

新世纪计算机基础教育丛书

丛书主编 谭浩强

# Delphi 程序设计

周果宏 罗述谦 罗起 编著

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

Delphi 是一种面向对象的可视化开发工具。它的特点是程序设计简单易学,所见即所得,执行代码速度快,程序开发效率高。作为一本系统的 Delphi 程序设计教程,本书对 Delphi 的基本知识,例如 IDE 集成开发环境、基本构件、Object Pascal 编程语言和数据库等进行了详细介绍;为注重实用,对可视化构件的使用以及制作新构件、多媒体及网络连接技术、制作安装程序等内容也进行了介绍,并且可帮助读者迅速接触实际应用;此外,还给出了类、OLE、COM 与 ActiveX 等概念。每章后均附有习题。

本书可作为高等学校及各类计算机培训班的教材,也可供读者自学参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Delphi 程序设计

作 者: 周果宏 罗述谦 罗 起 编著

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www .tup .tsinghua edu .cn](http://www.tup.tsinghua.edu.cn)

印刷者: 世界知识印刷厂

发行者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 23.75 字数: 546 千字

版 次: 2001 年 4 月第 1 版 2001 年 4 月第 1 次印刷

书 号: ISBN 7-302-04177-6 TP · 2465

印 数: 0001 ~ 5000

定 价: 29.00 元

# 第一部分 Delphi 入门

---

## 第 1 章 Delphi 概述

Delphi 是 Borland 公司的产品。它是一个运行在 Windows 下的可视化编程环境,可以创建 Windows 应用程序。Borland 公司以其高质量的软件开发工具独树一帜,其用户遍布世界各地、各个行业。1995 年 2 月, Borland 公司推出了 Delphi 1.0, 它提供给用户一种全新的可视化环境,用于快速开发 Windows 3.1 的应用程序。1996 年 1 月, Borland 公司正式宣布了 Delphi 2.0 的诞生。Delphi 2.0 是在 Windows 95 和 Windows NT 平台上的一套既方便又功能强大的开发工具,此外,还增加了如数据模块(data module)等多种高效的数据重用功能。在美国国家软件测试实验室 1996 年 3 月进行的软件评估中, Delphi Client/ Server Suite 2.0 获得了最高评价。Delphi 客户机/ 服务器开发工具在高性能的快速应用程序开发领域已经成为新的标准。随后, Borland 又推出 Delphi 的第三代版本——Delphi 3.0 多层 client/ server 的 32 位快速开发工具。

1998 年 8 月,在完成了必要的法律程序后,美国著名软件厂商 Borland 公司正式更名为 INPRISE 公司。1998 年 8 月 20 日, INPRISE 公司在北京的新闻发布会上发布了其备受关注的企业级开发工具——Delphi 4.0。该产品系列包括标准版、专业版、客户/ 服务器版三个版本,可全面支持 Oracle 8 对象关联扩展功能及 CORBA 分布式对象技术。本书以 Delphi 4.0 标准版为基础,介绍使用 Delphi 的基本知识和软件开发技术,力求使读者能快速掌握这一先进的面向对象的可视化开发工具。

### 1.1 Delphi 的主要特点

Delphi 不仅是编辑器和编译器,还是一种使用构件进行编程的工具,可使软件开发工作更加简单。限于篇幅这里只介绍最常用和最有用的几个特性,其中将涉及许多全新的技术和概念,读者不可能仅阅读本节便弄清所有概念,还需要在今后的反复学习、反复实践中逐渐加深理解。

Delphi 作为一种面向对象的可视化开发工具,其主要特点如下:

#### (1) 应用程序的快速开发

Delphi 是基于 Object Pascal 的开发工具,它具有高性能的 32 位本地优化代码编译器。其应用程序像用 C++ 语言编写的应用程序一样,可直接运行,能够最终生成可单

独执行的 DLL 与 EXE 文件。

实现了对 Win32 API, Multi-threading, OLE, OLEDB, COM, DCOM, ActiveX, ISAPI, NSAPI, MAPI, DirectX 等最新技术的完全支持。

充分支持 Windows 95/ NT 应用程序。

(2) 高效的可视化构件库与面向对象的架构。

(3) 具有集成的快速报表生成工具和集成的图表构件, 能将企业数据转换成决策信息。

(4) 能可视化地创建构件, 以及通过鼠标拖放生成构件模板; 具有对 Windows 95/ NT OLE Automation Controller 和 Server 的完全支持。

(5) 具有多种操作向导, 可加速程序编写和减少语法错误。

(6) 具有开放式数据库架构, 可轻松连接企业内的各式数据库结构。

(7) 具有集成的数据库开发工具、强大的客户/ 服务器运算开发功能。

(8) 具有可伸缩的多层面数据库架构, 便于维护和增加重用性。

(9) 具有 Web 数据库应用程序开发的能力。

### 1.1.1 Delphi 4.0 的一些新特性

Delphi 4.0 包含 10 个新的调试工具, 一个带有工程管理器的可定制集成开发环境, 以及对于 Windows 98/ 95 的完全支持。Delphi 4.0 的一些主要特点如下:

#### 1. 一步生成 COM 和 CORBA 对象

COM 是构件对象模型 (component object model) 的缩写。COM 是微软为生成和实现可重用构件所确立的标准。CORBA 是公共对象请求代理体系结构 (common object request broker architecture) 的缩写, 是分布式对象管理规范。在 CORBA 下, 服务器端应用程序实现的对象通过特定用户界面可以被客户端应用程序使用。与 COM 不同的是, CORBA 可以应用于 Windows 以外的其他平台。

CORBA 使开发人员摆脱了异构平台、异构数据库、继承应用等烦琐工作, Delphi 又使开发者和 IT 部门避开了 COM 和 CORBA 之间的差异带来的麻烦。Delphi 一步生成 COM 和 CORBA 组件的功能使应用程序可同时支持多种标准, 因此开发者可以集中精力于应用逻辑的开发, 而不会纠缠于烦琐的分布式 API 对象和协议中。

#### 2. 与分布式 Java 的完全互操作性

目前 Delphi 4.0 所生成的所有对象都符合 CORBA 标准。Java 应用和 Delphi 组件能在调用方式下互操作。使用全新的 MIDAS for Java 界面, Java 和 Delphi 开发者能在分布式环境下迅速分享信息。MIDAS 对 Java 的完全支持使 Delphi 开发者能迅速创建和 Java 应用立即互操作的分布式数据库应用, 从而在 NT, UNIX 和 Internet 应用之间架起了桥梁。

#### 3. 先进的分布式数据管理

在一个企业中, 对所有数据源提供全局访问需要一个复杂的网络和数据库驱动程序的组合。对于数据库驱动程序 (比如 ODBC), 存在仅支持数据库服务器中某一功能子集的弊病。Delphi 4.0 完全支持通过 ODBC 数据库的访问, 同时, 它还提供了性能优化

的本地驱动程序,提供对数据库的高效访问,并支持各种数据库的所有特性。Delphi 4.0 目前支持与 Oracle, DB2, Informix, Sybase, InterBase, MS SQL Server, dBASE, Paradox, Access 和 FoxPro 的本地连接,同时提供对 AS/400 的本地连接。

#### 4. 先进的开发工具

Delphi 简化了应用开发的复杂工作,能自动处理重复性的开发工作,大大提高了开发效率。Delphi 通过构件提高了代码的重用性,新的应用逻辑一经创建,就能打包成构件并在应用开发中使用。由于创建 Delphi 构件非常容易,因此 Delphi 的开发者可迅速得到对新技术的支持。

### 1.1.2 Delphi 产品系列

Delphi 为分布式应用开发提供了一整套集成开发工具。从功能上来讲,Delphi 产品主要有以下几种类型:

(1) Delphi 集成开发工具(IDE):现在,只有 Delphi 能够将用于编译本地代码的高效编译器、项目管理和团队(team)开发支持、可视化双向工具以及高级调试(debug)功能集成于一体。

(2) Delphi 可视化构件库(VCL):VCL 提供了成百个用于用户接口、数据库访问、数据处理、数据分析、报表和 Internet 开发的预建构件。

(3) Borland 数据库引擎(BDE)和数据库工具:除了具有数据库管理的多数功能外,BDE 还是一个可与数据库服务器进行高速连接的数据库引擎。

(4) MIDAS:MIDAS 是一个集成于 Delphi 的分布式应用的强大集合,能够确保对商业逻辑的智能集中管理和企业交易的高效处理。

## 1.2 Delphi 中的一些常用术语和概念

在学习 Delphi 的过程中,会遇到一些新的名词、术语和概念。为便于读者理解有关章节内容,这里对书中常见到的一些术语和概念作一简单介绍。

### 1.2.1 类、对象、属性与事件

关于类的概念,在第 6 章有详细介绍,这里仅给出一个笼统的定义。类定义了一个包含域(fields)、方法(method)和属性(property)的抽象数据结构。类的实例称做对象。例如, TObject 是一个类,它在 Delphi 中是所有其他类的祖先。 TComponent 也是一个类,它是 TObject 的后裔,也是在 Delphi 中所有构件对象的共同祖先。在设计阶段,放置到窗体上的每一个构件都是一个具体对象,是相应构件类的一个实例。例如,在某程序单元的类型说明部分有如下程序行:

```
Label1: TLabel;  
Button1: TButton;  
Edit1: TEdit;
```

这里, Label1, Button1, Edit1 是构件类的实例,是放到窗体上的具体构件;而 TLabel, TButton, TEdit 是构件类,它们都是 TComponent 类的后裔。

在 Delphi 中,程序设计所面临的一切事物都被视为对象(object)。按钮是对象,图标、窗口是对象,有些不可见的事物或事件也是对象。比如,异常(exception)就是对象。对象的结构由它们的类类型定义。类的域、方法和属性称为对象的元素或成员。Delphi 对象由方法组成,多数对象还有属性和事件。属性代表包含在对象中的数据。方法是指对象可以完成的动作,代表着一个对象中的过程,可以在该对象的外部被调用。事件是对象能够做出反应的条件,在这个特定条件下,执行与之相关的代码过程。

举一个普通的例子:一个窗体上包含两个按钮,它们都是 TButton 类的实例。由于它们都是对象,所以都有若干属性和事件。例如,它们各自的 Left 和 Top 属性值确定了它们在窗体上的位置;而它们的 Caption 属性可赋予不同的值,使得一个按钮标记 OK,另一个按钮标记 Cancel。给它们的 OnClick 事件句柄编写不同的代码,可使这两个 TButton 有不同的行为。典型的构件事件有 OnChange, OnKeyDown 等。典型的方法有 Hide, Refresh, BringToFront, Show 等,这些方法可以直接调用。例如:

```
Form1 .Hide;  
Form2 .Show;
```

## 1 2 2 构件

构件(components)是在项目设计时可以使用的、处理的可视对象。Delphi 用构件来辅助应用开发。构件是那些用于生成用户界面以及为程序提供附加功能模块的对象。在 Delphi 中,绝大多数构件都可以从集成开发环境中的构件板上找到。所有的构件都是可视构件库的组成部分。在这个构件类的大家族中, TComponent 类是所有其他构件类的共同祖先。TComponent 提供了一个构件能够在 Delphi 环境中工作所必须具有的属性和事件。虽然每一个构件都有自己的特性,但所有构件都分享了它们共同的祖先 TComponent 的特性。

放置到窗体上的每一个构件都将成为一个用户接口对象,或者一个数据库对象,或者是一个系统函数。只要把构件放置到窗体上,就创建了应用程序的用户界面。

构件包含三种类型的信息:

(1) 状态信息:是关于构件的目前条件,称为构件的属性。属性的值可以由用户或应用程序读取和设置。典型的构件属性有:Color, Height, Text, 以及上一节提到的 Left, Top, Caption 等。在程序中读取或设置属性值的例子如下:

```
Edit2 .Text := Edit1 .Text;  
Shape1 .Color := clBlue;
```

(2) 动作信息:构件通常可以完成一定的动作。这些动作由方法定义。方法实际上是一些过程(procedure)或函数(function),它们可以在程序中被调用,以告知构件要做什么。Delphi 构件有很多方法可以被调用。除上一节提及的例子外,常用的构件方法还有 Click(注意:不是 OnClick), Create, Destroy, DragOver, Free 等。

(3) 反馈信息: Delphi 提供了对发生在构件上的特定事件(event)编写代码的机会,这使得构件成为完全交互式的。通过对事件的响应, Delphi 为应用程序赋予了“生命”。构件的事件可以从对象观察器中找到,在第 2 章有详细介绍。

在可视构件库中约有 177 个预创建构件。用户可以使用这些构件来开发应用程序,也可以创建自己的新构件。创建新构件实际上是创建新的类,而应用程序开发人员使用的是类的实例。

### 1 2 3 可视化程序设计

Delphi 提供了一套可视化的工具,通过它们可以及时观察界面设计中的变化。这种设计过程基本上是“所见即所得”的。Delphi 4.0 的可视化开发环境将在第 2 章中介绍。

除了 Delphi 之外,还有相当数量的其他可视化开发环境。其中最流行的有 PowerBuilder (Sybase 公司出品), Visual Basic 和 Visual C++ (均为 Microsoft 公司出品)。Visual Basic 和 Delphi 十分相近。

### 1 2 4 面向对象的程序设计

面向对象方法的实质是强调从客观世界中的固有规律出发来构造系统,用人类在现实生活中常用的思维方式来认识、理解和描述客观事物。

编写面向对象的程序设计使开发人员能够重复使用工作成果并使程序维护更为容易,并且不易出错。原因在于它们都是很容易分离开来的对象。对象的继承性是指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辈对象的所有事件、属性及方法,并且在衍生对象添加新过程方面有很大的灵活性。

### 1 2 5 事件与事件驱动

事件就是所发生的事情。按动一个键是一个事件,移动鼠标是一个事件。计算机有一个内部时钟,时钟每跳动一次也是一个事件。一个事件就是程序应该响应的、由用户或计算机完成的事情。事件驱动的编程方法与传统的线性编程方法不同。

过去,程序员按如下形式编程:

```
begin
    做某件事;
    做另一件事;
    读取用户输入的数据;
    处理数据,执行下一个任务。
```

end

事件驱动的编程形式如下:

```
begin
    如果一个事件发生,那么,适当处理这一事件;
    否则,继续当前的(任何)活动。
    如果用户没有要求结束程序,则返回,并监视下一个事件的发生。
```

end

这就是事件驱动程序的概念。在此,事件驱动意味着等待某件事情的发生,如鼠标,单击一个按钮等,然后对这些事情做出反应。当然,也有一些事件是系统或程序员生成的。在 Delphi 中,一个由系统引发而无须用户介入的事件是属于 Timer 控件的 OnTimer 事件。该事件能够在达到程序员指定的时间间隔后自动启动。

## 1 2 6 Object Pascal 和 Delphi

什么是 Delphi? 简单地说,它是功能强大的 Pascal 编译器。Object Pascal 是 Delphi 的基础语言。而 VCL(可视化构件库)是 Object Pascal 对象集,这些对象可用来设计应用程序。因此 Delphi 是基于 Object Pascal 的可视化开发环境。

Delphi 使用 Object Pascal 作为底层编程语言。Object Pascal 是一种高级的编译语言,它支持结构化和面向对象编程。它的代码易读性强,编译快,使用多个 unit 文件以支持模块化程序设计。Object Pascal 具有支持 Delphi 构件框架和 RAD(快速应用开发)环境的特性。程序员可以在 Delphi 的集成开发环境 IDE 中编写 Object Pascal 程序,使用可视构件库来创建应用程序。IDE 帮助开发者处理许多在建立工程文件和源文件中的细节问题,例如,检查和维持程序模块单元之间的依赖问题。

Delphi 应用程序通常被分成若干源代码模块,称做 units(单元)。每个程序单元存储在一个独立的文件中,并且分别编译。编译后的单元( DCU 文件)互相链接起来,生成一个应用程序。Delphi 的这种程序结构允许把大的程序分成若干小的模块,分别编写和编译;允许创建可以在不同程序间共享的程序单元库;允许其他开发者使用单元库,而不必提供源代码。在常规的 Pascal 程序中,所有的程序代码,包括主程序,都存储在后缀为 .PAS 的文件中。而 Delphi 用一个项目文件( .DPR)存储主程序,其他的程序代码存储在单元文件(后缀为 .PAS)中。每一个应用项目可以包含一个项目文件、一个或多个单元文件。

## 1 3 Delphi 4 0 的安装与启动

本节介绍 Delphi 4 0 的安装、启动的方法以及系统需求。它的安装与启动方法与其他 Windows 95/ 98 环境下的应用软件没有大的区别。

### 1 3 1 最低系统需求

安装 Delphi 4 0 的最低计算机系统需求如下:

- (1) Intel 486DX/ 66 MHz;
- (2) Microsoft Windows 95/ 98, 或 Windows NT 4 .0 (Service Pack 3 以上)操作系统;
- (3) 内存容量 32MB(RAM);
- (4) 硬盘空间 60MB(仅对紧凑式安装够用);
- (5) CD-ROM 驱动器;

- (6) VGA 或更高分辨率的显示器;
- (7) 鼠标器;
- (8) Microsoft Windows 95 或 Windows NT 4.0 以上版本的网络操作系统。

### 1 3 2 Delphi 的安装

Delphi 4.0 和它的安装程序都是 32 位应用程序,所以,必须安装 Windows NT 或 Windows 95/98,才能安装 Delphi 4.0。安装 Delphi 请遵循如下步骤:

- (1) 启动 Windows NT 或 Windows 95。
- (2) 把 Delphi 光盘插进光盘驱动器插槽。
- (3) 打开光盘上 Install 目录下的 INSTALL.TXT 文件,并仔细阅读。
- (4) 从 Windows 95 的控制面板上打开“添加/删除程序”对话框,单击“安装”按钮,然后单击“下一步”,在“运行安装程序”对话框中单击“浏览”按钮,找到光盘上 Install 目录下的 SETUP.EXE 程序,再单击“完成”按钮,便开始运行安装程序。
- (5) 遵照安装程序的提示进行操作。

安装结束时,程序会提示阅读 README.TXT。该文件中是有关 Delphi 的重要信息,首次安装时最好阅读它。

### 1 3 3 启动 Delphi

启动 Delphi 像启动其他 Windows 应用程序一样,可以采用如下三种方式:

- (1) 从“开始”菜单指向“程序 | Borland Delphi 4”程序组,然后单击 Delphi 4 可执行文件的图标,如图 1-1 所示。

菜单启动 Delphi

(2) 用 Windows 资源管理器 (Explorer) 找到可执行文件 DELPHI32.EXE,并双击它。如果安装 Delphi 时没有修改路径,那么该文件位于 \Program Files \Borland \Delphi4\Bin 目录下。

(3) 选择 Windows 95 的“开始”菜单中的 Run,在弹出的对话框中填入文件名 DELPHI32.EXE 及其全路径,然后单击 OK(或“确定”)按钮即可。

## 1 4 写作本书的目的

随着计算机技术的迅猛发展和更新,程序设计和软件开发的观念也发生了根本的变革。从结构化程序设计到面向对象的可视化程序设计;从用户不能控制程序的执行流程到事件驱动、用户控制程序。面向对象编程(OOP)是结构化编程的自然扩展,它使程序设计更为简单易行,便于移植和维护。对象的可重用性不仅节省了程序员的时间,而且提高了工作效率。

写作本书的目的是为高等院校的学生和广大科技工作者提供一本学习面向对象可视

化编程的教材,Delphi 无疑是最好的选择之一。与其他同类开发工具(如 Visual Basic, PowerBuilder 等)相比,Delphi 在快速应用程序开发、可伸缩的数据库连接功能、可重用对象以及 Web 数据库应用开发能力等方面有其独特之处。

本书将侧重于阐明学习 Delphi 中需要了解的一些基本概念,介绍用 Delphi 进行软件开发的基本方法和编程技能;同时,也深入浅出地介绍了 Delphi 4 .0 的新构件、新功能。根据作者的教学实践,一个非计算机专业的学生可以在很短的时间内初步掌握这一得心应手的开发工具,有了这样的基础,在实践中继续深入钻研,成为一名优秀的专业软件开发人员也是不难实现的。

## 第 2 章 Delphi 4.0 集成开发环境及应用程序设计

在以往的程序设计环境中,创建应用程序用户接口是一项烦琐的工作。为了设计用户操作界面,必须为每一个屏幕元素设置每一项属性,这需要在编辑器中输入代码然后再进行编译,将成千上万条代码转换成可执行程序,还要查看接口运行结果,修改后再次编译运行,并使用其他工具对程序进行调试。这不仅费事费力,而且常常错误百出。Delphi 的集成开发环境(IDE)可以解决上述问题。该系统使用一组窗口、菜单和相关应用程序来设计可视化界面,将代码与每一项屏幕元素相连,在设计过程中所见即所得,从而使应用开发变得容易、快速而有趣。

本章通过一个简单的应用实例讲述 Delphi 集成开发环境及其操作方法。对从未使用过 Delphi 的人来说,本章内容特别重要。通过本章的学习,可以初步掌握使用 Delphi 的方法。此后边实践边学习,就可逐步掌握其编程技巧。

### 2.1 Delphi 4.0 集成开发环境

每次启动 Delphi 时,屏幕上都会显示 4 个窗口,这就是 Delphi 集成开发环境(Delphi integrated development environment),如图 2-1 所示。

图 2-1 Delphi 4.0 集成开发环境

其中包括:

- (1) 标题为“ Delphi 4-Project1 ”的 Delphi 主窗口(图 2-1 上部)。
- (2) 标题为“ Object Inspector ”的对象观察器窗口(图 2-1 左下部)。
- (3) 标题为“ Form1 ”的窗体窗口(图 2-1 右下部)。
- (4) 标题为“ UNIT1 .PAS ”的代码编辑器以及项目管理器窗口,刚启动时其大部分都隐藏在窗体窗口之下。

#### 2.1.1 主窗口

Delphi 的主窗口位于屏幕的上端,包括菜单、快捷条和构件选项板(图 2-2)。快捷条位于 Delphi 主窗口中的菜单下面,包括 16 个执行各种动作的按钮,这些按钮是菜单功能的快捷方式(图 2-3)。构件选项板位于快捷条右端。构件选项板内包含在 Delphi 应用开发中可以利用的大部分构件。这些构件按照其相似的功能分为若干页面。构件选项板的上端是页面标签。用鼠标单击这些标签,可以看到每个页面下都有若干不同的构件。

图 2-2 中所示是 Standard 页面中包含的构件。其中有在应用程序的用户界面中制作主菜单的构件、制作弹出式菜单的构件、标签构件、按钮构件等。这些构件板上的构件实际上是 Delphi 预先编制好的一些功能模块,就像建筑工地上不同形状、不同功用的预制板一样。它们可能是用户界面对象,或是数据库对象,或是系统函数(比如,系统定时器)。进行软件开发时,用户利用构件选项板来选择需要的构件并放到窗体窗口中,再通过对象观察器设置它们的属性值,并在单元窗口中对在程序运行过程中可能发生在它们身上的事件编写事件句柄代码。这就是在 Delphi 集成开发环境中开发应用程序的简单概念。

下面介绍 Delphi 4.0 主窗口中的快捷条上各按钮的功能和有关概念。

#### hi 4.0 快捷条

(1) 新选项(New)按钮:打开 New Items 对话框(图 2-4)。该对话框提供 Delphi 对象库的各类对象和模板,这里一一介绍。一个普通的用法是,通过它向项目中加进新的窗体(form)或新的单元(unit)。

#### 选项对话框

(2) 打开文件(Open)按钮:调出打开文件对话框打开一个已有的项目、窗体、单元、或文本文件。

(3) 保存(Save)按钮:保存当前文件,快捷键为 Ctrl + S。

(4) 全部保存(Save All)按钮:保存全部打开的文件,包括当前的项目和模块。

(5) 打开项目(Open Project)按钮:这个按钮的图标是一个从硬盘驱动器指向文件夹的箭头。允许打开一个项目。在 Delphi 中开发的一个应用程序称为一个项目。一个项目可以包含多个窗体(forms)和单元文件(units)。这个按钮等同于 File 菜单中的 Open Project,其快捷键为 Ctrl + F11。

(6) 添加文件(Add file to project)按钮:用于向项目中加进一个已存在的单元文件和与之关联的窗体。进行这项操作后,Delphi 自动在项目文件的 uses 语句中加进这个单元文件的名字。

(7) 删除文件(Remove file from project)按钮:允许在单击这个按钮后弹出的对话框中选择一个单元(及其相关的窗体),然后单击 OK 按钮,把它从项目中删除。Delphi 自动把这个单元文件从 uses 语句中删除,但是并不从磁盘上删除该文件。

(8) 帮助(Help)按钮:打开帮助窗口。

(9) 考察单元(View Unit)按钮:从相应对话框中选择当前项目的一个单元窗口打开(快捷键为 Ctrl + F12)。

(10) 考察窗体(View Form)按钮:从相应对话框中选择当前项目的一个窗体,使之

成为当前窗口(快捷键为 Shift + F12)。

(11) 转换窗体/单元(Toggle Form/ Unit)按钮:在当前窗体和它的单元窗口之间转换(快捷键为 F12)。

(12) 新窗体(New Form)按钮:在当前打开的项目中加进一个空白窗体。

(13) 运行(Run)按钮:编译并运行应用程序(快捷键为 F9)。

(14) 暂停(Pause)按钮:暂时中止执行一个正在运行的程序。

(15) 跟踪(Trace Into)按钮:调试器执行被运行指针标识的一行代码。如果运行指针指向一个函数调用,则调试器将运行指针移向定义这个被调用函数的第一行代码。如果被调用函数不包含调试信息,则调试器按正常速度运行该函数(快捷键为 F7)。

(16) 单步(Step Over)按钮:按一次执行一程序,用于调试过程(快捷键为 F8)。

## 2.1.2 构件选项板和构件

Delphi 4.0 构件选项板包含大量构件。用户可以增加、删除和重新安排构件板上的构件,也可以创建由多个构件组成的构件模板,并把这模板加到构件选项板上。构件选项板按相似的功能将构件分组为不同页面。表 2-1 列出了 Delphi 4.0 构件选项板的标准页面和它们所包含的构件类型,有关这些构件的详细信息可参阅本书第 4 章。

表 2-1 标准的 Delphi 4.0 构件选项板页面

页面名称	内容
标准(Standard)	标准 Windows 控件,菜单
附加(Additional)	定制控件
Win32	Windows 95/ NT 4.0 常用控件
Internet	用于 client/ server 应用的构件: Internet 通信协议、处理低级 TCP/ IP 的 sockets、以及用于构建 Web 服务器应用的对象
数据库读写(Data Access)	用于读写数据库的不可视构件: tables, queries, reports
数据库控件(Data Controls)	可视的数据控件
对话框(Dialogs)	Windows 常用对话框
系统(System)	系统级访问构件和控件,包括定时器、文件系统、多媒体、DDE
ActiveX	几个可用的 OLE 控件
Win3.1	与 Delphi 1.0 项目兼容的构件
示例(Samples)	定制构件示例
快速报表(Qreport)	快速报表构件,用于快捷生成内置报表

构件是在 Delphi 中构建应用项目的基本元素,它们包括一个项目的所有可视部分,如对话框、按钮、窗体、数据库的字段;也包括那些运行时不可见部件,如系统定时器,或动态数据交换(DDE, dynamic data exchange)服务器。通过选择和连接这些元素就建成了自己的应用界面。从面向对象程序设计(OOP)的观点来看,构件也就是对象。构件包含

三种重要信息:

(1) 状态信息: 又叫属性(property)。它表明构件的当前状态条件。用户可以对这些属性进行阅读和设置。典型的构件属性如高度和颜色。

(2) 动作信息: 构件通常能按照命令完成一定动作。这些动作由方法(methods)定义。方法也就是过程(procedure)或函数(function), 可以在编程时调用它们, 以告知构件做什么。典型的方法实例是 Hide 和 Refresh。

(3) 反馈信息: 构件给程序员提供了对事件(event)进行编码控制的可能, 这样构件可以对事件进行响应。典型的构件事件是 OnClick 和 OnKeyDown。

添加到窗体上的构件的属性和事件可以从对象观察器中进行观察和操作。

### 2.1.3 对象观察器窗口

对象观察器窗口(object inspector)位于主窗口的左边, 快捷条下面。它包含一个组合框、一个 Properties(属性)页面和一个 Events(事件)页面。在属性页面里显示着有关窗体窗口中当前被选择对象的属性信息, 并允许用户改变该对象的属性。利用事件页面, 可以配合编写程序代码来定义对象接收到某个事件时要执行的动作。首次启动 Delphi 时, 对象观察器窗口中显示着当前窗体的属性(见图 2-1 左边)。从位于该窗口顶部的组合框中选择其他对象, 可以改变当前选中的对象。也可以在 form 窗口内用鼠标单击选择所要的对象, 以改变当前对象。每选择一个新对象, 对象观察器窗口的内容都会更新。图 2-5 为在窗体中用鼠标选中了 Button1 的情形, 对象观察器中显示的是 Button1 的属性和事件。

选中 Button1 的界面

### 2.1.4 窗体窗口

窗体窗口(Form)(图 2-1 右下)是最常用的构件, 也是特殊的对象, 窗体窗口是开发应用程序的载体, 可以把各种所需构件(如按钮、标签、对话框等)放置在窗体窗口中, 并可任意排列。以后在程序运行时, 出现在用户面前的窗口基本上和设计时看到的情形相同。如果把 Delphi 应用程序比做建筑物, 则窗体即是这个建筑物的地基。虽然窗体也是构件, 但它不包含在构件选项板上。如前面所述, 向项目中加进新的窗体, 可以使用快捷条上的“新窗体”按钮, 也可以在 Delphi IDE 主窗口的 File 菜单中选择 New Form。

### 2.1.5 单元窗口

启动 Delphi 时, 与 Form1 同时产生的, 还有一个代码编辑器窗口, 叫做 Unit1, 它包含了用户所创建窗体的源代码。Delphi 的一大特点是程序员基本上不必动手编写用户界面代码。虽然仍有人工编码, 但构件被放到窗体窗口时, Delphi 会自动生成大部分用户界面代码。剩下要做的工作是编写把这些构件结合起来的“粘接剂”代码, 以及完成某些特

殊功能的程序模块,或对特定事件(如用户按下某个按钮)进行响应的事件句柄(event handler)代码(图 2-6)。

#### 事件句柄中写进一行代码

Unit1 和 Form1 被存储为两个独立的文件,Unit1 窗口中的代码被存于文件 UNIT1 .PAS 中(PAS 源于 Pascal)。Form1 文件含有屏幕上画出的窗体和它所包含的构件的描述,叫做 UNIT1 .DFM(DFM 源于 Delphi Form)。当然,用户可以更改 Unit1 这个名字。

## 2.1.6 项目管理器

除了图 2-1 中所示的 4 个窗口外,Delphi 软件开发环境的另一个有用工具是项目管理器(Project Manager)。用 Delphi 开发的应用软件称做“项目”,即 Project。一个项目中可包括多个窗体和多个单元。通过项目管理器,可以跟踪和读写组成 Delphi 项目的文件。在打开了一个项目的前提下,从 Delphi 主窗口的 View 菜单选择 Project Manager,会打开项目管理器窗口(图 2-7)。Delphi 4.0 的项目管理器窗口与先前版本形式有所不同,它将相互依赖的项目结合为一个项目组,其中分层列出组成应用软件的全部窗体和单元。它类似于 Windows 95 中资源管理器的形式,可对项目进行管理。

#### 管理器窗口

用项目管理器可以浏览当前打开的项目所包含的文件的的内容和状态信息。如图 2-7 所示,鼠标指向任一个对象,单击鼠标右键,会弹出一个与之相关的操作选项菜单来,从而可以在项目组中增加或删除项目,在项目中增添或删除代码单元及其相应的窗体等。也可使用弹出式菜单来打开单元或窗体窗口。窗口上面的 New 和 Remove 快捷按钮也可用于向项目中增添或删除代码单元及其相应的窗体。当项目组中包含两个以上项目时(例如图 2-7 中包含 StasisP .exe 和 Project1 .exe 两个项目),Active 按钮用于激活被选项目,以便对其进行操作。

要查看一个单元,只要从项目管理器中双击这个单元的图标或文件名字。要查看一个窗体,也可用同样的方法。

当项目中包含的文件很多时,要查找一个特定的文件,只要在键盘上逐个字母键入文件名字,项目管理器便会指向相匹配的文件。如果要在不同的项目之间共享文件,则最好使用项目管理器,因为它可以显示每一个文件在项目中的位置。

通过项目管理器可以把协同工作的项目结合为一个项目组。从而使用户可以方便地组织和工作在相互依赖的项目中,如 DLLs 和若干可执行文件。

## 2.1.7 快捷命令菜单

快捷命令菜单提供了使用当前可用命令的快捷方式。要显示快捷命令菜单,需将鼠

标指在一个选定的对象上,单击鼠标右键即可。图 2-8 中窗体上的弹出式菜单列出了对于 DBGrid 构件可用的命令,这个菜单是用左键选中 DBGrid 这个构件,再单击右键而弹出的。

## 快捷菜单

在 Delphi 集成开发环境中,窗体、构件板、对象观察器、快捷条和项目管理器都支持快捷命令菜单。

## 2.1.8 数据库管理器

从 Delphi 4.0 的主菜单中选择 DataBase | Explorer,将打开 DataBase Explorer 或 SQL Explorer 窗口(即数据库管理器)。数据库管理器是一个用来对数据库类对象(如数据表、字段、存储过程、触发器、参照关联和索引等)进行浏览和修改的图示管理器,以使用户在开发过程中与远程数据库建立紧密联系。数据库管理器提供以下功能:

- (1) 创建和维护数据库别名(database aliase)。
- (2) 查看数据库中的图示数据,如数据表、存储过程、触发器。
- (3) 查看数据表对象,如字段、索引等。
- (4) 生成、查看和修改数据表中的数据。
- (5) 键入 SQL 命令对任意数据库进行直接检索。
- (6) 创建和维护用于储存属性设置的数据字典。

如图 2-9 所示,在 SQL Explorer 窗口的左半边 Databases 页面中列出了硬盘上所有已定义的有效数据库别名。选中一个别名,窗口右半边会显示该数据库别名的有关定义。要连接到某个别名所指定的数据库,可用如下两种方法之一:

- (1) 在数据库管理器窗口中选中这个别名后,再选择 Object | Open。
- (2) 从数据库管理器的快捷按钮中选择 Open。

当已经连接到一个数据库时,它的别名的图标被一绿色方框所环绕。从图 2-9 可以看到,DBDEMOS 和 Jwuchu 两个数据库别名图标的背景呈灰色(因为不是彩色打印)。

## 数据库管理器

在数据库管理器中逐层单击对象左边的“+”号,展开其中所包含的对象,然后选中一个对象(数据库,表,字段,索引等),便可在窗口右半边查看其内容和结构定义。图 2-10 所示为在窗口左半边选中 DBDEMOS 中的二维表 animals .dbf,窗口右半边的三个页面中包含这个对象的结构定义及所包含的记录(数据),用户可对它们进行修改,并可利用 Enter SQL 页面键入一条 SQL 语句,来对数据表进行查询。

要实现 SQL 操作,需要选择 Enter SQL 页面标签,在下面的编辑区域键入 SQL 语句,然后单击 Excute Query 按钮。SQL 语句执行后,其结果显示在下面的数据表格中。图 2-11 是一个 SQL 语句执行后的情况。

中执行 SQL 语句的例子

数据库管理器还有许多其他功能,许多操作都可通过弹出式菜单中的选项进行。

## 2.1.9 在线帮助

Delphi 的在线帮助提供了大量的 Delphi 使用信息。从主窗口的帮助菜单 (Help) 中可以获得使用 Delphi 的在线帮助内容和关于 Windows 95 API 的参考材料。帮助窗口如图 2-12 所示。

### 在线帮助

从 Delphi 4 程序组也可以直接打开帮助窗口。从 Windows 95/98 桌面状态栏上单击“开始”,指向“程序”|“ Borland Delphi 4 ”|“ Help ”后,可看到 Help 下面还有大约 17 个子菜单项。要查找程序设计过程中遇到的一般问题的帮助,可在子菜单中选择 Using Delphi。要查看专项题目的帮助,则可从子菜单中选择合适的标题。选择 Visual Component Library Reference,可查看 Delphi 可视构件库所包含的所有构件和例程(函数)的帮助信息。选择 Creating Custom Components,可查看关于创建新构件的帮助信息。单击帮助窗口上端的“索引”页面标签,显示索引页,可在页面顶端的编辑框中输入关键词,按关键词查找有关帮助内容。

## 2.2 一个简单的 Delphi 应用程序设计

Delphi 程序开发从根本上不同于以前形式的程序开发。一个 Delphi 程序首先是一个程序框架。这就是当启动 Delphi 时,或从 Delphi 主窗口的 File 菜单中选择 New Application 时,所出现的 Form1 和 Unit1(窗体窗口和代码编辑器窗口)。从构件选项板上选取构件到窗体上,再对构件的属性进行设置,编写事件句柄代码,使各构件对不同事件做出不同的反应,完成相应功能,这就是在 Delphi 中开发应用程序的大体概念。Delphi 易学易用。下面以一个简单实例,让读者获得一些用 Delphi 进行程序开发的直观印象和入门知识。

这个例子是一个关于数据库应用的小软件,它用来帮助某大学研究生部管理本部门的财务账目。假如已用 Paradox 建立了一个数据库,也就是说,已经建立了若干个二维表 (Tables)。图 2-13 是二维表 Income .db 的字段结构,用于记录各项经费来源及金额。图 2-14 是二维表 Outcome .db 的字段结构,用于记录各项经费支出及金额。图 2-15 和图