高职高专“十一五”规划教材

Delphi 2005 程序设计实用教程

何定华　贺体刚　编著

化学工业出版社
·北 京·

需要全本请访问 www.ertongbook.com

本书详细讲述了 Delphi 2005 程序设计，内容包括：Delphi 基础知识，Pascal 语言，编程基础（包含程序的控制结构、枚举、子界和集合、数组与记录以及过程与函数），界面设计（包含窗体与基本组件、对话框、菜单、工具栏、状态栏以及图形图像程序设计）、文件编程，数据库编程（包含数据库基础、BDE 数据库应用程序开发、BDP 数据库引擎应用）以及 ASP.NET 编程。本书不仅内容丰富、翔实，而且重点突出。相信通过本书的学习，读者一定能够快速地掌握 Delphi 这门优秀的开发工具，并能轻松地开发出实用的软件和数据库应用程序。

作者将为购买本书的读者免费提供全书例题源程序，以方便读者讲课或者学习之用。

本书不仅可以作为高等学校教材，也可以作为社会培训班教材和参考书。对于那些希望快速学会 Delphi 2005 开发工具的初学者，本书也是一本不可多得的好教材。

图书在版编目(CIP)数据

Delphi 2005 程序设计实用教程 / 何定华，贺体刚编著.
北京：化学工业出版社，2008.7
高职高专“十一五”规划教材
ISBN 978-7-122-03123-5

Ⅰ. D… Ⅱ. ①何… ②贺… Ⅲ. 软件工具-程序设计-高等学校：技术学院-教材 Ⅳ. TP311.56

中国版本图书馆 CIP 数据核字（2008）第 090481 号

责任编辑：王听讲　　文字编辑：陈　元
责任校对：战河红　　装帧设计：刘丽华

出版发行：化学工业出版社（北京市东城区青年湖南街 13 号　邮政编码 100011）
印　　刷：大厂聚鑫印刷有限责任公司
装　　订：三河市前程装订厂
787mm×1092mm　1/16　印张 18½　字数 478 千字　　2008 年 8 月北京第 1 版第 1 次印刷

购书咨询：010-64518888（传真：010-64519686）　售后服务：010-64518899
网　　址：http: // www. cip. com. cn
凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：29.00 元

版权所有　违者必究

前　言

Delphi 是一种优秀的基于 Windows 的应用软件开发工具。它具有可视化的开发环境、简洁明快的编程语言、强大的数据库功能、可扩展的组件库等特点，Delphi 正在成为一种越来越成熟的软件开发工具。聪明的程序员使用 Delphi，这是业界广为所知的一句名言，因此 Delphi 正在赢得越来越多的用户。

本书针对较新的 Delphi 2005 版本，是 Delphi 入门级的学习教程。无论读者是否具有编程基础与编程思想，本书都是非常适合的教材与参考书。编者长期从事 Delphi 程序设计的教学工作，本教材是根据编者多年的教学经验编写而成的。本书具有如下一些特点。

1．结构合理

本书将 Delphi 基础、Pascal 语言等内容介绍得比较详细，很多内容都独立编写成章。这样可以充分照顾到入门者的实际情况，而对于有学习基础的读者，也方便他们对某些章节进行合理的取舍。本书例题丰富、讲述清楚、界面美观，力图深入浅出。本书各个章节的先后顺序也经过编者精心设计，读者还可以根据自己的喜好和习惯，改变本书学习和讲授的原有顺序。例如，可以将分支结构和单选按钮、复选按钮一起学习，将循环结构和列表框、组合框一起学习等。

2．内容丰富

本书内容丰富，它讲述了 Delphi 语言的基础知识，包括第 1 章、第 2 章、第 4 章、第 5 章、第 6 章和第 7 章。这些章节对于有编程思想和编程基础的读者可以适当进行一些取舍，教师在讲授的时候课时分配也可以酌情减少。第 3 章讲述窗体和基本组件，舍弃了一些比较复杂的组件，讲述的组件都是常用的和简单的，适合高校学生的实际情况，这些组件的讲述采用案例式的方法，配备有丰富的实例讲解，利于学生消化和吸收。第 8 章、第 9 章和第 10 章都是界面编程，所开发的软件界面美观是 Delphi 的一大特点。

3．重点突出

Delphi 的数据库功能非常强大，这是 Delphi 的特点。虽然本书内容丰富，讲述的知识点也比较多，但是本书的重点非常突出。数据库是本书的重点，本书花了大量篇幅讲述数据库知识。第 12 章讲述数据库基础，第 13 章讲述 BDE 数据库应用程序开发，第 14 章讲述 BDP 数据库引擎应用。各个章节例题丰富、讲述清楚，不仅有详细的代码而且还有可视化的工具软件操作，界面清晰。

4．配备教学资源

本书中的每一道例题，编者都保留了原始文件，且每道例题在 Delphi 2005 环境下都能正常运行，本书的所有例题源代码都免费赠送给读者，需要例题源程序的读者请与编者联系。编者联系方式，E-mail：hedinghua@qq.com，QQ：287125761。

本书由武汉职业技术学院的何定华和荆楚理工学院的贺体刚编著。何定华编写第 1 章～第 9 章，贺体刚编写第 10 章～第 15 章，全书由何定华统稿。本书可以作为高等院校教材和

参考书，也可以作为社会培训班教材。教师可以根据学生的实际情况对某些章节进行增删。

在本书的整个编写、出版过程中，我们参考了有关文献著作，得到了化学工业出版社的支持与帮助，我们在此特向有关作者和化学工业出版社的有关人员表示衷心的感谢。

由于时间仓促，加之编者水平有限，不足之处在所难免，恳请读者批评指正。

编　者

2008 年 6 月于武汉

第 1 章　Delphi 基础知识

Delphi 是快速应用软件开发工具 RAD（Rapid Application Development）。Delphi 使用了当今世界上很多最为先进的程序开发思想，使用 Delphi 开发软件无疑会大大提高软件开发的效率。Delphi 2005 是针对微软公司 Framework 技术而开发的，其目的是为了使 Delphi 开发者能够轻松使用多种语言（如 C#语言和 Pascal 语言）开发.NET 应用。

从 1995 年到现在，Delphi 经历了多代发展历程，本书采用较新版本 Delphi 2005 作为开发平台。

1.1　Delphi 2005 简介

Delphi 2005 是 Windows 系统下的可视化集成开发工具，它提供了强大的可视化组件 VCL（Visual Component Library）功能，使程序员可以快速、高效地开发出 Windows 系统下的应用程序。和其他软件开发工具相比 Delphi 在网络编程、数据库编程、程序界面方面更胜一筹，而且增加了对 Linux 平台下应用程序开发的更强大的支持。Delphi 2005 使用的语言是 Object Pascal 语言，但是它已经支持 win 32 和.NET 框架了。

1.1.1　Delphi 的优点

1．可视化的集成开发环境

Delphi 提供了可视化的集成开发环境 IDE（Integrated Development Environment）。可视化环境是指用户在设计程序界面的时候，无需为程序界面编写代码，只需要将相应的组件添加到窗体上，调节其大小和位置即可，Delphi 会自动生成相应的代码。Delphi 的集成开发环境不仅可以非常方便地设计出用户的程序界面，而且程序的界面设计、属性设置、代码编写、程序调试、运行、生成可执行文件等操作都可以在这个集成开发环境中进行，使得软件设计变得非常快捷、高效。

2．真正的面向对象

面向对象的程序设计 OOP（Object Oriented Programming）是 Delphi 诞生的基础。像 Visual Basic 这样的语言是伪面向对象的，它不支持封装、继承性和多态性等面向对象的技术。相对于 Visual Basic 等语言而言 Delphi 是完全面向对象的程序设计。它以 Pascal 语言为基础，通过将常用功能封装为组件或者类，获得了极高的程序开发效率。

3．可扩充的 VCL 库

VCL（Visual Component Library）组件库是 Delphi 的重要的组成部分。Delphi 提供了丰富的组件库，这些组件是程序开发强有力的工具。Delphi 的 VCL 组件库具有良好的可扩充性，它允许程序使用第三方组件，用户将第三方组件添加到 Delphi 中，然后就可以像使用自带的 VCL 组件一样方便地使用这些组件了。

4．高效的、完全的编译器

许多 Windows 应用开发工具采用不完全编译或者生成伪代码。伪代码是机器不可执行代码，它必须在运行时翻译成可执行代码，这就大大降低了系统的性能。Pascal 编译器以编译速度快而

需要全本请访问 www.ertongbook.com

著名，Delphi 正是建立在此基础之上的，Delphi 的编译速度是其他开发工具无法比拟的。Delphi 使用的是完全编译器和连接器，产生 100%的本地可执行代码。

5．强大的数据库开发功能

Delphi 提供了 DataBase Desktop 和 BDE Administrator 等工具，使得利用 Delphi 开发数据库应用程序变得非常容易。

提供访问 Inter Base、Oracle、SQL Server、Sybase、Infomix 和 DB2 数据库服务器的 SQL Links BDE 驱动器，并且允许无限制地分发这些驱动程序。

SQL 数据库浏览器可以浏览和编辑特定服务器的数据。图形化查询建立工具 SQL Builder 具有较强的数据处理和转换输出能力。SQL 监视器可以监视与 SQL 服务器的通信，从而可以调整 SQL 应用程序的性能。

6．支持分布式数据库开发

Delphi 支持多种分布式模式的开发，从简单的消息通信程序到庞大的多层次应用。在 Delphi 中可以方便地建立客户机-服务器结构的二层分布式应用，还可以方便地建立客户机-应用服务器-数据库服务器结构的多层分布式应用程序。

7．Web 技术扩展

Delphi 从 AtoZed 软件公司引入了 IntraWeb 技术，IntraWeb 类组件可以使用户以窗体工具的形式使用 IntraWeb 技术来设计 Web Broker，Web Snap 以及 Web 服务器应用程序。

Delphi 还支持 Apache 2 Web 服务器作为 Web Broker、Web Snap 和 SOAP 的目标环境。

8．对开发 Linux 平台应用程序更强大的支持

Inprise 公司最新的 Linux 平台上的 RAD 工具 Kylix 是 Delphi 的 Linux 版。从集成开发环境上比较，Kylix 与 Delphi 最大的区别在于 Kylix 采用的是支持跨平台的 CLX 组件库，而 Delphi 采用的是 VCL 类库，CLX 与 VCL 有很多相似之处，但 CLX 不仅是一个可视化组件库，更是一个跨平台的组件库。Kylix 上所使用的支持跨平台的 CLX 组件库是基于 Qt 的，而 Delphi 采用的 VCL 类库是基于 Win 32 API 的。Borland 公司称，在下一个 Delphi 版本中，将会同时支持 CLX 和 VCL，就是说 Delphi 开发的应用程序只需要进行必要的修改就可以移植到 Linux 平台上。

9．支持.NET 技术

Delphi 2005 支持 ASP.NET 技术和 ADO.NET 技术。用户使用 Delphi 2005 可以非常方便地开发出 Web Application 和 Web Services 程序，还可以开发出 ADO.NET 数据库应用程序。

1.1.2 关于对象的一些基本概念

在面向对象的编程（OOP）中，对象是一个比较时髦的词汇，使用非常频繁。和 Visual Basic 等语言相比，Delphi 是真正的面向对象的程序设计语言。那么什么是对象呢？我们可以把对象理解成为一种编程结构，这种编程结构将数据和函数封装成一个单元，而外界可以通过对象的接口（对象的属性、方法和事件）来访问对象。在本节，将主要讲述对象、属性、方法和事件。

1．类和对象

对象（Object）和类（Class）是在面向对象的编程中经常使用到的概念。类是对象的抽象和概括，而对象则是类的一个实例。比如说人类就是一个类，而张三则是一个具体的实例，张三就是一个对象。人类就是对很多人的一个抽象。如果把类细分还可以分为基类、派生类等。比如人类可以分为黄色人种、黑色人种、白色人种和棕色人种等。那么各个人种又可以看成是不同的几个类。这些类是人类的派生类，它们具有人类这个类的一些特性（属性），

也就是说这些类继承了人类这个类的属性、方法和事件。张三是黄色人种这个类的一个实例，张三是一个具体的对象。Tom 是白色人种的一个实例，Tom 也是一个对象。每个类都有自己的属性、方法和事件。

Delphi 中的对象有标签、按钮、列表框、图像框等。

2．对象的属性

属性（Property）是用来描述对象的。比如对象的高度、宽度、颜色、位置等。属性的表现形式为数据。比如说张三的高度为 172cm，张三的体重为 65kg，张三的皮肤颜色为黄色等。可以这样来表示：

```
张三.高度:=172;
张三.体重:=65;
张三.皮肤颜色:="黄色";
```

一般来说给一个对象的属性赋值的格式为：

```
对象.属性:=属性值（或者是表达式）;
```

以 Delphi 中的对象为例来说明如何为对象的属性赋值。例如，设置标签 Label1 的显示内容为“属性举例”。标签 Label1 是对象，标签有一个属性是 Caption，表示在标签上要显示的内容。因此，可以这样设置属性：

```
Label1.Caption:="属性举例";
```

在 Delphi 集成环境中，当用户输入对象的名称 Label1 后，打一个点号，则系统马上弹出一个下拉列表框，在这个下拉列表框中显示了 Label1 这个类的所有属性和方法等内容。用户只需要在里面选择即可，非常方便。

Delphi 中常见的属性有 Caption（标题）、Name（名称）、Font（字体）等。请看下例：

```
Label1.Caption:= '聪明的程序员使用 Delphi。';
Label1.Font.Size:=10;
Label1.Font.Name:= '隶书';
```

3．对象的方法

方法（Method）是封装在对象中的一段代码，用来实现预先规定好了的功能。方法是对象能够执行的动作，它由 Delphi 内部定义，用户不用编写代码即可实现某种功能，如 Show（显示）、Move（移动）、Line（划线）。不同的对象有自己的方法集合。在程序中调用方法的语法为：

```
对象.方法（方法需要的参数）;
```

例如以一个 Delphi 对象为例，调用编辑框的方法 SetFocus 让编辑框得到焦点。代码为：

```
Edit1.SetFocus;
```

4．对象的事件

事件（Event）是作用在对象上，并且能够被对象识别的动作。用户不能建立新的事件，

为此 Delphi 提供了大量的事件，以满足程序员的需要。比如，用鼠标点击命令按钮，命令按钮能够识别这个动作，并且程序会立即去执行存放在命令按钮的单击事件过程里面的代码，这样就完成了用户和程序之间的交互。

在面向过程的程序设计中，程序是按照预先设置好的顺序执行的。这就意味着程序与用户之间的交互相当有限。在面向对象的程序设计中引入了一个新的概念，也就是事件驱动机制。程序的执行不再按照某个固定不变的顺序进行。程序代码放在事件过程（Event Procedure）中。一个事件发生后，程序会自动执行对应的事件过程。

事件的触发就是事件是怎样发生的。事件的触发有以下 4 种方式。

① 用户通过交互方式触发事件。比如，用户单击命令按钮就触发了命令按钮的单击事件。

② 时间触发。在 Delphi 中有一个组件 Timer，它有一个属性 Interval，这个组件能够每隔 Interval 毫秒自动触发 OnTimer 事件。

③ 系统触发。例如，OnCreate 事件在窗体加载到内存的时候自动发生。OnClose 事件在窗体从内存中卸载的时候自动触发。

④ 可以用代码调用使事件发生。例如执行 form1.Close；语句后程序自动触发 Onclose 事件关闭窗体。

1.2 Delphi 2005 集成开发环境

Delphi 的集成开发环境 IDE（Integrated Development Environment）是设计程序界面、编写程序代码、调试程序、生成可执行文件的软件环境。Delphi 2005 的集成开发环境使得开发 Delphi 应用程序变得非常方便。

1.2.1 Delphi 集成开发环境组成

Delphi 集成开发环境由多个部分构成：主窗口、窗体设计器、对象查看器、结构浏览器、组件栏、代码编辑器等构成，如图 1-1 所示。代码编辑器没有显示出来，双击窗体设计器可以进入代码编辑器。

集成开发环境的各个部分是一个有机体，它们协同工作。

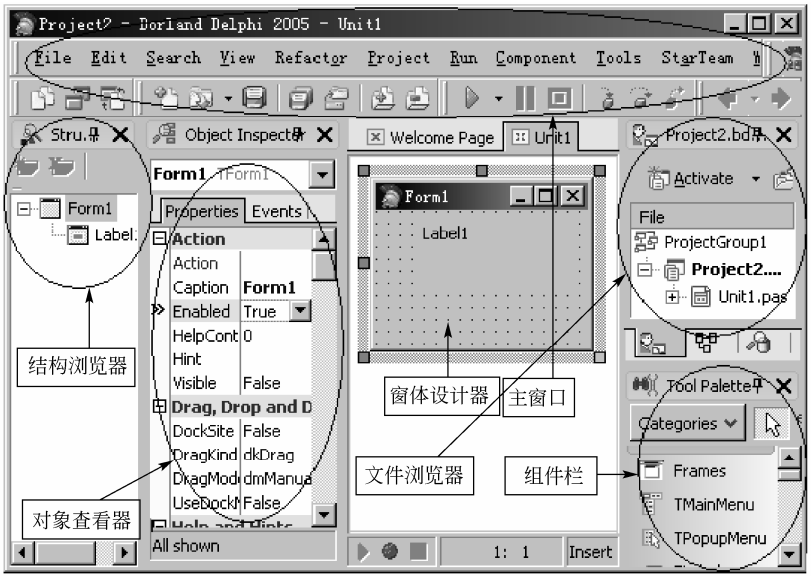

图 1-1 Delphi 2005 集成开发环境

1．Delphi 主窗口

Delphi 主窗口是 Delphi 集成开发环境的控制中心，它具有 Windows 应用程序的风格，Delphi 主窗口包括 3 个部分：标题栏、主菜单和工具栏。

（1）主菜单　主菜单提供了所有进行程序开发所需要的命令和功能，包括创建、打开、保存各种类型的文件、编辑程序、视图查看、修改选项等功能。Delphi 共有 12 个主菜单项，参见表 1-1 所示。

表 1-1　Delphi 的主菜单项

主菜单名称	主 要 功 能
File	新建、打开、保存各种文件
Edit	编辑项目中的各种文件
Search	实现文本的查找
View	打开项目管理器、对象查看器、显示调试信息、查看窗体、单元，设置工具栏
Project	管理项目文件
Run	提供运行、调试、中断等各种与运行有关的命令
Component	安装第 3 方组件、设置组件面板
Refactor	用于实现 Delphi 重构功能
Tools	提供 Delphi 开发环境的设置
Starteam	用以检视软件开发的各个阶段
Windows	切换 Delphi 中的窗口
Help	提供 Delphi 帮助信息

（2）工具栏　Delphi 工具栏（ToolBar）上的每个按钮都可以在菜单中找到，把菜单中经常使用的菜单项做成工具栏的形式，可以使用户操作更加方便，用户无需在菜单中查找菜单，只需直接单击工具栏按钮即可。Delphi 提供的工具栏有 6 项，分别是：Standard、View、Debug、Custom、Browser 和 Desktop。如图 1-2 所示为 Delphi 的工具栏。

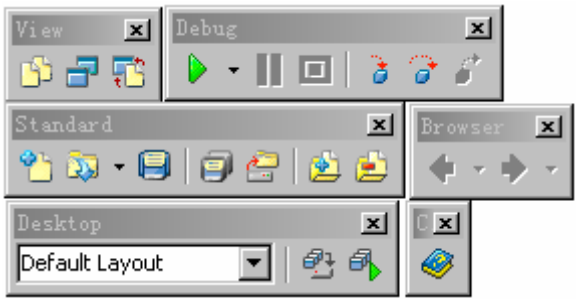

图 1-2　工具栏

2．Delphi 窗体设计器

Delphi 窗体设计器是设计应用程序界面的工作区域，运行时程序界面完全与设计时的界面一样。系统自动产生用户界面代码，用户只需要完成相应功能的程序段。

窗体在运行时称为窗口，一般来说一个应用程序不止一个窗口。窗口是由标题栏、工作区构成。标题栏给出窗体的标题，右边还有控制按钮，用于在运行期间关闭、最大化、最小化窗口。工作区是添加组件的位置，工作区上有栅格（Grids），用于组件对齐使用，栅格的密度可以调节，在运行期间栅格是不可见的。

3．对象查看器（Object Inspector）

利用对象查看器可以设置窗体中各个组件以及窗体的属性，或者使各个组件或窗体响应

不同的事件。选择菜单“View | Object Inspector”命令可以显示对象查看器。

属性（Property）是对象的一些数据，用于描述对象的颜色、大小、字体。

事件（Event）是一种消息处理机制，它能够捕捉某种动作并做出一些响应。例如鼠标单击事件、鼠标移动事件、窗体改变大小事件。

对象查看器总是显示当前选中的组件或者窗体。对象查看器的上面是一个对象组合框，用户可以在这个组合框中选择不同的对象。对象查看器是多页结构，包括属性页和事件页。点击选项卡可以在两个页面之间进行切换，如图 1-3 所示。其中，最上面的组合框是对象组合框，对象组合框用于选择不同的对象，左边图为属性页，右边图为事件页。

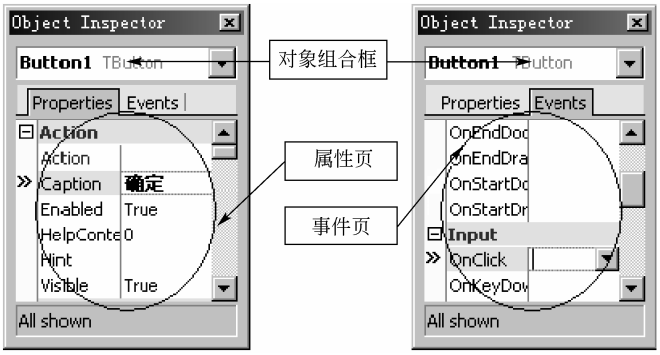

图 1-3　对象查看器

（1）Properties 页(属性页） Properties 页用于在设计时查看或者修改对象的相关属性。对象的属性可以在程序运行时通过代码改变，也可以在 Properties 页中改变，也就是程序运行前设置初始值。根据属性类型的不同，可以采用不同的属性设置方法，以下给出了属性设置的不同方法。

- 属性为数值类型或者字符串类型。只需要直接输入属性值即可，例如 Name、Caption 属性就可以直接在属性右边的编辑框中直接输入属性值。
- 属性为布尔型或者枚举型。选择该属性，在该属性右边的组合框中选择适当的属性值即可，例如 Label 的 Visible 属性，BitBtn 的 Kind 属性就属于这种类型。
- 通过对话框设置属性。有些属性的右边有按钮…，单击该按钮出现对话框，在对话框中可以设置该对象的属性值，例如 Font、Picture、Items、Glyph 等属性就是通过这种方法设置的。
- 直接调节对象的大小和位置也可以设置对象的部分属性，例如 Left、Top、Width 和 Height 等。
- 子属性设置。有的属性还有子属性。例如 Font，它的左边有一个⊞符号，单击该符号则⊞变成⊟号，同时 Font 的子属性被展开，此时用户可以设置其子属性 Color、Name、Size 等。
- 属性为集合类型。单击该属性左边的⊞符号，该⊞号变成⊟号，然后再设置该属性的集合元素。例如 Anchors 就属于这种类型。

（2）Events 页(事件页） 在 Events 页面中可以设置对象或者窗体的响应事件。不同对象预置了不同的一些事件，例如 OnClick、OnActivate、OnKeyPress 等。在编写程序的时候要为某些预置的事件编写相应的代码程序。

4．结构浏览器（Structrue）

结构浏览器以树状结构的形式显示窗体中各个组件之间的逻辑关系。结构浏览器和对象

查看器是同步协调工作的。在设置属性或者编写事件过程的时候，可以在结构浏览器选择好某个对象，然后再在对象查看器中设置属性或者编写事件过程，这样远比直接在对象查看器的对象组合框中选择对象方便。结构浏览器如图 1-4 所示。

如果结构浏览器树被关闭，选择菜单“View | Structure”可以显示结构浏览器。

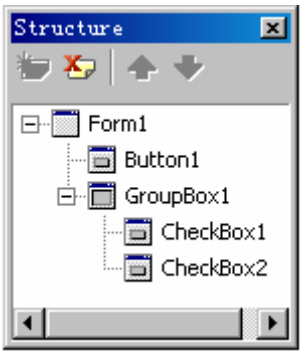

图 1-4　结构浏览器

5．代码编辑器

代码编辑器是程序代码输入和编辑的工具，尽管可视化编程技术自动生成了一些代码，但是用户仍然有很多代码需要手工完成，因此代码编写仍然是程序设计的核心工作。Delphi 的代码编辑器是一个功能强大、使用方便的代码编写工具。

（1）代码编辑器的组成　代码编辑器是一个多页的文本编辑器。有时候应用程序由几个窗体构成，此时单元就不止一个。通过选项卡可以选择不同的页面，每个页面就是一个单元文件。代码编辑器的标题显示单元文件名。如图 1-5 所示为代码编辑器。

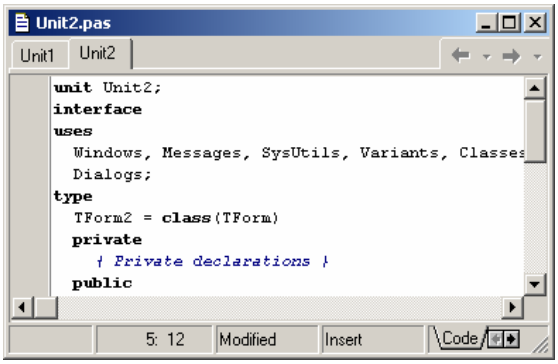

图 1-5　代码编辑器

（2）代码编辑器的功能　Delphi 的代码编辑器具有友好的提示和帮助功能，它主要表现在如下几点。

- 代码完善功能：在用户输入已创建的对象的名称和点号的时候，稍作停顿，系统将自动弹出提示列表框，列出该对象的所有属性和方法。用户无需自己敲入，只需在列表中选择合适的属性和方法，则该属性或者方法就自动添加到程序行中，如图 1-6 所示。
- 参数提示功能：在输入代码的时候，用户输入对象的方法或者标准的自定义的函数或者过程的时候，当用户输入左括弧，稍作停顿，系统会给出包括参数个数、参数类型在内的参数提示。参数提示功能如图 1-7 所示。

需要全本请访问 www.ertongbook.com

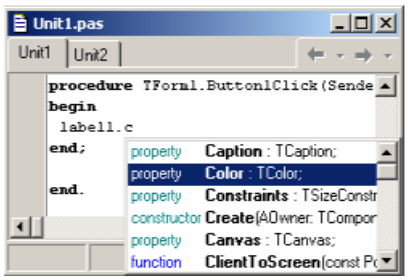

图 1-6　代码编辑器的代码完善功能

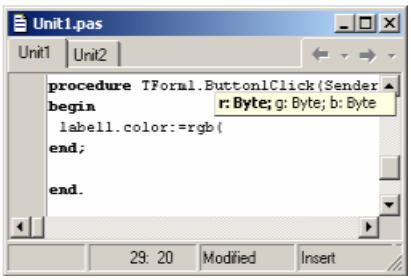

图 1-7　代码编辑器的参数提示功能

• 代码模板功能：代码模板用于提供语句模板以帮助代码编写。例如用户输入“for”，然后按“Ctrl-J”，系统自动显示 for 循环语句的语法供用户参考，防止代码出错。如图 1-8 所示为代码模板功能。左图列表框给出 for 语句的两种格式，第一种是带 begin 和 end 的格式，第二种是不带 begin 和 end 的格式，右图是选择第一种 for 循环格式，并按回车键后显示的 for 语句格式。

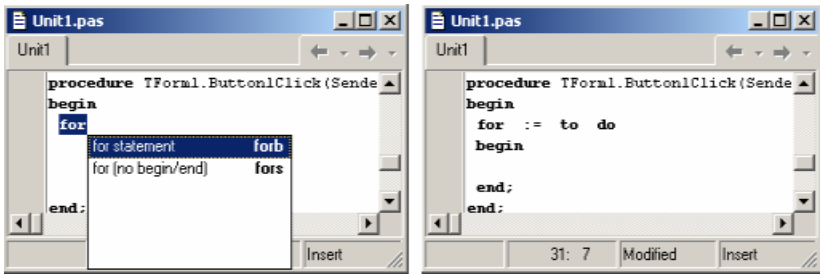

图 1-8　代码模板功能

• 符号洞察功能：将鼠标置于 Delphi 标识符（函数名、过程名、组件名、类型名和变量名）上，稍作停顿，系统自动给出该标识符的相关信息，如图 1-9 所示。

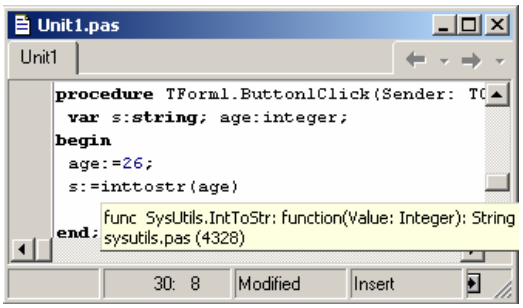

图 1-9　代码编辑器的符号洞察功能

6．组件栏

组件栏是建立 Delphi 应用程序的要素，Delphi 2005 提供了丰富的组件库，并将这些组件按照功能分类并显示在不同的选项卡中。常见的选项卡如图 1-10 所示。单击每个组件选项卡左边的“＋”号就可以展开该组件选项卡，并显示该组件选项卡中的所有组件供用户选用。如图 1-11 为展开后的 Standard 组件选项卡。使用组件栏右上角的两个图标 可以在多个组件中进行切换。

【例 1-1】 编制一个简单的 Win32 应用程序。通过本程序读者可以初步了解 Delphi 集成环境的使用方法。



图 1-10　最常见的组件选项卡

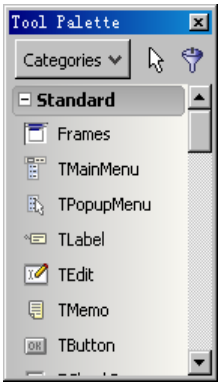

图 1-11　展开后的 Standard 组件选项卡

说明：本程序保存在“d:\delphi2005\ch1\eg1-1”文件夹中，以后的程序也保存在相应的文件夹中。请读者在编写 Delphi 程序的时候养成新建文件夹的习惯，本书中，编写程序前新建文件夹这一步骤一律省略。

步骤如下：

① 选择菜单“File”→“New”→“VCL Forms Application－Delphi for Win32”。在窗体上添加按钮 Button1 和标签 Label1，调节好窗体 Form1、Button1 和 Label1 的大小和位置。

② 设置 Button1 的 Caption 属性为“确定”。界面如图 1-12 所示。

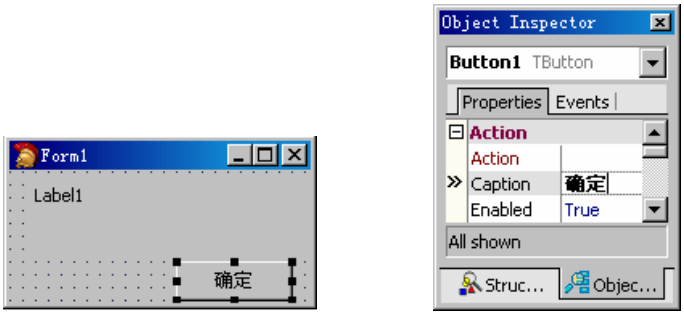

图 1-12　设置属性

③ 编写代码，选中 Button1，此时在 Events 中显示的是 Button1 对象，在对象查看器中选中 Events 页，并找到 OnClick 事件，用鼠标双击右边空白栏，如图 1-13 所示。此时打开代码编辑器，系统自动在窗体 Form1 的单元 Unit1 的 Interface（接口）部分插入该过程的声明，同时在 Implementation（实现）部分插入该过程的框架。如图 1-14 所示。

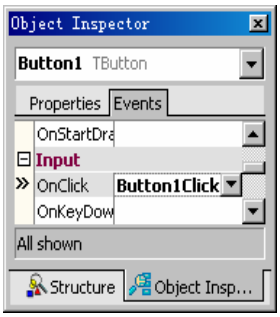

图 1-13　双击 Events 页中 OnClick 事件右边空白栏

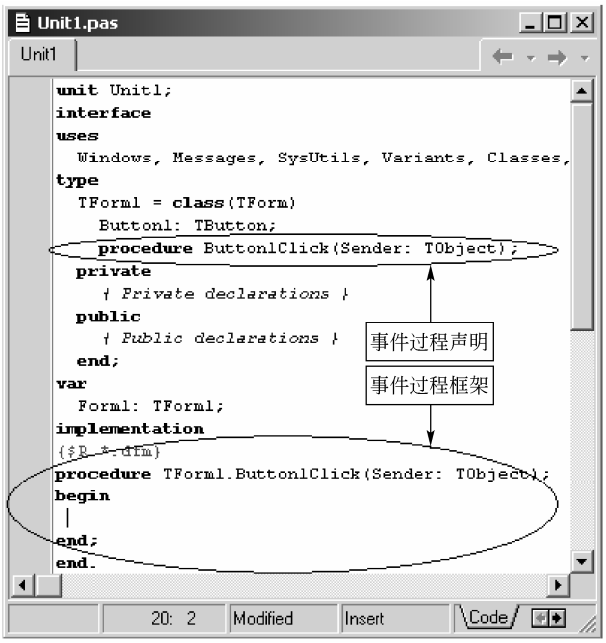

图 1-14　在代码编辑器中输入事件过程

④ 在事件过程框架中也就是在 Begin 和 End 之间（光标所在处）输入代码：

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  label1.Caption:='聪明的程序员使用 Delphi!';      //本行为输入的代码
end;
```

⑤ 按 F9 键运行程序，运行结果如图 1-15 所示。



图 1-15　程序运行结果

1.2.2　Delphi 应用程序所包含的文件

一个 Delphi 应用程序由多个文件构成。表 1-2 给出了 Delphi 中的一些常见的文件类型及含义。

表 1-2　Delphi 应用程序所包含的文件

文 件 类 型	扩　展　名	说　　明
项目组文件	bgp	
包文件	dpk	
项目文件	dpr	记录应用程序所包含的单元
单元文件	pas	Object Pascal 源代码文件
窗体描述文件	dfm	描述窗体和组件的性质

续表

文 件 类 型	扩　展　名	说　　明
单元编译文件	dcu	单元编译后生成的文件
选项文件	dof	记录项目的选项设置
资源文件	res	
动态连接库文件	dll	
配置文件	cfg	保存项目的配置
可执行文件	exe	
备份文件	~pas、~dfm、~dpr	单元、窗体、项目文件的备份文件
工程文件	bdsproj	

以【例 1-1】为例来看看 Delphi 中的一些主要文件。由于上次编写【例 1-1】程序保存文件时，系统只会自动保存项目文件和单元文件，而其他一些文件是在编译、运行、生成 exe 文件过程中形成的。先打开【例 1-1】中的项目文件和单元文件，并编译该项目和生成 project1.exe 文件。此时系统形成如下几个文件：Project1.bdsproj、project1.cfg、project1.dof、project1.dpr、project1.exe、project1.res、unit1.dcu、unit1.dfm 和 unit1.pas。如果读者对这些文件不是特别熟悉，建议不要删除其中任何文件，另外养成每个应用程序占用一个文件夹的习惯。这样应用程序的文件就不会张冠李戴了。

1．单元文件

单元是扩展名为 pas 的源程序代码，单元中可以定义或者声明常量、变量、函数、过程、对象、组件等。设计单元的好处是，单元可供其他应用程序共享。单元还可以独立编译为扩展名为 dcu 的文件，编译后的 dcu 文件可以连接到可执行文件中。单元文件不能够独立运行。单元文件分为如下几种：

① 与窗体相关的单元文件，这种单元文件与窗体一一对应；

② 用于存储公用函数、过程、常量、变量的单元文件，例如 math；

③ 组件文件，该单元提供组件开发接口。

单元文件是有一定结构的，一般一个单元文件由名称、接口、实现和初始化 4 个部分组成，各个部分说明如下。

- unit 子句：标识单元名，单元名与单元文件的文件名相匹配。
- interface 接口部分：说明本单元哪些部分可以供其他单元或者程序访问，在此声明的函数、过程、类型、常量、变量可供其他单元或者程序共享。接口部分只有声明，不能有定义，定义在实现部分完成。
- implementation 实现部分：接口部分声明的函数、过程、类型、常量、变量等的具体实现在此完成，实现部分也可以说明函数、过程、类型、常量、变量等，但是它们仅供本单元使用。
- initialization 初始化部分：可选部分，用于初始化程序所需数据。

请看【例 1-1】中的单元文件 project1.pas。

```
unit Unit1;                                    //说明单元

interface                                      //接口部分

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls,math;
```

```
type                                                    //类型说明
  TForm1 = class(TForm)
    Label1: TLabel;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);            //过程声明
  private
    { Private declarations }                            //私有部分
  public
    { Public declarations }                             //公有部分
  end;

var                                                     //变量说明
  Form1: TForm1;

Implementation                                          //实现部分

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);         //具体实现上面声明过的过程
begin
 label1.Caption:='聪明的程序员使用 Delphi!';
end;

end.                                                    //end 之后有句号，是程序结束标志
```

说明：在接口部分的 uses 短语后有一些单元名称，例如 Windows、Messages、SysUtils 等，这些单元是系统的标准单元或者是其他程序的单元，本单元要使用它们，因此需要在接口部分的 uses 短语后指明这些单元。

2．窗体描述文件

每个窗体都包含一个扩展名为 dfm 的窗体描述文件，该窗体描述文件用于描述窗体及窗体内组件的属性。鼠标右键单击窗体选择“View as Text”菜单可以显示窗体描述文件。请看【例 1-1】中的窗体描述文件 Unit1.dfm。

```
object Form1: TForm1
  Left = 192
  Top = 107
  Width = 205
  Height = 104
  Caption = 'Form1'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
```

```
    Left = 16
    Top = 8
    Width = 153
    Height = 25
    AutoSize = False
    Caption = 'Label1'
  end
  object Button1: TButton
    Left = 120
    Top = 48
    Width = 65
    Height = 25
    Caption = '确定'
    TabOrder = 0
    OnClick = Button1Click
  end
end
```

细心的读者一定可以从窗体文件中看到窗体及窗体内组件的一些属性。窗体描述文件就是在设计应用程序界面时系统自动生成的一些代码。鼠标右键单击窗体描述文件选择“View as form”可以重新以界面的形式显示窗体。

3．项目文件

项目文件的扩展名为 dpr。项目文件是应用程序的宏观框架，它管理着应用程序的多个单元。

以【例 1-1】为例，请看项目文件 project1.dpr。选择菜单“View | Units…”出现 View Unit 对话框，在对话框中选择项目文件“project1”按“OK”按钮。View Unit 对话框如图 1-16 所示。

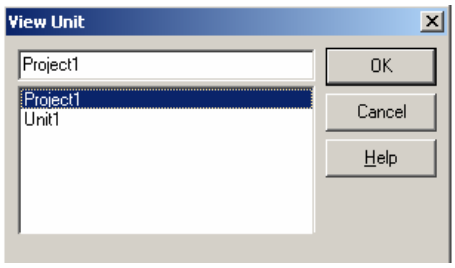

图 1-16　View Unit 对话框

project1.dpr 文件如下：

```
program Project1;

uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};
{$R *.res}

begin
  Application.Initialize;
```

需要全本请访问 www.ertongbook.com