值：它的形状可以是矩形、圆形、圆角矩形、方形等几种。这是我们设计后续功能的基础。

选中 Button1 按钮，此时 Object Inspector 已经显示出此按钮的相应属性。将它的 Caption 属性改为“&Rectangle”，“&”号使得 Delphi 特殊处理它后面的字符，在这里，按钮中的 R 字母被做了下划线处理，运行时，可以用“Alt-R”热键来按动此按钮。同样，您可以将其他的两个按钮 Button2 和 Button3 的 Caption 属性改成需要的形状指示，譬如“&RoundRec”、“&Ellipse”。

2. 设置窗体的缺省按钮

可以把某个按钮作为窗口上的缺省按钮，Delphi 会为按钮加上有黑色的边缘。运行时，

用户按下回车键即相当于该按钮被按下。只需将此按钮的 Default 属性从 False 改成 True，即将它设为窗体的缺省按钮。点动 Default 属性，在右端的值后面双击左键，或从下拉菜单中选取 True，即可改变此属性。Delphi 中有许多只有 True、False 两个属性的组件，双击左键可以在这两个值之间切换。

3. 汉化界面及字体选取

如果您的 Windows95/98 系统支持中文，那么对界面做汉化是极其方便的。例如，您可以将 Button1 的 Caption 属性改成“&R 矩形”，同样地可改变其他组件的属性，将窗体做成中文的操作界面。

若对中、西文字体不满意，则可以调整 Font 属性以满足您的要求。Font 属性的前面，有一个小小的“+”号，这说明它表征的是集属性，也即属性不再是一个单值，而是一个属性的集合。双击 Font，Object Inspector 将在下面扩充显示它的其他属性，如图 1-9 所示。Color 用来表示文本的颜色，Name 定义了字体名，如 System，MsSerif，Arial，宋体，黑体等。Style 下又拥有四个属性：fsBold（加粗），fsItalic（斜体），fsUnderline（下划线）和 fsTrikeOut（删除线）。如果想让字体有其中的某种风格，可把相应的属性值设成 True。

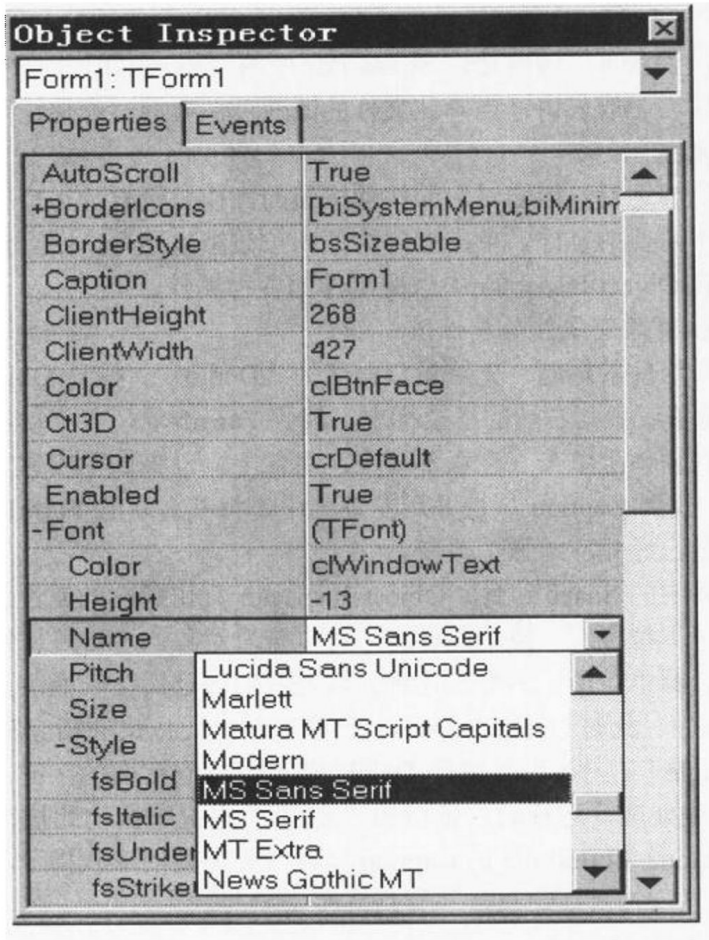

图 1-9 Object Inspector 中的 Font 属性