

高职高专计算机系列规划教材

Delphi 程序设计实用教程 (第 2 版)

陈 瑞 叶核亚 编著

電子工業出版社·

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

Delphi 是一种优秀的、成熟的、功能全面的 Windows 应用程序开发工具。

本书主要内容包括 Delphi 的可视化开发环境、Object Pascal 语言、窗体和各种功能组件、面向对象程序设计、多文档界面设计、异常处理、图形和图像处理、目录与文件及流、多媒体技术、数据库操作与编程、网络编程。这些内容是构成 Delphi 应用程序的基本要素和必备知识。本书的任务就是介绍怎样利用这些功能来设计开发 Windows 界面的应用程序。

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。

本书适合作为高职高专院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。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

Delphi 程序设计实用教程/陈瑞, 叶核亚编著.—2 版. —北京: 电子工业出版社, 2004.3

(高职高专计算机系列规划教材)

ISBN 7-5053-9668-4

.D... . 陈... 叶... .软件工具—程序设计—高等学校: 技术学校—教材 .TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 010206 号

责任编辑: 洪国芬

印 刷:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787 × 1 092 1/16 印张: 21.75 字数: 557 千字

印 次: 2004 年 3 月第 1 次印刷

印 数: 7 000 册 定价: 26.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。
联系电话: (010) 68279077。质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

Delphi 是一种优秀的基于 Windows 的开发工具。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 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。

本书是 Delphi 程序设计的入门书，对于没有 Pascal 语言基础的读者，可先详细阅读第 2 章。

本书以 Delphi 7 为开发环境，主要介绍了构成 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、窗体和各种功能组件、面向对象程序设计、多文档界面设计、异常处理、图形和图像处理、目录与文件及流、多媒体技术、数据库操作与编程、网络编程。

本书力求使读者能快速、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。在“理论必需、够用为度”的前提下，加强实用性，加强实践性环节，为此本书每章都配有习题，还根据教学需要，安排了与课程衔接的大量实验，特别编写了第 14 章，并给出大型综合应用程序设计的选题和要求，最后结合作者多年从事教学和科研开发的经验，编写了一个大型综合应用程序的实例（学生管理信息系统）。

本书由陈瑞、叶核亚编著。其中叶核亚编写第 1、2、3、4、5、6、7、14 章，陈瑞编写第 8、9、10、11、12、13 章。在本书的编写过程中得到了廖雷、陈立、陈建红、阚建飞、沈晨鸣、王少东、刁翔、李林广、王青云等老师的帮助，在此深表感谢。

尽管作者写作时非常认真和努力，但由于水平有限，时间紧迫，书中难免有疏漏之处，恳请广大读者批评指正。

作者 E-mail 地址：yeheya@x263.net。

编 著 者
2004 年 1 月

目 录

| | |
|-------------------------|------|
| 第 1 章 Delphi 简介及其集成开发环境 | (1) |
| 1.1 Delphi 简介 | (1) |
| 1.1.1 特点 | (1) |
| 1.1.2 历史回顾 | (1) |
| 1.1.3 Delphi 7 的功能 | (2) |
| 1.2 Delphi 的集成开发环境 | (4) |
| 1.2.1 集成开发环境的组成 | (4) |
| 1.2.2 一个应用程序中所包含的文件 | (8) |
| 1.2.3 使用系统功能和帮助 | (12) |
| 1.2.4 创建控制台应用程序 | (14) |
| 1.2.5 程序调试技术 | (15) |
| 1.2.6 IDE 的其他功能 | (19) |
| 习题 1 | (20) |
| 实验 1 熟悉 Delphi 集成开发环境 | (20) |
| 第 2 章 Object Pascal 语言 | (22) |
| 2.1 语言成分 | (22) |
| 2.1.1 程序组成 | (22) |
| 2.1.2 数据类型 | (23) |
| 2.1.3 运算符与表达式 | (25) |
| 2.1.4 标准函数 | (29) |
| 2.1.5 常量与变量 | (30) |
| 2.2 语句与流程控制 | (32) |
| 2.2.1 基本语句 | (32) |
| 2.2.2 流程控制的概念 | (35) |
| 2.2.3 复合语句 | (36) |
| 2.2.4 条件语句 | (36) |
| 2.2.5 循环语句 | (40) |
| 2.2.6 转向语句 | (48) |
| 2.3 自定义数据类型 | (49) |
| 2.3.1 类型定义 | (49) |
| 2.3.2 枚举型与子界型 | (49) |
| 2.3.3 数组类型 | (50) |
| 2.3.4 字符串类型 | (59) |
| 2.3.5 集合类型 | (61) |
| 2.3.6 记录类型 | (65) |

| | | |
|--------------|-------------------------------------|---------------|
| 2.3.7 | 文件类型 | (67) |
| 2.3.8 | 指针类型 | (67) |
| 2.4 | 过程与函数 | (70) |
| 2.4.1 | 过程的说明与调用 | (70) |
| 2.4.2 | 函数的说明与调用 | (71) |
| 2.4.3 | 变量的作用域 | (73) |
| 2.4.4 | 参数传递 | (76) |
| 2.4.5 | 递归的过程与函数 | (78) |
| | 习题 2 | (80) |
| | 实验 2 调试控制台应用程序 | (81) |
| 第 3 章 | 窗体与基本组件 | (82) |
| 3.1 | 面向对象概念初步 | (82) |
| 3.1.1 | 类与对象 | (82) |
| 3.1.2 | 类的继承 | (84) |
| 3.1.3 | 组件与 TComponent 类 | (85) |
| 3.2 | 窗体 | (87) |
| 3.3 | 文本显示与编辑组件 | (90) |
| 3.3.1 | Label 标签 | (90) |
| 3.3.2 | Edit 单行编辑框 | (91) |
| 3.3.3 | MaskEdit 格式编辑框 | (91) |
| 3.3.4 | Memo 多行文本编辑框 | (93) |
| 3.3.5 | RichEdit 多行 RTF 文本编辑框 | (94) |
| 3.3.6 | SpinEdit 渐变编辑框 | (95) |
| 3.4 | 按钮 | (95) |
| 3.4.1 | Button 按钮 | (95) |
| 3.4.2 | BitBtn 图片按钮 | (96) |
| 3.4.3 | SpeedButton 加速按钮 | (96) |
| 3.5 | 选项按钮 | (100) |
| 3.5.1 | CheckBox 复选框 | (100) |
| 3.5.2 | RadioButton 单选按钮与 GroupBox 组框 | (100) |
| 3.5.3 | RadioGroup 单选按钮组 | (100) |
| 3.6 | 列表组件 | (101) |
| 3.6.1 | ListBox 列表框 | (101) |
| 3.6.2 | ComboBox 组合框 | (102) |
| 3.6.3 | CheckListBox 复选列表框 | (105) |
| 3.7 | 组件的布局 | (107) |
| 3.7.1 | 布局工具 | (107) |
| 3.7.2 | Panel 面板与控件的 Parent 属性 | (109) |
| 3.7.3 | 布局属性 Align 与 Anchor | (109) |
| 3.7.4 | Splitter 分隔条 | (110) |

| | | |
|--------------|------------------------|----------------|
| 3.8 | Timer 定时器 | (112) |
| | 习题 3 | (113) |
| | 实验 3 多项选择与数据移动 | (113) |
| 第 4 章 | 菜单、工具栏和对话框 | (115) |
| 4.1 | 菜单 | (115) |
| 4.1.1 | MainMenu 主菜单 | (115) |
| 4.1.2 | PopupMenu 弹出式菜单 | (118) |
| 4.1.3 | 菜单图像与 ImageList 组件 | (119) |
| 4.2 | ActionList 行为列表组件 | (119) |
| 4.3 | 工具栏与状态栏 | (122) |
| 4.3.1 | ToolBar 工具栏 | (122) |
| 4.3.2 | StatusBar 状态栏 | (125) |
| 4.4 | 消息框与输入框 | (127) |
| 4.4.1 | 消息框与信息函数 | (127) |
| 4.4.2 | 输入框与输入函数 | (128) |
| 4.5 | Windows 通用对话框 | (129) |
| 4.5.1 | 打开与保存文件对话框 | (129) |
| 4.5.2 | 字体与颜色对话框 | (131) |
| 4.5.3 | 查找与替换对话框 | (132) |
| 4.5.4 | 页面设置、打印与打印设置对话框 | (137) |
| | 习题 4 | (139) |
| | 实验 4 增加菜单等功能 | (139) |
| 第 5 章 | 使用复杂组件 | (140) |
| 5.1 | TreeView 树形视图 | (140) |
| 5.2 | ListView 图标视图 | (144) |
| 5.3 | 滚动控件 | (148) |
| 5.3.1 | ScrollBar 滚动条 | (149) |
| 5.3.2 | TrackBar 带刻度的滚动条 | (149) |
| 5.3.3 | UpDown 上下按钮 | (149) |
| 5.3.4 | ScrollBox 可滚动区域 | (151) |
| 5.4 | 日期类控件 | (152) |
| 5.4.1 | TDateTime 日期时间类 | (152) |
| 5.4.2 | DateTimePicker 日期和时间控件 | (153) |
| 5.4.3 | MonthCalendar 月历 | (153) |
| 5.4.4 | Calendar 日历 | (153) |
| 5.5 | PageControl 多页控件 | (153) |
| 5.6 | ProgressBar 进程条 | (157) |
| 5.7 | StringGrid 显示网格 | (157) |
| | 习题 5 | (159) |
| | 实验 5 设计电话号码簿 | (160) |

| | |
|----------------------------|---------|
| 第 6 章 面向对象程序设计 | (161) |
| 6.1 定义类及使用对象 | (161) |
| 6.2 类的封装 | (163) |
| 6.2.1 什么是封装 | (163) |
| 6.2.2 如何封装 | (164) |
| 6.2.3 self、as 和 is | (166) |
| 6.2.4 属性 | (167) |
| 6.2.5 方法的重载 | (169) |
| 6.3 类的继承与多态 | (173) |
| 6.3.1 继承的概念 | (173) |
| 6.3.2 TObject 类 | (174) |
| 6.3.3 定义子类 | (176) |
| 6.3.4 多态的概念 | (181) |
| 6.4 可视化继承 | (182) |
| 习题 6 | (187) |
| 实验 6 设计银行账户类及存取款操作方法 | (187) |
| 第 7 章 多文档界面设计 | (188) |
| 7.1 建立 MDI 应用程序 | (188) |
| 7.1.1 创建子窗口 | (188) |
| 7.1.2 动态建立子窗体 | (189) |
| 7.1.3 MDI 的属性和方法 | (190) |
| 7.1.4 MDI 的菜单设计 | (191) |
| 7.2 系统对象 | (195) |
| 7.2.1 Screen 变量 | (195) |
| 7.2.2 Application 变量 | (196) |
| 习题 7 | (198) |
| 实验 7 设计类似 Excel 的多文档表格处理程序 | (198) |
| 第 8 章 异常处理 | (200) |
| 8.1 程序错误及处理 | (200) |
| 8.2 异常处理机制 | (203) |
| 8.2.1 异常处理的概念 | (203) |
| 8.2.2 异常响应 | (204) |
| 8.2.3 异常保护 | (206) |
| 8.3 异常类 | (208) |
| 8.3.1 Exception 类 | (208) |
| 8.3.2 运行时间库异常 | (208) |
| 8.3.3 对象异常类 | (211) |
| 8.3.4 组件异常类 | (213) |
| 习题 8 | (214) |
| 实验 8 异常响应和异常保护 | (214) |

| | |
|---|---------|
| 第 9 章 图形、图像处理 | (215) |
| 9.1 图形设计 | (215) |
| 9.1.1 Canvas 画布对象 | (215) |
| 9.1.2 绘图方法 | (217) |
| 9.1.3 窗口绘图事件 | (220) |
| 9.1.4 响应鼠标事件 | (221) |
| 9.2 图像处理 | (224) |
| 9.2.1 图形、图像组件 | (224) |
| 9.2.2 常用图像类 | (227) |
| 习题 9 | (231) |
| 实验 9 图形设计 | (231) |
| 第 10 章 目录、文件和流 | (233) |
| 10.1 文件管理组件 | (233) |
| 10.1.1 Win3.1 选项卡上的文件管理组件 | (233) |
| 10.1.2 Samples 选项卡上的文件管理组件 | (236) |
| 10.2 文件操作 | (237) |
| 10.2.1 文件类型 | (238) |
| 10.2.2 类型文件的操作 | (238) |
| 10.2.3 文本文件操作 | (242) |
| 10.2.4 与文件目录相关的标准过程和函数 | (244) |
| 10.3 流 | (248) |
| 10.3.1 理解数据流 | (249) |
| 10.3.2 TStream 类 | (249) |
| 10.3.3 TFileStream 类 | (251) |
| 10.4 使用类方法进行文件存取 | (253) |
| 10.4.1 LoadFromFile 和 SaveToFile 方法 | (253) |
| 10.4.2 LoadFromStream 和 SaveToStream 方法 | (253) |
| 习题 10 | (254) |
| 实验 10 自动判题 | (254) |
| 第 11 章 多媒体技术 | (255) |
| 11.1 多媒体的基本术语 | (255) |
| 11.2 Animate 组件 | (256) |
| 11.3 MediaPlayer 组件 | (260) |
| 习题 11 | (264) |
| 实验 11 显示文件复制的过程 | (264) |
| 第 12 章 数据库操作和编程 | (265) |
| 12.1 数据库编程基础 | (265) |
| 12.1.1 关系数据库的基本概念 | (265) |
| 12.1.2 数据库体系结构简介 | (267) |
| 12.1.3 Delphi 的 BDE 数据库管理结构框架 | (269) |

| | | |
|---------------|-----------------------------|----------------|
| 12.2 | 数据库基本操作组件 | (272) |
| 12.2.1 | 基于 BDE 的数据库操作组件概述 | (272) |
| 12.2.2 | 数据集组件 Table | (274) |
| 12.2.3 | 数据源组件 DataSource | (276) |
| 12.2.4 | 数据控制组件 DBGrid 与 DBNavigator | (278) |
| 12.3 | 操纵表中字段 | (279) |
| 12.3.1 | 字段对象 | (279) |
| 12.3.2 | 操纵字段的数据控制组件 | (285) |
| 12.4 | SQL 与 Query 组件 | (285) |
| 12.5 | Database 组件 | (291) |
| | 习题 12 | (292) |
| | 实验 12 数据库操作 | (292) |
| 第 13 章 | 网络编程 | (293) |
| 13.1 | TCP/IP 编程 | (293) |
| 13.1.1 | 在 Delphi 中使用 TCP/IP 编程 | (293) |
| 13.1.2 | TCPServer 和 TCPClient 组件 | (293) |
| 13.2 | WebSnap 基础 | (297) |
| 13.3 | Web App Debugger | (306) |
| | 习题 13 | (307) |
| | 实验 13 制作 Web 应用程序 | (307) |
| 第 14 章 | 综合应用设计与实习 | (308) |
| 14.1 | 综合应用设计实例——学生管理信息系统 | (308) |
| 14.1.1 | 创建 Access 数据库 | (308) |
| 14.1.2 | 使用 ODBC 或 BDE 配置数据库 | (308) |
| 14.1.3 | MDI 主窗口 | (310) |
| 14.1.4 | 输入子窗口 | (312) |
| 14.1.5 | 浏览子窗口 | (316) |
| 14.1.6 | 条件查询子窗口 | (319) |
| 14.1.7 | 统计汇总子窗口 | (323) |
| 14.2 | 实践性环节 | (325) |
| 14.2.1 | 课程设计目的与要求 | (325) |
| 14.2.2 | 课程设计选题 | (325) |
| 附录 A | 标准过程与函数 | (327) |
| 附录 B | VCL 组件的属性、方法与事件 | (329) |
| 附录 C | 出错信息 | (332) |

第 1 章 Delphi 简介及其集成开发环境

本章首先简要介绍 Delphi 的特点、功能及发展历程，使读者对 Delphi 有一个初步认识，然后详细介绍 Delphi 集成开发环境的功能及使用方法。

1.1 Delphi 简介

1.1.1 特点

Delphi 是一种优秀的、成熟的、使用广泛的 Windows 应用程序开发工具，具有如下特点：

- 高性能的集成开发环境（Integrated Development Environment，IDE）；
- 结构严谨、体系完备的面向对象程序设计语言；
- 速度快、效率高、稳定性强的编译器；
- 灵活的、可扩展的数据库结构；
- 可扩充的 VCL 组件设计与操纵模式。

1.1.2 历史回顾

从核心上说 Delphi 其实是一个 Pascal 编译器。Borland 公司从 1983 年推出 Turbo Pascal 1.0 开始，就一直推动着 Pascal 及 Delphi 向前发展，至今走过了近 20 年艰苦创作的历程，并完成了从 DOS 到 Windows 的质的飞跃。

1. DOS 时代

Turbo Pascal 以其结构化的简练和真编译器的性能，综合了高级语言易于使用和汇编语言效率高的优势，具有稳定、优雅以及编译速度快等特点，成为代表性的编程环境，并使 Pascal 成为当时最广泛使用的语言之一。

2. Windows 时代

(1) Delphi 1

1995 年，Borland 公司推出了基于 Windows 平台的全新的面向对象的 Delphi 1，成为 Windows 开发工具的新突破。Delphi 1 提供了一种完全不同于以往的开发方法，它是第一个综合了集成开发环境、优化的源代码编译器、可扩展的数据库访问引擎的 Windows 开发工具，吸引了很多程序员。

其后 Borland 公司以每年一个版本的速度对 Delphi 进行功能扩充和性能提高。

(2) Delphi 2

1996 年，Delphi 2 在 32 位的操作系统 Windows 95 和 Windows NT 下实现了原有的一切功能，还增加了许多功能，诸如，32 位的编译器能生成速度更快的应用程序，对象库得到进一步丰富和扩展，完善数据库支持，改进字符串处理，支持 OLE 和可视化窗体继承，与 16 位

Delphi 兼容，增强数据库和客户机/服务器的功能。

(3) Delphi 3

1997年，Delphi 3的出现使本来极其复杂的COM、ActiveX、WWW应用程序，“瘦”客户应用程序，多层数据库系统体系结构等开发技术变得非常容易使用。Delphi 3的代码内视（Code Insight）技术简化了代码编写过程。

(4) Delphi 4

1998年，Delphi 4中Module Explore技术的引入使程序员能够以一致的图形界面浏览和编辑代码。代码导航和类自动生成的功能使程序员只需关注应用程序本身，而不必在输入代码上花费太多精力。IDE经过重新设计，支持浮动和可停靠的工具栏和窗口，调试器也做了改进。Delphi 4的MIDAS、DCOM和CORBA等技术使其应用范围扩展到企业级。

(5) Delphi 5

1999年下半年推出的Delphi 5，在简化程序编写、增加Internet开发功能、增强稳定性等方面取得了较大进步。

Delphi 5增加了更多的功能，使程序的编写更简单，程序员可以把精力都集中在想写什么而不是怎样写上。这些新功能包括：进一步增强IDE和调试器的功能，提供了TeamSource小组开发软件和转换工具等。

为简化Internet的开发而增加了许多新功能，例如，Active Server Object Wizard用于创建ASP，InternetExpress组件用于支持XML和新的MIDAS功能，使Delphi成为Internet的一个通用数据平台。

(6) Delphi 6

大约两年后，Borland的年度巨献Delphi 6终于在世人的面前揭开神秘的面纱。Delphi 6不但在传统的开发能力方面具有持续的进步，使程序员能够享有更具生产力的开发环境，而且还支持最新的信息技术，例如XML/XSL、SOAP、Web Service等，使程序员能够快速地使用这些新技术。

1.1.3 Delphi 7 的功能

2002年Borland用Delphi 7 Studio首次开辟了通往.NET的独立道路，它是应用于Windows平台上的具有跨平台功能的快速应用程序开发环境。Delphi 7 Studio的特征还包括企业应用程序设计与分发，允许开发者利用企业应用程序开发方法，通过新的UML设计器与模型驱动结构技术来更快地将概念转换为产品。

1. Delphi 7 的基本功能

(1) 集成开发环境（IDE）

包括编辑器、调试器、窗体设计器和许多其他功能。

(2) 可视化组件库（VCL）

包括组件板上的几百个组件。

(3) 全面支持 Win32 API

包括COM、GDI、DirectX、多线程以及Microsoft和第三方软件开发包（SDK）。

(4) 数据库支持

包括Borland数据库引擎（BDE）、本地表的BDE驱动、数据访问组件（用来将其他的

数据库引擎嵌入到 VCL 中)、数据控制组件、数据库浏览器、数据共享库、支持 ODBC 数据源以及 InterBase Express 本地 InterBase 组件。

提供访问 InterBase、Oracle、Microsoft SQL Server、Sybase、Informix 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。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。

(5) COM 组件生成向导

包括 ActiveX 控件、Active 窗体、Automation 服务器以及属性页,直接支持 ADO(ActiveX 数据对象)。

(6) 开发 Internet 应用程序

包括 WebBroker 和 FastNet 向导和组件,Web 发布功能可以方便地在 Web 上分发 ActiveX 项目。Active Server Object Wizard 用于创建 ASP,InternetExpress 组件用于支持 XML 和新的 MIDAS 功能,使 Delphi 成为 Internet 的一个通用数据平台。

(7) 支持多层分布式应用程序的开发

CORBA 和 MIDAS 的支持和开发许可,使多层应用程序的开发大大简化。

(8) 制作工具

包括安装应用程序制作工具 InstallShield Express 和快速数据迁徙工具 Data Pump Expert。

2. Delphi 7 Studio 的特征与优点

(1) 企业应用程序 MDA 开发

允许开发者在应用程序的设计与分发的全过程中复用一个应用来加速开发过程,同时明显地减少代码量与缩短开发时间。

(2) RAD 可视化 Web 开发

RAD 可视化 Web 开发不仅允许开发者使用 Delphi 7 Studio 环境来可视化地建立 Web 应用程序,也允许开发者使用应用程序模式框架来显式地处理会话管理,以减少通用服务器端的开发任务。

(3) 为 Linux 内建的跨平台支持

Delphi 7 Studio 与 Delphi 语言版本的 Borland Kylix 3 共同发布。Kylix 3 是第一个为 Linux 操作系统快速地建立数据库、GUI 及 Web 服务应用程序的高性能的可视化的集成开发环境。

(4) 企业型报告能力

Delphi 7 Studio 允许开发者建立跨平台的、有助于确定应用程序运行效率的报表。

(5) Windows XP 应用程序

Delphi 7 Studio 包括了对 Windows XP 主题的支持,允许开发者建立使用 Windows XP 用户界面主题的应用程序。

(6) 多层应用程序分发

Delphi 7 Studio 允许开发者无缝地将单层、客户机/服务器应用程序扩展为多层应用程序,无须额外的运行费用。

3. 增强的网络应用功能

Delphi 7 引入了 IntraWeb 技术,用户利用 Delphi 7 提供的窗体工具来使用 IntraWeb 技术

开发网络服务程序以及 WebBroker、WebSnap 应用程序的页面。

使用新增加的组件，用户可轻松地使用 SOAP (Simple Object Access Protocol) 组件来编写网络服务程序以及访问这些网络服务的客户端应用程序。Delphi 7 的 SOAP 技术已成功地应用于 Windows 和 Linux 操作系统中，为跨平台的分布式应用提供了良好的基础。

此外，Delphi 7 还加强了对 WebServices 的支持，主要体现在以下几点。

(1) 新的 UDDI 浏览器

Delphi 7 提供新的 UDDI 浏览器，使用户可通过 UDDI 接口方便地从 WebServices 中导出 WSDL 文档。

(2) SOAPheaders

SOAP 记录了用户在客户端和服务端传递数据时的一些信息，Delphi 7 提供一些新的类和接口，使用户能方便地从中提取和插入头节点。

(3) Attachments

Attachments 作为多窗体的一部分采用 SOAP 编码的信息进行传递，当一个应用程序接收到传递过来的 Attachments 后，会自动将其保存为一个临时文件，然后就可以在应用程序中使用。

4. 新增组件

(1) 新增单元文件

新增名为 DBClientActns 的单元文件，用于开发与客户端数据集相关的数据库应用程序，其中包含 TClientDataSetApply、TClientDataSetUndo 和 TClientDataSetRevert 3 个新的动作组件。

(2) 新增组件

- dbExpress 组件板上新增 TSimpleDataSet 组件，用于开发简单的 two-tiered 数据库应用程序。
- Dialogs 组件板上新增 TPageSetupDialog 组件，为应用程序提供 Windows 标准页面设置对话框。
- Additional 组件板上新增 TXPColorMap、TStandardColorMap 和 TTWilightColorMap 3 个组件，用于为菜单和工具条设置颜色。

1.2 Delphi 的集成开发环境

Delphi 的集成开发环境 (IDE) 提供编辑、编译、运行和调试 Delphi 应用程序的软件环境。

1.2.1 集成开发环境的组成

Delphi 的集成开发环境主要包括 5 个部分：主窗口、窗体设计器、对象监视器、代码编辑器和代码浏览器，如图 1.1 所示。

集成开发环境的各组成部分是协同工作的。当用户设计窗体时，Delphi 在后台自动为正在窗体中操纵的控件生成代码。在编辑器中，用户编写程序的同时，也可以设置断点和监控点来调试程序。

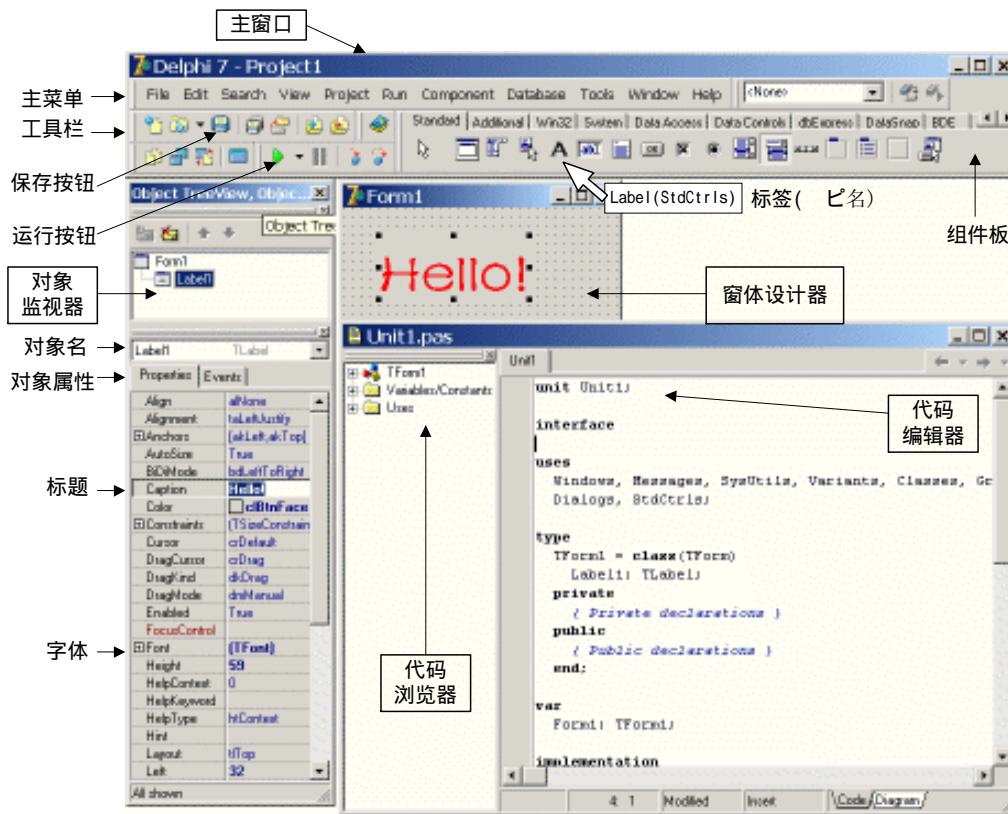


图 1.1 Delphi 7 的集成开发环境

1. 主窗口

主窗口是 Delphi 集成开发环境的控制核心，具有 Windows 应用程序的窗口风格和功能。主窗口包括 3 个部分：主菜单、工具栏和组件板。

(1) 主菜单

主菜单 (Menu) 提供了所有进行程序开发所需的命令和功能，包括创建、打开或保存文件、调用向导、查看其他窗口、修改选项等。Delphi 主菜单共有 11 项，如表 1.1 所示。

表 1.1 菜单名称和主要功能

| 主菜单名称 | 主要功能 |
|----------------|------------------------------------|
| File (文件) | 新建应用程序；新建、打开、保存单元或项目文件 |
| Edit (编辑) | 实现标准的编辑功能以及设置项目特性 |
| Search (查询) | 实现文本查找、替换功能 |
| View (浏览) | 打开项目管理器、对象查看器，显示调试信息，查看窗体和单元，设置工具栏 |
| Project (项目) | 管理项目文件、编译项目和项目设置 |
| Run (运行) | 提供运行、调试程序的各种命令 |
| Component (组件) | 建立、安装组件及设置用户需要的组件板 |
| Database (数据库) | 提供数据库应用程序的各种工具 |
| Tools (工具) | 进行 Delphi 开发环境的设置等 |
| Window (窗口) | 切换窗口 |
| Help (帮助) | 提供全面的 Delphi 帮助信息 |

(2) 工具栏

工具栏 (Toolbar) 上的每个按钮实现 Delphi 的某项功能, 使用它们可以快速执行命令, 而不必到菜单项中去查找, 例如打开文件、保存文件、运行程序等。除了组件板, Delphi 中默认显示的工具栏有 6 个: Standard、View、Debug、Custom、Component Palette 和 Desktops。工具栏上各按钮的功能说明如图 1.2 所示。

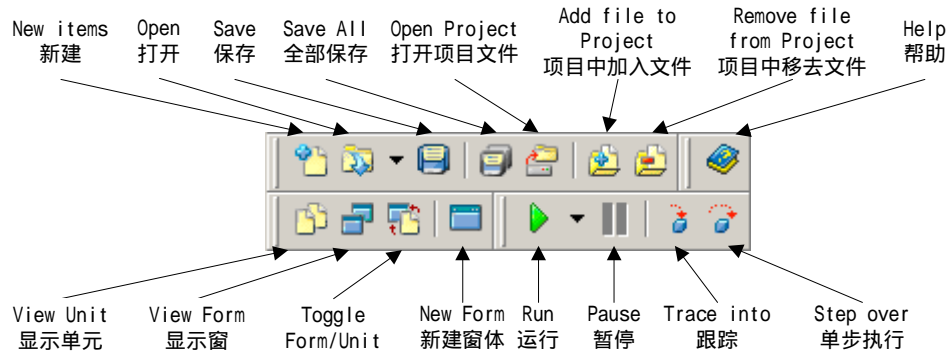


图 1.2 工具栏上各按钮的功能说明

(3) 组件板

组件是构成 Delphi 应用程序界面的基本元素。Delphi 提供了大量丰富的组件, 如按钮、编辑框和组合框等。

组件板 (Component Palette) 包含 Delphi 中安装的所有组件和 ActiveX 控件。它是一种多页结构, 将组件按功能分成多个页面 (或称为选项卡), 共有 Standard、Additional、Win32、System 等 32 个页面, 每页上放置具体的组件。单击页上的标签名, 可以切换页面; 使用组件板上的滚动箭头, 可以查看窗口上未显示的页面。

2. 窗体设计器

窗体设计器, 简称窗体 (form), 是应用程序在设计期间的操作界面。窗体上除标题栏以外的区域称为窗体的客户区。窗体的客户区可看做是艺术家的画布, 在上面可以绘制出各种各样的 Windows 应用程序。

Delphi 使用窗体设计器实现应用程序的用户界面。窗体启动时是一个空白的窗口, 可在其上放置组件。方法很简单, 只要在组件板上选择一个组件并将它放于客户区中, 就能够在窗体上看见该组件, 此时可以用鼠标调整组件的位置和大小, 用对象监视器来改变组件的属性, 还可用代码编辑器为组件编写相应的事件处理程序等。

窗体与窗口是同一对象在不同时期的两种称号, 设计时称为窗体, 运行时称为窗口。

3. 对象监视器

利用对象监视器 (Object Inspector) 可以修改窗体或组件的属性, 或者使它们能够响应不同的事件。选择主菜单 “View | Object Inspector” 命令可显示对象监视器。

属性 (property) 是一些数据, 如高度、颜色、字体等, 它们决定了组件在屏幕上的外观。

事件 (event) 是一种消息处理机制, 它能够捕捉某种情况的发生并做出反应。例如, 鼠标单击事件、窗口尺寸改变事件等。

作为一个动态的工具，对象监视器显示的总是当前选中组件的属性，而组件名及其类名显示在对象监视器上方的列表框中。

对象监视器是多页结构，包括属性页 Properties 和事件页 Events。单击页上的标签名，可以切换页面。

(1) Properties 页

Properties 页用于在设计时查看和改变一个对象的相关属性。窗体及组件的属性值可在设计时手工修改，也可以在运行时通过程序动态地修改。在对象监视器上设置属性有 4 种方法。

- 直接输入法。有些属性只是一个简单的数值或字符串，如 Name、Width、Caption 等，对这些属性的修改可以采用直接输入法，即在属性右侧的编辑框中直接输入字符串或数值。
- 使用组合列表框。有些属性的取值只能是几个固定值中的一个，系统将属性的所有可能取值放在组合列表框中，使用时从中选取某一项即可。例如，设置 Visible 属性，单击 Visible 属性框右边的下拉箭头可弹出一个下拉列表，其中只有 true 和 false 两个值，单击所需的属性值即可。
- 通过对话框选取。有些属性编辑框的右端有一个带有三个圆点的省略号按钮（【...】）表示将通过对话框选取属性值，如单击 Font 属性右端的【...】按钮，则弹出一个对话框，选择相应值设置属性。
- 子属性设置。有些属性是复合属性，如 Font，它左侧有一个“+”标记，表示下面还有子属性。单击“+”标记，展开子属性列表，标记变为“-”，这时可以修改子属性；再单击“-”标记，则收起子属性列表。

(2) Events 页

Events 页上列出了当前组件能够响应的事件。

编写某组件的某个事件处理程序，首先要在窗体上选中某组件，使之成为当前组件，然后在 Events 页上双击该组件要响应的事件，系统就会在代码编辑器中自动生成一个事件处理过程框架，供用户输入源代码。

一个组件的某个事件处理程序也可以执行另一个事件处理程序，方法是从相应事件的下拉列表中选择一个事件处理程序。

【例 1.1】创建并运行 Windows 应用程序。

本例演示在 Delphi 的集成开发环境中创建一个应用程序。基本操作如下：

- 新建一个应用程序，单击组件板上的“标签”按钮，放置在窗体 Form1 上。修改标签的属性，Caption 属性框中输入“Hello!”，Font 属性框中修改字体、字号、颜色等，如图 1.1 所示。
- 单击保存按钮【】或执行“File | Save”菜单，保存单元文件为“Fhello.pas”。
- 单击程序运行按钮【】或执行“Run | Run”菜单。程序运行窗口如图 1.3 所示。

至此，我们已使用 Delphi 的 IDE 创建了一个完整的 Windows 应用程序，并实现了程序的保存和运行。本例中我们未写一行程序代码，全部代码都是由系统自动生成的！

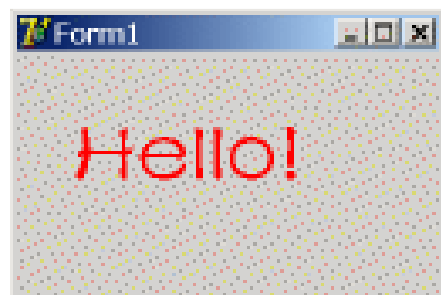


图 1.3 程序运行窗口

4. 代码编辑器

代码编辑器是输入代码编写程序的地方，也是 Delphi 自动生成代码的地方。代码编辑器也是一个多页结构，每一页对应着一个源代码编辑器或文件。单击每页的标签名，就可在多个单元文件之间进行切换。

Delphi 启动时，默认的单元文件为“Unit1.pas”。例 1.1 中保存单元文件后，系统设置该代码页名为“Fhello”。当应用程序中加入一个窗体时，Delphi 就会自动创建一个新的单元文件（Unit2.pas），并显示在代码编辑器中。如果要编辑一个与窗体无关的单元文件，可以使用“View | Unit”菜单打开“View Unit”对话框，从单元列表中选择一单元文件（.pas）。

5. 代码浏览器

代码浏览器以树形视图方式显示列在代码编辑器中的单元文件。通过代码浏览器，可以方便地在单元文件中漫游，或在单元文件中加入新的元素，或将已有的文件改名。代码浏览器和代码编辑器是一一对应的。

1.2.2 一个应用程序中所包含的文件

一个 Delphi 的应用程序由多个文件组成，其中必须有一个项目文件（.dpr）和至少一个单元文件（.pas）。主要文件类型如表 1.2 所示。

表 1.2 主要文件类型

| 文件类型 | 扩展名 | 说明 | 例 1.1 生成的文件 |
|--------|------|---------------------|-------------|
| 项目组文件 | .bpg | | |
| 包文件 | .dpk | | |
| 项目文件 | .dpr | 记录应用程序中包含哪些单元 | Phello.dpr |
| 单元文件 | .pas | Object Pascal 源代码文件 | Fhello.pas |
| 窗体描述文件 | .dfm | 描述窗体和组件的属性 | Fhello.dfm |
| 编译文件 | .dcu | 单元编译后生成 | Fhello.dcu |
| 选项文件 | .dof | 记录项目的选项设置 | Phello.dof |
| 资源文件 | .res | | Phello.res |
| 动态链接库 | .dll | | |
| 配置文件 | .cfg | 保存项目配置 | Phello.cfg |
| 执行文件 | .exe | | Phello.exe |
| 单元文件备份 | ~pas | | Fhello.~pas |
| | ~dfm | | Fhello.~dfm |
| | ~dpr | | Phello.~dpr |

在例 1.1 中，保存单元文件为“Fhello.pas”，用于存储事件处理程序。同时系统生成同名的窗体描述文件 Fhello.dfm，保存用户对窗体和组件设置的属性。再执行“File | Save Project As”菜单，保存项目文件为“Phello.dpr”。

用户在保存文件时，只需为单元文件和项目文件命名，且文件名自动成为单元名，其他文件由 Delphi 自动生成。程序编译、运行时，系统将单元文件编译生成 Fhello.dcu 文件，将