

零点起航



—— 计算机编程系列教材

Delphi 7

Borland

Delphi™
Enterprise

基础教程

©零点工作室 宋一兵 赵景波 李春艳 等编著

Unregistered ©2002 Borland Software Corporation



机械工业出版社
CHINA MACHINE PRESS

零点起航 —— 计算机编程系列教材

Delphi 7 基础教程

零点工作室 宋一兵 赵景波 李春艳 等编著



机械工业出版社

Delphi 7 是 Inprise 公司推出的面向对象的可视化编程语言, 它提供了大量 VCL 组件, 具有强大的数据库开发和网络编程能力, 极大地提高了应用系统的开发速度, 是目前最优秀的前端开发平台之一。

本书从基础入手, 由浅入深, 全面系统地介绍了 Delphi 7 的基本知识。通过实例, 详细讲解了常用组件的用法。全书共分 13 章, 分别讲述了 Delphi 7 的集成开发环境、Object Pascal 语法、用户界面设计、多媒体应用、数据库开发和网络编程等方面的知识。

本书内容翔实、图文并茂, 强调“边学边做”, 具有良好的操作性, 适合 Delphi 初级开发人员使用, 也可以作为广大计算机爱好者的参考资料。

图书在版编目 (CIP) 数据

Delphi 7 基础教程/ 宋一兵等编著. —北京: 机械工业出版社, 2005.1

(零点起航——计算机编程系列教材)

ISBN 7-111-15414-2

I. D... II. 宋... III. 软件工具—程序设计—教材 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 105647 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: 商红云

封面设计: 陈沛 责任印制: 石冉

保定市印刷厂印刷·新华书店北京发行所发行

2005 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·21.5 印张·519 千字

定价: 29.50 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、88379646

封面防伪标均为盗版

前 言

Delphi 7 是 Inprise 公司（原 Borland 公司）推出的一个面向对象的快速应用程序开发工具，它以可视化的开发环境、简洁明快的编程语言、功能强大的组件、优化的源代码编译器、可扩展的数据库访问引擎、稳定性和高效性等特点，成为一个成熟的功能全面的产品，赢得了越来越多的用户。

Delphi 7 继承了 Pascal 语言的严谨结构和优雅风格，以开放式的环境，完全地支持面向对象程序设计等诸多特性，成为 Windows 环境下首选的开发工具。

本书主要介绍了开发 Delphi 应用程序的多种基本要素和必备知识，包括 Delphi 的可视化开发环境、Object Pascal 语言、面向对象程序设计的基本概念、窗体和各种功能的组件、MDI、调试技术与异常处理、图形图像处理、多媒体应用、数据库开发和 Internet 编程等。

本书从基础入手，深入浅出，在功能讲解的同时，配合大量的实例，力求使读者能快速、轻松地学会用 Delphi 编程，能充分理解面向对象程序设计的内涵，能用本书提供的实例解决实际问题。

根据应用性人才的培养目标，本书注重理论与实践相结合，注重基础知识的理解与基本技能的培养。在理论“必需、够用为度”的前提下，突出实用性，突出实践性环节。强调“边学边做”，使读者每学习一点儿知识，就能够通过具体的编程练习得到锻炼，同时也加深了对内容的理解。

本书是 Delphi 7 程序设计的入门书，特别适合于高校教师、学生和初级程序设计人员使用。读者使用本书不需要预先具有任何编程经验，但是如果对 Pascal、C、C++ 或者 Visual Basic 等有所了解，那么将会对本书的学习非常有益。

本书 1~4 章由李春艳老师编写，8~12 章由赵景波老师编写，其余内容由宋一兵编写并负责全书统稿。本书在编写过程中，得到了许多同事和朋友的支持，隋成城、张书钦、张宪海、汪学清、刘丽莉、王臣业、林新志、杨兴河、张忠林、李忠伟等参与了内容的编写和程序的调试工作，在此一并表示感谢。

由于时间匆忙、水平有限，书中难免存在错误和疏漏之处，恳请读者批评指正。

书中实例程序的源码可以从零点工作室网站 (<http://www.zerobook.net>) 获得。

编者

2004 年 6 月

系列教材序言

随着计算机技术的飞速发展,各种应用件也在社会和领域得到了广泛应用,软件的设计和开发也成为日常工作中的一部分。对于从事计算机方面工作的人员,掌握一种或几种编程工具是其必须具备的专业技能,也是胜任工作的基本条件。同时,很多高校学生、电脑爱好者也希望能掌握基本的软件设计方法,以满足未来工作和学习的需要。

零点起航——计算机编程系列教材选择了目前常用的编程软件,涵盖了应用程序设计、数据库开发和网络编程开发工具,着眼于大专院校教师、学生和工程技术人员的自学和培训。整套教材由7本书组成:

- (1)《Visual Basic.NET 基础教程》
- (2)《Visual C++.NET 基础教程》
- (3)《PowerBuilder 9.0 基础教程》
- (4)《Java 2 基础教程》
- (5)《C++Builder 6.0 基础教程》
- (6)《Delphi 基础教程》
- (7)《Visual FoxPro 7.0 基础教程》

整套教材本着从零末始的思想,从基础培训的角度入手,在内容的选限和章节的设置上充分考虑了初学者的衬际需要,力求简明清晰、通俗易懂。在详细讲解软件功能和用法的同时,引导读者练习一些针对性、衬用性很强的程序实例,以加深内容的理解。在每章的最后,都附带了一些习题,通过对这些习题的思考和练习,读者可以对该章所学内容有更深刻的认识。该套教材的作者都是长期从事计算机教学和软件设计的高校教师和专业人员,具有丰富的理论和实距经验,在写作过程中融入了多年的经验和体会,为初学者提出了许多有益的建议。

零点起航——计算机编程系列教材中的各教材自成体系,读者可以根据自己的实际需要选择。

希望这套教材对您的学习、工作和生活有所帮助。

零点工作室网站: <http://www.zerobook.net>

主编邮箱: guandianzhu@qdcnc.com

零点工作室
2004年6月

目 录

系列教材序言

前言

第 1 章 Delphi 基础知识	1	第 5 章 基本窗体设计	81
1.1 Delphi 的特点	1	5.1 Form (窗体) 组件	81
1.2 Delphi 7 的集成开发环境	2	5.2 向窗体中添加组件	86
1.3 实例——Hello, world!	5	5.3 单文档窗体	89
1.4 小结	9	5.4 多文档窗体	91
第 2 章 Object Pascal 语言基础	10	5.5 小结	92
2.1 Delphi 的编程风格	10	第 6 章 基本组件的应用	93
2.2 标识符	12	6.1 组件的概念	93
2.3 数据类型	14	6.2 文本输入组件	96
2.4 赋值语句	21	6.3 按钮及分类组件	103
2.5 过程与函数	23	6.4 列表组件	110
2.6 控制语句	25	6.5 滚动组件	121
2.7 实例——随机加减法测试	29	6.6 表格组件	125
2.8 小结	35	6.7 日期和时间组件	127
第 3 章 面向对象编程	36	6.8 多选项卡组件	130
3.1 OOP 基本概念	36	6.9 小结	133
3.2 类的基本概念	37	第 7 章 用户界面设计	135
3.3 类的封装	40	7.1 菜单设计	135
3.4 类的继承性	47	7.2 工具栏与状态栏	144
3.5 类的多态性	49	7.3 信息对话框	148
3.6 异常处理	52	7.4 对话框组件	152
3.7 小结	56	7.5 实例——文档编辑器	164
第 4 章 应用程序开发框架	57	7.6 小结	170
4.1 Delphi 的文件结构	57	第 8 章 图形图像技术	171
4.2 单元文件的内部结构	61	8.1 画布技术	171
4.3 项目管理	64	8.2 图形图像的和组件	180
4.4 应用程序类 TApplication	68	8.3 图形图像编程应用	183
4.5 项目的编译和调试	71	8.4 使用鼠标绘制图形	196
4.6 小结	80	8.5 小结	198

第 9 章 多媒体技术	199	11.2 数据控制组件	269
9.1 多媒体基础知识	199	11.3 人事管理系统开发	278
9.2 多媒体组件	200	11.4 小结	294
9.3 多媒体编程	214	第 12 章 报表和图表设计	295
9.4 小结	220	12.1 数据报表的设计	295
第 10 章 数据库基础	221	12.2 图表设计	299
10.1 数据库简介	221	12.3 小结	304
10.2 SQL 结构化查询语言	223	第 13 章 网络编程技术	305
10.3 Delphi 访问数据库的机制	227	13.1 网络基础知识	305
10.4 Delphi 的数据库管理工具	232	13.2 TCP/IP 编程	306
10.5 建立第一个数据库应用程序	247	13.3 使用网络函数编程	322
10.6 小结	248	13.4 小结	336
第 11 章 开发数据库应用程序	249	参考文献	337
11.1 数据集组件	249		

第 1 章 Delphi 基础知识

作为一款优秀的应用程序开发工具，Delphi 受到了广大程序设计人员的青睐，在各个领域得到了广泛应用。Delphi 7 是 Borland/Inprise 公司的主要作品之一，是基于对象 Pascal 语言的 RAD (Rapid Application Development, 快速应用程序开发) 工具，具有良好的可视化应用程序开发环境和强大的可扩展数据库功能。利用 Delphi 7 提供的 VCL (Visual Component Library, 可视化组件库) 进行编程，可以快速、高效地开发出基于 Windows 环境的各类应用程序。

1.1 Delphi 的特点

人们如此形容 Delphi: 真正的程序员用 C++, 聪明的程序员用 Delphi。与其他编程软件相比，特别是在数据库和网络应用方面，Delphi 具有简便易用、灵活高效的特点。

1. 强大的可视化编程环境

Delphi 的集成开发环境设计得非常简洁、明快，整个屏幕的各种窗口都经过精心安排，分布紧凑合理；而且用户可以定制桌面设置，建立一个符合自己风格的用户界面。

Delphi 7 具有非常优秀的窗体设计器，它是建立在一个真正面向对象的框架结构基础之上。这样，对基类所作的改变都将会传递给所有的派生类。在窗体设计器中进行设计时，Delphi 7 会自动在后台为窗体中的组件生成代码。

Delphi 7 的编辑器和其他工具的编辑器基本功能相差不多，但其 Code Insight 技术却省去了许多输入工作的麻烦，避免了记忆组件种类繁多的方法和属性。而且系统还能够自动对用户输入的函数和过程的参数列表给出提示。

Delphi 7 的调试器功能可与 Visual C++ 相媲美，除了能在编辑器中通过设置断点和监控点等来调试程序外，还具备了许多先进的功能，如远程调试、过程关联、DLL 和包调试、自动本地监控以及 CPU 窗口等。

2. 高效的编译器

开发软件通常要经过修改源代码、编译、测试、再修改、再编译、再测试等操作，这些操作形成了一个开发循环，所以快速的编译器可以大大降低开发周期。如果读者用过 Visual C++ 或者 C++ Builder，对其比较慢的编译速度可能会有较深的印象。

Pascal 编译器以其编译速度快特点而闻名，而 Delphi 正是建立在这种高效编译器的基础之上，是针对 Windows 的最快的高级语言代码编译器。即便是 C++ 编译器近年来取得了很大的进步，增加了链接和各种缓存策略，其编译速度仍远较 Delphi 低。

Delphi 和 C++ Builder 共享同一种编译器后端，因此生成的代码也十分精练高效，使 Delphi 程序的运行速度更快。

3. 丰富的 VCL

VCL 是 Delphi 7 最重要的组成部分。VCL 包含丰富的不同种类的组件，大大方便了软件的开发工作。读者还可以通过网络获得很多其他类型的组件，并将其添加到组件面板上，如同 Delphi 7 自带的组件一样使用。

在程序设计时操作组件、创建组件、使用面向对象技术继承其他组件的行为和能力，是决定 Delphi 程序开发效率的关键因素。

1.2 Delphi 7 的集成开发环境

Delphi 7 是目前的最新版本，它所提供的可视化集成开发环境（IDE）使程序员能够轻松设计出图形化的应用程序。

启动 Delphi 7 后，其集成开发环境如图 1-1 所示，主要包括菜单栏、工具栏、组件面板、窗体设计器、代码编辑器、对象监视器、对象树形查看器等几种。

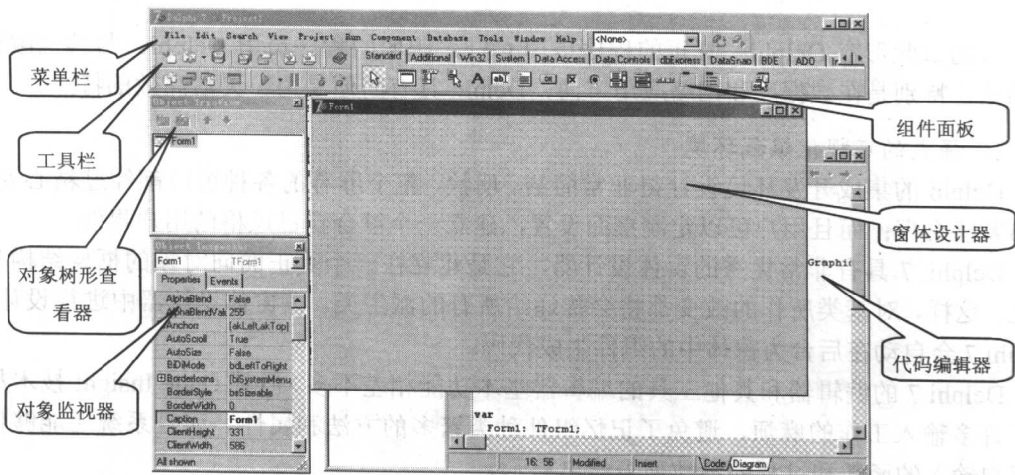


图 1-1 Delphi 7 的集成开发环境

1. 菜单栏

菜单栏提供了 Delphi 集成开发环境的所有功能。菜单栏除包含标准 Windows 程序通常所具有的【File】、【Edit】、【Help】等菜单外，还有一些与 Delphi 7 项目、运行、辅助工具和环境配置等相关菜单。其中的菜单命令将在后面的学习中结合具体应用来讲解。

2. 工具栏

工具栏中包含了一些常用操作的快捷菜单，各个按钮的功能如图 1-2 所示。鼠标指针在某个工具按钮上停留片刻就会出现 Tooltip（工具提示）。

工具栏中的快捷按钮可以由用户按自己的需要自由添加或删除。工具栏也可以进行个性化设置。选择【View】/【Toolbars】/【Customize】菜单命令，会出现如图 1-3 所示的对

话框。通过该对话框，用户可以自定义工具栏中的栏目。

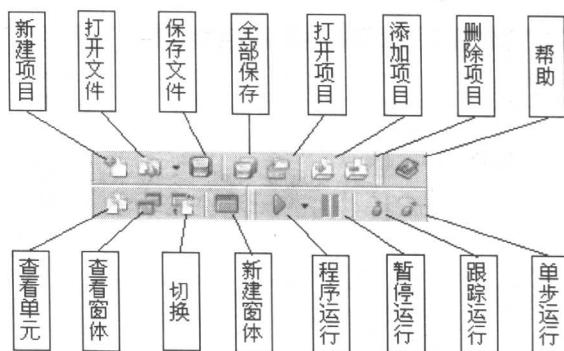


图 1-2 工具栏中按钮的功能

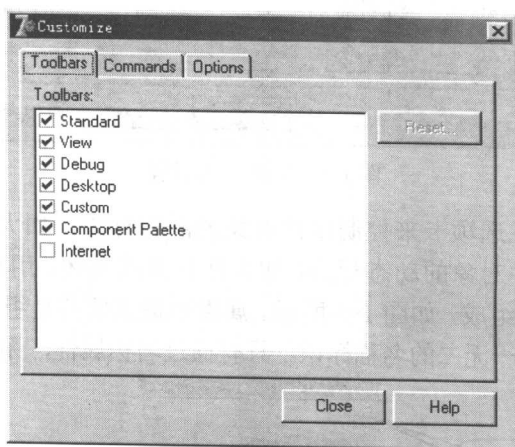


图 1-3 自定义工具栏栏目

3. 组件面板

组件面板是一个以选项卡形式显示的工具栏，其中包含了 Delphi 7 自带的 VCL 组件和用户安装的 ActiveX 组件，主要有【Standard】、【Additional】、【Win32】等选项卡，每个选项卡又包括若干以图标形式表现的组件。组件是建立一个应用程序最常用的元素。

用户可以自己定制组件栏的页面和各页面所包含的组件。选择【Tools】/【Environment Options】菜单命令，打开系统的环境设置对话框，其【Palette】页面如图 1-4 所示，页面左边为组件面板的各个选项卡，右边是该选项卡包含的组件。通过对话框中的几个按钮，可以添加和删除选项卡，也可以对选项卡进行更名操作。除了隐藏选项卡中的组件，还可以添加组件到指定的选项卡，如果是其他选项卡中的组件，可以简单地通过鼠标的拖放操作来完成。如果是从组件库中选择组件，就需要先构造和安装组件包。

4. 【Object Inspector】(对象监视器)

对象监视器是用来描述组件及窗体对象的属性特征和行为事件的。它是应用程序设计

过程中最重要的一个工具。其功能主要有以下两方面，一是修改窗体及其包含的对象的属性，二是通过自动代码生成机制使对象响应相应的事件。

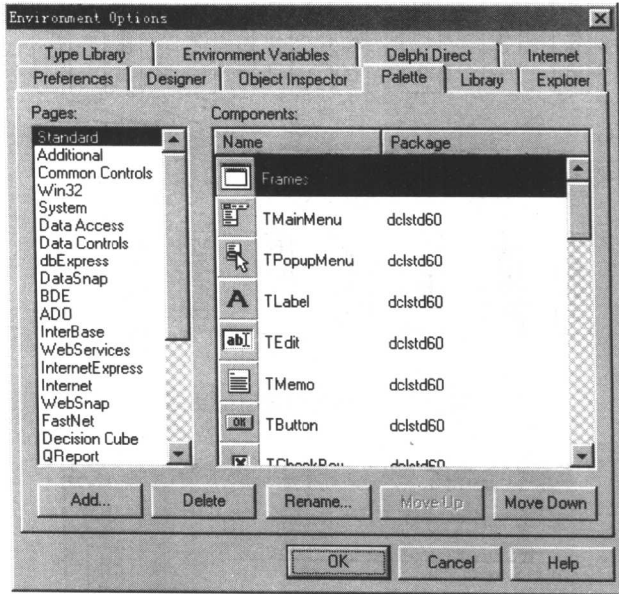


图 1-4 系统环境设置

对象监视器通过属性选项卡来控制组件对象的静态行为，如大小、色彩和名称等；通过事件选项卡来控制组件对象的动态行为，如组件打开或关闭、用户在组件上单击鼠标等。

对象监视器由 3 部分组成，如图 1-5 所示。属性中最重要的是组件的【Name】（名称）属性，每个对象都有自己独一无二的名称标识。只有通过名称属性，系统才能识别不同的对象。

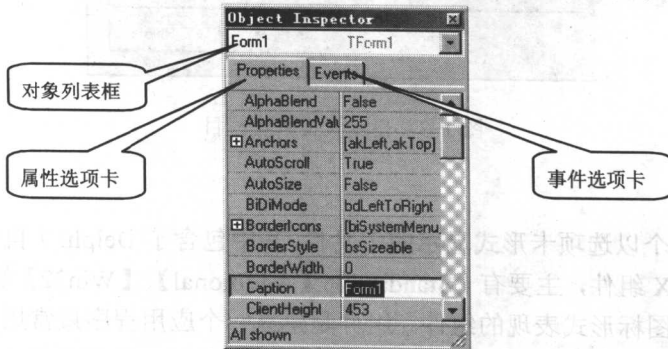


图 1-5 对象监视器窗口

5. 窗体设计器

窗体是放置 VCL 中各类组件对象的地方。通过所见即所得的窗体设计器，程序员可以在窗体中绘制各式各样的程序界面，用户通过界面操作，就可与应用程序交互。

窗体设计的主要步骤就是向窗体中添加可以完成应用程序任务的组件；更改组件的属性使之满足程序的设计要求；将组件的事件与程序代码相联系，用代码实现组件的活动。

6. 【Object Treeview】(对象树形查看器)

对象属性查看器能够以目录树的形式显示整个工程中使用到的对象。在这里选择对象，就如同在窗体设计器中选择对象一样。

7. 代码编辑器

代码编辑器又称单元窗口，是编写程序代码的地方，如图 1-6 所示。它包括两个密切相关的部分：单元管理窗口显示的是程序的结构组成，包括类定义、变量和常量的定义部分以及引用声明部分；代码编辑窗口是用来编辑源程序的地方。

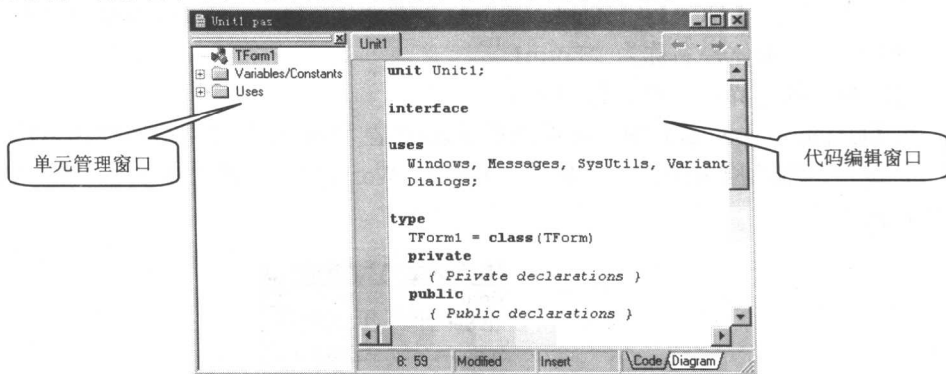


图 1-6 代码编辑器窗口

代码编辑器一般隐藏在窗体设计器下面，可以通过以下几种方法显示它：

- 选择【View】/【Code Explorer】菜单命令或【View】/【Toggle Form/Unit】菜单命令；
- 单击快捷工具栏中的【Form/Unit】转换按钮；
- 在集成开发环境中用鼠标单击代码编辑器窗口。

在窗口内单击鼠标右键会弹出快捷菜单，其中包含了代码编辑和调试的一些命令。

1.3 实例——Hello, world!

Delphi 7 应用程序开发的基本步骤为：

- 建立窗体；
- 在窗体上添加组件；
- 检查和设定对象属性；
- 编写响应事件处理程序；
- 保存文件；
- 编译、运行程序。

下面以一个简单的实例来说明 Delphi 程序的结构。

【例 1-1】 Hello, world!

程序效果：程序中包含两个按钮，单击“问候”按钮，会出现“Hello, world!”的文字；单击“退出”按钮，就结束程序。画面如图 1-7 所示。

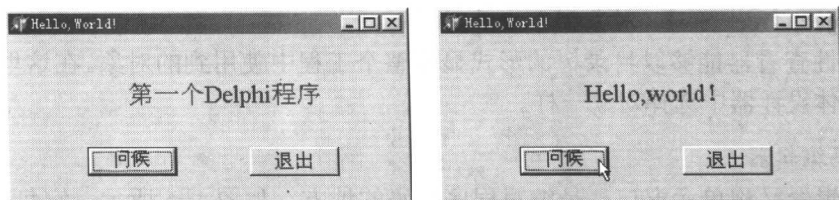


图 1-7 第一个 Delphi 程序

(1) 选择【File】/【New】/【Application】菜单命令，建立一个新的应用程序。此时，Delphi 自动生成一个窗体，名称为“Form1”。

(2) 拖动窗体的边界，调整窗体的大小。

(3) 在【Object Inspector】中，可见当前显示的是“Form1”对象的属性和方法。单击【Caption】属性，删除原来的文字，输入“Hello, World!”，如图 1-8 所示。修改窗体标题栏中的文字内容。

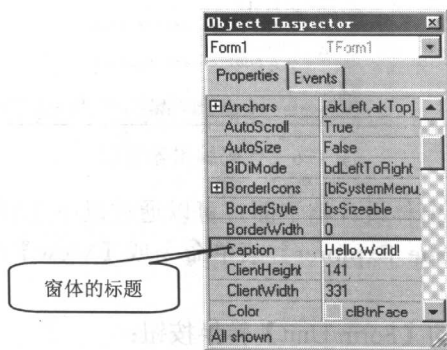


图 1-8 修改窗体标题栏中的文字内容

(4) 在组件面板的【Standard】选项卡中，选择“Button”组件；然后在窗体中，按住鼠标，拖出一个虚线方框；松开鼠标，则按钮出现，可见按钮上的文字为“Button1”，如图 1-9 所示。

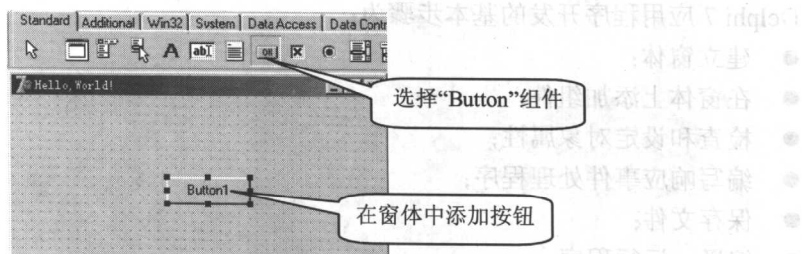



图 1-9 在窗体中添加按钮

提示：在“Button”组件上双击鼠标，会直接在窗体中建立一个组件对象。

(5) 选择窗体中的按钮，在【Object Inspector】中，可见当前显示的是“Button1”对

象的属性和方法。单击【Caption】属性，删除原来的文字，输入“问候”，修改按钮上显示的文字。

(6) 在【Font】属性后面单击按钮，会出现一个设置字体的对话框。在这里设置按钮上显示的文字样式，如图 1-10 所示。

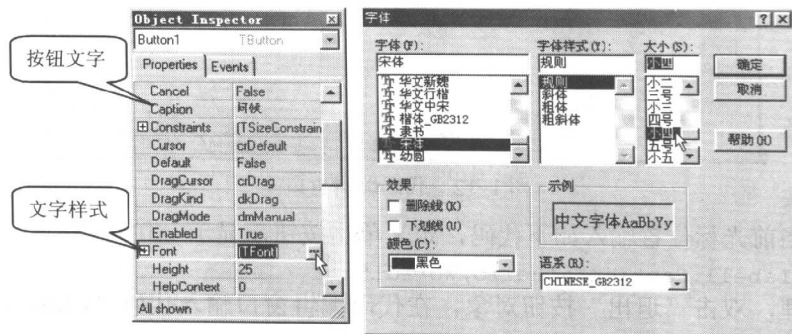


图 1-10 设置按钮文字和文字样式

(7) 同理，再在窗体中添加一个按钮，并修改按钮文字为“退出”，字体样式同“问候”按钮。

(8) 在组件面板的【Standard】选项卡中，选择“Label”（标签）组件，然后在窗体中建立一个标签对象；修改对象的文字为“第一个 Delphi 程序”，并设置文字字体和颜色，如图 1-11 所示。

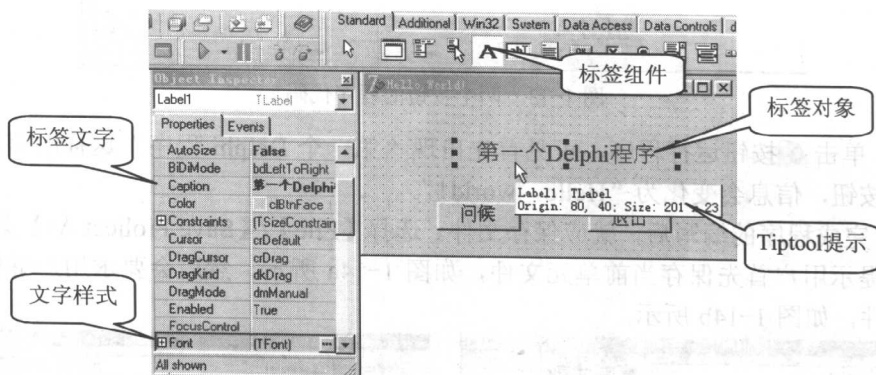


图 1-11 建立标签对象并修改其属性

(9) 在窗体上双击“问候”按钮对象，会显示出代码编辑窗口，如图 1-12 所示。可见当前已经有一些程序代码，这是系统自动创建的程序结构代码；而且，虽然前面修改了按钮的文字，但是并没有改变按钮的名称，所以按钮名称仍为默认的“Button1”。

提示：在【Object Inspector】窗口选择【Event】选项卡，双击“OnClick”事件右侧区域，也会打开代码编辑窗口；而且这里有更多的事件可供选择。

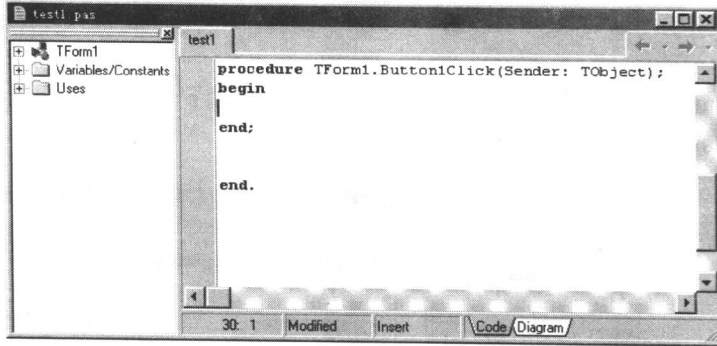


图 1-12 代码编辑窗口

(10) 在当前光标位置输入如下代码，定义单击按钮后显示一句问候：

```
Label1.Caption:='Hello,World!'
```

(11) 同理，双击“退出”按钮对象，在代码编辑窗口输入代码“Close”，如图 1-13 所示。定义单击按钮退出程序。

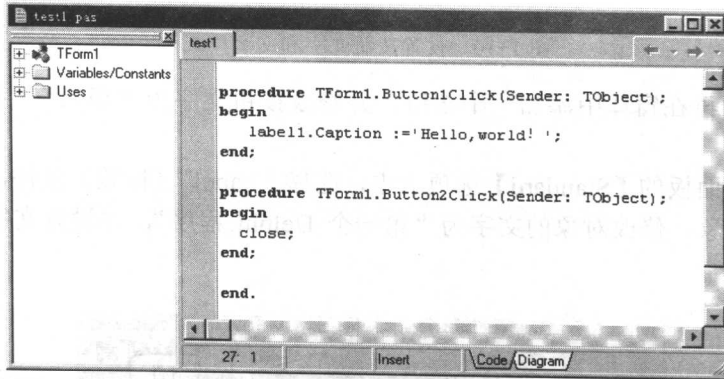



图 1-13 两个按钮的程序代码

(12) 单击  按钮运行程序，可见首先出现“第一个 Delphi 程序”这样的信息；单击“问候”按钮，信息会变化为“Hello, world!”。

(13) 完成程序的编写后，就应保存文件。选择【File】/【Save Project As】菜单命令，Delphi 会提示用户首先保存当前单元文件，如图 1-14a 所示。然后会要求用户继续保存当前工程文件，如图 1-14b 所示。

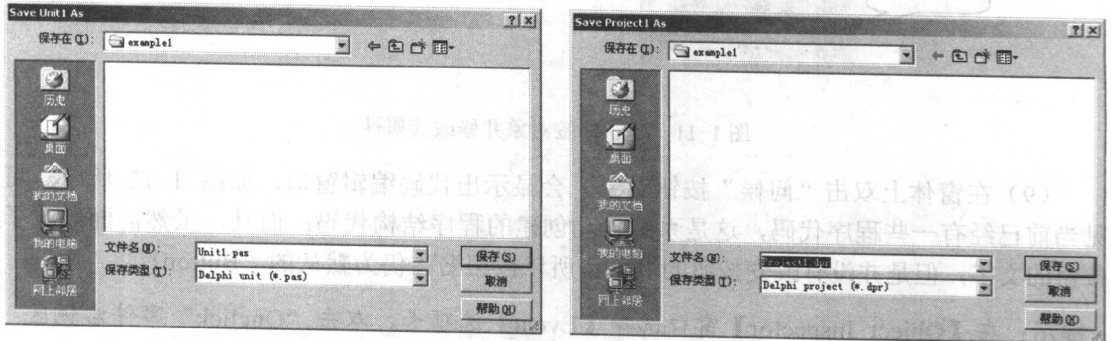


图 1-14 保存单元文件和工程文件

提示：工程文件和单元文件的名称不能一样，否则会出现错误警告。

程序最终代码如下。为了与系统自动建立的代码相区别，这里将我们自己添加的代码用带底纹的斜体文字表示出来。

```
unit test1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
  Label1.Caption := 'Hello, world!'; //添加的代码
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
  close; //添加的代码
end;
end.
```

1.4 小结

本章主要介绍了 Delphi 7 的基本知识，使读者对 Delphi 7 有了一个大致的了解。通过一个简单的实例，使大家对 Delphi 的简单易用有了感性的认识。集成开发环境是 Delphi 的操作界面，认识和了解 IDE 的组成及各部分的作用是进行程序设计的基础，对于后面的学习至关重要。

第 2 章 Object Pascal 语言基础

Object Pascal 是在 Pascal 的基础上发展起来的，它不仅继承了 Pascal 语言语法结构严谨、编译代码效率高等优点，而且使程序更加易学易用；同时，使用编译器创建的应用程序只生成单个可执行文件（.EXE）。正是这些特点，使得 Object Pascal 很适合作为基础的软件开发语言，从而成为 Delphi 这种先进开发工具的编程语言。

本章将讨论 Object Pascal 的基本知识，并讲解如何在事件处理过程和其他应用程序中，使用它来编制程序代码。讲解的重点在于 Object Pascal 的语法基础，而不是 Pascal 语言的一切细节。

2.1 Delphi 的编程风格

所谓编程风格，就是指程序员常用的编程习惯和文本样式。Pascal 程序具有良好的可读性，但是通过遵循一定的编程风格，会使程序更加便于理解、便于维护。

2.1.1 注释语句

Object Pascal 支持三种类型的注释：

- 花括号注释：组合符号“{”和“}”的成对使用表示它们之间的内容是注释部分。
- 圆括号+星号注释：组合符号“(”和“)”的成对使用表示它们之间的内容是注释部分。上面两种注释样式又称为块注释。
- C++风格的双斜杠注释：符号“//”的单独使用表示后面的内容是注释部分。这种注释样式又成为行注释。

请看下面的例子：

```
{ 花括号注释 }  
(* 圆括号+星号注释 *)  
// C++风格的注释
```

前两种注释在本质上是相同的，编译器把处于限定符头和限定符尾中间的内容当作注释。花括号和“圆括号+星号”相比，较适合在大段注释时使用。如果在“{”或“(”后面是一个“\$”符号时，表示该句为一个编译器指令，与普通的注释不同，通常用来对编译过程进行设置。

对于 C++风格的注释来说，双斜杠后面到行尾的内容被认为是注释。比较适用于单行和少量几行注释的情况。

虽然不同类型的注释在语法上是合法的，但不建议这样做。另外，相同类型的注释不能嵌套使用。下面是一些注释语句嵌套的例子：