

Delphi 3.0

使用与开发指南

冯永红 王军 宋天平 编
史惠康 审校



人民邮电出版社
PEOPLE'S POSTS &
TELECOMMUNICATIONS
PUBLISHING HOUSE

内 容 提 要

Delphi 3.0 是 Borland 公司推出的全新的可视化编程环境, 完全面向对象, 能方便、快捷地开发 Windows 应用程序。Delphi 3.0 使用了 Windows 图形用户界面的许多先进特性、目前世界上最快的编辑器、最为领先的数据库技术, 是一种不可多得的可视化开发工具。本书从浅到深讲述了 Delphi 的使用环境和编程技巧, 适合于自学 Delphi 编程的初、中级读者。

Delphi 3.0 使用与开发指南

- ◆ 编 冯永红 王 军 宋天平
- 审 校 史惠康
- ◆ 人民邮电出版社出版发行 北京崇文区夕照寺街 14 号
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 19.75
字数: 482 千字 1998 年 5 月第 1 版
印数: 1—6 000 册 1998 年 5 月北京第 1 次印刷

ISBN 7-115-07098-9/TP·665

定价: 26.00 元

前 言

Delphi 3.0 是 Borland 公司推出的全新的可视化编程环境,是一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想,采用了面向对象程序语言(Object-Oriented Language)、当今世界上最快的编辑器和最为领先的数据库技术。

面向对象的程序设计(Object-Oriented Programming, 简记为 OOP)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。面向对象的编程语言使得复杂的工作条理清晰、编写容易。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软件重用的开发工具,从而具有强大的吸引力。

Delphi 的语言基础是 Pascal,但是它与传统的 Pascal 语言有着根本性的差别。Delphi 编程环境自动为 Delphi 应用程序生成应用程序框架,这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西,仍可以严格地按照设计运行。编程人员所要做的工作只是在“骨架”中加入程序代码,以实现自己的目的。缺省的应用程序是一个空白的窗体(Form),运行它,就可以得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质:可以被放大缩小、移动、最大最小化等。因此,可以说应用程序框架通过提供所有应用程序共有的东西,为用户应用程序的开发打下了良好的基础。Delphi 已经做好了一切应用程序的基础工作——程序框架本身就是一个已经完成的可运行的应用程序,只是不处理任何事情。编程人员所需要做的,只是在程序中加入完成功能所需的代码而已。

可以这么说,Delphi 就是 Visual Pascal。它继承了传统的 Pascal 的优美的语言风格,而且也发展 Pascal,使用的是面向对象的 Pascal(Object Pascal)。

本书针对 Delphi 的使用方法,由浅入深介绍了 Delphi 3.0 的编程技术,同时为了照顾 Delphi 1.0 和 2.0 的用户,在第一章中给出了 Delphi 3.0 新增功能的介绍。除了对 Delphi 本身编程方法的讲述外,书中还对 Delphi 在数据库、Active X、Internet 方面的应用做了详细阐述。对没有接触过 Pascal 的用户,我们在附录中给出了相应的基础知识。本书的具体阅读方法,详见随后的导读部分。

本书由史惠康策划编写,第一章由刘铭编写,第二章由王松编写,第三、四、五章由冯永红编写,第六、七、八章由王军编写,第九章由郭立明编写,第十章由范云峰编写,第十一、十二章由宋天平编写,第十三章由章广为编写。李雨编制了附录并帮助编制了部分程序。冯永红、王军和宋天平负责各章节的组织工作。全书由史惠康审阅。

本书在编写中得到了清华大学雷霆的大力帮助,特此表示感谢。

由于作者经验有限,纰漏在所难免,敬请读者批评指正。

编者

1998 年 2 月

全书导读

本部分内容旨在提纲挈领地全书的各个章节的内容作简要介绍，以方便读者从中快速找到自己所需要的内容。下面几句话只是编者愚见，仅供参考。随后是各章节的内容简介。

读者情况	建议阅读方法
使用过 Delphi 的以前版本	你已经具备面向对象的知识和 Delphi 的基本使用方法，可以跳过第一部分，直接阅读后面内容。
没使用过 Delphi，但接触过其他面向对象的程序设计方法	第一章的内容可略读，用以了解 Delphi 的基本开发步骤。
没接触过任何面向对象的程序设计方法	建议你从头读起。第一、二章尤为重要。
不了解 Pascal，但接触过其他结构化程序的设计方法	你最好先阅读附录，了解 Pascal 语言的基础知识。建议你学习本书时手边备一本关于 Pascal 的书籍，若无 Pascal 书籍，关于 C 程序设计的亦可。
不懂结构化程序设计方法	如果你属于这种情况，本书不适合你。建议你先学习一门结构化程序设计语言（C、Pascal 均可），然后再考虑学习本书。

本书内容简介

第一部分 入门篇

第一章我们将了解如何利用 Delphi 进行应用程序的开发的简单步骤，并会见到第一个 Delphi 应用程序。更重要的是，我们要了解什么是面向对象的编程思想及 OOP 的一些基本概念。

在第二章和第三章，我们将介绍如何利用 Delphi 的 IDE 进行工作。第二章我们将重点介绍 IDE 的组成及功能，第三章我们将通过一个计算器的例子介绍如何利用前一章的 IDE 工作。

通过第一部分的学习，我们将对 Delphi 有一个大体的概念，为以后的应用程序的开发打下基础。

第二部分 基础篇

第一部分使得我们对 Delphi 3.0 有了一些概念，这一部分将在前面的基础上具体介绍 Delphi 3.0 的大部分功能。当然，要掌握功能，仅仅把功能罗列成表，然后去了解它是不可能真正学会的。在讲解这些内容时，我们配合了较多的程序。

编辑器的设计，以及后来对编辑器从单文档到多文档的改进是这一篇内容的关键，它将具体地教会你如何开发一个完整的应用程序。

多文档及多页对象、文件的管理是在第四、第五章的基础上进行的，从第四、五章学

到的方法会在这里经常用到。

多媒体的制作能够极大地激发你的创造力。第八章介绍关于图像、声音及动画的制作方面的内容，相信通过这一章你能够做出自己满意的多媒体作品来。

第六章着重介绍了面向对象编程的基本概念，如果想对 Pascal 中的基本内容有所了解，可以参看附录。

同时，这一篇也将教会你在程序出问题如何去调试它。

第三部分 提高篇

本篇我们将重点讲述 Delphi3.0 的新特性，为我们更高一级的使用 Delphi3.0 提供一个方法。本篇我们首先介绍数据库应用程序的开发，重点介绍数据库中数据的操作及查询。然后，我们将重点介绍 Delphi3.0 可视组件的开发以及 ActiveX 组件的简单生成，最后我们还介绍了 ActiveForm 在 Internet 上的应用。在本篇的最后一章，我们介绍了 Internet 应用程序的创建，重要讲述了 Internet 组件的使用，并以 THTML 组件为例具体讲述了一个浏览器的生成；然后我们讲述了如何利用服务器模型创建 CGI 应用程序，最后进一步讲述了 ActiveForm 在 Internet 应用程序的开发中的应用。

附录 Pascal 语言的基础知识

讲解 Pascal 语言的基础知识。

目 录

第一章 Delphi 基础	1
1.1 Hello! Delphi	1
1.2 Delphi 2.0 到 3.0 的飞跃.....	4
1.3 可视化及 OOP 概念.....	5
1.3.1 什么是对象.....	5
1.3.2 什么是属性.....	6
1.3.3 什么是事件.....	6
1.4 什么是面向对象编程 (OOP)	7
1.4.1 对象的封装性、继承性.....	7
1.4.2 对象的多态性.....	8
1.4.3 对象的可见性.....	8
1.4.4 如何访问对象的域和方法.....	9
1.5 本章小结.....	10
第二章 Delphi 集成环境开发	11
2.1 主窗口	11
2.1.1 菜单栏 (MenuItem)	11
2.1.2 快速按钮 (SpeedBar)	14
2.1.3 组件板 (Component Palette)	14
2.2 工程管理器 (Project Manager)	14
2.3 表单窗体 (Form)	15
2.4 对象查看器 (Object Inspector)	16
2.5 代码窗口 (Code Editor)	17
2.6 本章小结.....	20
第三章 如何利用 IDE 创建一简单加减计算器程序	21
3.1 计算器的功能.....	21
3.2 项目的创建的途径.....	22
3.3 创建用户界面.....	23
3.3.1 如何选取所需的部件.....	23
3.3.2 如何实现部件的调整.....	24
3.4 如何设计部件的属性.....	26
3.5 计算器功能的实现——代码的编写.....	29
3.5.1 计算功能的实现.....	31
3.5.2 清除功能的实现.....	34
3.6 程序的运行.....	35
3.7 程序的改进.....	37
3.8 程序的简单调试.....	38

3.8.1	断点的设置.....	38
3.8.2	变量值的观察.....	39
3.8.3	程序的运行控制.....	39
3.9	项目的管理.....	40
3.9.1	如何设置自己的图标和标题.....	40
3.9.2	帮助文件的联接.....	42
3.10	本章小结.....	42
第四章	构建简单的编辑器程序.....	43
4.1	编辑器的功能.....	43
4.2	编辑器的输入.....	46
4.2.1	Edit 的使用.....	46
4.2.2	RichEdit 的改进.....	51
4.3	如何实现对字体的控制.....	56
4.3.1	单选框与复选框的使用.....	57
4.3.2	用组合框改进.....	59
4.4	菜单设计.....	61
4.4.1	菜单设计.....	62
4.4.2	菜单项功能的实现.....	65
4.5	对话框的使用.....	68
4.5.1	普通对话框.....	68
4.5.2	如何打开文件.....	69
4.5.3	如何存储文件.....	72
4.5.4	文件的打印.....	73
4.5.5	字体对话框.....	78
4.6	本章小结.....	80
第五章	窗体风格的设计.....	81
5.1	Form 对象.....	81
5.2	编辑器的 About 窗口.....	84
5.2.1	窗体的添加与删除.....	84
5.2.2	为编辑器写软件说明.....	85
5.3	让编辑器支持右键.....	86
5.4	对编辑器菜单的改进.....	88
5.4.1	级联菜单的设计.....	88
5.4.2	复选标记设置.....	89
5.4.3	菜单项的灰显.....	90
5.5	工具条的添加.....	91
5.5.1	工具条的使用.....	91
5.5.2	快速按钮设计.....	92
5.6	窗口状态栏的设计.....	95

5.7 窗体的分割.....	99
5.7.1 用 Panel 组件分割.....	99
5.7.2 用 HeaderControl 组件分割.....	101
5.7.3 用 Header 组件分割.....	103
5.8 本章小结.....	104
第六章 OOP 程序设计风格.....	105
6.1 Delphi 代码的结构.....	105
6.2 类和对象.....	107
6.3 对象系统的三个特性.....	110
6.4 方法重载.....	112
6.5 本章小结.....	112
第七章 多页对象与多文档界面设计.....	115
7.1 多页对象概述.....	115
7.2 与多页对象有关的组件.....	115
7.2.1 Notebook 与 TabSet.....	115
7.2.2 TabbedNotebook 与 TabControl.....	120
7.2.3 PageControl 与 TabSheet.....	121
7.3 多文档界面的结构.....	122
7.4 多文档界面建立.....	122
7.4.1 建立 MDI 父窗体.....	122
7.4.2 建立子 MDI 子窗体.....	123
7.4.3 动态建立子窗体的方法.....	124
7.5 子窗体菜单设计.....	125
7.5.1 合并菜单.....	126
7.5.2 代码的更改.....	127
7.5.3 对子窗体的管理.....	127
7.6 MDI 窗体的关闭.....	129
7.7 MDI 模板.....	130
7.8 本章小结.....	132
第八章 图像、图形与多媒体.....	133
8.1 Delphi 图形基础.....	133
8.1.1 画布 (TCanvas).....	133
8.1.2 颜色 (Tcolor) 对象.....	133
8.1.3 坐标与像素.....	134
8.1.4 画笔 (Tpen) 对象.....	136
8.1.5 画刷 (Tbrush) 对象.....	139
8.2 用 Canvas 编写图形应用程序.....	140
8.2.1 响应鼠标事件.....	142
8.2.2 用鼠标绘制各种图形.....	143

8.2.3 TPaintBox 组件	145
8.3 图像的处理.....	146
8.3.1 图像基础.....	147
8.3.2 如何显示一个已存在的图像.....	148
8.3.3 如何在 Delphi 中编辑图像.....	151
8.4 如何在程序中使用声音.....	154
8.4.1 用 MediaPlayer 组件播放声音	154
8.4.2 调用 API 函数播放声音	155
8.5 视频文件的播放.....	156
8.6 对媒体播放器的方法控制.....	157
8.7 本章小结.....	160
第九章 文件系统的管理	161
9.1 文件的类型和属性.....	161
9.2 编辑文本文件的操作.....	162
9.2.1 文件的打开与关闭.....	162
9.2.2 文件的读写.....	163
9.3 文件系统的管理控制.....	164
9.3.1 File ListBox 和 FilterComboBox	164
9.3.2 DirectoryListbox 和 DriveComboBox.....	168
9.3.3 DirectoryOutline	171
9.4 开发一个文本与图像浏览器.....	172
9.5 本章小结.....	175
第十章 灵活运用集成调试器	177
10.1 如何设置环境.....	177
10.2 控制程序运行方式.....	178
10.3 断点的设置及使用.....	180
10.4 跟踪程序过程.....	182
10.5 异常处理.....	185
10.5.1 Delphi 异常处理机制与异常类.....	185
10.5.2 常见的几类异常.....	185
10.5.3 异常保护.....	188
10.5.4 异常响应.....	190
10.6 本章小结.....	192
第十一章 开发数据库应用程序.....	193
11.1 数据库的建立.....	193
11.1.1 关系数据库.....	193
11.1.2 Delphi 数据库特性.....	194
11.2 建立一个新的数据库表.....	194
11.2.1 Database Desktop 工具.....	194

11.3 如何访问数据库表.....	198
11.3.1 Form Wizard 创建简单数据库应用程序.....	198
11.3.2 数据访问组件(Data Access)的应用.....	201
11.3.3 数据控制组件(Data Control)的应用.....	203
11.4 运行期间访问数据库表.....	207
11.4.1 数据集的打开与关闭.....	207
11.4.2 数据库表记录的定位.....	207
11.4.3 对数据库表数据的操作方法.....	208
11.5 记录的排序与查询.....	211
11.5.1 索引的建立.....	212
11.5.2 利用索引排序.....	213
11.5.3 精确查找某一记录.....	213
11.5.4 利用 GotoNearest 和 FindNearest 执行不精确查找.....	216
11.5.5 限制记录范围.....	217
11.6 SQL 语言与 TQuery 组件.....	217
11.6.1 TQuery 组件.....	217
11.6.2 在 TQuery 组件中使用静态 SQL.....	218
11.6.3 在 TQuery 组件中使用动态 SQL 语句.....	220
11.7 多数据库表的查询.....	221
11.7.1 链接两数据库表.....	222
11.7.2 数据集的过滤(Filter).....	223
11.8 创建学生信息管理应用程序.....	224
11.8.1 数据模块(Data Module)使用.....	225
11.8.2 界面设计.....	226
11.8.3 有关代码编写.....	227
11.9 客户/服务器结构.....	230
11.9.1 客户机概述.....	230
11.9.2 中间件.....	231
11.10 本章小结.....	231
第十二章 创建自己的可视组件和 ActiveX 组件.....	233
12.1 Windows 的动态链接库 DLLs.....	233
12.1.1 如何创建一个新的 DLLs.....	233
12.1.2 如何调用 DLLs.....	234
12.2 创建自己的可视组件.....	238
12.2.1 包(Package).....	238
12.2.2 如何在组件板里加入自己的组件.....	238
12.3 为组件添加属性、方法和事件——TMinMax 组件.....	242
12.3.1 利用 Component 创建骨架单元.....	242
12.3.2 Delphi 对象类的访问类型.....	243

12.3.3	TMinMax 属性的添加	243
12.3.4	方法的添加——CompareAB 方法	245
12.3.5	事件 OnEqual 的添加	245
12.3.6	事件的触发.....	246
12.3.7	构造函数.....	246
12.3.8	MinMax 组件的安装.....	248
12.3.9	组件 MinMax 的应用.....	249
12.4	对组件 MinMax 的改进.....	252
12.4.1	通过函数方法访问属性值.....	252
12.4.2	从已有组件继承组件.....	253
12.4.3	组件的更新.....	256
12.5	带参数事件的添加.....	256
12.5.1	新事件类型的定义.....	256
12.5.2	触发事件和响应事件.....	257
12.6	ActiveX 组件	257
12.6.1	创建一基于 TButton 组件的 ActiveX 组件.....	258
12.6.2	ActiveX 组件的注册	262
12.6.3	添加新方法.....	262
12.7	ActiveForm 创建 Internet 应用程序	263
12.8	本章小结.....	265
第十三章	Internet 应用程序的开发.....	267
13.1	网页浏览器的创建.....	267
13.1.1	界面设计.....	267
13.1.2	THTML 组件的应用.....	268
13.2	Web Server Application 的应用.....	273
13.2.1	Hello! 应用程序的编写.....	273
13.2.2	网络投票应用程序.....	274
13.3	利用 ActiveForm 重写投票应用程序	278
13.4	本章小结.....	281
附录	Pascal 简介	283
A.1	代码格式与程序风格.....	283
A.2	一般数据类型.....	286
A.3	常量型.....	288
A.4	定义数据类型.....	288
A.5	流程的控制.....	293
A.5.1	if 语句	293
A.5.2	case 语句.....	294
A.6	循环.....	295
A.6.1	repeat 语句.....	295

A.6.2	while 语句.....	295
A.6.3	for 语句.....	296
A.7	过程与函数.....	296
A.7.1	过程与函数的概念.....	296
A.7.2	参数的传递.....	297

第一章 Delphi 基础

Delphi 是 Borland 公司推出的全新的可视化编程环境，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 Windows 图形用户界面的许多先进特性和设计思想，采用了面向对象程序语言(Object-Oriented Language)、当今世界上最快的编辑器和最为领先的数据库技术。本章将就一个简单的 Delphi 程序例子向读者介绍 Delphi3.0 的特性及最基本的可视化编程基础知识。

1.1 Hello! Delphi

从 Windows95 启动菜单运行 Delphi3.0 应用程序，如图 1-1 所示。

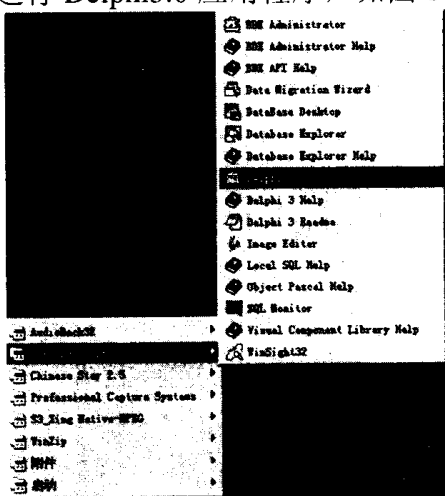


图 1-1 Delphi3.0 应用程序启动画面

1. 启动完毕出现 Delphi 3.0 集成开发环境 (IDE)，单击 File 菜单的 New Application 选项,创建一简单 Delphi3.0 应用程序，如图 1-2 所示。

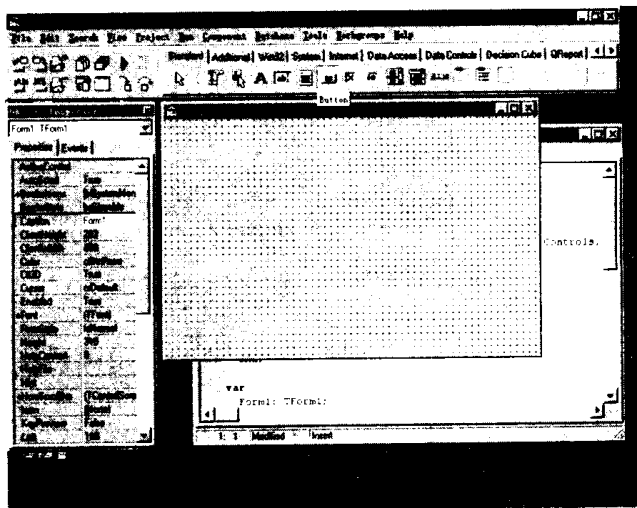


图 1-2 Delphi 集成开发环境

2. 用鼠标左键在菜单栏右下组件板（图 1-3）单击按钮控件。

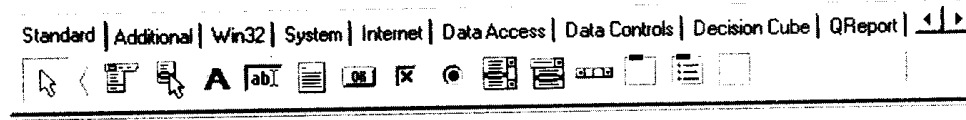


图 1-3 组件板

按住鼠标左键在窗体(form)中拖放该按钮控件，出现图 1-4 所示的窗体。

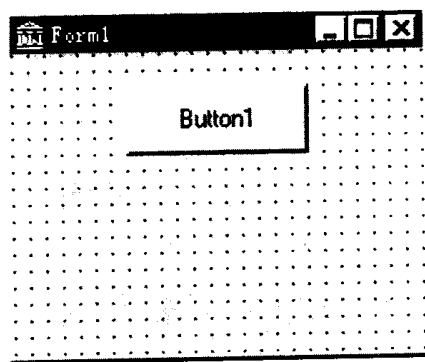



图 1-4 Form 窗体

3. 鼠标左键单击按钮控件，在 Object Inspector  窗口中设置按钮(Button1)Caption 属性为 Hello(如图 1-5 所示)。单击窗体，设置窗体(form1)Caption 属性为 Hello! Delphi3.0。我们会看到按钮上的 Button1 变为 Hello，窗体标题由 Form1 变为 Hello! Delphi3.0。

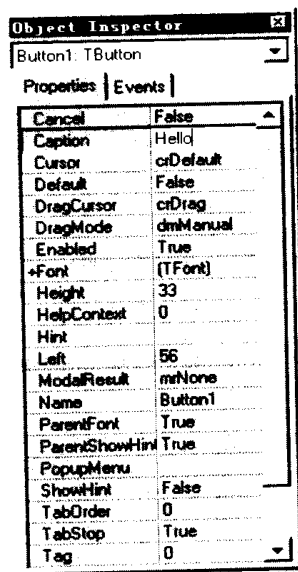



图 1-5 Object Inspector 窗口

重复步骤 2 和 3，从组件板中选取编辑框控件 ，用同样的方法在窗体中安放该编辑框控件，并设置 text 属性为空。

经过以上步骤，该程序界面设计业已完成，最后界面如图 1-6 所示。

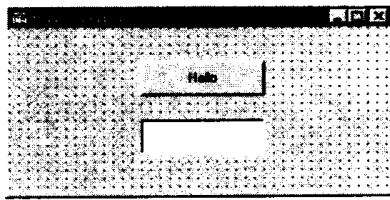


图 1-6 Hello!Delphi3.0 最后界面

4. 以上完成了程序界面设计, 虽还没有编写代码, 但 Delphi 业已完成了类与变量的说明, 这就是可视化编程带给程序员的方便之处。可视化编程让我们把精力放在了程序功能的实现上, 而界面上的工作由 Delphi 本身自动完成。类与变量的说明如下所示(不明白之处参考后面章节及附录编程语言部分)。

程序清单 1-1

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type
  TForm1 = class(TForm) {窗体的类型说明}
    Button1: TButton;
    Edit1: TEdit;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1; {说明一个窗体变量}
implementation
  {$R *.DFM}
end.

```

鼠标左键双击按钮控件, 会弹出代码编辑窗口, 如图 1-7 所示。

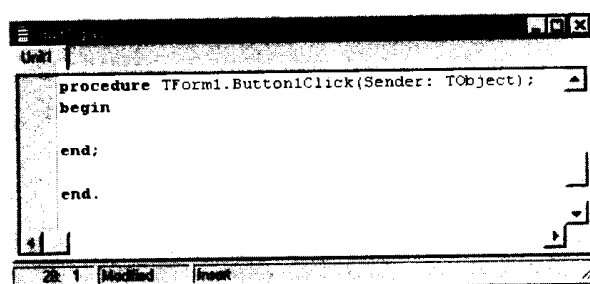



图 1-7 代码编辑窗口

在光标停留处输入如下代码：

```
edit1.text:= 'Hello! Delphi3.0!';
```

5. 鼠标左键单击主窗口程序运行  快捷键，即可看到该程序运行结果。单击 Hello 按钮，我们会看到编辑框中会出现“Hello! Delphi3.0!”问候语,如图 1-8 所示。

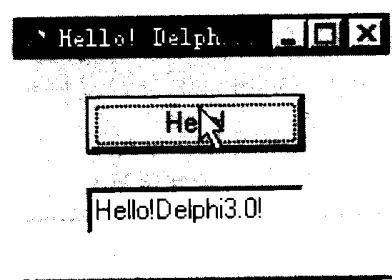


图 1-8 单击 Hello 按钮后的问候语

从这个简单的例子代码可以看到，Delphi 实际上是基于 Pascal 语言的一种版本，但它与传统的 Pascal 语言有天壤之别。如果我们使用过 Pascal 语言编程的话，要编写一个界面会让我们花大量的精力。而一个 Delphi 程序首先是应用程序框架，在框架上即使没有附着任何东西，仍可以严格地按照设计运行。我们可以在未编写任何代码的情况下运行以上这个例子。缺省的应用程序是一个空白的窗体(Form)，我们可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以被放大缩小、移动、最大最小化等，但我们却没有编写一程序。因此，可以说应用程序框架通过提供所有 Windows 应用程序共有的东西，为用户应用程序的开发打下了良好的基础。Delphi 已经为我们做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，只是不处理任何事情。我们的工作只是在框架中加入我们的程序，完成我们要实现的功能。

另外我们也可以看出，Delphi 是面向对象的编程语言。以上这个例子中，按钮对象接受响应用户的输入，即 click 事件，响应处理为

```
edit1.text := 'Hello! Delphi3.0!';
```

该语句实现的功能是让编辑框(edit1)的 text 属性为“Hello! Delphi3.0!”。响应处理过程中的 Windows 编程的回调、句柄处理等繁复过程，都由 Delphi 帮我们完成了。

从这个例子可以看出，Delphi 是一种面向对象的可视化编程语言。

1.2 Delphi 2.0 到 3.0 的飞跃

Delphi3.0 与 Delphi2.0 尽管在界面 (IDE) 上看似几乎完全一样，但是它们还是有很大区别的。下面就 3.0 在调试器、编辑器、网络以及数据库方面做的改进给以简单的说明。

1. IDE 调试器和编辑器功能的增强。Delphi 3.0 具有自动代码完成(automatic code

completion)和代码参数专家(code paramater expert)功能。

2. 关于 ActivieX 控件。在 3.0 中, 可以创建自己的 ActivieX 控件, 同时也可以使用在 Delphi 工程中预先编写好的控件(关于 ActiveX 在后面的章节中会具体介绍)。

3. 网络方面可以同时创建 Web 客户和 Web 服务器程序, 这显然会使 Delphi 3.0 在网络开发工具方面的竞争中更具优势。

4. 数据库体系结构发生了完全的改变, 它采取了多层化结构, 而非传统的客户/服务器设计, 这使得开发小而精巧的程序成为可能。

5. 主动窗体。我们可以将整个 Delphi 3.0 的应用程序作为一个 ActiveX 控件来运行。可以明确地在 Web 页运行完整的应用程序。

6. 线程安全。很多的图形例程将成为线程安全例程。

7. 包。现在我们可以自由地选择将运行库包在可执行文件中还是将其做为一个独立的 DLL 来使用(静态与动态链接)。这样的特征使独立开发者将所有的信息都包含在一个可执行文件中, 以便发行。同时, 具有这个特征之后它使开发者可以为所有桌面系统分发一个标准的.运行, 且只是很小的 EXE 文件。

1.3 可视化及 OOP 概念

面向对象的程序设计(Object-Oriented Programming, 简记为 OOP)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。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软件重用的开发工具, 从而具有强大的吸引力。

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 的推出, 填补了这项空白。我们不必自己建立对象, 只要在提供的程序框架中加入完成功能的代码, 其余的都交给 Delphi 去做。欲生成漂亮的界面和结构良好的程序丝毫不必绞尽脑汁, Delphi 将帮助我们轻松地完成。它允许在一个具有真正 OOP 扩展的可视化编程环境中, 使用它的 Object Pascal 语言。这种革命性的组合, 使得可视化编程与面向对象的开发框架紧密地结合起来。

1.3.1 什么是对象

面向对象的编程语言最重要的特性是对象。它含有数据及处理该数据的代码, 对象就像记录一样, 是一个数据结构。可以按最简单的理解为一记录。Hello!Delphi 例子中窗体(TForm1)、按钮(Button1)和编辑框(Edit1)就是三个对象, 我们可以看看它们的声明: