

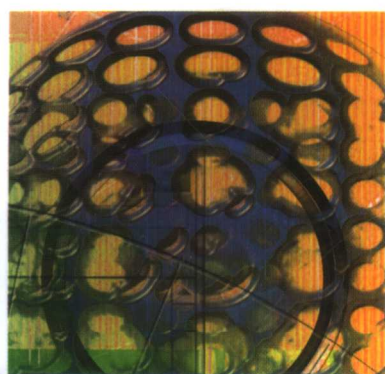
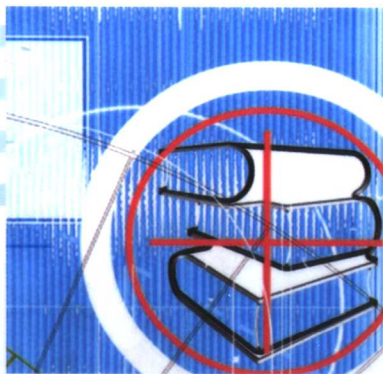
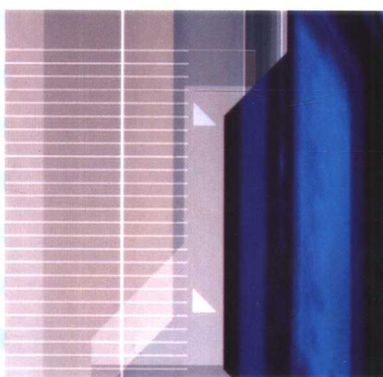


21 世纪高等学校应用型教材

Delphi

面向对象程序设计

□ 李俊平 主 编
□ 薛海燕 副主编



高等教育出版社
Higher Education Press

21 世纪高等学校应用型教材

Delphi 面向对象程序设计

李俊平 主 编
薛海燕 副主编

高等教育出版社

内 容 简 介

本书从实例出发,由浅入深,全面系统地介绍 Delphi 7 的基础知识以及开发中方方面面的技巧及应用,展示 Delphi 7 强大的功能及易用性。全书从面向对象的角度,在充分理解 VCL 组件体系结构的基础上,详细阐述 Delphi 的集成开发环境、Object Pascal 语言体系、面向对象的编程机制、VCL 体系结构、基于常用 VCL 组件的编程、文件操作和多媒体编程、异常处理机制以及数据库编程等知识。本书以面向对象作为主线,用丰富而贴近实际的例程来诠释 Delphi 强大的编程能力。更为重要的是,本书贯穿企业化的编程规范和理念,引导读者形成正确的编程风格。

根据应用型人才的培养目标,本书注重理论与实践相结合,注重基本知识的理解与基本技能的培养。书中内容丰富,章节安排合理,讲解深入浅出、通俗易懂,示例典型实用、易学易用,程序结构严谨、规范。本书适合作为应用型本科、高职高专院校、成人高校计算机及相关专业的 Delphi 程序设计教材,也可作为初学者的入门参考书。

本书所配电子教案及书中相关程序源代码均可从高等教育出版社的计算机教学资源网下载,网址为 <http://www.hep-st.com.cn> 或 <http://cs.hep.com.cn>。

图书在版编目(CIP)数据

Delphi 面向对象程序设计 / 李俊平主编. —北京:高等教育出版社,2005.2

ISBN 7-04-016066-8

I. D... II. 李... III. 软件工具-程序设计

IV. TP311.56

中国版本图书馆 CIP 数据核字 (2005) 第 009344 号

策划编辑 雷顺加 责任编辑 雷顺加 特约编辑 吴炳林
市场策划 韩 飞 封面设计 王凌波 责任印制 韩 刚

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>

经 销 北京蓝色畅想图书发行有限公司
印 刷 高等教育出版社印刷厂

开 本 787×1092 1/16
印 张 21.75
字 数 530 000

版 次 2005 年 2 月第 1 版
印 次 2005 年 2 月第 1 次印刷
定 价 28.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 16066-00

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序、保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010) 58581897/58581896/58581879

传 真：(010) 82086060

E - mail：dd@hep.com.cn

通信地址：北京市西城区德外大街4号

高等教育出版社打击盗版办公室

邮 编：100011

购书请拨打电话：(010)58581118

前 言

Delphi 是 Borland 公司继 Borland C++ 之后又一个使其步入辉煌的开发工具。Delphi 不仅具备超越 Visual Basic 和 PowerBuilder 的可视化快速开发能力,拥有能和 Java 和 C#媲美的面向对象的开发机制,而且有着和 C++ 一般的对 Windows 系统底层的良好控制能力。面向对象的程序设计(OOP)理念已成为当前程序设计的主流,它改变了以往传统的程序设计模式,并日益显示出其强大的生命力。Delphi 由于其强大的功能、快捷方便的开发模式,已成为一个极具代表性的面向对象的可视化开发工具。它将面向对象的程序设计方法与组件技术、数据库技术、网络技术以及可视化、代码自动生成等先进技术完美地融合在一起,给编程人员提供了一个超强高效的开发环境。Delphi 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 开发工具中首选的教学环境。

本书从实例出发,由浅入深,全面系统地介绍 Delphi 7 的基础知识以及开发中方方面面的技巧及应用,展示 Delphi 7 强大的功能及易用性。全书共分 13 章,从面向对象的角度,在充分理解 VCL 组件体系结构的基础上,详细阐述 Delphi 的集成开发环境、Object Pascal 语言体系、面向对象的编程机制、VCL 体系结构、基于常用 VCL 组件的编程、文件操作和多媒体编程、异常处理机制以及数据库编程等知识。本书以面向对象作为主线,用丰富而贴近实际的例程来诠释 Delphi 强大的编程能力。更为重要的是,本书贯穿企业化的编程规范和理念,引导读者形成正确的编程风格。

根据应用型人才的培养目标,本书注重理论与实践相结合,注重基本知识的理解与基本技能的培养。书中内容丰富,章节安排合理,讲解深入浅出、通俗易懂,示例典型实用、易学易用,程序结构严谨、规范。

本书尽量避免冗长无味的说明,代之以具体实用例子演示。通过例子,引导读者把握 Delphi 的精髓所在。在每一节中都有一到两个精心挑选的实例,并将其具体开发方法一步步展示出来,便于读者的实践。实例贴近实际项目且具有一定的趣味性,力图提升读者对于编程的兴趣。工程和单元文件名以及代码中变量、窗体、组件的命名都严格按照企业规范,体现了良好的编程风格;在每章的结尾讲解一个综合的例子,把一章中所有涉及到的内容包括在其中。通过实际的练习将知识融会贯通;将面向对象的编程思想融入到各个章节中,深入剖析 VCL 组件背后的机制,使读者知道怎么做,也知道为什么这么做,避免很多开发工具教学中“依葫芦画瓢”而不求甚解的弊病。

本书由李俊平主编,薛海燕任副主编。其中李俊平编写了本书第 1、3、4、5、6、8、10、11 和 12 章,薛海燕编写了第 2、7、9 和 13 章。全书由李俊平统稿,薛海燕和李俊平对全书进行了校对。尽管编者兢兢业业,唯恐误人子弟,但由于水平和时间所限,错误和疏漏在所难免,望读者见谅。

在本书酝酿和编写过程中,得到徐人凤老师、王明福老师和周光明老师的许多关心和帮助,在编程实例的选择和程序调试过程中,顾明、杨丽娟、蒋韶生、袁梅冷、张霞、梁雪萍、范新灿、肖正

兴等老师提出了许多宝贵的建议并参与了部分调试工作,在此表示衷心感谢。同时感谢深圳职业技术学院软件工程系的全体同仁对于本书编写工作的大力协助,感谢家人对无数个不眠之夜的理解和支持。

编 者

2004 年 12 月

目 录

第 1 章 Delphi 基础知识	(1)	2.5.1 算术运算符(+、-、*、/、div、mod)	(40)
1.1 Borland 公司和 Delphi	(1)	2.5.2 逻辑运算符(not、and、or、xor)	(40)
1.2 Delphi 的特点	(2)	2.5.3 关系运算符(=、<>、<、>、<=、>=)	(41)
1.2.1 选择什么开发工具?	(2)	2.5.4 位运算符(not、and、or、xor、shl、shr)	(41)
1.2.2 Delphi 能做什么?	(2)	2.5.5 指针运算符(^、+、-、=、<>)	(41)
1.3 Delphi 7 的集成开发环境	(3)	2.5.6 集合运算符(+、-、*、=、<>、in、<=、>=)	(41)
1.3.1 主窗体	(3)	2.6 语句	(42)
1.3.2 对象树状视图窗口	(4)	2.6.1 赋值语句	(42)
1.3.3 对象观察器窗口	(4)	2.6.2 条件语句	(43)
1.3.4 窗体设计器窗口	(5)	2.6.3 循环控制语句	(45)
1.3.5 代码编辑器窗口	(5)	2.6.4 with 语句	(47)
1.4 Delphi 编程的一般步骤	(7)	2.7 过程与函数	(48)
1.4.1 控制台应用程序	(8)	2.7.1 过程和函数的声明	(49)
1.4.2 可视化应用程序	(9)	2.7.2 过程和函数的参数	(50)
1.5 工程、窗体和单元文件	(12)	2.7.3 过程和函数的调用	(51)
1.5.1 工程文件	(12)	习题二	(52)
1.5.2 窗体文件	(13)	第 3 章 Delphi 面向对象编程思想	(55)
1.5.3 单元文件	(14)	3.1 面向对象的程序设计(OOP)的基本概念	(56)
习题一	(15)	3.1.1 封装性	(56)
第 2 章 Object Pascal 语言基础	(16)	3.1.2 继承性	(56)
2.1 保留字和标识符	(16)	3.1.3 多态性	(56)
2.1.1 保留字(Reserved Words)	(16)	3.2 类和对象	(57)
2.1.2 指令符(Directives)	(17)	3.3 类的声明	(57)
2.1.3 标识符(Identifiers)	(17)	3.3.1 类的数据成员(域)	(58)
2.2 注释	(18)	3.3.2 类的方法成员	(58)
2.3 常量与变量	(18)	3.3.3 类的属性成员	(58)
2.3.1 常量	(18)	3.3.4 类成员的可见性	(58)
2.3.2 变量	(20)	3.4 类的实现	(59)
2.4 数据类型	(21)	3.5 类的使用	(60)
2.4.1 简单数据类型	(22)	3.6 构造方法和析构方法	(65)
2.4.2 字符串类型	(29)		
2.4.3 结构类型(Structured Type)	(30)		
2.4.4 记录类型和动态数组的综合举例	(34)		
2.4.5 指针类型(Pointer)	(38)		
2.5 运算符与表达式	(39)		

3.6.1 构造方法	(65)	5.2 组框组件(GroupBox)和面板组件 (Panel)	(115)
3.6.2 析构方法	(66)	5.2.1 GroupBox 组件	(115)
3.7 类的继承	(67)	5.2.2 Panel 组件	(116)
3.7.1 类继承的语法	(67)	5.3 复选框(CheckBox)和单选按钮 (RadioButton)	(118)
3.7.2 类继承的应用	(68)	5.3.1 复选框(CheckBox)	(118)
3.7.3 VCL 的类库层次	(73)	5.3.2 单选按钮(RadioButton)	(118)
3.8 多态性初步	(74)	5.3.3 CheckBox 和 RadioButton 的 应用	(119)
3.8.1 静态方法	(75)	5.4 位图按钮组件(BitBtn)和加速按钮 组件(SpeedButton)	(121)
3.8.2 虚拟方法	(78)	5.4.1 位图按钮(BitBtn)	(121)
3.8.3 动态方法	(80)	5.4.2 加速按钮(SpeedButton)	(121)
3.9 抽象类和抽象方法	(81)	5.5 单选按钮组组件(RadioGroup)	(125)
3.10 类操作符	(83)	习题五	(128)
3.10.1 类型判断操作符 is	(83)	第 6 章 VCL 组件应用(续一)	(129)
3.10.2 类型强制转换操作符 as	(84)	6.1 形状组件(Shape)	(129)
习题三	(86)	6.2 列表框组件(ListBox)和下拉列表框 组件(ComboBox)	(130)
第 4 章 VCL 组件基础	(87)	6.2.1 ListBox 组件	(130)
4.1 VCL 基础	(87)	6.2.2 ComboBox 组件	(134)
4.1.1 VCL 组件的属性	(88)	6.2.3 ListBox 和 ComboBox 的综合 实例	(137)
4.1.2 VCL 组件的方法	(90)	6.3 计时器组件(Timer)	(140)
4.1.3 VCL 组件的事件	(91)	6.3.1 Timer 组件的属性和事件	(140)
4.1.4 VCL 组件库	(92)	6.3.2 小球碰撞的实例	(143)
4.2 窗体	(94)	6.4 一个综合实例——彩票投注开奖 模拟	(146)
4.2.1 窗体的属性	(94)	习题六	(155)
4.2.2 窗体的事件	(95)	第 7 章 异常处理与程序调试技术	(156)
4.2.3 窗体的方法	(95)	7.1 异常处理概述	(156)
4.2.4 静态创建窗体	(96)	7.1.1 传统的错误处理方式	(157)
4.2.5 动态创建窗体	(97)	7.1.2 使用异常进行错误处理	(157)
4.3 Parent、Owner、Sender 和 Self	(98)	7.2 异常处理的语法	(159)
4.3.1 Parent 与 Owner	(98)	7.2.1 try...except...end	(159)
4.3.2 Sender	(100)	7.2.2 try...except On...do...end	(161)
4.3.3 Self	(101)	7.2.3 异常的重引发	(163)
4.4 基本 VCL 组件应用——按钮、标签和 文本框	(102)	7.2.4 try...finally	(163)
4.4.1 按钮(Button)组件	(102)	7.2.5 资源保护与异常处理的混合 使用	(165)
4.4.2 标签(Label)组件	(103)	7.2.6 自定义异常	(166)
4.4.3 文本框(Edit)组件	(105)		
4.4.4 文本框事件的一般方法	(109)		
习题四	(110)		
第 5 章 VCL 组件应用	(111)		
5.1 备注框组件(Memo)	(111)		
5.1.1 Memo 组件的常用属性	(111)		
5.1.2 Memo 组件的应用	(112)		

7.3 程序的调试	(166)	习题九	(210)
7.3.1 程序中的错误	(166)	第10章 菜单和工具栏	(211)
7.3.2 执行程序	(167)	10.1 主菜单组件(MainMenu)和弹出式菜单 组件(PopupMenu)	(211)
7.3.3 断点(Breakpoint)的使用	(167)	10.1.1 主菜单组件(MainMenu)	(211)
7.3.4 运行到光标处	(168)	10.1.2 弹出式菜单组件 (PopupMenu)	(216)
7.3.5 单步调试	(168)	10.2 工具栏组件(ToolBar)和状态栏组件 (StatusBar)	(219)
7.3.6 观察窗(Watch List)的使用	(169)	10.2.1 工具栏组件(ToolBar)	(219)
7.3.7 Evaluate/Modify 对话框	(170)	10.2.2 状态栏组件(StatusBar)	(224)
习题七	(171)	10.3 动作列表组件(ActionList)	(226)
第8章 VCL 组件应用(续二)	(172)	10.4 CoolBar 组件和 ControlBar 组件	(229)
8.1 树状视图组件(TreeView)和列表视图 组件(ListView)	(172)	10.4.1 CoolBar 组件	(230)
8.1.1 树状视图组件(TreeView)	(172)	10.4.2 ControlBar 组件	(231)
8.1.2 TreeView 组件的属性、方法和 事件	(173)	10.5 动态菜单项和工具栏按钮	(232)
8.1.3 列表视图组件(ListView)	(177)	习题十	(234)
8.1.4 列表视图组件(ListView)的属性、 方法和事件	(178)	第11章 文件操作和管理	(235)
8.2 跟踪条组件(TrackBar)和进度条组件 (ProgressBar)	(180)	11.1 文件系统组件和外壳(Shell)组件	(235)
8.2.1 跟踪条组件(TrackBar)	(180)	11.1.1 文件系统组件	(235)
8.2.2 进度条组件(ProgressBar)	(184)	11.1.2 外壳(Shell)组件	(239)
8.3 Tab 组件(TabControl)和多页组件 (PageControl)	(186)	11.2 文本文件和有类型文件	(241)
8.3.1 Tab 组件(TabControl)	(186)	11.2.1 文本文件	(241)
8.3.2 多页组件(PageControl)	(188)	11.2.2 有类型文件	(245)
习题八	(194)	11.3 无类型文件	(251)
第9章 对话框	(195)	11.4 文件流(TFileStream)对象的操作	(254)
9.1 对话框函数和过程	(195)	11.5 注册表读/写操作	(258)
9.1.1 ShowMessage、ShowMessageFmt 和 ShowMessagePos 过程	(195)	习题十一	(263)
9.1.2 MessageDlg 和 MessageDlgPos 函数	(196)	第12章 图形、图像和多媒体技术	(264)
9.1.3 InputBox 和 InputQuery 函数	(198)	12.1 画布对象(TCanvas)	(264)
9.1.4 MessageBox 函数	(200)	12.1.1 TCanvas 对象的方法	(264)
9.2 公用对话框	(201)	12.1.2 TPen 和 TBrush 对象	(267)
9.2.1 “打开”对话框(OpenDialog)	(201)	12.2 图像编程技术	(270)
9.2.2 “另存为”对话框(SaveDialog)	(203)	12.2.1 图像组件(Image)	(271)
9.2.3 “字体”对话框(FontDialog)	(204)	12.2.2 JPEG 图像的显示	(271)
9.2.4 “颜色”对话框(ColorDialog)	(205)	12.2.3 图像格式转换	(273)
9.3 系统提供的 About 窗体	(206)	12.3 多媒体编程技术	(277)
9.4 自定义对话框	(207)	12.3.1 动画组件(Animate)	(277)
		12.3.2 多媒体播放组件 (MediaPlayer)	(279)
		12.3.3 RM 格式媒体的播放	(283)
		习题十二	(286)

第 13 章 数据库编程	(287)	开发	(320)
13.1 Delphi 自带的数据库	(287)	13.4.1 常用组件简介	(320)
13.2 利用 FormWizard 进行简单的 数据库编程	(290)	13.4.2 ADO 编程	(321)
13.3 基于 BDE 的数据库应用 程序开发	(292)	13.5 基于 dbExpress 的数据库应用 程序开发	(327)
13.3.1 BDE 简介	(292)	13.5.1 常用组件介绍	(327)
13.3.2 常用组件简介	(294)	13.5.2 使用 dbExpress 连接 SQLServer 数据库	(328)
13.3.3 使用 Table 组件进行数据库 应用程序开发	(295)	13.6 使用 Rave 组件制作报表	(330)
13.3.4 使用 Query 组件进行数据库 应用程序开发	(313)	13.6.1 常用组件介绍	(330)
13.4 基于 ADO 的数据库应用程序		13.6.2 利用向导创建报表	(331)
		习题十三	(336)
		参考文献	(337)

第1章

Delphi 基础知识

本章内容

- Borland 公司和 Delphi 的传奇历史
- 编程思想和编程工具的辩证关系
- Delphi 7 集成开发环境
- Delphi 编程的一般方法和步骤
- Delphi 工程、窗体和单元文件

1.1 Borland 公司和 Delphi

在 20 世纪 80 ~ 90 年代,中国的“老”程序员们可能没有不知道 Borland 公司的,从大学计算机房的 Turbo Pascal 和 Turbo C 到风靡世界的 Borland C++,都会引起我们对那个时代的回忆。中国程序员很多都有深深的“Borland 情结”。

Borland 公司是美国一家以生产开发工具为主的软件供应商,1983 年由 Philippe Kahn 创立,随后的 Turbo Pascal 和 Turbo C 的成功使 Borland 迅速崛起,而 1993 年的 Borland C++ 3.1 使得 Borland 公司的事业达到顶峰,那时候几乎全世界的程序员都用 Borland C++ 编程。1995 年 Borland 公司由于决策上的失误,没能及时投入资源开发 C++ 的 Framework,因而使 Borland C++ 在那场惨烈的 C++ 大战中败给了微软的 Visual C++,逐渐失去市场,Borland 公司也因此开始走下坡路,十年辉煌岁月就此终结。挽救大厦将倾的救世主就是 Delphi,一个可视化的 RAD 开发工具,1995 年 Delphi 一经推出即获得广泛好评,被称为“VB 杀手”,Borland 公司也因此翻开新的一页。目前,Borland 公司仍然是所有与微软正面竞争而硕果仅存的软件开发工具厂商,其主要产品 Delphi、C++ Builder、JBuilder 以及最新推出的 C# Builder 在中国拥有大量的用户群,“Borland 情结”还将延续下去。



图 1.1 Borland 公司总部

在很多程序员的眼里, Borland 就是技术的代名词。Borland 公司崇尚技术, 在微软公司的 C++ 项目组中有 60% 的成员都是从 Borland 挖来的, 其技术力量之雄厚由此可见一斑。Borland 公司 20 年的发展历程中涌现出众多的技术天才, Microsoft .NET 架构设计师和 C# 语言的缔造者 Anders Hejlsberg (见图 1.2) 就是其中的代表人物, 要知道, 这位软件领域的一代宗师就是当年的“Delphi 之父”。

才华横溢的 Anders 主持设计的 Delphi 造就了 Borland 公司新的传奇, Delphi 是古希腊一座神殿的名字 (见图 1.3), 而最著名的数据库产品 Oracle 名称的含义是神使。参与 Delphi 最初版本开发的资深工程师 Danny Thorpe 解释说: 如果你要和神使 (Oracle) 交谈, 那么去 Delphi 神殿吧。暗示 Delphi 是连接 Oracle 等数据库服务器的最好的客户端开发工具, 一个多么神秘而又富有哲理的名字。

Delphi 诞生之后便好评如潮, 与当时的 PowerBuilder 和 Visual Basic 并称为“C/S 三杰”, 随着 PowerBuilder 被 Sybase 收购而日渐衰落, 而微软全面转向 .NET 战略使得 VB 程序员升级到 VB.NET 门槛太高 (VB.NET 已经是一种面向对象的全新的语言, 只是继承了 VB 的语法而已), Delphi 因此成为 Windows 原生开发工具中硕果仅存的佼佼者。随后推出的 Linux 操作系统下的 Kylix 使 Delphi 成功实现了跨平台的开发, 而即将问世的 Delphi.NET 势必将把 Delphi 推向一个新的高峰。Borland 公司的天才们的智慧使 Delphi 时刻走在技术潮流的前沿, 成为开发工具领域一颗永远璀璨的明珠。

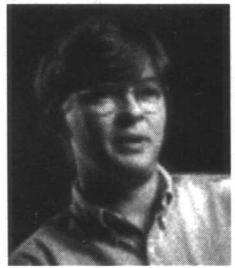


图 1.2 Delphi 之父

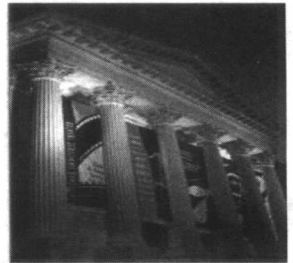


图 1.3 Delphi 神殿

1.2 Delphi 的特点

1.2.1 选择什么开发工具?

Delphi 的横空出世使得开发工具市场的格局发生了深刻的变化, 也造就了成千上万的 Delphi 程序员。我们为什么选择 Delphi 呢? 所有的 Delphi 程序员都会说: “因为 Delphi 是最好的!”; 当然, 我们在这里无意去继续那些关于开发工具孰优孰劣的争论, 这些争论在网上的程序员社区总是最热门且永无结论的话题, 谁也别想说服谁。其实, 开发工具终究是一种工具, 就像剑客手中的剑。高手之所以是高手, 不是因为他的剑, 而是因为他的剑招, 而剑招并不是由剑的好坏决定的, 但一把好剑绝对可以激发剑客的灵感和激情以把剑招发挥到极致。开发工具之于程序员如同剑之于剑客, 而程序员的“剑招”就是自身的编程修养、思想和理念等。只有扎扎实实地打好基础, 具备良好的编程思想, 才能成为软件开发的高手。

1.2.2 Delphi 能做什么?

Delphi 目前的最新版本是 Delphi 7.0, 采用严谨的面向对象的 Object Pascal 语言, 有世界上

最快的后端代码优化编译器,内置功能完善而强大的数据库引擎(BDE、ADO 和 dbExpress),加上出色的可视化框架类库 VCL,使得 Delphi 从众多的开发工具中脱颖而出。Delphi 7 全面支持 Windows 各种平台下的窗体应用程序、多媒体、数据库客户端、网络通信以及 COM、CORBA 和 Internet 等各方面应用的开发;它包含了大量内置的组件和设计工具,为编程人员提供完备的开发和调试环境,能大大缩短程序开发周期;通过 CLX 组件技术,Delphi 还成功实现了跨平台的项目开发;Delphi 7 中还包含 Delphi for .NET 的预览版开发工具,为将来全面支持 .NET 平台做好了准备。

与以往的版本相比,Delphi 7 突出了对 .NET、Web Services、Internet 应用以及建模技术的支持,主要表现在以下几个方面:

- (1) 对 .NET 结构的开发进行了准备,为将来全面支持 Microsoft .NET 打下基础。
- (2) 引入模型驱动架构(Model Driven Architecture, MDA),整合著名建模工具 ModelMaker,用户可以在集成环境中使用 UML 建模,并实现了模型和代码之间的无缝集成。
- (3) 引入 IntraWeb 技术,更方便地开发可视化的 Web 应用。
- (4) 增强对 Web Services 支持,简化基于标准 SOAP/XML Web Services 技术的应用开发。
- (5) 通过引入 Rave Reports 技术来代替 Delphi 以前版本中的 QuickReport,使程序员可以快速开发跨平台的报表。
- (6) 提供支持 Windows XP 主题风格的 VCL 组件,使 Delphi 开发的程序界面视觉效果更好。
- (7) 加强 DataSnap 技术,使应用程序可以高性能地与 Oracle、Microsoft SQL Server、IBM DB2、MySQL 等绝大多数数据库系统进行连接,成为数据库前端开发的绝佳工具。

以上这些优秀的特性使 Delphi 无可争议地成为 Windows 原生开发工具的瑰宝,并孕育着良好的发展前景。

1.3 Delphi 7 的集成开发环境

Delphi 7 的集成开发环境(IDE)由五个相对独立的窗口组成,分别是主窗体、对象树状视图、对象观察器、窗体设计器和代码编辑器,如图 1.4 所示。这五个窗口可以按照用户的编程习惯任意摆放。Delphi IDE 提供了编辑、调试和管理 Delphi 应用程序的各种命令和工具。软件开发的整个过程,包括程序设计、代码编译、调试、最后形成发布程序的全部工作都可在其中完成。Borland 公司的主打产品 Delphi、C++ Builder、Kylix 以及 JBuilder 都有着极为类似的开发环境界面,使其中任一种开发工具的程序员都能很快熟悉其他的开发工具。Delphi 的 IDE 也是作者所用过的开发工具中最强大也最舒适的一种开发环境。

1.3.1 主窗体

主窗体主要包括三部分:主菜单、工具栏和组件板,如图 1.5 所示。主菜单是 Delphi 各种命令的集合,包括开发过程中的控制命令以及 IDE 自身的设置命令;工具栏将一些常用的菜单命令以图标按钮的方式显示出来,便于快速访问;组件板是 Delphi 可视化开发的核心部分,包含了多个按功能分类的组件页:Standard 页、Additional 页、Data Access 页等,每个组件页含多个不同用途

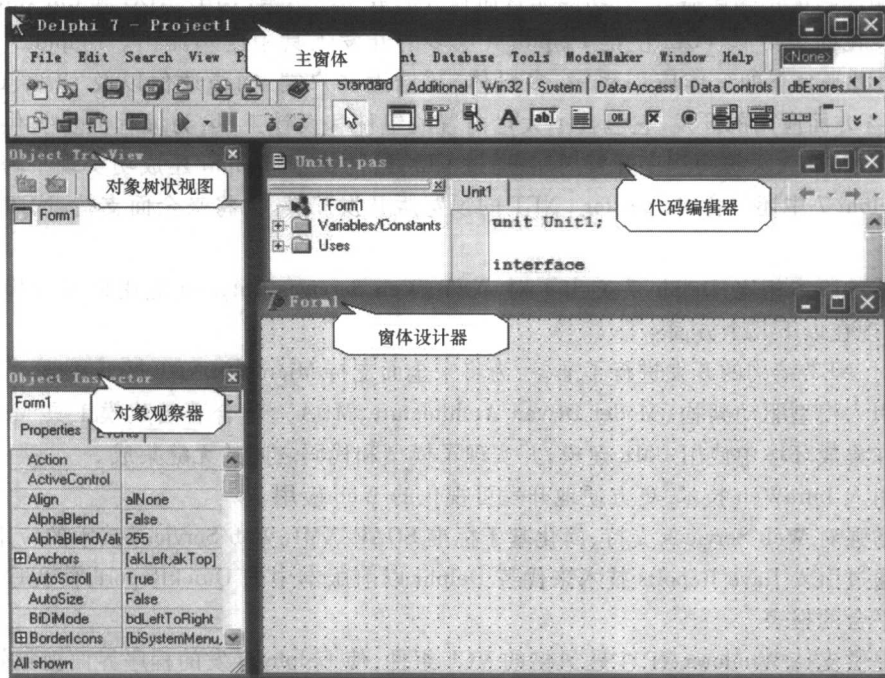


图 1.4 Delphi 7 的 IDE 即环境

的组件。Delphi 的内置组件也是所有开发工具中最丰富的。

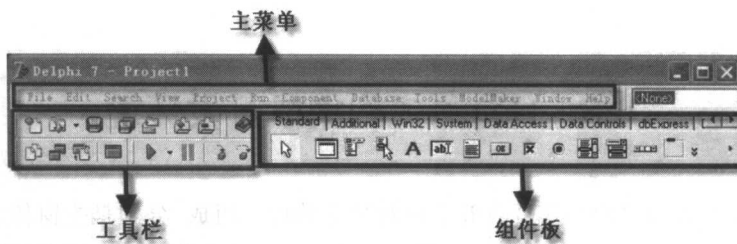


图 1.5 Delphi 的主窗体

1.3.2 对象树状视图窗口

对象树状视图 (Object TreeView) 是 Delphi 6 开始新增的一个窗口, 它显示了工程中的所有对象和它们之间的从属关系, 如图 1.6 所示。按快捷键 Shift + Alt + F11 或使用菜单 View/ObjectTreeView 可以激活该窗口。

1.3.3 对象观察器窗口

对象观察器 (Object Inspector) 的功能是显示和编辑组件的属性以及设置组件事件响应代码。它由一个下拉列表框和两个标签页组成 (见图 1.7), 下拉列表框显示当前窗体组件名列表,

默认显示为当前组件名; Properties 页设置窗体和窗体中组件的属性, 又分为属性名和属性值两栏; Events 页设置窗体和窗体中组件的事件, 又分为事件标题栏和事件处理过程栏。

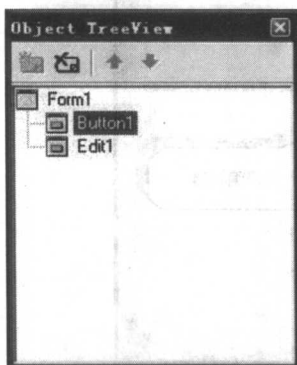


图 1.6 对象树状视图窗口

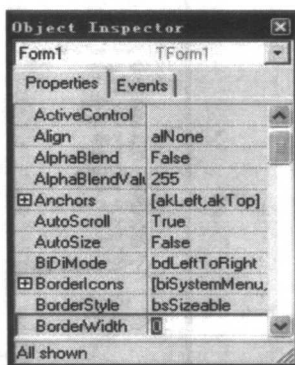


图 1.7 对象观察器窗口

1.3.4 窗体设计器窗口

窗体设计器(见图 1.8)用于设计程序界面。每个窗体都有一个窗体文件(.dfm)和一个单元文件(.pas)。Delphi 每次启动时, 自动生成一个新的工程, 其中包含一个空白窗体(默认名为 Form1)。一个工程可含多个窗体。

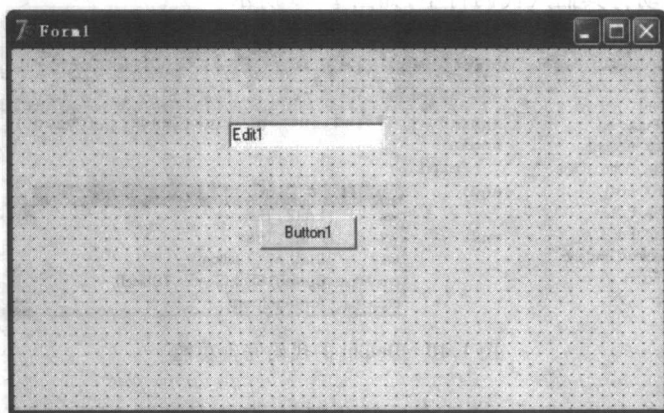


图 1.8 窗体设计器

1.3.5 代码编辑器窗口

代码编辑器窗口包含代码浏览器和代码编辑器两部分, 如图 1.9 所示。代码浏览器显示源程序中所有的类型、类、属性、方法及全局变量、全局函数、引用单元等的树形结构; 代码编辑器为代码的输入和编辑提供了一个方便的环境。每打开一个文件, 代码编辑器窗口的上部就会增加一个标签。可以通过单击标签在不同的文件中切换。图 1.9 中只打开了一个 Delphi 单元文件 Unit1.pas。

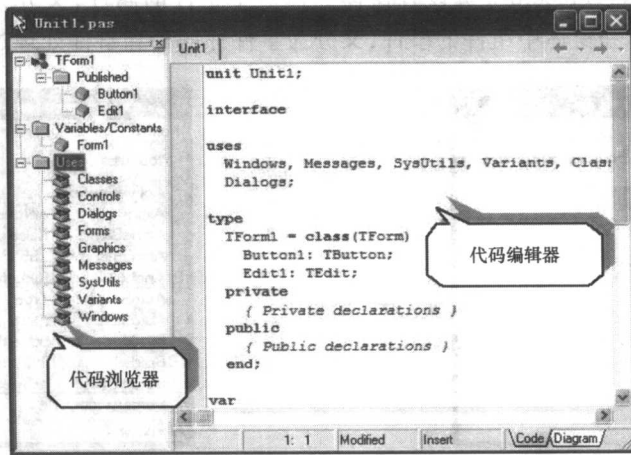


图 1.9 代码编辑器窗口

Delphi 的代码编辑器支持代码洞察 (Code Insight) 技术, 这项技术最初是在 Visual Basic 中首先推出的, 但 Delphi 的 Code Insight 技术更加完善和体贴。因为 Visual Basic 的 Code Insight 技术是基于 IDE 的, 而 Delphi 是基于编译器的。代码洞察技术包括如下的信息显示工具:

代码完善 (Code Completion) 在编写代码时, 只要输入已创建对象的名称和句点“.”, 稍作停顿, 系统将自动弹出一个提示列表框, 列出该对象的所有属性和方法。选择所需的属性和方法, 代码编辑器会自动将其插入代码行中, 如图 1.10 所示。

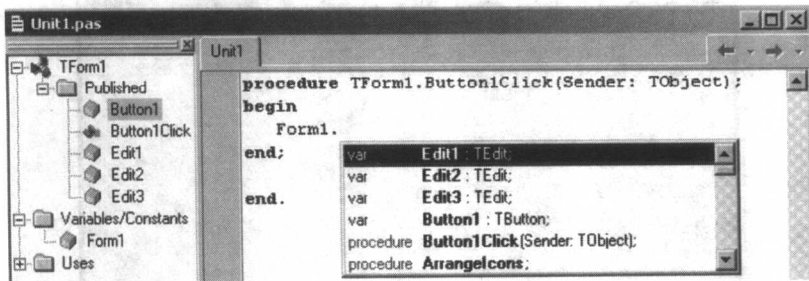


图 1.10 Delphi 的代码完善功能

代码参数提示 (Code Parameters) 在编写代码时, 只要输入已创建对象的方法 (内部过程或函数) 名或已经声明的子程序 (过程或函数) 名并输入一个左括号, 稍作停顿, 系统将自动弹出一个参数提示框, 提示每个参数的类型, 如图 1.11 所示。

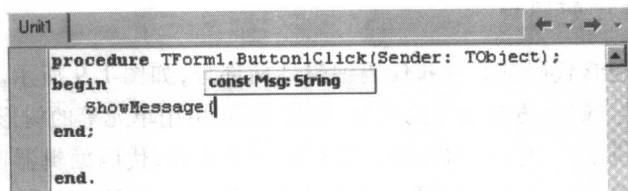


图 1.11 Delphi 的代码参数提示功能

代码模板专家(Code Templates) 用于提供语句模板,以帮助代码的编写。例如,输入“For”,按组合键 Ctrl+J,系统将自动弹出有关的代码模板工程,选中一种模板即可添加到代码中,既节省了代码输入时间,又避免了输入时的语法错误,如图 1.12 所示。

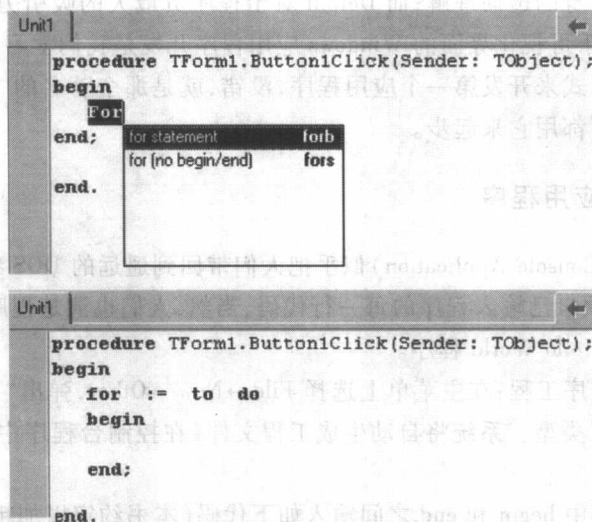


图 1.12 Delphi 的代码模板功能

符号洞察(Tooltip Symbol Insight) 在编写代码时,将鼠标置于一个标识符(类型名、变量名、函数名、过程名)上,稍作停顿,系统将自动弹出一个黄色提示框。提示框中显示该标识符的类型(变量以 var 表示,函数以 func 表示,过程以 proc 表示)、所属的单元名、声明的位置(行号)等信息,如图 1.13 所示。

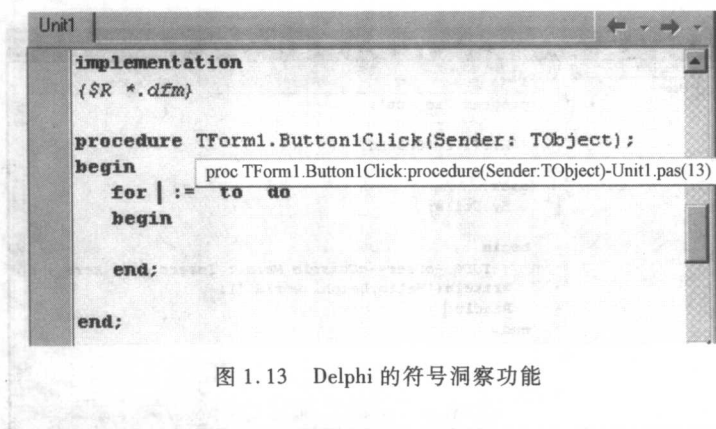


图 1.13 Delphi 的符号洞察功能

1.4 Delphi 编程的一般步骤

在这一节将开始 Delphi 编程的梦幻之旅,你可以充分享受 Delphi 集成开发环境带给你的编程的惬意和舒适,编程因此将成为一种艺术的创作。