

Delphi 7 全方位教程

计算机教育图书研究室 总策划
Computer Education Books



尚志宏 石蔚云 主编



Delphi 7

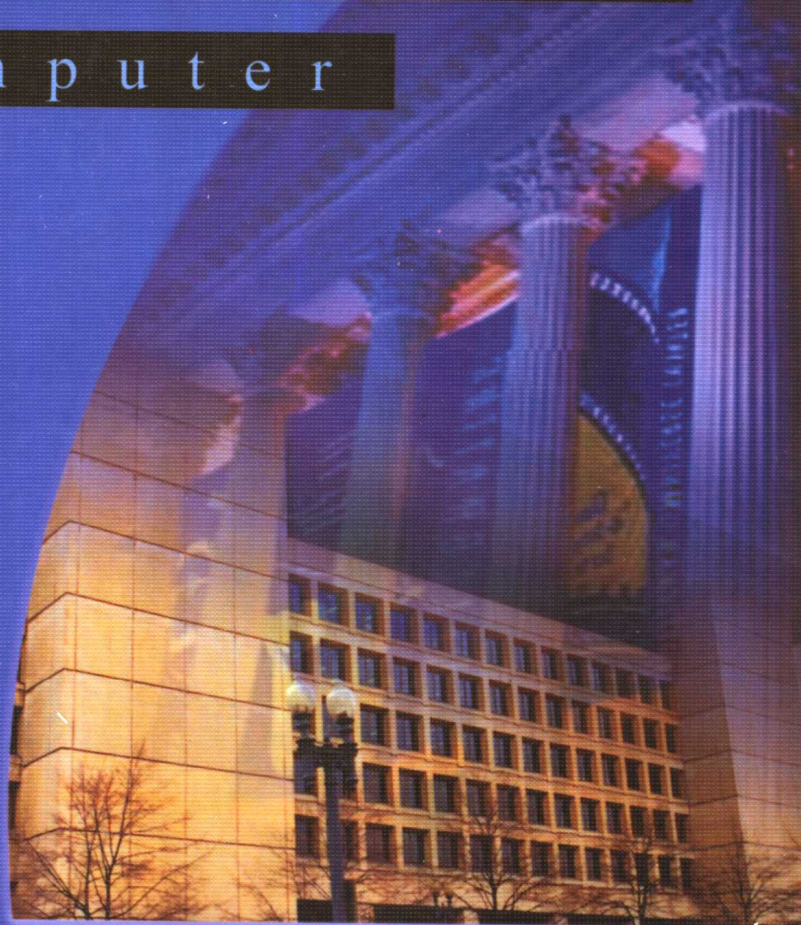
全方位教程

Completely Windows tutorial
Computer

■ 本书内容

- Delphi 入门
- Object Pascal 语言
- Windows 界面程序设计
- 编写多线程应用程序
- Windows 系统编程
- 图形图像编程
- 数据库程序开发

航空工业出版社



Delphi7全方位教程



计算机教育图书研究室
Computer Education Books

总策划

主 编 尚志宏 石蔚云

副主编 张 霞 睢 丹

编 委 崔慧勇 林 峰

任立功 魏 霞



航空工业出版社

内 容 提 要

本书融合了利用 Delphi 7.0 开发 Windows 应用程序的理论和实践,全面介绍了利用 Delphi 开发应用程序的方法和技巧,并附带大量的程序开发实例,有很强的实用性。

本书共包括 13 章,系统介绍了 Delphi 7.0 的基础知识、Object Pascal 语言、基本的控件和编程方法、用户界面设计、图形图像编程方法、多媒体编程、多线程应用、自定义控件、数据库的基本知识、开发数据库应用程序、网络编程、Web Service 编程和系统集成。

本书内容丰富、详实,涵盖了 Delphi 编程的各个方面,适合于 Delphi 的初中级开发人员使用,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 全方位教程 / 尚志宏主编. — 北京:
航空工业出版社, 2003.10
ISBN 7-80183-217-5

I .D… II.尚… 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 (2003) 第 083243 号

航空工业出版社出版发行

(北京市安定门外小关东里 14 号 100029)

北京市燕山印刷厂印刷

全国各地新华书店经售

2003 年 10 月第 1 版

2003 年 10 月第 1 次印刷

开本: 787×1092 1/16

印张: 25.00

字数: 406 千字

印数: 1-6000

定价: 29.80 元

本社图书如有缺页、倒页、脱页、残页等情况,请与本社发行部联系调换。联系电话: 010-65934239 或 84917422

前 言

Delphi 7 是 Inprise 公司（原 Borland 公司）最新推出的企业级应用程序开发工具。它具有面向对象编程、支持团队开发、提供工程管理、对数据库的良好支持等特性，它所提供的大量可重用组件和用户自建模板技术，极大地提高了应用系统的开发速度。业界如此形容 Delphi：真正的程序员用 C，聪明的程序员用 Delphi。自从它面世以来，便深受广大程序员的喜欢。

Delphi 7 的语言基础为 Pascal，它继承了 Pascal 语言严谨的优点，如代码结构清晰、可读性好和代码执行效率高等。Delphi 从 1.0 版本发展到 7.0 版本，性能在很多方面有了很大的提高，包括数据库的体系结构、ActiveX 控件的开发、Web 应用程序和安全线程等，从而使得 Delphi 的功能更加强大，使用起来也更加方便和灵活，提高了应用程序的开发效率。

总之，不论是初学 Windows 程序设计的学生或是开发关键性任务、大型应用系统的企业开发人员，Delphi 完整的产品线都能满足这些人员的需求，为他们提供了极为方便的技术和服务。

Delphi 7 轻松跨越 Windows/Linux 双平台，能够高效地开发面向对象的企业级应用程序，是当前 Windows 平台上第一个全面支持最新 Web 服务的快速开发工具。无论是企业级用户，还是个人开发者，都能够利用 Delphi 7 轻松、快捷地构建新一代电子商务应用软件。

本书共包括 13 章：第 1 章介绍了 Delphi 7 的基础知识；第 2 章讲述了 Object Pascal 语言；第 3 章讲述了 Windows 程序设计方法；第 4 章讲述了多线程程序设计；第 5 章讲述了文件目录的操作方法；第 6 章介绍了系统编程；第 7 章讲述了 Delphi 自定义控件；第 8 章讲述了图形图像编程；第 9 章讲述了多媒体编程；第 10 章讲述了网络编程；第 11 章和第 12 章讲述了数据库编程技术；第 13 章讲述了 Web 编程。

本书由尚志宏、石蔚云主编，参加本书编写与制作的老师还有：张霞、睢丹、崔慧勇、任立功、林锋、魏霞等。本书主要面向初中级的 Delphi 用户，也可作为高级用户的参考手册。由于作者水平有限，书中难免存在不足和疏漏之处，恳请读者批评指正。

<http://www.china-ebooks.com>

编 者
2003 年 8 月

目 录

第 1 章 Delphi 7 入门 1	
1.1 Delphi 概述..... 1	
1.1.1 Delphi 简介..... 1	
1.1.2 Delphi 的特点..... 1	
1.1.3 Delphi 7 的新特点..... 2	
1.2 Delphi 集成开发环境..... 3	
1.2.1 用户界面..... 3	
1.2.2 主窗口..... 5	
1.2.3 对象监视器..... 8	
1.2.4 窗体窗口..... 8	
1.2.5 代码编辑器..... 9	
1.3 开发第一个应用程序..... 10	
1.3.1 编程步骤简述..... 10	
1.3.2 启动 Delphi..... 11	
1.3.3 创建程序界面..... 12	
1.3.4 添加程序代码..... 13	
1.3.5 编译生成程序..... 15	
1.4 Delphi 各类文件简介..... 15	
1.4.1 工程文件 (.dpr)..... 15	
1.4.2 源代码文件 (.pas)..... 16	
1.4.3 窗体与数据模块 (.dfm)..... 16	
1.4.4 配置与选项文件 (.dof)..... 17	
1.4.5 中间编译单元 (.dcu)..... 17	
1.4.6 备份文件 (.~pas)..... 18	
1.4.7 包文件 (.dpk)..... 18	
1.4.8 应用程序文件 (.dll, .exe, .ocx)..... 18	
1.5 单元文件..... 18	
1.5.1 单元文件内部结构简介..... 19	
1.5.2 Unit 部分..... 19	
1.5.3 接口部分..... 19	
1.5.4 实现部分..... 20	
1.5.5 定义 Uses 子句..... 20	
1.5.6 Type 子句..... 20	
1.5.7 变量部分..... 21	
1.5.8 资源声明..... 21	
1.5.9 Initialization 部分..... 21	
1.5.10 Finalization 部分..... 21	
小 结..... 22	
第 2 章 Object Pascal 语言 23	
2.1 Object Pascal 语言基础..... 23	
2.1.1 标识符、常量与变量..... 23	
2.1.2 变量的赋值..... 25	
2.1.3 数据类型..... 26	
2.1.4 注释语句..... 35	
2.1.5 过程与函数..... 36	
2.2 流程控制语句..... 38	
2.2.1 判断语句..... 38	
2.2.2 循环语句..... 39	
2.2.3 Break 语句..... 41	
2.2.4 Continue() 语句..... 41	
2.2.5 With 语句..... 41	
2.3 Object Pascal 的面向 对象技术..... 42	
2.3.1 Delphi 与面向对象..... 42	
2.3.2 类与对象..... 42	
2.3.3 类的继承..... 46	
2.3.4 类与单元..... 46	
2.4 异常处理..... 48	
2.4.1 异常控制语句..... 49	
2.4.2 raise 语句..... 51	
2.4.3 异常类..... 51	
小 结..... 52	
第 3 章 Windows 界面程序设计 ... 53	
3.1 窗体 (Form)..... 53	
3.1.1 窗体的属性..... 53	
3.1.2 窗体的方法..... 56	



3.1.3 窗体的事件	57	3.9.3 TreeView 控件	100
3.2 菜单 (Menu)	58	3.9.4 ListView 控件	103
3.2.1 使用菜单设计器		3.9.5 ImageList 控件	104
建立菜单	59	3.10 特殊控件	105
3.2.2 定制各种类型的菜单	60	3.10.1 ScrollBar 控件	105
3.2.3 动态改变菜单	62	3.10.2 TrackBar 控件	107
3.3 工具栏 (Tool Bar)	67	3.10.3 UpDown 控件	108
3.4 信息对话框 (MessageBox)	68	3.10.4 HotKey 控件	108
3.4.1 MessageBox 函数	68	3.11 状态显示控件	111
3.4.2 MessageDlg 函数	69	3.11.1 ProgressBar 控件	111
3.4.3 MessageDlgPos 函数	71	3.11.2 StatusBar 控件	112
3.4.4 MessageDlgPosHelp 函数	71	3.12 表格控件	112
3.4.5 ShowMessage 函数	71	3.12.1 StringGrid 控件	112
3.4.6 InputBox 函数	72	3.12.2 DrawGrid 控件	114
3.5 单文档窗体 (SDI) 和多文档窗体 (MDI)	72	3.13 标签页控件	116
3.5.1 单文档窗体	72	3.13.1 TabControl 控件	116
3.5.2 多文档窗体	75	3.13.2 PageControl 控件	117
3.6 专用对话框	77	小 结	119
3.6.1 字体对话框	77	第 4 章 编写多线程应用程序	120
3.6.2 查找对话框	78	4.1 线程的概念	120
3.6.3 替换对话框	79	4.1.1 线程的基本概念	120
3.6.4 打开对话框	80	4.1.2 Delphi 7 对线程的支持	122
3.6.5 颜色对话框	82	4.2 TThread 对象	122
3.6.6 打印对话框	83	4.2.1 TThread 基础	122
3.7 文本控件	85	4.2.2 建立 TThread 对象	123
3.7.1 Edit 控件	85	4.2.3 TThread 对象的属性	124
3.7.2 Memo 控件	87	4.2.4 TThread 对象的方法	125
3.7.3 MaskEdit 控件	89	4.2.5 TThread 对象的事件	126
3.7.4 RichEdit 控件	91	4.3 线程控制	126
3.8 按钮控件	92	4.3.1 线程同步	126
3.8.1 Button 控件	93	4.3.2 设置线程优先级	127
3.8.2 BitBtn 控件	94	4.3.3 线程的挂起和唤醒	127
3.8.3 SpeedButton 控件	94	4.3.4 线程的执行时间	127
3.8.4 CheckBox 控件	95	4.4 多线程应用实例	128
3.8.5 RadioButton 控件	96	4.4.1 利用多线程实现	
3.9 列表控件	97	算法比较	128
3.9.1 ListBox 控件	97	4.4.2 利用多线程进行	
3.9.2 ComboBox 控件	98	图形处理	135

4.4.3 利用多线程操作数据库	138	小 结	192
小 结	143	第 7 章 Delphi 自定义控件	193
第 5 章 文件目录操作	144	7.1 控件基础	193
5.1 Delphi 的文件类型	144	7.1.1 控件概述	193
5.1.1 文本文件	144	7.1.2 控件的类型	194
5.1.2 记录文件	146	7.1.3 控件的结构	194
5.1.3 无类型文件	147	7.2 自定义控件开发	195
5.2 文件的基本操作	148	7.2.1 自定义控件简介	195
5.2.1 打开和关闭	148	7.2.2 祖先类的选择	196
5.2.2 目录操作	149	7.2.3 创建控件单元	197
5.2.3 文件名操作	152	7.2.4 控件的添加	198
5.2.4 获取文件属性	153	7.2.5 控件的删除	200
5.3 文件操作控件	155	7.2.6 更改控件图标	200
5.4 高级技巧	156	7.3 开发非可视控件	202
5.4.1 查看驱动器类型	156	7.4 开发 Active X 控件	205
5.4.2 启动控制面板中的内容	157	7.4.1 Active X 简介	206
5.4.3 打开和关闭光驱	160	7.4.2 使用 Active X	206
5.5 文件操作实例	160	7.4.3 创建 Active X 控件	209
5.5.1 程序功能	161	小 结	211
5.5.2 程序功能的实现	162	第 8 章 图形图像编程	212
小 结	165	8.1 图形对象概述	212
第 6 章 Windows 系统编程	166	8.1.1 画布对象 (TCanvas Object)	212
6.1 注册表操作	166	8.1.2 画刷对象 (TBrush Object)	218
6.1.1 注册表简介	166	8.1.3 画笔对象 (TPen Object)	219
6.1.2 TRegistry 类	167	8.1.4 颜色对象 (TColor Object)	221
6.1.3 用注册表创建系统快捷方式	168	8.2 图形图像控件	222
6.2 托盘编程	172	8.2.1 Image 控件	222
6.2.1 托盘程序基础	173	8.2.2 Shape 控件	222
6.2.2 制作托盘程序	173	8.2.3 PaintBox 控件	223
6.3 使用剪贴板	177	8.3 图形图像的显示	224
6.3.1 剪贴板基础	177	8.3.1 放大图片	224
6.3.2 使用剪贴板	178	8.3.2 将图片颠倒显示	226
6.4 对象的链接与嵌入	181	8.4 曲线绘制实例	227
6.4.1 OLE 简介	181	小 结	228
6.4.2 Delphi 中的 OLE 控件	182		
6.4.3 OLE 对象应用实例	186		



第9章 多媒体编程	229
9.1 MediaPlayer 控件	229
9.1.1 MediaPlayer 控件的属性	230
9.1.2 MediaPlayer 控件的方法	235
9.1.3 MediaPlayer 控件的事件	238
9.1.4 播放声音文件	239
9.1.5 播放 VCD 文件	240
9.1.6 播放 CD	241
9.2 高级多媒体技巧	249
9.2.1 显示声音控制属性窗口	249
9.2.2 禁止光驱自动播放	249
9.2.3 判断光驱中是否有 CD	250
9.2.4 全屏播放电影	252
小 结	253
第10章 网络编程	254
10.1 Delphi 的网络控件	254
10.1.1 TCPServer 控件和 TCPClient 控件	254
10.1.2 用控件实现网络聊天	256
10.1.3 WebBrowser 控件	258
10.1.4 Ping 操作的实现	262
10.2 调用系统函数进行 网络编程	264
10.2.1 信使服务程序	264
10.2.2 判断本机是否联网	267
10.2.3 取得 Modem 状态	268
10.2.4 启动 Outlook Express	269
10.2.5 检测局域网中的计算机	270
小 结	272
第11章 数据库编程基础	273
11.1 数据库系统概述	273
11.1.1 数据库管理系统 (DBMS)	274
11.1.2 数据库应用程序	275
11.1.3 数据库的建立	276
11.1.4 数据库的开发步骤	276
11.2 Delphi 与数据库开发	277
11.2.1 数据记录显示控件	277
11.2.2 数据选择控件	278
11.2.3 Delphi 7 访问数据库 的机制	278
11.3 Delphi 的数据库管理	280
11.3.1 DataBase DeskTop 控件	280
11.3.2 数据库引擎	286
11.3.3 数据字典	288
11.3.4 Data Pump 工具	289
11.4 简单数据库程序实例	291
小 结	292
第12章 数据库程序开发	293
12.1 数据库控件	293
12.1.1 TTable 控件	293
12.1.2 TQuery 控件	299
12.1.3 TDataSource 控件	302
12.1.4 TStoredProc 控件	304
12.1.5 TDatabase 控件	306
12.1.6 TSession 控件	309
12.2 数据感知控件	311
12.2.1 使用数据感知控件的 一般步骤	311
12.2.2 TDBGrid 控件	312
12.2.3 TDBCtrlGrid 控件	315
12.2.4 TDBNavigator 控件	318
12.2.5 TDBText 控件	319
12.2.6 TDBEdit 控件	320
12.2.7 TDBMemo 控件	320
12.2.8 TDBImage 控件	321
12.2.9 TDBListBox 控件	323
12.2.10 TDBComboBox 控件	326
12.2.11 TDBCheckBox 控件	326
12.2.12 TDBRadioGroup 控件	327
12.3 Delphi 报表设计	329
12.3.1 数据源设置	329
12.3.2 设计 Rave 报表	330
12.3.3 报表设计实例	332
12.4 数据库应用开发实例	333
12.4.1 设计思路	333
12.4.2 数据库设计	333

12.4.3 程序编写	334	13.1.8 Cookie 的设置	356
小 结	340	13.2 使用 WebBroker 控件	358
第 13 章 Web 编程	341	13.2.1 TPageProducer	358
13.1 Web 服务器编程	341	13.2.2 TDataSetPageProducer	360
13.1.1 WebBroker 概述	341	13.3 WebSnap 应用	362
13.1.2 TWebModule	343	13.3.1 简单的 WebSnap 程序	362
13.1.3 TCustomWebDispatcher	344	13.3.2 WebSnap 控件	364
13.1.4 TWebActionItem	345	13.4 ActiveForm 与网络编程	365
13.1.5 通过网络传送信息	352	13.5 Web 程序调试	369
13.1.6 获取客户输入	353	小 结	370
13.1.7 网络安全验证	355	附录 常用 API 函数	371

第 1 章 Delphi 7 入门

Delphi 是著名的 Borland（现在已和 Inprise 合并）公司开发的可视化软件开发工具，是基于 Windows 的面向对象的可视化软件开发系统，具有高效、优化和可扩展数据库技术的优点。Delphi 被称为第四代编程语言，它具有简单、高效、功能强大的特点。和 VC 相比，Delphi 更简单、更易于掌握，而在功能上却丝毫不逊色；和 VB 相比，Delphi 则功能更强大、更实用。可以说 Delphi 同时兼备了 VC 功能强大和 VB 简单易学的特点。它一直是程序员至爱的编程工具。

本章主要介绍 Inprise 公司的最新推出的 Delphi 7。

1.1 Delphi 概述

Delphi 7 是 Delphi 的最新版本，它的功能更强大，操作更方便，深受人们的喜爱和使用。

1.1.1 Delphi 简介

Delphi 这个名字源于古希腊的城市名。它集中了第三代语言的优点，以 Object Pascal 为基础，扩充了面向对象的能力，并且完美地结合了可视化的开发手段。Delphi 自 1995 年 3 月一推出就受到了人们的关注，并在当年一举夺得了多项大奖。

Delphi 实际上是 Object Pascal 语言的一种版本，但与传统的 Pascal 语言相比，有天壤之别。一个 Delphi 程序首先是应用程序框架，而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。程序员的工作只是在“骨架”中加入适当的代码。缺省的应用程序是一个空白的窗体（Form），可以直接运行，结果将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、最大化和最小化等，但却没有编写一程序。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 已经为程序员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的应用程序，只是不处理任何事情。程序员所需要做的，只是在程序中加入完成所需功能的代码而已。

在空白窗口的背后，应用程序的框架正在等待用户的输入。由于并未告诉它接收到用户输入后做何反应，窗口除了响应 Windows 的基本操作（移动、缩放等）外，它只是接受用户的输入，然后再忽略。Delphi 把 Windows 编程的回调、句柄处理等繁复过程都封装到了对象里面，这样就可以不为它们所困扰，从容地对可视控件进行编程。

1.1.2 Delphi 的特点

“真正的程序员用 C，聪明的程序员用 Delphi”，这句话是对 Delphi 最经典、最实在的描述。到底什么使得 Delphi 如此优秀？和别的编程工具相比，为什么程序员更愿意选择

Delphi 呢？这应该归结于它的高效性。要创建 Windows 应用程序，使用 Delphi 是目前能够找到的最为简捷的途径。决定一个软件开发工具效率的因素可以归结为以下五点：可视化开发环境的性能、编译器的速度和已编译代码的效率、编程语言的功能及其复杂性、数据库结构的灵活性和可扩展性和框架对设计和使用模式的扩充。另外还包括如配置、文档、第三方的支持等。

Delphi 和 Visual Basic 在完善窗体设计器的功能方面展开了激烈的竞争。它们的新版本的功能一个比一个强。Delphi 的窗体设计器的与众不同之处在于 Delphi 是建立在一个真正面向对象的框架结构基础之上的。这样，对基类所做的改变都将会传递给所有的派生类。这里涉及的一项关键技术就是 VFI (visual form inheritance)，即可视化窗体继承。VFI 技术使程序员能够动态地继承当前项目或对象库中的任何其他窗体。一旦基窗体发生改变，派生的窗体会立即予以更新。

Delphi 快速编译器可以使程序员逐步递进地开发软件，经常地修改源代码、重新编译、测试、再修改、再编译、再测试……，从而形成这样一个良好的开发循环。如果编译速度很慢，开发者就不得不分批地修改代码，每次编译前进行多处修改以适应一个低效率的循环过程。Delphi 在提高运行效率、节约运行时间、生成的二进制代码更为短小等这些优越性上是不言而喻的。Pascal 编译器最著名的特点就是速度快，而 Delphi 正是建立在这种编译器的基础之上的。事实上，它可能是针对 Windows 的最快的高级语言本地代码编译器。以往速度很慢的 C++ 编译器在近年来取得了很大的进步，增加了链接和各种缓存策略，尤其是在 Visual C++ 和 C++ Builder 中。但即便如此，C++ 的编译器还是比 Delphi 的编译器慢了几倍。因此，选择 Delphi 编程是最确切、最简单的一种编程方式。

Delphi 的主要特性如下：

- 基于窗体和面向对象的方法，高速的编译器，强大的数据库支持，与 Windows 编程紧密结合，强大而成熟的控件技术，这些都是 Delphi 的基本特性。但最重要的还是 Object Pascal 语言，它才是一切的根本。Object Pascal 语言是在 Pascal 语言的基础上发展起来的，简单易学。

- Delphi 提供了各种开发工具，包括集成环境、图像编辑 (Image Editor)，以及各种开发数据库的应用程序，如 DesktopDataBase Expert 等。除此之外，还允许用户挂接其他的应用程序开发工具，如 Borland 公司的资源编辑器 (Resource Workshop)。

- 在 Delphi 众多的优势当中，它在数据库方面的特长显得尤为突出，从而适用于多种数据库结构：从客户机 / 服务器模式到多层数据结构模式；高效率的数据库管理系统和新一代更先进的数据库引擎；最新的数据分析手段和提供大量的企业控件。

1.1.3 Delphi 7 的新特点

Delphi 发展至今，从 Delphi 1.0、Delphi 2.0 到现在的 Delphi 7，不断添加和改进各种特性，功能越来越强大。Delphi 5.0 添加了针对 IDE (集成开发环境) 的很多改进的新特性：扩展了数据库支持 (ADO 和 InterBase 数据库)，带有 Internet 支持的 MIDAS 改进版，TeamSource 版本控制工具，转换功能，框架概念。

Delphi 7 又增加了很多新特点。与以往版本相比，Delphi 7 突出对 .NET、Web Services、跨平台开发的准备和支持，Delphi 7 提供了开发电子商务 (E-business) 应用程序

的众多新特征，可概括为：

- 提供了一些工具，对 NET 架构进行准备。
- 使用模板驱动架构（Model Driven Architecture, MDA）。
- 引入 AToZed 公司的 IntraWeb 技术。
- 通过 Borland Kylix 3 for Delphi 版本，重新编译并分发给 Linux 平台应用。这体现了 Delphi 7 的跨平台的兼容性。
- 通过 Delphi 7，可以使用用户的商业开发与 Web Services 进行整合。
- 使用 Delphi 7，可以浏览可用的 UDDI 目录，采用网络指纹技术。
- Delphi 7 的 BizSnap 技术，强有力地扩展了使用商业 XML 的 Windows 操作系统与 Web Services 兼容能力。这项技术也使程序更容易交换、转移和操作 XML 文档。因此，可以在 Delphi 7 中开发与商业伙伴所使用 Web Services 的平台无缝结合的应用程序，包括支持 Microsoft 的 .NET、BizTalk，以及 SUN System 的 ONE 等。
- 在 Delphi 7 中，引入了由 MIDAS 发展而来的 DataSnap 技术。
- 在 Delphi 7 中，引入 Rave Reports 技术，使程序员可以快速开发跨平台的报表。
- 在 Delphi 7 中，提供了支持 Windows XP 主题风格的 VCL 控件。

1.2 Delphi 集成开发环境

Delphi 7 与 Visual Basic 一样，是用来创建 Windows 应用程序的一种快速应用开发工具、一种可视化编程环境，可以更加直观地使用图形化工具来创建 Windows 的应用程序。它使程序员从完全以代码的形式创建可视化应用程序的繁琐且复杂的工作中解脱出来，使用户可以不用掌握太多的编程专业知识就能够创建出图形化的高难度的应用程序，这一切都得益于 Delphi 的集成开发环境（IDE）。

Delphi 7 的 IDE 与 Delphi 以前的版本一样，使用一组窗口、菜单和应用程序来设计可视化界面，将代码以事件的形式与界面的每一元素建立联系，并可以对整个应用程序进行调试。

1.2.1 用户界面

启动 Delphi 7 后，程序窗口如图 1-1 所示。

整个 Delphi 7 的窗口包括五部分，分别是：主窗口、对象监视器、窗体、代码编辑器和对象树形图，下面介绍各部分的构成和功能。

主窗口（Main Form）

Delphi 的主窗口位于屏幕的上端，包括 Menu（菜单栏）、Speed Bar（工具栏）和 Component Panel（控件面板）。Menu 是下拉式主菜单，Speed Bar 位于主窗口的左下端，由两排共 14 个加速按钮组成。这些按钮是菜单功能的快捷方式，各种图标直观地表示了它能执行的动作。Component Panel 由一行、若干页对象按钮所组成，利用它可以选择需要的控件并将它放到窗体中去。



图 1-1 Delphi 的程序界面

【Object Inspector】(对象监视器)

4

【Object Inspector】窗口含有两个标签页：【Properties】标签页显示窗体中当前被选择控件的属性信息，并允许改变对象的属性；【Events】标签页列出了当前控件可以响应的事件。单击【Object Inspector】窗口下端的【Events】标签页，使得【Events】标签页可见，然后双击指定的事件后边的空白处，可以定义对象接收到相应事件时执行的动作。首次启动时，【Object Inspector】窗口显示的是当前窗体 Form1 的属性。【Object Inspector】窗口根据对象属性的多少，决定是否有滚动条。移动滚动条，可以查看当前对象的全部属性。

此外，位于【Object Inspector】窗口上方的下拉式菜单中显示了窗体上所有控件的名称和类型，也包含窗体本身，可以很容易地在窗体的各个控件之间切换，也可以快速地回到窗体本身。当窗体中含有较多的对象时，就会发现这是切换对象尤其是回到窗体的最快捷的途径。



技巧：

想使【Object Inspector】窗口一直可见，可将鼠标指针移到【Object Inspector】窗口上，单击鼠标右键，在弹出的快捷菜单中选择【Stay On Top】选项。这对初学者常是一个很重要的设置方式。

窗体窗口

Forms 窗口是开展大部分设计的工作区域。首次启动 Delphi 时显示的是窗体 Form1。可以把控件放在窗体中，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控件的容器。窗体上有栅格 (Grids)，供放置控件时对齐位置用，在程序运行时 Grids 是不可见的。

一个真正的应用程序可能有不止一个窗口，可以选用不同的窗体进行设计。其他窗体可以是对话框 (Dialog Box)、数据录入框等。

代码窗口

代码窗口一开始处于窗体窗口之下。因为在 Delphi 中，设计用户界面直接在窗体中进行，运行结果和设计样板完全一致。当控件被放到窗体上时，Delphi 会自动生成大部分的用户界面代码。程序员应做的只是在它生成的框架中加入完成所需功能的程序段而已。单击 Form1 的状态行可使代码窗口可见。

对象树形图窗口

该窗口用来显示用户使用的所有可视化控件，对于管理程序、明确各控件间的关系有很大帮助。

1.2.2 主窗口

主窗口的界面如图 1-2 所示。

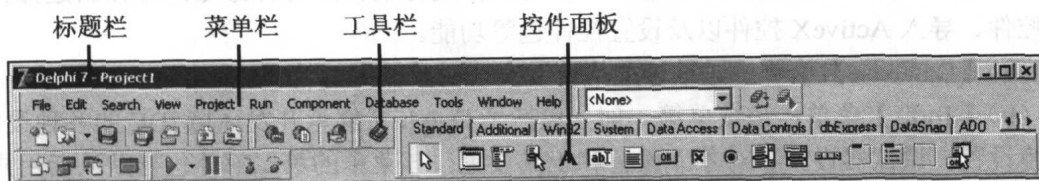


图 1-2 Delphi 的主窗口

主窗口有四部分：标题栏、菜单栏、工具栏和控件面板。下面简要介绍各部分的功能。

标题栏

主窗口最上部为标题栏，上面显示的是当前工作的 Project (项目) 的名称。所谓项目，就是一个独立应用程序中所包含的文件的总称。

菜单栏

菜单栏在标题栏下面，与一般应用程序的菜单栏形状相同，包括【File】、【Edit】、【Search】等 11 项下拉式菜单，这些菜单融合了 Delphi 7 程序开发的全部命令和功能。

(1) 【File】菜单：和普通的 Windows 菜单类似，它列出了 Delphi 7 文件标准的存取命令，如【Open】、【Save】及【Close】等，除此之外，还有创建新的应用程序、窗体和数据模块等命令。

(2) 【Edit】菜单：主要是对文本或控件进行编辑操作。该菜单分为两部分，前两部分为【Undelete】、【Cut】、【Copy】等，它们的功能与一般应用软件相同；第二部分用来对控件进行控制，包括设置控件大小以及设置它们的排列方式等。

(3) 【Search】菜单：该菜单中的命令用于对代码编辑器中的文本、错误、对象、变量和符号进行定位操作。其中：

- 【Find】命令用于打开一个搜索对话框，用于搜索用户指定的文本内容，并且切换

到编辑窗口中，以高亮方式将第一次出现该文本的内容显示出来。

- **【Find in Files】** 命令用于在指定文件中进行搜索。
- **【Replace】** 命令用于文本替换操作，与一般应用程序中的替换命令相同。
- **【Search Again】** 命令用于重复上一次的**【Find】**或**【Replace】**命令，当选择该命令时，上次在**【Find Text】**或**【Replace Text】**对话框中设置的内容将有效。
- **【Incremental Search】** 命令可以越过**【Find Text】**对话框而直接将光标移到用户键入文本的地方，并且在代码编辑框的状态栏里出现用户键入的文本。
- **【Go to Line Number】** 命令可以指定光标所在的行。
- **【Find Error】** 命令用于在程序运行产生错误时，显示产生错误的地址。
- **【Browse Symbol】** 命令用于浏览用户指定的符号。

(4) **【View】** 菜单：使用**【View】**菜单，用户可以隐藏或显示 Delphi 中 IDE 环境的不同元素，如项目管理器、对象监视器等，也可以打开继承调试器的窗口。

(5) **【Project】** 菜单：该菜单包含了用于编译和建立用户应用程序的各种命令。比如添加单元、删除单元、添加项目、删除项目等。

(6) **【Run】** 菜单：提供了 Delphi 7 中程序调试、编译和运行的各种命令。

(7) **【Component】** 菜单：该菜单包含对控件进行编辑的各种命令，比如新建控件、添加控件、导入 ActiveX 控件以及设置控件包等功能。

(8) **【Database】** 菜单：通过该菜单可以实现对数据库的管理。

(9) **【Tools】** 菜单：该菜单第一部分包括了 Delphi 7 继承开发环境 IDE 的各种设置，如环境设置、调试器设置、对象库设置和工具设置。该菜单的第二部分包括默认后用户自己设置的常用工具，包括：**【DataBase Desktop】**、**【Image Editor】**、**【Rave Designer】**等。

(10) **【Window】** 菜单：该菜单主要用于在 Delphi 的不同窗口间切换，并且其菜单项随创建的窗体的不同而有所变化。

工具栏

工具栏位于主窗体的左下部，其中包含许多可执行动作的快捷按钮，这些按钮是最常用菜单命令和功能的快捷方式，如图 1-3 所示。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。

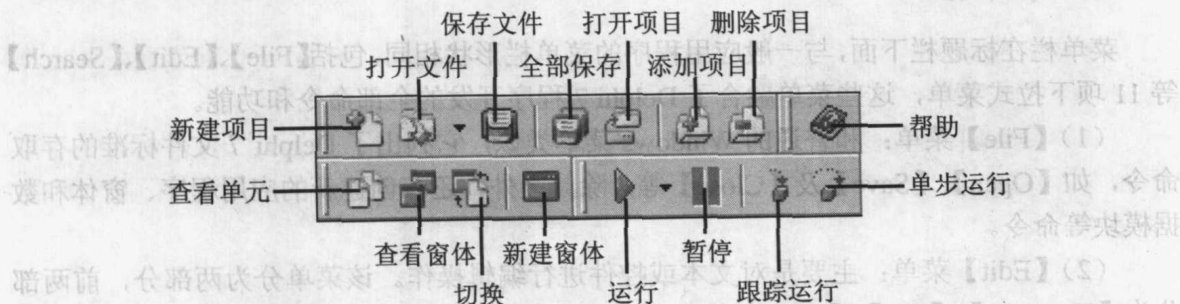


图 1-3 工具栏的功能

对工具栏可以进行个性化设置，即用户可以选择自己最常用的快捷按钮放到桌面上。具体方法是：单击**【View】|【Toolbars】|【Customize】**菜单项，将会弹出如图 1-4 所示的窗口，在其**【Toolbars】**标签页上可以选择自己常用的按钮组。

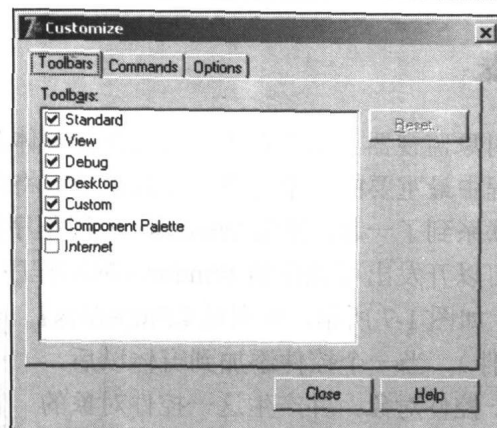


图 1-4 自定义工具栏

控件面板

控件面板实际上是一个标签页盒，位于菜单栏的右下方，包括【Standard】、【Additional】和【Win32】等 30 多个标签页，如图 1-5 所示。每个标签页又包括若干控件，它们都是以图标形式显示出来。控件是建立一个应用程序最常用的元素。



图 1-5 控件面板

用户可以自己定制控件面板的标签页和各个标签页所包含的控件，方法是：单击【Tools】|【Environment Options】菜单项激活如图 1-6 所示的环境设置对话框，选择其中的【Palette】标签页，左边为控件面板的各个标签页，右边是该标签页包含的控件。通过单击【Add】和【Delete】按钮可以添加或删除标签页，也可以单击【Rename】按钮对标签页进行更名操作。除了隐藏标签页中的控件外，还可以添加控件到指定的标签页，如果是其他标签页的控件，可以简单地通过鼠标的拖放操作来完成。如果是从控件库中选取控件，就需要先构造和安装控件包。

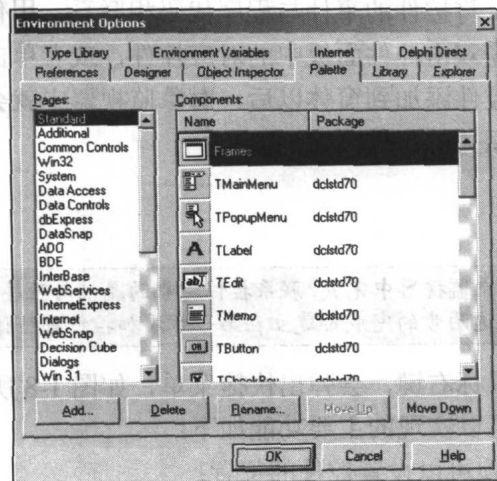


图 1-6 设置控件面板

1.2.3 对象监视器

【Object Inspector】(对象监视器)用来描述控件对象及窗体对象的属性特征和行为事件。它是应用程序设计过程中最重要的一个工具。正是由于它将生动友好的可视化窗体与原本枯燥无味的程序代码联系到了一起,使得 Windows 应用程序的开发变得如此轻松。无需探究乏味的编程语言就可以开发出可视化的 Windows 应用程序。

对象监视器包括两个,如图 1-7 所示,分别是【Properties】(属性)和【Events】(事件)。当一个控件添加到窗体以后,对象监视器中就会添加这一控件对象,并产生这一控件对象的属性列表和与这一控件对象相关的事件列表。属性列出了控件的 Published 属性。对象监视器中的属性和事件列表与窗体设计器中选中的对象是一一对应的。部分属性前有“+”号,称之为多重属性,表示这一属性里包含了好几项。例如: +Filter Options 等等。如果不想该窗口被覆盖,可在该窗口中单击鼠标右键,从弹出的快捷菜单中选择【Stay On Top】选项即可。属性中最重要的是控件的 Name 属性,每个对象都有自己独一无二的 Name 属性,即使是同种控件,因为它是每个控件的标识。例如:有两个控制按钮,其 Name 分别为 Cmd1 和 Cmd2。

Cmd1.Caption 代表的是第一个按钮的标题,而 Cmd2.Caption 代表的是第二个按钮的标题。只有通过 Name 属性,系统才能识别不同的对象。

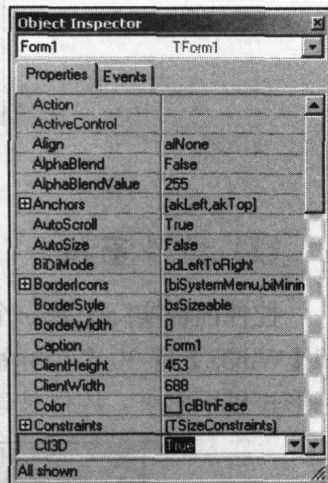


图 1-7 对象监视器

1.2.4 窗体窗口

窗体设计器作为 IDE 的一个部分,主要通过向窗体中添加控件以及更改窗体及控件的属性来完成窗体的设计。应用程序的实现正是用事件的形式将窗体及其中的控件的属性与程序代码结合起来。关于事件的概念,在介绍对象监视器时会详细叙述。

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控件;更改控件属性使之满足程序设计要求;将控件的事件与程序代码相联系,用代码实现控件的活动。其中,添加控件到窗体中只需双击控件面板中的对应控件,或者单击对应控件后再单击窗体中该控件将放置的位置。控件添加到窗体以后,对象监视器中就会产生这一控件对象的属性列表和与这一控件对象相关的事件列表。



更改控件属性在对象监视器中完成,联系控件对象的属性与程序代码由对象监视器与代码编辑器共同完成,而这两步的完成都建立在第一步添加控件到窗体的基础上。

在窗体设计器上单击鼠标右键,会弹出快捷菜单,如图 1-8 所示。快捷菜单中的各选项可以实现的主要功能如下:

- 【Position】: 该选项可以设置控件的位置: