

金飞弘道计算机系列丛书

融会贯通——

Delphi 4.0

北京华杰同科技有限公司 策划

弘道工作室 编著

曹 伟 执笔

人民交通出版社

内 容 提 要

本书从一个开发者的切身经验出发,系统地介绍了使用 Delphi 4.0 这一优秀的可视化工具进行应用程序设计的方法,内容既涉及 PASCAL 语法、菜单的设计、文件操作、数据库的开发等常用概念,又精辟地阐述了多媒体和网络应用程序的编写,以及动态库、自定义组件、嵌入汇编、资源、多线程、串行通信、程序调试等比较深入的部分,同时还以相当篇幅介绍了帮助文件的编写和安装程序的制作。

本书适用于 Delphi 的初学者以及具有一定基础希望有所深入的用户。

图书在版编目(CIP)数据

融会贯通 Delphi 4.0/弘道工作室编.-北京:人民交通出版社,1998.2

(金飞弘道计算机系列丛书)

ISBN 7-114-03184-X

I.融… II.弘… III.Delphi 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(98)第 40818 号

融会贯通——Delphi 4.0 RONGHUIGUANTONG Delphi 4.0

弘道工作室 编著

曹 伟 执笔

责任印制:孙树田

人民交通出版社出版发行

(100013 北京和平里东街 10 号)

各地新华书店经销

北京牛山世兴印刷厂印刷

开本:787×1092 1/16 印张:19.25 字数:480 千

1999 年 1 月 第 1 版

1999 年 1 月 第 1 版 第 1 次印刷

定价:48.00 元

ISBN 7-114-03184-X

TP · 00050

前 言

Delphi 是 Inprise（即原来的 Borland 公司）公司推出的一个基于 Object Pascal 语言的可视化软件开发平台。它提供了优秀的可视化设计工具，采用面向对象的方法将 Windows 编程的复杂性封装起来，实现了将可视化设计与 Object Pascal 语言的有机集成，具有对网络连接的内部支持。配有 Database Engine，可通过 SQL Links、ODBC 等轻松访问多种数据库，并且提供了强大的开发基于客户/服务器模式的数据库应用的能力。所有这些，使得 Delphi 具有功能强大、使用方便、运行速度快、开发效率高等卓越特性，一经推出就受到世界各地编程者的钟爱，成为最受青睐的开发工具之一。

Delphi 的每一个新的版本都给用户提供了更新的功能。例如，Delphi 1.0 版本，提供给用户一种全新的可视化的环境，能够快速开发 Windows 3.1 的应用程序；Delphi 2.0 版本重新改写了 Delphi 的核心，不仅仅是从 16 位到 32 位的转变，而且全面支持 Windows 95&NT 的 OLE Automation、OCX、多线程、Unicode 和 MAPI 等功能，还增加了如数据模块(Data Module)等多种高效的数据重用功能；Delphi 3.0 则是一个多层 C/S 的 32 位快速开发工具，提出了三种 Broker 和新一代的数据库引擎，来适应新一代数据库管理观念 n-Tier 的需求，全面支持 Internet(FTP、HTTP、NNTP、POP、SMTP、TCP、UDP)和 Web 上的 Client/Server 技术，提供了新的编译技术 VCL Package 和创建 ActiveX 构件及文档的能力，提供对 COM(Component Object Model)的自动支持；1998 年推出的 Delphi 4.0，除了界面上更加美观之外，还对 Object Pascal 语言进行了一些扩展，加强了调试的功能，加入了对 2000 年问题的改进以及对 CORBA 的支持。

鉴于 Delphi 的优秀性能和广大开发者的迫切需要，我们在积累了大量编程经验的基础上编写了这本书。本书共分 14 章，系统地讲述了 Delphi 程序设计的基本概念、Object Pascal 语法和面向对象的思想、菜单的设计、文件操作、多媒体、动态库和自定义组件的编写、网络应用程序开发、资源的使用、多线程、串行通信、帮助文件的生成、程序的调试、数据库应用程序设计以及安装程序的制作、Delphi 4 中的“复活节彩蛋”等。

在本书的编著过程中，我们尽量注意减少冗长无味的说明，而代之以具体生动的实例演示。通过实例，引导读者把握 Delphi 的精髓所在。本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发，适合于各个层次的 Delphi 用户。对初学者来说，可迅速加入 Delphi 高级用户的行列；对有一定使用经验的读者，也可通过本书掌握 Delphi 更深层次的开发方法，学会用更巧妙的办法开发出高水平的 Delphi 应用软件，真正达到精通 Delphi 4.0 的水平。

目 录

第一章 Delphi 快速入门	1
1.1 Delphi 的特点	1
1.2 Delphi 4.0 快速入门	3
1.3 常用组件介绍	10
第二章 Object Pascal 语言和面向对象编程	20
2.1 程序的结构	20
2.2 变量和数据类型	22
2.3 过程与函数	30
2.4 Delphi 的基本语句	33
2.5 面向对象的编程	40
第三章 菜单和 MDI 窗体的设计	45
3.1 MDI 的基本概念	45
3.2 创建 MDI 的主窗体	46
3.3 创建 MDI 的子窗体	46
3.4 设计下拉菜单	48
3.5 设计弹出菜单	49
3.6 编写代码	49
3.7 程序源代码	58
第四章 文件操作	64
4.1 Delphi 中的文件类型	64
4.2 文件操作的基本函数	67
第五章 用 Delphi 开发多媒体应用程序	70
5.1 关于图象编程	70
5.2 抓取屏幕的程序	76
5.3 多媒体开发中涉及的主要概念	89
5.4 MediaPlayer 组件	90
5.5 Animate 组件	94
5.6 使用 MediaPlayer 组件编写多媒体程序	97

第六章 动态连接库和组件的开发	103
6.1 调用动态连接库	103
6.2 编写动态连接库	107
6.3 自定义组件的编写	116
第七章 网络应用程序的设计	138
7.1 Internet 组件页	138
7.2 数据的发送与接收	141
7.3 WWW 浏览器	145
7.4 Message 服务	147
第八章 其它话题	158
8.1 资源文件的使用	158
8.2 多线程编程	162
8.3 串行通信的使用	169
8.4 打印	183
8.5 Delphi 附带的工具	188
第九章 帮助文件的编写	195
9.1 帮助文件的结构	195
9.2 使用 Word 编写帮助	196
9.3 使用 HelpScribble 编写帮助	197
9.4 连接程序和帮助	206
9.5 自定义组件的帮助	207
第十章 程序的调试和异常处理	212
10.1 调试方法概述	212
10.2 使用 Browser	221
10.3 其它调试工具	223
10.4 异常处理	224
第十一章 数据库应用	231
11.1 数据库系统概述	231
11.2 数据库应用程序的开发	241
11.3 SQL 的使用	253
11.4 C/S 应用的开发	256
第十二章 软件的分发-安装程序的制作	258
12.1 InstallShield 简介	258
12.2 IS for Delphi 4.0 的使用	259

12.3	InstallShield 远东版.....	274
12.4	Wise 的使用.....	280
第十三章	Internet 上的 Delphi 资源	287
13.1	国内的 Delphi 站点	287
13.2	国外的 Delphi 站点	294
13.3	电子杂志的订阅	297
第十四章	结束语	298

第一章 Delphi 快速入门

Delphi 是最优秀的可视化编程环境之一，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows95 图形用户界面的许多先进特性和设计思想，采用了弹性可重复利用的完整的面向对象程序语言(Object-Oriented Language)，集成了当今世界上最快的编译器、最为领先的数据库技术、对 Internet 的杰出支持，具有运行速度快、代码效率高、掌握时间短、开发迅速等卓越特性，令众多编程工具无法望其项背。对于广大的软件设计人员来说，使用 Delphi 开发应用软件，无疑会大大地提高编程效率和软件质量。一旦真正精通 Delphi，您将会发现编程已经不再是原来那种枯燥无味的工作了，一切都变得如此轻松和充满乐趣。Delphi 的每个细节，都会给开发者带来一份欣喜和享受！

1.1 Delphi 的特点

Delphi 具有如下特点：

1. 可视化的集成环境

Delphi 是一个运行于 Windows 环境下的可视化开发工具，提供易于使用的图形化工具供编程者创建程序中的可视化部分，从而使你用最少的工作量来得到最高的开发效率。开发者只需要单击一下鼠标，就可以把选择的按钮和组件放到窗口中，再根据自己的习惯和用户的要求，安排窗口的外观，调整各组件的位置。对于那些习惯于使用非可视化工具进行软件开发的人员来说，这些工作必须通过手工编写一行行代码来实现，然后运行程序，根据实际效果返回再修改源程序，直至满足要求为止。可视化的出现则把这一过程简化到只需要鼠标的几次点击，在设计时中看到的窗口和菜单等就是在运行时出现的效果，真正达到了“所见即所得”。这样一来，开发者就不必把精力放在界面的调整上，而是主要用来实现程序的功能模块，从而大大节约了编程人员的时间，提高了程序开发的效率。

2. 面向对象的编程概念

面向对象的程序设计(Object-Oriented Programming，简称为 OOP)是 Delphi 诞生的基础。Delphi 是以面向对象的方式设计程序代码。

面向对象的方式立意于创建软件代码的可重复利用性，具有高度继承性，具备更好地模拟现实世界环境的能力，因而被认为是自上而下编程方法中的佼佼者。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，使得复杂的工作条理清晰、编写容易、维护方便，更提供了极强的扩展性。编程人员可以在已有对象类型的基础上创建新的对象，这个新对象将继承其先前对象的所有事件、属性、方法，从而重复利用开发者自己或是其他人已有的成果，而且，还可以根据需要，在对象中添加新的过程和方法，非常灵活。不同的对象，即使有共同的祖先，通过不同方式的扩展，也可以在同一过程中做完全不同的

事情。

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，程序员仍然要编写大量的代码。Delphi 则把可视化和面向对象这两个优秀的概念结合起来，使生成美观的界面和结构良好的程序已经不再是一件痛苦的事情。

Delphi 所采用的底层编程语言是 Object Pascal，它风格简洁、语法严格、结构清晰、可读性强、代码效率高，尤其适合于可视化的程序设计。

总之，可视化、面向对象、Object Pascal 这种革命性的组合，使得 Delphi 从众多的编程工具中脱颖而出，受到越来越多开发人员的青睐。

3. 创建新的模板和组件进行扩展的能力

Delphi 提供了 VCL(Visual Component Library, 可视化组件库)。VCL 指构成 Delphi 组件结构基础的对象类层次，包括可视化部件、非可视化部件、不能作为组件的类以及不能视为对象的通用函数等。用户可以定制组件模板(Component Palette)，为其增加页面或者新的组件。

Delphi 模块化、可重用的特点使得开发者可以把自己认为有意义，以后还可能用到的程序以模板的形式保存成窗体表单，或以 DLL 的形式编译该程序，从而在以后的工作中受益匪浅。

4. 卓越的数据库引擎和快速报表工具

Delphi 结合了传统的编程语言 Object Pascal 和数据库语言的强大功能，既可以用于传统的数值运算类编程又可以用于数据库编程。利用 Delphi 强大的数据库功能，不需要编写任何代码便可以创建一个简单的数据库应用。

Delphi 支持开发桌面或者客户机/服务器(Client/Server)结构的数据库应用程序和报表。通过窗体和报表，允许用户创建能访问 dBASE、Paradox 和 Local InterBase 服务器的数据库应用，也可以访问远程数据库服务器上的数据库(如 ORACLE、SyBase、InterBase、Microsoft SQL Server、Informix 等客户/服务器数据库中的数据库)，或任何经 ODBC(Open Database Connectivity) 可访问的数据库管理系统中的数据库。

利用 Delphi 建立数据库管理系统，开发人员不必过分操心数据库本身是在本地硬盘上还是位于一个远程的 SQL 服务器中，提高了软件开发的效率。

5. 对 Internet 的全面支持

Delphi 4.0 提供了多达 26 个 Internet 相关的组件，同时支持 Internet Explore 的 ISAPI 和 NetscapeD 的 NSAPI 动态链接扩充协议，具有对 TCP、UDP、POP、NNTP、POP3、SMTP、FTP、HTTP 等协议的内在支持。使用这些组件可以在很短的时间内完成 TELNET、FTP、HTTP、FINGER、MAIL、MESSAGE 等等客户端和服务端应用程序的开发，而不必去花大量的精力钻研相关的协议知识。

6. 优秀的消息处理机制

Windows 本身是一个消息驱动的操作系统，它也通过消息的发送、接收和处理来和应用程序打交道。Delphi 提供了对 Windows 消息处理的方法，各条消息都有一个便于记忆的名称，而且这个名字与 Windows SDK 中基本相同，它通过为一个应用程序的所有组件注册一个 MainWndProc 的方法，将其作为窗体过程进行消息流跟踪。与 VB 等可视化工具相比，

Delphi 的消息要全面得多，而且开发者还可以根据自己的需要，另外自定义非标准消息，因而更加靠近 Windows 的底层，能完成更强劲的功能。仅此一点，就足以引起众多编程者的瞩目，把他们吸引到 Delphi 的阵线上来。

7. 真正独立的可执行文件和动态链接库的创建

用 Delphi 创建的 EXE 和 DLL 文件不需要任何额外的 OCX、VBX 或者其他 DLL，除非在程序中非要使用这些文件不可。Delphi 的库编译以后是打包到可执行文件中的，因此生成的 EXE 拷贝到任何机器上都可以直接运行，不需要再附带其他任何文件。这与用其他很多可视化工具开发出的伪代码相比，具有显而易见的优越性。

鉴于 Delphi 的众多特性，使用这一工具进行开发的编程者已经越来越多。值得一提的是，随着 Internet 的迅猛发展，网上的 Delphi 站点如雨后春笋般不断出现，因而可利用的资料、组件应有尽有，其中又有一半以上是 Delphi 的爱好者们以源码的形式提供的，这些资源不论对 Delphi 的初学者还是高手都是一笔宝贵的财富。

1.2 Delphi 4.0 快速入门

1.2.1 安装 Delphi 4.0

Delphi 4.0 的安装非常简单。把光盘插入驱动器后，运行盘上的 Install.exe 文件进行安装。启动画面如图 1-1 所示。单击画面上的“Delphi 4”选项，根据安装过程中的提示，选择需要的部件和合适的路径进行安装。

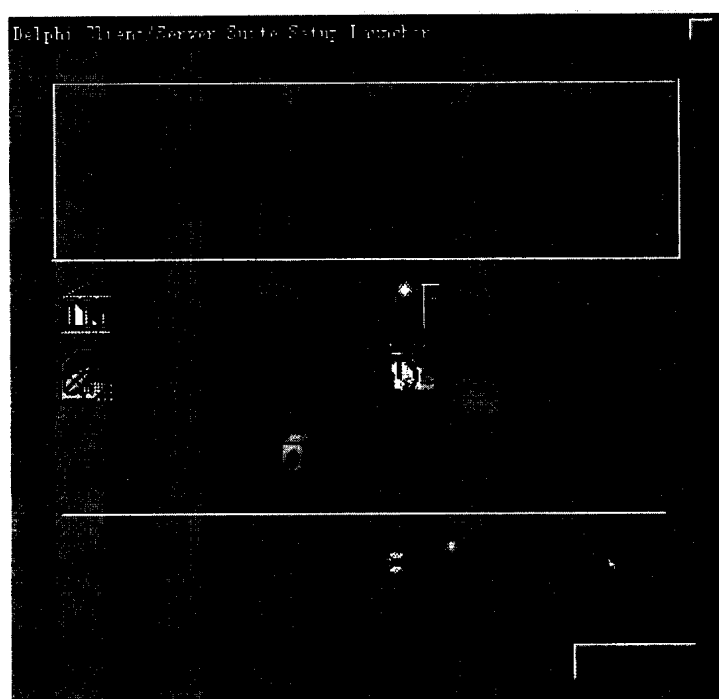


图 1-1 Delphi 4.0 安装启动画面

1.2.2 熟悉 Delphi 4.0 的集成环境

启动 Delphi 4.0 主程序后, 出现如图 1-2 所示的集成环境界面。

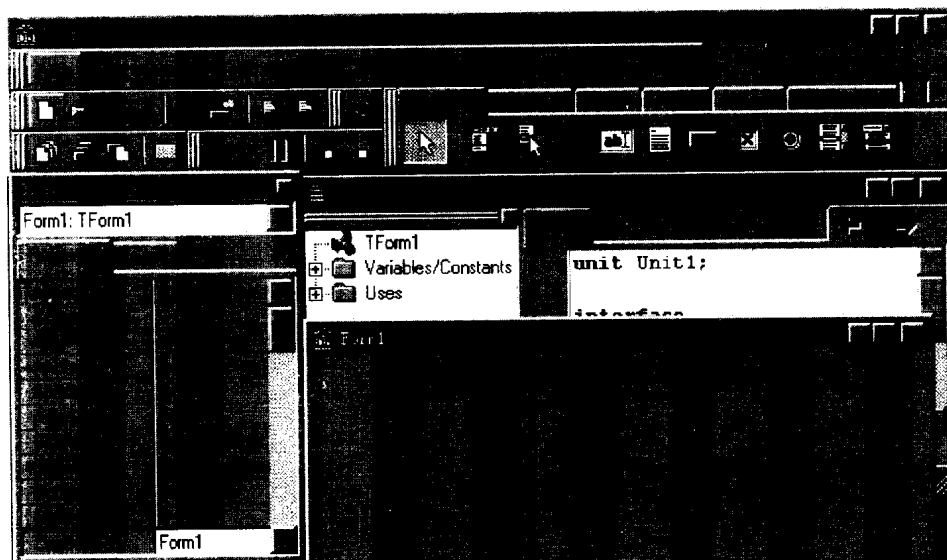


图 1-2 Delphi 4.0 的集成环境

第一次运行 Delphi 4.0, 屏幕上出现四个窗口:

- 标题为“Delphi 4-Project1”的 Delphi 主窗口
- 标题为“Object Inspector”的对象监视器窗口
- 标题为“Form1”的窗体(Form)窗口
- 标题为“Unit1.pas”的代码编辑窗口

下面对这几个窗口分别介绍。

1. 主窗口

Delphi 的主窗口位于屏幕的上端, 它由菜单、加速条和组件模板等几部分组成。

下拉式主菜单可完成以下功能:

- 文件(File)菜单。负责工程文件、表单文件、单元文件的创建、打开、存储、打印等管理工作;
- 编辑(Edit)菜单。负责剪切、复制、粘贴、删除、撤消等编辑功能以及对已创建的组件位置、排列顺序的安排;
- 查找(Search)菜单。负责完成文本的查找、替换、移动以及查错和符号浏览的功能;
- 察看(View)菜单。负责工程管理、资源察看、对象察看、加速条设置、调试窗口的显示/隐藏等;
- 工程(Project)菜单。负责工程中文件的添加、删除, 工程的编译、链接、语法检查以及对工程的若干设置工作;
- 运行(Run)菜单。负责工程的运行和调试, 如断点设置、单步运行、变量值监测、跟踪等;
- 组件(Component)菜单。包括新组件的创建、已有组件和 Package 的安装、ActiveX

控制的引入、组件控制板的配置等；

- 数据库(Database)菜单。负责数据库相关功能，如数据库的管理、数据库表单的创建；
- 工具(Tools)菜单。负责 Delphi 4.0 环境的设置、调试器设置以及图形编辑器等外部程序的运行；
- 帮助(Help)菜单。包括目录、如何使用帮助、主体搜索及 Delphi、Inprise 公司主页连接。

Delphi 4.0 的快捷工具栏是 Office 风格浮动式的快捷工具栏，位于主窗口的左下端，由两排共 16 个快捷按钮组成，如图 1-3 所示。这些按钮是菜单功能的快捷方式，依次分别为：新建文件、打开文件、保存文件、保存所有文件、打开工程文件、向工程中加入文件、从工程中删除文件、在线帮助、从列表中选择单元、从列表中选择窗体、单元与窗体的切换、新建窗体、程序运行、程序暂停、跟踪、单步执行等。



图 1-3 Delphi 4.0 的加速条

组件模板由多个组件页组成，组件(Component)是建立 Delphi 应用程序的要素。Delphi 为用户提供了丰富的组件库，既有可视的部件(如编辑框、按钮)等，也有不可视的部件(如系统定时器、数据表等)。它们按照功能分别排列在 Component Panel 的各页上，包括 Standard、Additional、Win32、System、Internet、Data Access、Data Controls、Midas、Decision Cube、QReport、Dialogs、Win31、Samples 等。利用组件模板可选择需要的部件，并将它放到窗体中去。

2. 对象监视器(Object Inspector)窗口

Object Inspector 窗口如图 1-4 所示。它含有两个标签页：属性(Properties)页用于设置窗体和窗体上当前被选择对象的属性，如按钮的名称、位置、大小、颜色、字体、字型等；事件(Events)页列出了当前对象可以响应的事件。鼠标单击 Object Inspector 的“Events”页标签，显示 Events 页。双击事件后边的空白处，可以定义该对象接受到相应事件时应该执行的动作。

3. 窗体窗口

窗体(Form)是编程者进行设计工作的区域。首次打开 Delphi 4.0 时，该窗体是空的，使用者可以在上面放置所需要的对象，如按钮、菜单等，也可以利用 Delphi 本身提供的窗体模板进行设计。可以把窗体想象成一个放置其它部件的容器。窗体上有栅格(Grids)，供放置部件时对齐位置用。在设计时可以显示或隐藏这些栅格，在程序运行时栅格不可见。

4. 代码编辑窗口

每个窗体都对应一个单元文件，里面包含了变量的定义、程序代码等内容。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。开发者所需要做的只是在它生成的框架中加入程序段。双击窗体，就会自动弹出该窗体对应的单元文件。缺省产生的空白窗体对应的代码单元窗口如图 1-5 所示。窗口的左边一部分是该单元内容的一个索引，双击里

面的选项，就可以直接跳到右边窗口中对应的代码处。

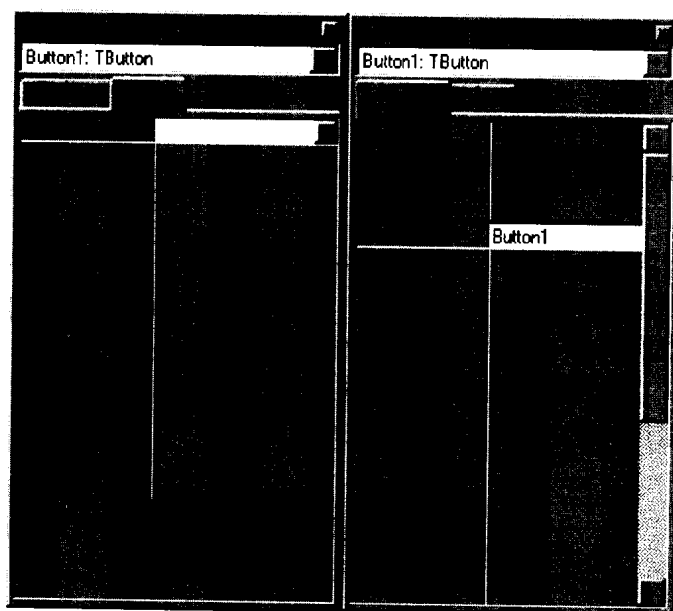


图 1-4 对象监视器窗口

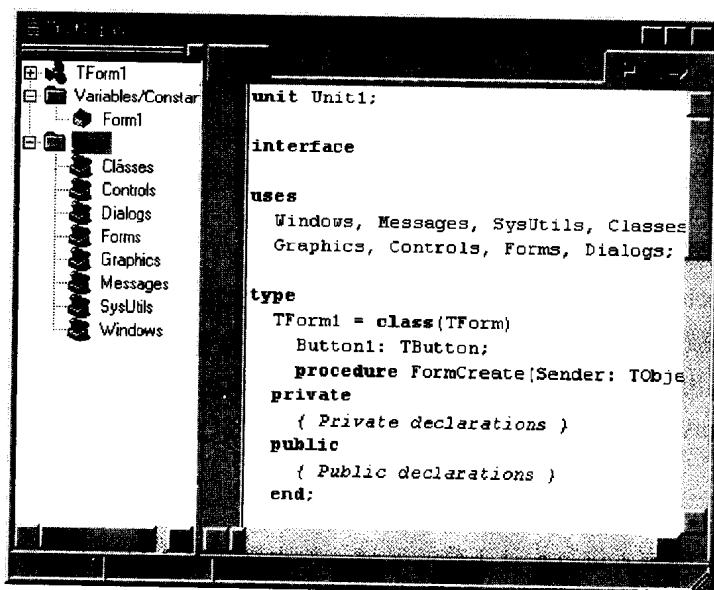


图 1-5 代码编辑窗口

1.2.3 Delphi 4.0 编程基本过程

在这一节中，我们将通过一个简单的例子来说明如何使用 Delphi 进行应用程序的设计和开发。这个例子只有一个窗口，窗口中包含一个标签、一个编辑框和两个按钮。在编辑框中输入文字，然后单击一个按钮，使窗口的标题变成所输入的内容，单击另一个按钮关闭窗口。步骤如下：

1. 创建新工程

在菜单上选用 **File/New Application** 菜单项来生成新的工程文件。这时，屏幕上出现图 1-2 所示的窗体，属性页中出现 **Form1** 的属性。将鼠标移动到窗体的边框处可以调整窗体的大小。

单击属性页中的 **name** 项，把主窗体的名称改为 **MainForm**，此时，窗体的标题(**Caption**)自动变成 **MainForm**。单击 **Caption** 属性，输入“测试程序一”，改变窗体的标题。

2. 保存文件

单击加速条中的“**Save All**”按钮，将单元存为 **MainUnit.pas**，将工程保存为 **test1.dpr**。虽然 **Delphi** 在运行时会自动保存所做的修改，但在程序编写过程中随时存盘是一种良好的编程习惯，可以避免一些不必要的损失。

3. 放置组件

单击组件模板上“**Standard**”页上的“按钮”组件，然后，将鼠标移动到 **MainForm** 窗体上，在合适的位置处按下左键，该组件被放到窗体中。按钮的轮廓线上会显现八个黑色小方块，称为尺寸调整器(**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**”键可以将该部件删除。在属性编辑窗口中，将该按钮的 **Name** 改为 **ApplyBtn**，**Caption** 改为“设置”。这时会发现标题的字体非常小，看起来很不舒服。双击 **ApplyBtn** 的 **Font** 属性，弹出字体对话框，将字体改为宋体，大小改为 12，选择“确定”后关闭该对话框。很多组件都有一个称为 **Hint** 的属性，**ApplyBtn** 也不例外。**Hint** 的作用是当鼠标在该组件上停留 2 到 3 秒时，弹出一个黄色的小方框，提示该组件的作用。我们把 **Hint** 属性设置为“改变窗口标题”，同时还需要把 **ShowHint** 属性设置为 **True**。**ApplyBtn** 还有一个 **Default** 属性，它的值可以为 **True** 或 **False**，为 **True** 时表示它是缺省按钮，运行时，回车即相当于被按下。我们把 **ApplyBtn** 的 **Default** 属性设置为 **True**。

在组件模板上，单击写有“**Additional**”的页标签，切换到 **Additional** 页，再移动鼠标，逐个用观看 **Hint** 的办法查看组件的名称，选择一个“**BitBtn**”的按钮(图标是有绿色对号的 **OK** 按钮)，把它放置在 **MainForm** 中，名称为 **CloseBtn**，标题为“关闭”，字体仍为宋体，大小为 12，**Hint** 为“关闭窗口，退出程序”，**ShowHint** 属性设置为 **True**，**Kind** 属性设置为“**bkClose**”，这是 **Delphi** 自带的一种位图按钮。

为了使这两个按钮的大小保持一致，需要进一步的调整。按住 **Shift**，单击 **ApplyBtn** 按钮，这样就同时选中了 **ApplyBtn** 和 **CloseBtn**，在任何一个按钮上单击鼠标右键，在弹出的菜单上选择“**Size...**”项，出现如图 1-6 所示的组件大小对话框。对于 **Width** 和 **Height** 均选择“**Grow to Largest**”项，使按钮宽度和高度保持统一。

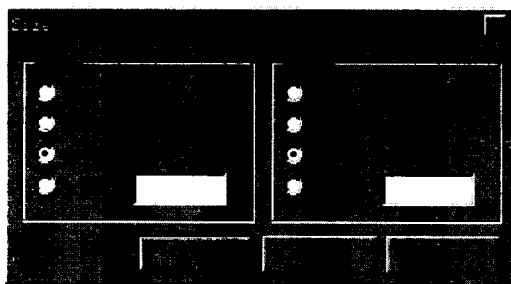


图 1-6 组件大小对话框

为了将这两个按钮对齐，同样单击鼠标右键，选择“Align...”选项，弹出如图 1-7 所示的组件对齐对话框。选择水平方向右对齐。

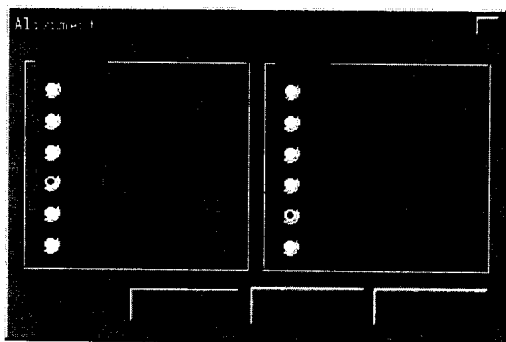


图 1-7 组件对齐对话框

再在 MainForm 中放置一个“Standard”页中的“Label”组件和“Edit”组件。前者的 Name 属性为 CaptionLabel，Caption 为“新标题:”，Font 同上；后者的名称为 CaptionEdit，Font 同上，Text 属性设置为空。

此时，MainForm 的窗体外观如图 1-8 所示。

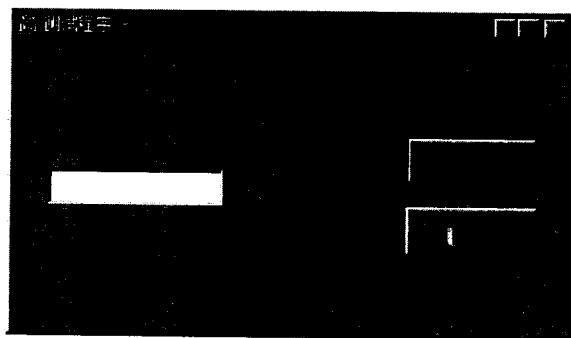


图 1-8 MainForm 的外观

4. 运行程序

单击加速条中的“Run”按钮，程序开始运行。正如我们所期待的那样，出现了一个窗口，上面有刚才设计时所放置的组件。鼠标在按钮上停留几秒钟会出现提示框，单击“关闭”按钮或窗口标题条上的叉号，可以退出程序。但“设置”按钮目前还没有任何作用。下面我们将为其添加代码。

5. 编写事件处理的代码

代码的目的是单击“设置”按钮后，把窗口的标题改为编辑框中所输入的文字。双击“设置”按钮，进入代码编辑窗口。可以看到，现在处于 ApplyBtn 的 OnClick 事件中。键入如下的代码：

```
MainForm.Caption := CaptionEdit.Text;
```

键入组件名称和后面的圆点之后，停留 2 秒钟，Delphi 会自动弹出一个列表框。这里面有该组件的所有属性名称、类型以及可调用的过程、函数，可以用箭头键浏览，然后按空格键选中。

保存所有文件，运行结果如图 1-9 所示。在编辑框中键入一些文字，然后单击“设置”按钮，可以看到窗口的标题条文字如你所愿地发生了改变。



图 1-9 程序运行时的界面

通过以上的实践，我们已经完成了一个简单的应用程序。尽管它本身没有什么太多的实际意义，但却可以直观地给您一个用 Delphi 进行开发的印象。事实上，总共键入的字符数只有寥寥几个，比较一下程序所完成的功能和编程的工作量，就足以理解 Delphi 为编程者提供了多大的方便。

1.2.4 Delphi 的文件类型

Delphi 的文件类型繁多，主要有下面几种：

- DPR 文件

DPR(Delphi Project)是 Delphi 工程的源文件。

- DOF 文件

工程选项(Delphi Options)文件。这是一个文本文件，里面包含了工程文件的选项，诸如编译器和链接器设置、路径、命令行参数等。多数情况下，这些是在 ProjectOptions 菜单中进行设置。

- CFG 文件

工程配置(Project Configuration)文件。这是编译器可读的文件，保存工程文件的配置。

- DSK 文件

桌面配置(Desktop Settings)文件。这是一个文本文件，保存了工程文件的状态信息，如哪些窗体是打开的、各窗体的位置等。

- PAS 和 DCU 文件

每个单元对应一个 PAS 文件，编译后产生一个 DCU 文件。

- DFM 文件

每个窗体对应一个 DFM 文件，记录该窗体及其所包含组件的属性信息。

- DPK 和 BPL 文件

Package 是 Delphi 所使用的一种特殊 DLL 文件，它的源代码文件以 DPK 结尾，编译以后生成 BPL(Borland Package Library)文件。

- RC 和 RES 文件

这是 Delphi 的资源(Resource)源文件和编译后生成的资源文件。

● 后缀以“~”开头的文件
如~dp, ~df, ~pa等, 这些是 Delphi 在编译过程中产生的一些临时文件, 一般不需要考虑。

1.3 常用组件介绍

1.3.1 Form

在介绍常用组件之前, 首先对 Form 的主要属性和事件做一阐述。

Form 具有的主要属性如下。

1. AutoScroll 属性

是一个布尔量。如果 AutoScroll 为 True, 那么当窗体太小而无法显示整个区域时, 会自动出现一个水平或竖直的滚动条。

2. BorderIcons 属性

用于窗体的控制菜单显示, 有四个选项可供选择:

- BiSystemMenu 是否显示系统菜单的按钮
- biMinimize 是否显示最小化按钮
- biMaximize 是否显示最大化按钮
- biHelp 是否在标题条上显示求助按钮。在 biHelp 为 True 的情况下, 如果 BorderStyle 属性为 bsDialog 或者 biMinimize 和 biMaximize 属性为 False, 则会在窗体的标题条出现一个问号, 用以选择在线帮助的对象。

3. BorderStyle 属性

指定窗体的边框风格。有六种风格可供选择:

- bsDialog 窗体大小不可更改, 对话框边框
- bsNone 窗体大小不可更改, 没有标题条, 无边框
- bsSingle 窗体大小不可更改, 单线条边框
- bsSizeable 窗体大小可以更改, 标准窗体边框
- bsSizeToolWin 与 bsSizeable 类似, 但显示的是一个窄标题条, 且没有系统菜单
- bsToolWindow 与 bsSingle 类似, 但显示的是一个窄标题条, 且没有系统菜单

4. Caption 属性

设置在窗体的标题中出现的文字。

5. Constraints 属性

设置窗体的最大、最小宽度和高度。如果窗体中组件超出窗体最大宽度或高度的范围, 会自动出现滚动条。设置为 0 时表示可以任意调整。

6. FormStyle 属性

设置窗体的类型。有四种类型可供选择:

- fsNormal 正常的文档界面窗体
- fsMDIChild MDI 子文档窗体

- **fsMDIForm** MDI 父文档窗体
- **fsStayOnTop** 正常的文档窗体，但始终位于所有窗体的顶部，即是一个“挡不住”的窗体。

7. HorzScrollBar 属性

设置水平滚动条的属性，例如是否可见、滚动条颜色、滚动范围、位置、大小、增量、类型、滑块尺寸等。

8. Icon 属性

指定该窗体对应的图标(Icon)。

9. PopupMenu 属性

指定该窗体的弹出菜单。

10. Position 属性

设置窗体在应用程序中显示时的位置和大小。有六种类型可供选择：

- **poDesigned** 由编程者在设计时指定窗体的位置和大小
- **poDefault** 由 Windows 确定运行时窗体的位置和大小
- **poDefaultPosOnly** 由编程者在设计时指定窗体的大小，而由 Windows 指定窗体的位置
- **poDefaultSizeOnly** 由编程者在设计时指定窗体的位置，而由 Windows 指定窗体的大小
- **poScreenCenter** 由编程者在设计时指定窗体的大小，运行时位于屏幕的中心位置，在有多个显示器的情况下，此位置自动调整到缺省的那个显示器中心处
- **poDesktopCenter** 由编程者在设计时指定窗体的大小，运行时位于屏幕的中心位置，对于多显示器的应用，不作调整

11. PrintScaled 属性

设置如何使用 **Form.Print** 方法进行打印。

12. Scaled 属性

有些情况下，系统的显示特性与设计时所使用系统的显示特性不同，此时，**Scaled** 属性设置确定窗体是否进行缩放。如果将 **Scaled** 设置为 **True**，则应将 **AutoScroll** 属性设置为 **False**。

13. Tag 属性

存储一个整数值。**Tag** 属性在 Delphi 中没有预先定义其具体的含义，因而设计者可以根据自己的需要作不同的用途。例如，对于同样的几个组件，可以定义一个数组来统一表示，用 **Tag** 来进行区分。

14. VertScrollBar 属性

设置竖直滚动条的属性，例如是否可见、滚动条颜色、滚动范围、位置、大小、增量、类型、滑块尺寸等。

15. WindowState 属性

设置窗体的初始状态。有三种状态可供选择：

- **wsNormal** 缺省的正常状态
- **wsMaximized** 窗体处于最大状态
- **wsMinimized** 窗体处于最小状态