



通用程序设计案例丛书

Delphi 程序开发 案例解析

杨富国 主编

陈立俊 唐巧琪 曹宇 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

通用程序设计案例丛书

Delphi

程序开发案例解析

杨富国 主编

陈立俊 唐巧琪 曹宇 编著

清华大学出版社
北京交通大学出版社

·北京·

内 容 简 介

本书通过系统而完整的程序设计案例向读者介绍如何使用 Delphi 开发各种较为复杂的应用系统。着重分析各种应用系统中典型案例设计的思路、方法、技巧和经验, 试图为那些具有一定编程基础而又缺少系统开发经验的读者开辟一条捷径, 使其迅速上手、触类旁通。

本书内容翔实、案例丰富、实践性强, 可作为广大程序员和编程爱好者学习 Delphi 程序开发和应用系统设计的教材, 也可作为相关专业大学生及程序开发人员的参考书。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13501310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再表的方法识别真伪。

图书在版编目 (CIP) 数据

Delphi 程序开发案例解析/杨富国主编; 陈立俊, 唐巧琪, 曹宇编著. —北京: 清华大学出版社; 北京交通大学出版社, 2006. 6

(通用程序设计案例丛书)

ISBN 7-81082-739-1

I. D… II. ①杨… ②陈… ③唐… ④曹… III. 软件工具-程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2006) 第 039507 号

责任编辑: 刘 洵

出版者: 清华大学出版社 邮编: 100084 电话: 010-62776969

北京交通大学出版社 邮编: 100044 电话: 010-51686414

印刷者: 北京瑞达方舟印务有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 24.25 字数: 605 千字

版 次: 2006 年 6 月第 1 版 2006 年 6 月第 1 次印刷

书 号: ISBN 7-81082-739-1/TP·272

印 数: 1~5 000 册 定价: 35.00 元

本书如有质量问题, 请向北京交通大学出版社质监局反映。对您的意见和批评, 我们表示欢迎和感谢。
投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@center.bjtu.edu.cn.

编委会成员名单

主 编：杨富国

编 委：（按姓氏笔画排序）

王 浩 王 倩 王 健

吕江英 邢 霞 陈立俊

杨茂龙 唐巧琪 曹 宇

前言

本书不是泛泛地介绍程序设计语言的枯燥理论，也不是简单地罗列各式各样的零散程序片断，而是从开发平台的特点出发，依照软件工程设计规范，对每个案例的需求分析、系统建模、界面设计、模块开发说明、系统实现与编译等进行介绍，涉及软件开发的整个过程。

书中所有案例的难度和规模与计算机专业本科生毕业设计的要本相当，案例范围涉及信息管理系统、网络应用系统、图形图像系统、多媒体系统、远程控制系统等常用领域。本书提供了案例的完整代码和可执行程序，读者可通过北京交通大学出版社网站下载。具体方法为，访问北京交通大学出版社网址：<http://press.bjtu.edu.cn>，在“FTP 下载专区”单击“下载图书辅助资料”，在其中寻找本书案例。

通过阅读本书，读者可以进一步掌握软件工程的规范，增加系统开发经验。能够着眼于完整的、较复杂的应用程序，运用面向对象原则和软件工程的规范，由全局到细节地进行设计和开发，而不是一开始就陷入编程的旁枝末节困扰，实现从初级程序员向具备一发系统分析和系统设计能力的高级程序员的转变。

对于系统设计和开发人员来说本书是很好的参考书，系统设计和开发人员可以从本书了解到 Delphi 开发平台的特性，了解如何应用 Delphi 开发平台实现各种应用系统的方法和思路。本书提供的代码可以作为借鉴和引用。

本书按照 Delphi 7 编程基础知识和案例设计划分，共包括 7 章内容和 3 个附录。读者可以根据自己的需要，选择阅读相关的章节，如果对前面的内容很熟悉，可以跳过这些内容继续阅读。

本书第 1 章主要概述 Delphi 7 编程基础知识，涵盖了软件编程中常用的知识要点；第 2 章介绍软件工程基本理论，为从工程化角度进行软件开发打下基础；第 3 章介绍数据库信息管理系统的开发所需要的相关知识，结合物流系统案例的开发实践，掌握数据库管理系统的开发流程；第 4 章介绍网络数据通信的概念和主要的协议接口及 IntraWeb 技术，主要内容包括俄罗斯方块网络游戏和基于 IntraWeb 技术开发 Web 数据服务程序两个案例；第 5 章介绍基于 Windows 平台的 GDI 和 DirectX 图形图像处理技术，并且通过绘图程序和战斗机游戏两个软件开发案例具体讲解两种技术的应用方法；第 6 章介绍基于 Windows

平台的多媒体指令接口 MCI 及多媒体应用程序的开发方法，并提供了媒体播放器程序设计案例，讲解如何利用第三方控件开发多媒体软件；第 7 章介绍计算机远程控制理论，并通过客户机-服务器模式远程控制系统开发案例进行具体讲解实践；附录 A 介绍了本书案例中使用的控件的获取和安装方法；附录 B 介绍了 Delphi 7 常用的 API 函数接口的使用；附录 C 介绍了常用软件功能的开发方法和技巧。

本书适合以下读者对象：软件开发的初学人员；从事应用软件开发的人员、分析员和项目管理人员；对应用软件开发感兴趣的技术人员。

编者
2006 年 6 月



| | | |
|--------------|--------------------------|----|
| 第 1 章 | Delphi 编程基础 | 1 |
| 1.1 | Delphi 编程特点 | 1 |
| 1.1.1 | Delphi 的发展 | 1 |
| 1.1.2 | 事件驱动程序 | 2 |
| 1.1.3 | 窗体控件的作用 | 3 |
| 1.1.4 | 面向应用的开发 | 4 |
| 1.2 | Delphi 集成开发环境 | 4 |
| 1.2.1 | Delphi 集成开发环境组成 | 4 |
| 1.2.2 | 集成调试器 | 8 |
| 1.2.3 | Object Pascal 语言基础 | 8 |
| 1.3 | 常用构件介绍..... | 22 |
| 1.3.1 | 构件 | 23 |
| 1.3.2 | 构件分类..... | 23 |
| 1.3.3 | 构件的常用属性、事件和方法 | 23 |
| 1.3.4 | 构件简介..... | 25 |
| 1.4 | 编程举例..... | 36 |
| 1.4.1 | “Hello World” 程序 | 36 |
| 1.4.2 | 程序编译、打包与安装发布 | 37 |
| 1.5 | 小结..... | 42 |
| 第 2 章 | 软件开发过程 | 43 |
| 2.1 | 概述..... | 43 |
| 2.1.1 | 瀑布模型..... | 44 |
| 2.1.2 | 原型模型..... | 44 |
| 2.1.3 | 螺旋模型..... | 45 |
| 2.2 | 软件定义及计划..... | 46 |
| 2.2.1 | 问题定义..... | 46 |
| 2.2.2 | 可行性分析 | 46 |
| 2.2.3 | 制订开发计划 | 46 |
| 2.3 | 需求分析..... | 47 |
| 2.3.1 | 需求分析任务 | 47 |
| 2.3.2 | 需求分析过程 | 47 |
| 2.3.3 | 常用需求分析方法 | 48 |

| | | |
|------------|------------------|-----|
| 2.3.4 | 需求文档 | 49 |
| 2.4 | 软件设计 | 49 |
| 2.4.1 | 设计目标 | 49 |
| 2.4.2 | 设计任务 | 50 |
| 2.4.3 | 软件设计过程 | 50 |
| 2.4.4 | 软件设计方法 | 52 |
| 2.5 | 编码 | 52 |
| 2.5.1 | 概述 | 52 |
| 2.5.2 | 程序设计风格 | 53 |
| 2.5.3 | 编码标准 | 53 |
| 2.6 | 软件测试 | 54 |
| 2.6.1 | 概述 | 54 |
| 2.6.2 | 测试目的和原则 | 54 |
| 2.6.3 | 测试方法 | 55 |
| 2.6.4 | 软件测试过程 | 56 |
| 2.7 | 软件维护 | 57 |
| 2.7.1 | 概述 | 57 |
| 2.7.2 | 影响维护工作量的因素 | 57 |
| 2.7.3 | 软件的可维护性 | 58 |
| 2.8 | 小结 | 59 |
| 第3章 | 数据库信息管理案例 | 60 |
| 3.1 | 管理信息系统基础 | 60 |
| 3.1.1 | 管理信息系统的概述 | 60 |
| 3.1.2 | 管理信息系统的特点 | 62 |
| 3.1.3 | 管理信息系统的开发 | 62 |
| 3.2 | SQL与ADO概述 | 63 |
| 3.2.1 | SQL语言 | 63 |
| 3.2.2 | ADO数据对象 | 69 |
| 3.3 | 鹰眼物流跟踪管理系统 | 73 |
| 3.3.1 | 功能分析 | 74 |
| 3.3.2 | 系统设计 | 74 |
| 3.3.3 | 数据库设计 | 75 |
| 3.3.4 | 界面设计 | 77 |
| 3.3.5 | 系统实现 | 90 |
| 3.3.6 | 系统运行与测试 | 107 |
| 3.4 | 小结 | 108 |
| 第4章 | 网络数据通信案例 | 109 |
| 4.1 | 网络信息系统编程基础 | 109 |

| | | |
|--------------|----------------------------------|------------|
| 4.1.1 | 网络信息系统概述 | 109 |
| 4.1.2 | 网络信息系统的特点 | 109 |
| 4.2 | Socket 编程概述 | 110 |
| 4.2.1 | WinSock 基本概念 | 110 |
| 4.2.2 | 客户机-服务器模型 | 124 |
| 4.3 | 案例 1: 俄罗斯方块游戏软件 | 127 |
| 4.3.1 | 方案分析 | 127 |
| 4.3.2 | 系统设计 | 127 |
| 4.3.3 | 数据库设计 | 128 |
| 4.3.4 | 界面设计 | 129 |
| 4.3.5 | 系统实现 | 136 |
| 4.3.6 | 程序运行结果 | 177 |
| 4.4 | 案例 2: 利用 IntraWeb 组件开发人力资源网络应用软件 | 179 |
| 4.4.1 | Delphi IntraWeb 开发基础 | 179 |
| 4.4.2 | 人力资源网络应用软件设计 | 185 |
| 4.4.3 | 人力资源数据库设计 | 185 |
| 4.4.4 | 人力资源网络应用软件界面设计 | 188 |
| 4.4.5 | 应用软件系统实现 | 202 |
| 4.4.6 | 程序安装运行与配置 | 236 |
| 4.5 | 小结 | 238 |
| 第 5 章 | 图形图像处理案例 | 239 |
| 5.1 | 图形图像处理系统基础 | 239 |
| 5.1.1 | 图形图像处理系统概述 | 239 |
| 5.1.2 | 图形图像处理系统的特点 | 240 |
| 5.1.3 | 图形图像处理系统的开发 | 240 |
| 5.2 | GDI 与 DirectX 技术 | 240 |
| 5.2.1 | GDI 图形图像技术 | 240 |
| 5.2.2 | DirectX 技术 | 243 |
| 5.3 | 矢量图绘制程序 | 246 |
| 5.3.1 | 案例分析 | 246 |
| 5.3.2 | 软件系统设计 | 247 |
| 5.3.3 | 软件界面设计 | 247 |
| 5.3.4 | 代码分析与实现 | 250 |
| 5.3.5 | 程序运行与测试 | 261 |
| 5.4 | 飞机战斗游戏的开发 | 262 |
| 5.4.1 | 案例分析 | 262 |
| 5.4.2 | 软件系统设计 | 262 |
| 5.4.3 | 软件界面设计 | 262 |

| | | |
|--------------|------------------------------------|------------|
| 5.4.4 | 代码分析与实现 | 264 |
| 5.4.5 | 程序运行与测试 | 290 |
| 5.5 | 小结 | 291 |
| 第 6 章 | 多媒体应用案例 | 292 |
| 6.1 | 计算机多媒体概述 | 292 |
| 6.1.1 | 计算机多媒体的类型 | 292 |
| 6.1.2 | 多媒体应用系统的特点 | 293 |
| 6.1.3 | Windows 的多媒体体系结构 | 294 |
| 6.2 | MCI 编程简介 | 296 |
| 6.2.1 | MCI 设备与时间格式 | 296 |
| 6.2.2 | MCI 函数与命令 | 297 |
| 6.2.3 | MCI 函数与命令的使用 | 298 |
| 6.3 | 多媒体播放器 | 301 |
| 6.3.1 | 案例分析 | 301 |
| 6.3.2 | 软件系统设计 | 301 |
| 6.3.3 | 软件界面设计 | 301 |
| 6.3.4 | 代码分析与设计 | 303 |
| 6.3.5 | 程序运行结果 | 309 |
| 6.4 | 小结 | 310 |
| 第 7 章 | 远程控制应用案例 | 311 |
| 7.1 | 远程控制基础 | 311 |
| 7.1.1 | 远程控制的原理 | 311 |
| 7.1.2 | 远程控制的实现 | 312 |
| 7.1.3 | 远程控制的应用 | 312 |
| 7.1.4 | 远程控制的安全性 | 312 |
| 7.2 | 远程控制程序 | 313 |
| 7.2.1 | 案例分析 | 313 |
| 7.2.2 | 程序系统设计 | 313 |
| 7.2.3 | 程序界面设计 | 313 |
| 7.2.4 | 代码分析与实现 | 316 |
| 7.2.5 | 远程控制运行与测试 | 325 |
| 7.3 | 小结 | 326 |
| 附录 A | 常用控件的安装与使用 | 327 |
| A.1 | RealAudio ActiveX 控件的安装 | 327 |
| A.2 | ShockWaveFlash ActiveX 控件的安装 | 329 |
| A.3 | VCL Skin 控件的安装 | 329 |
| A.4 | BetterAdoDataSet VCL 控件的安装 | 330 |
| A.5 | DirectX for Delphi 开发包的安装 | 331 |

| | | |
|-------------|---------------------------------------|-----|
| 附录 B | ◀ 常用 API 函数一览 | 333 |
| | B.1 数据类型转换函数 | 333 |
| | B.2 字符串、数组操作函数 | 334 |
| | B.3 文件、磁盘操作函数 | 339 |
| | B.4 内存、指针操作函数 | 345 |
| | B.5 数学运算函数 | 346 |
| | B.6 日期函数 | 348 |
| 附录 C | ◀ Delphi 基础开发技巧 | 351 |
| | C.1 网络邻居复制文件 | 351 |
| | C.2 产生鼠标拖动效果 | 351 |
| | C.3 取得 Windows 目录 | 352 |
| | C.4 在 FORM 或其他容器上画线 | 352 |
| | C.5 字符串列表使用 | 352 |
| | C.6 简单的剪贴板操作 | 353 |
| | C.7 关于文件、目录操作 | 353 |
| | C.8 处理文件属性 | 353 |
| | C.9 执行程序外文件 | 353 |
| | C.10 取得系统运行的进程名 | 354 |
| | C.11 关于汇编的嵌入 | 354 |
| | C.12 关于类型转换函数 | 354 |
| | C.13 字符串的过程和函数 | 355 |
| | C.14 关于处理注册表 | 356 |
| | C.15 关于键盘常量名 | 356 |
| | C.16 初步判断程序母语 | 357 |
| | C.17 操作 Cookie | 357 |
| | C.18 增加到文档菜单的连接 | 357 |
| | C.19 判断鼠标按键 | 357 |
| | C.20 设置窗体的最大显示 | 358 |
| | C.21 按键接受消息 | 358 |
| | C.22 截获 WM_QUERYENDSESSION 关机消息 | 358 |
| | C.23 获取网上邻居 | 359 |
| | C.24 获取某一计算机上的共享目录 | 361 |
| | C.25 得到硬盘序列号 | 362 |
| | C.26 MEMO 的自动翻页 | 362 |
| | C.27 DBGrid 中按 Enter 键转到下个位置 (Tab 键) | 363 |
| | C.28 如何安装控件 | 363 |
| | C.29 目录完全删除 (deltree) | 364 |
| | C.30 符 TMemo 控件当前光标的行和列信息获取到 Tpoint 中 | 365 |

| | | |
|-------|------------------------------------|-----|
| C. 31 | 读硬盘序列号 | 365 |
| C. 32 | VCL 制作简要步骤 | 365 |
| C. 33 | 动态链接库的装载 | 366 |
| C. 34 | 指针变量和地址 | 366 |
| C. 35 | 判断字符是汉字的一个字符 | 366 |
| C. 36 | Memo 的定位操作 | 367 |
| C. 37 | 获得双字节字符内码 | 367 |
| C. 38 | 调用 ADD 数据存储过程 | 367 |
| C. 39 | 将文件存到 blob 字段中 | 367 |
| C. 40 | 把文件全部复制到剪贴板 | 368 |
| C. 41 | 列举当前系统运行进程 | 369 |
| C. 42 | 根据 BDE Table1 建立新表 Table2 | 369 |
| C. 43 | 理解 DLL 建立和引用 | 370 |
| C. 44 | 动态读取程序自身大小 | 370 |
| C. 45 | 读取 BIOS 信息 | 370 |
| C. 46 | 动态建立 MS SQL 别名 | 371 |
| C. 47 | 播放背景音乐 | 372 |
| C. 48 | 接口和类的一个范例代码 | 372 |
| C. 49 | 在任务条中隐藏程序 | 373 |
| C. 50 | 按 Alt+Ctrl+Del 组合键隐藏程序 | 373 |
| C. 51 | 检测光驱符号 | 373 |
| C. 52 | 检测声卡 | 374 |
| C. 53 | 在字符串网格中画图 | 374 |
| C. 54 | SQL 显示多数据库/表 | 374 |
| C. 55 | RFC (Request For Comment) 相关 | 374 |
| C. 56 | TNMUUProcessor 的使用 | 375 |
| C. 57 | TFileStream 的操作 | 375 |

| | |
|---------------------|-----|
| 参考文献 ◀ | 376 |
|---------------------|-----|

第1章

Delphi 编程基础

1.1 Delphi 编程特点

Delphi 是 Borland 公司推出的基于 Windows 环境的快速程序开发工具。Delphi 的命名来源于古希腊城市，这个城市因为拥有阿波罗神殿而著名。本章将简要介绍 Delphi 的发展历史、语言特点、编程环境等相关知识。

1.1.1 Delphi 的发展

Delphi 是第四代编程语言，是 RAD (Rapid Application Development, 快速应用程序开发) 工具的代表。从核心上说，Delphi 是一个 Pascal 编译器。Delphi 语言的发展历程如下。

Delphi 1——1995 年，Borland 推出了 Delphi 1。Delphi 1 提供了一种完全不同的开发 Windows 程序的方法：可视化的开发环境、编译后的可执行软件、DDL、数据库，以及可以毫无限制地给可视环境命名。而且 Delphi 1 是第一个综合了可视化开发环境、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，它奠定了 RAD 的基础。

Delphi 2——1996 年，Delphi 2 在 32 位的操作系统 Windows 95 和 Windows NT 上实现了 Delphi 1 的一切功能。另外，Delphi 2 还增加了许多 Delphi 1 没有的功能，例如 32 位的编译器能生成速度更快的应用程序，对象库得到进一步的丰富和扩展，完善了数据库支持，改进了字符串处理，支持 OLE 对象链接与嵌入和可视化窗体继承，以及与 16 位的 Delphi 兼容等。Delphi 2 成为衡量其他 RAD 工具的标准。为了满足 IT 产业的需要，Delphi 2 增强了数据库功能和客户机-服务器的网络通信功能。

Delphi 3——1997 年，Borland 公司推出了 Delphi 3。Delphi 3 使本来极其复杂的 COM、ActiveX、WWW 应用程序开发、“瘦”客户应用程序、多层数据库系统体系结构等技术变得非常容易。虽然 Delphi 3 与 Delphi 1 编写应用程序的基本方法大都相同，但 Delphi 3 的 Code Insight (代码内视) 技术却简化了代码编写过程。

Delphi 4——1998 年推出的 Delphi 4 致力于使 Delphi 更易于使用。Module Explore (模块导航) 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 经过重新设计可支持浮动或可停靠的工具栏和窗口，调试器也做了改进。Delphi 4 不

愧为一个先进的开发工具，它的 MIDAS、DCOM 和 CORBA 等技术使 Delphi 4 的应用范围扩展到企业级用户。

Delphi 5——1999 年，Delphi 5 推出。Delphi 5 通过增加更多的功能使程序的编写更简单。新功能包括进一步增强了 IDE 和调试器的功能，提供了 Team Source 小组开发软件和转换工具等。Delphi 5 也为简化 Internet 的开发增加了许多新功能，包括用于创建 ASP 的 Active Server Object Wizard、用于支持 XML 和 MIDAS 功能的 Internet Express 组件，使 Delphi 成为 Internet 的一个通用数据平台。

Delphi 6——2001 年，Delphi 6 推出。Delphi 6 不但在传统的开发能力方面持续改进，让程序员能够享有更具生产力的开发环境，而且对于最新的信息技术，例如 XML/XSL、SOAP、Web Service 等也都有非常良好的支持，让程序员能够快速地使用这些新技术。此外，Delphi 6 在核心运行时函数库及编译器方面也有了不小改善，不但让程序员可以开发功能更强的窗口应用程序，而且可以更加平滑、顺利地移植到 Linux 平台，通过跨平台的新可视化组件 Framework CLX，即使是 Windows 程序员也可以快速地开发 Linux 平台上的应用系统。

由于 Delphi 6 的很多新特性还不是很成熟，Borland 公司很快推出了 Delphi 7，使这些新特性得到了很大的改善和提高。

1.1.2 事件驱动程序

基于 Windows 的应用程序一般由界面和代码两部分组成。程序界面为用户提供一个人机交互的接口，用户的意图通过对应用程序界面上的各种控件的操作来表达，然后控件对应的处理代码将用户的意图转化、处理和分发到相应的处理单元进行运算，最后运算的结果通常又要通过应用程序界面上的控件转化为用户可接受的表达形式。

事件驱动也称消息驱动，是 Windows 应用程序工作的重要方式。消息是 Windows 的重要概念。通过消息，Windows 处理用户输入，协调系统内的应用程序。Delphi 使用 Windows 消息，同时也使用自己定义的内部消息。Delphi 程序通过内在的消息机制处理来自 Windows 和自身的消息。

在 Windows 中消息被定义为如下形式的变体记录：

```
TMessage= record
    Msg: Cardinal;
    case Integer of
        0: (WParam: Word; LParam: Longint; Result: Longint);
        1: (WParamLo: Byte; WParamHi: Byte; LParamLo: Byte; LParamHi: Byte;
            ResultLo: Byte; ResultHi: Byte;);
    end;
```

最初，程序员必须了解每个参数的意义，后来微软公司对消息的参数作了命名，使每条消息的参数更容易理解，如键盘按下的消息：

```
type
    TWMKeyDown= record
        Msg: Cardinal;
```

```
CharCode:Word;  
Unused:Word;  
KeyData:Longint;  
Result:Longint;  
end;
```

虽然对每个参数都作了具体命名，但它的结构仍然是记录，占用 16 个字节。Delphi 的单元 Messages 定义了所有的 Windows 消息和 Delphi 消息。

当建立一个窗体、加入一个控件时，就自动注册了一个消息接收例程 MainWndProc，这个例程是每个构件和窗体所固有的。Delphi 内部使用这个例程接收来自于各方面的消息。MainWndProc 只接收消息，并不处理消息。在此例程内调用例程 WndProc（虚拟方法）来处理消息，并在边界提供了异常保护。如果有异常发生，将调用 Application.HandleException 方法处理异常。

WndProc 是一个虚拟方法，可以被自定义的消息处理例程覆盖。在此例程中可以将自己关心的消息截获、处理或丢弃。对于自己不处理的消息最好调用继承（Inherited）例程，让父类去处理。父类的 WndProc 如果被覆盖，消息同样要被筛选一次。不管父类 WndProc 是否被覆盖，在继承中都要调用祖先类的 WndProc，这样一直追溯到 Tcontrol 类的 WndProc。在 Tcontrol.WndProc 中调用最终祖先类 TObject 的 Dispatch 派发消息。假如 MainWndProc 收到一条消息，沿着继承的路线调用 WndProc，如果一路上的 WndProc 都未处理该消息，那么就会调用 TObject.Dispatch 去找一个例程来处理。

TObject.Dispatch 派发消息时也是按照继承的链逆流面上的。首先查找本类中用 Message 定义的消息处理方法，如果没有再查找本类的 DefaultHandler（虚拟方法）例程是否被覆盖，如果覆盖则调用它进行处理，否则继续查找父类。父类从 Message 定义的方法查起，接着再看看 DefaultHandler 是否覆盖，一直到 TObject.DefaultHandler。TObject.DefaultHandler 中，什么消息都不处理，只不过是走到消息处理的尽头。

Windows 的消息目录如下：

- 窗口管理消息——当一个窗口状态改变时，由 Windows 发送给应用程序的消息；
- 初始化消息——当应用程序建立一个菜单或者对话框时发送的消息；
- 输入消息——向 Windows 发送的鼠标、键盘、滚动条、时钟等消息；
- 系统消息——用户操作 Windows 系统的控制菜单、滚动条及边界时由 Windows 发给应用程序的消息；
- 剪贴板消息；
- 系统信息修改消息；
- 构件操作消息——Windows 发给子构件的消息，如按钮、列表、编辑框等；
- 构件通知消息——通知构件的父窗口发生在该控件内的动作的消息；
- 多文档 MDI 消息；
- 非客户区消息；
- 滚动条通知消息。

1.1.3 窗体控件的作用

Delphi 的窗体是应用程序的主要部分，用户通过在窗体中加入构件来建立自己的应

用程序。窗体是 TForm 类的构件，一般使用 Application.CreateForm 创建。在工程文件中，Application.Run 语句之前的窗体程序自动建立，建立之后，并不都显示在屏幕上，只显示主窗体，即第一个建立的窗体。如果要显示那些已经建立但没有激活的窗体，使用方法 Show 或 ShowModal，关闭窗口使用方法 Close 或者 CloseQuery。ShowModal 显示的窗体带有聚焦点，如果不处理完这个窗体，就不能操作程序中的其他窗体，该窗体的返回结果保存在属性 ModalResult 中。

比较庞大的 Windows 应用程序应该是包含多个窗体的。MDI 方法采用一个主窗体和若干子窗体的方式进行管理，并且所有的子窗体都保持在主窗体的范围之内。SDI 方法则使用各个窗体独立的形式来管理整个应用程序中的窗体组。

1.1.4 面向应用的开发

Delphi 向程序员提供了大量的控件，其中包括对 Windows 常用组件进行封装所得到的控件（例如打开文件对话框、文本编辑框等）和针对特定应用方向开发的控件（例如 Internet 服务控件、网络服务控件、Office 服务控件等）。当然，用户也可以根据特定的需要方便地开发可视化组件库（VCL，Visual Components Library）控件。程序员往往可以通过对这些控件进行简单的设置和处理设计出满足应用的软件系统。

1.2 Delphi 集成开发环境

和以往 Delphi 的每一个新版本一样，Delphi 7 继续改善了开发环境，帮助程序员提高开发效率。Delphi 7 在集成开发环境中加入了很多让程序员爱不释手的新功能。

1.2.1 Delphi 集成开发环境组成

Delphi 的集成开发环境主要由以下几部分组成。

1. 主窗口 (Main Window)

Delphi 的主窗口位于屏幕的上端，界面如图 1-1 所示，包括菜单 (Menu)、加速条 (Speed Bar) 和组件面板 (Component Panel)。Menu 是下拉式主菜单；Speed Bar 位于主窗口的左下端，由两排加速按钮组成，这些按钮是菜单命令的快捷方式，各种图标直观地表示了它能执行的动作；Component Panel 由一行、若干页对象按钮组成，利用它来选择需要的组件并将它放到窗体中去。

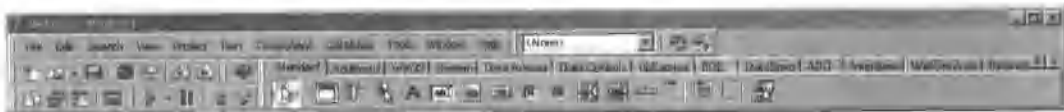


图 1-1 主窗口

2. 对象查看器窗口 (Object Inspector Window)

对象查看器 (Object Inspector) 窗口含有两个选项卡：Properties 选项卡显示窗体中

当前被选择组件的属性信息，并允许改变对象的属性，如图 1-2 (a) 所示；Events 选项卡列出了当前组件可以响应的事件，在某一事件后边的空白处，可以定义对象在相应事件发生时执行的动作。

首次启动时，Object Inspector 窗口显示的是当前窗体 Form1 的属性。Object Inspector 根据对象属性的多少，决定是否有滚动条显示。拖动滚动条可以查看当前对象的全部属性。

此外，Object Inspector 上还有 Object Selector (对象选择器)，它位于 Object Inspector 上方的下拉式菜单中。Object Selector 显示了窗体上所有组件的名称和类型，也包含窗体本身。用 Object Selector 可以很容易地在窗体的各个组件之间切换，也可以快速地回到窗体本身。当窗体中含有较多的对象时，这是切换对象，尤其是回到窗体的最快捷途径。

要设置 Object Inspector 为一直可见，可将鼠标移到 Object Inspector 上，右击鼠标启动 Object Inspector 的弹出式菜单，将其设置为 Stay On Top。这对初学者来说是一个很重要的设置方式。

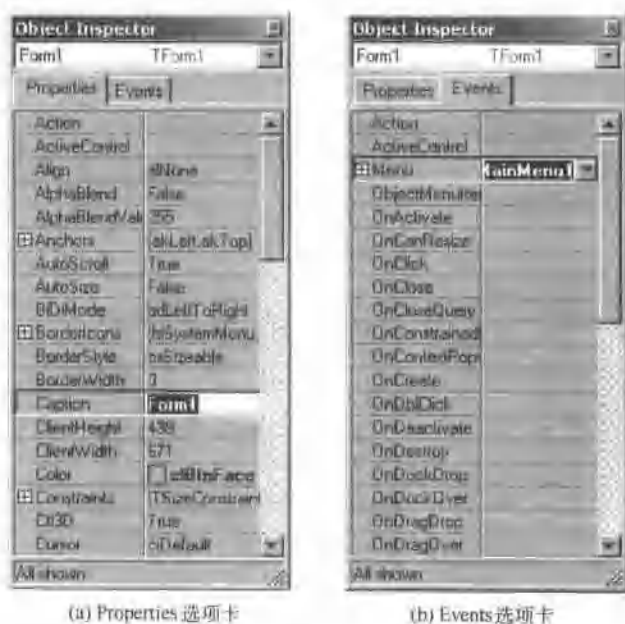


图 1-2 对象查看器

3. 窗体设计器窗口 (Form Designer Window)

窗体设计窗口也称窗体窗口，是开展大部分设计工作的区域。首次启动 Delphi 7 时显示的是系统默认窗体 Form1。可以把组件放在窗体中，通过移动位置、改变尺寸等操作随心所欲地安排它们，以此来开发应用程序的用户界面。可以把窗体想像成一个可以放置其他组件的容器。窗体上有栅格 (Grids)，供放置组件时对齐位置用，在程序运行时 Grids 是不可见的。如图 1-3 所示，窗体设计器上放置了媒体播放器、主菜单、时钟、打开文件对话框等控件。