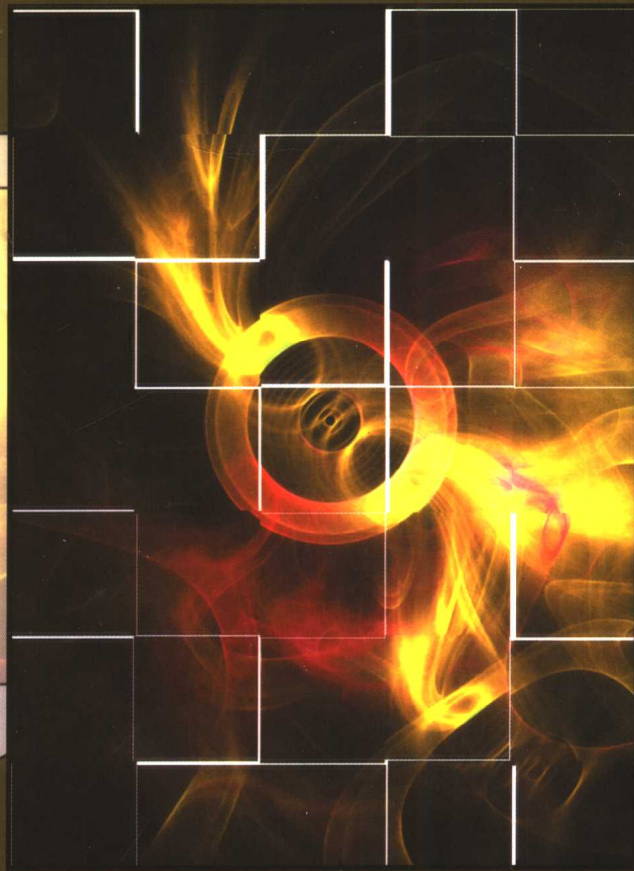


# Delphi 精要



罗小平 编著



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONIC INDUSTRY  
<http://www.phei.com.cn>

Borland In-Depth Series\Borland 大系

# Delphi 精要

罗小平 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 提 要

本书的核心是讲述 Object Pascal 和 VCL 的重、难点知识(即所谓“精要”),集中于第3章、第5章。第2章、第4章阐明相应的一些基本概念,是为初学者而设立的。

本书也是一本参考手册,包括了“IDE 的快捷键列表”(2.8节)、“编译指令”(3.4节)、“VCL 消息大全”(5.2.4节)、“常用函数和过程”(第8章)以及一些常用的、重要的开发技巧(第9章)等内容。

本书同时也较为全面地讲述了组件开发知识,集中于第6章、第7章。

第10章通过一个实例综合运用了全书内容。

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 精要/罗小平编著. —北京:电子工业出版社,2004.1

(Borland In-Depth Series\Borland 大系)

ISBN 7-5053-9412-6

I. D... II. 罗... III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆CIP数据核字(2003)第110153号

责任编辑:周 筠 陈元玉

技术编辑:韩 磊

印 刷 者:北京市增富印刷有限公司

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:787×980 1/16 印张:25.25 字数:600千字

印 次:2004年1月第1次印刷

印 数:6000册 定价:39.80元(含光盘1张)

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 zltis@ phei. com. cn,盗版侵权举报请发邮件至 dbqq@ phei. com. cn。

# 前 言

---

Delphi 是什么？它是一个让我们做菜的厨房。在这间厨房里，四周墙壁上镌刻着烹饪原理的众多细节，桌子上有经典的菜谱和齐全的调味品以及各种炊具。从来没有进过厨房的人来到这里，可以很快地煮出一碗可口面条；优秀的厨师在这里也可以烹制国宴。

你想成为一个优秀的厨师吗，抑或低一点要求——一个拉面能人、一个操刀高手？

本书带你去认识 Delphi 的精华。让我们拨开油烟审视墙上的真实字迹，潜到锅底透视炸制鸡腿的详细过程。

本书作者长期在 CSDN（中国程序员大本营）的 Delphi 论坛回答和讨论问题（CSDN 网名：lxpbuaa（桂枝香在故国晚秋）），由此了解了很多朋友在学习和使用 Delphi 中的不足和未能突破之处。

为什么我用 Delphi 很长时间了，还是感觉自己很菜？为什么我遇到一点新东西就不知所措？为什么别人用两行代码实现的功能我却要花十行的代价？为什么翻来覆去我都只会用简单的几个函数去拼凑复杂功能？不会开发组件、搞不懂消息，怎么办？我体会不到开发的乐趣，只有机械性劳作那种很累的感觉！我写的东西总是要无数次地修改！

这么多问题的根源在哪里？是因为没有突破基础理论的学习，没有掌握提升自己开发能力的方法，没有真正理解 Delphi，与 Delphi 的沟通存在障碍。

本书针对众多 Delphi 使用者的上述问题，结合作者的工作、学习体会，详细讲解了 Object Pascal 和 VCL 的重要概念、结构和知识要点，揭示了很多资料上没有涉及但是相当重要的东西，并在其中穿插大量实例演示这些重要知识。

本书的写作目的是帮助朋友们弄清重要原理，并掌握大量开发技巧，从而在 Delphi 学习和使用上有一个质的突破，并找到提升自己功力的有效方法。

阅读本书，可以澄清很多概念上的认识误区，也能掌握众多的开发技巧和设计细节。本书主要面向有一定使用 Delphi 开发经历的朋友，对于初学者也有较高的参考价值。

有了本书的帮助，你或许会发现不需要像下面这样学习 Delphi：每天费尽心机去琢磨它、讨好它，一点点抠每个组件的属性、方法、事件；你可能发现学会一个，就懂得了一打，举一可以反三。那时候你就会发现，学习和使用 Delphi 是一件很快乐的事情。

作者虽力求完美，但智识短浅，谬误之处已尽力避免，疏漏之处还请读者批评指正。

本书配套光盘中包含本书所有的源代码。

感谢电子工业出版社的郭立、周筠二位老师和张毅先生、陈元玉小姐对此书出版所做的支持；感谢 CSDN 的韩磊先生为本书技术验证所付出的辛勤劳动。

罗小平  
2003.10 于成都

# 目 录

<b>第 1 章 关于 Delphi 与本书</b> .....	(1)
1.1 Delphi 是什么.....	(1)
1.1.1 Delphi 的历史.....	(1)
1.1.2 选择 Delphi 意味着什么.....	(2)
1.1.3 如何使用 Delphi 编程.....	(3)
1.2 本书的特点.....	(4)
1.2.1 本书有哪些内容, 没有哪些内容.....	(5)
1.2.2 本书是如何写作的.....	(5)
1.2.3 如何阅读本书.....	(5)
<b>第 2 章 Object Pascal 入门</b> .....	(7)
2.1 运算符.....	(7)
2.1.1 赋值运算符.....	(7)
2.1.2 比较运算符.....	(7)
2.1.3 逻辑运算符.....	(8)
2.1.4 算术运算符.....	(8)
2.1.5 按位运算符.....	(9)
2.1.6 其他运算符和模拟运算.....	(9)
2.2 常量和变量.....	(12)
2.2.1 无类型常量和有类型常量.....	(13)
2.2.2 全局变量和局部变量.....	(13)
2.2.3 声明时使用编译时函数.....	(14)
2.2.4 资源字符串.....	(15)
2.3 过程和函数.....	(15)
2.3.1 过程和函数的区别.....	(15)
2.3.2 子过程.....	(16)
2.4 数据类型.....	(17)
2.4.1 常用类型和复杂类型.....	(17)
2.4.2 不同语言的数据类型对照表.....	(17)

2.5	程序流程控制	(18)
2.5.1	条件分支	(18)
2.5.2	循环	(19)
2.5.3	跳转	(20)
2.5.4	用过程辅助实现流程控制	(21)
2.6	单元的组织结构	(24)
2.6.1	Program 的组织结构	(24)
2.6.2	Unit 的组织结构	(25)
2.6.3	单元循环引用	(27)
2.7	with...do 语句的用法	(28)
2.8	IDE 的快捷键列表	(29)
<b>第 3 章</b>	<b>Object Pascal 精要</b>	<b>(31)</b>
3.1	数据类型及其相互关系	(31)
3.1.1	数据类型概述	(32)
3.1.2	变量的内存分配和释放	(48)
3.1.3	数据的内存结构	(52)
3.1.4	强数据类型与类型转化	(53)
3.2	过程和函数	(59)
3.2.1	作用域	(60)
3.2.2	参数传递	(62)
3.2.3	声明指令	(66)
3.3	类和类成员	(66)
3.3.1	类和类成员概述	(67)
3.3.2	深入认识方法	(69)
3.3.3	深入认识属性	(78)
3.3.4	深入认识事件	(84)
3.3.5	类成员重新声明	(86)
3.3.6	inherited 释疑	(88)
3.3.7	接口的真相	(89)
3.4	编译指令	(95)
3.4.1	开关指令	(96)
3.4.2	参数指令	(99)
3.4.3	条件指令	(102)



<b>第 4 章 VCL 入门</b> .....	(104)
4.1 VCL 概述.....	(104)
4.2 组件与控件的概念.....	(105)
4.3 使用 VCL.....	(105)
4.4 扩展 VCL.....	(107)
<b>第 5 章 VCL 精要</b> .....	(108)
5.1 揭开 VCL 的神秘面纱.....	(108)
5.1.1 VCL 架构.....	(108)
5.1.2 构造和析构的内幕.....	(111)
5.1.3 虚拟方法表和动态方法表.....	(115)
5.1.4 TObject 如何使用虚拟方法表.....	(118)
5.1.5 运行时类型信息.....	(122)
5.2 VCL 的消息机制.....	(130)
5.2.1 VCL 消息机制.....	(130)
5.2.2 处理消息的八种方法.....	(134)
5.2.3 选用什么方法发送消息.....	(140)
5.2.4 VCL 消息大全.....	(144)
5.3 多态性.....	(161)
5.3.1 多态性的概念.....	(162)
5.3.2 多态性和虚方法的关系.....	(164)
<b>第 6 章 组件开发实战</b> .....	(167)
6.1 三种组件开发方法.....	(167)
6.1.1 继承、聚合和子类化.....	(167)
6.1.2 接口、虚方法和辅助类的选择.....	(169)
6.2 文件拖放监视器.....	(169)
6.2.1 文件拖放原理.....	(169)
6.2.2 文件拖放实例.....	(170)
6.2.3 组件封装.....	(171)
6.3 托盘组件.....	(175)
6.3.1 装入托盘图标.....	(175)
6.3.2 在应用程序最小化时去掉状态栏的图.....	(177)
6.3.3 给托盘图标增加接收鼠标消息功能.....	(179)



6.3.4	处理鼠标消息	(181)
6.3.5	显示动画图标	(182)
6.3.6	设置程序的自动启动功能	(182)
6.3.7	组件封装	(183)
6.4	自动下拉的 TComboBox	(190)
6.5	开发数据敏感控件	(195)
6.5.1	数据敏感原理	(196)
6.5.2	开发日期敏感控件	(196)
6.6	开发聚合组件	(203)
6.6.1	开发 LabelDBDatePicker	(203)
6.6.2	加强 LabelDBDatePicker	(206)
6.7	开发图形图像控件	(211)
6.8	开发 QuickReport 组件	(218)
<b>第 7 章</b>	<b>组件开发相关工作</b>	<b>(225)</b>
7.1	包和包编译指令	(225)
7.2	创建组件图标	(226)
7.3	属性编辑器	(227)
7.4	组件编辑器	(231)
<b>第 8 章</b>	<b>常用函数和过程</b>	<b>(235)</b>
8.1	数据类型转化类	(235)
8.1.1	数值和字符串的相互转化	(235)
8.1.2	整数和字符串的相互转化	(236)
8.1.3	实数和字符串的相互转化	(236)
8.1.4	实数子类型的相互转化	(237)
8.1.5	布尔类型和字符串的相互转化	(237)
8.2	字符串处理类	(237)
8.2.1	字符串的分类	(240)
8.2.2	和字符串相关的类	(241)
8.3	流处理类	(245)
8.4	内存管理、程序流程控制类	(250)
8.4.1	内存管理	(250)
8.4.2	程序流程控制	(251)



8.5	文件操作类	(252)
8.5.1	使用文件句柄进行 I/O 处理	(252)
8.5.2	使用 Pascal 文件变量进行 I/O 处理	(253)
8.5.3	面向对象文件 I/O 处理	(255)
8.5.4	文件属性操作	(255)
8.5.5	其他函数和方法	(256)
8.6	日期时间类	(258)
8.6.1	获取/合成日期/时间	(258)
8.6.2	日期/时间和字符串的转换	(259)
8.6.3	日期/时间的运算	(259)
8.7	VCL 类	(260)
8.7.1	Classes 单元	(260)
8.7.2	Controls 单元	(262)
8.7.3	Dialogs 单元	(262)
8.8	位运算类	(264)
8.9	图形图像类	(266)
<b>第 9 章</b>	<b>高级开发技巧</b>	<b>(268)</b>
9.1	自定义窗口过程	(268)
9.2	自定义消息及其替代方法	(270)
9.3	自定义系统惟一消息	(272)
9.4	新颖的类工厂	(275)
9.5	使用对象库	(281)
9.6	非发布 (published) 数据的持久化	(287)
9.7	使用回调函数	(288)
9.8	使用递归算法	(290)
9.9	编写 NT 服务程序	(294)
9.10	编写只能惟一运行的程序	(295)
9.11	字段类型全家福	(298)
9.12	获取数据库结构信息	(300)
9.13	深入使用 TCanvas	(301)
9.14	指针列表类的使用	(308)
9.15	结构化存储技术	(312)
9.16	挂钩技术	(321)

9.17	TRichEdit 高级开发.....	(327)
9.18	用 TTreeView 分析数据表的结构.....	(334)
9.19	SQL 语句分析器.....	(339)
9.20	剪贴板高级编程.....	(344)
<b>第 10 章</b>	<b>综合例子——使用 Socket 传输多个文件.....</b>	<b>(350)</b>
10.1	Socket 简介.....	(350)
10.2	TServerSocket 和 TClientSocket.....	(351)
10.3	设计通讯协议.....	(354)
10.4	实现服务端.....	(357)
10.5	实现客户端.....	(367)
10.6	组件封装.....	(374)
10.7	自动下载技术在项目中的应用.....	(391)

目

录

# 第 1 章 关于 Delphi 与本书

---

在一本书的开头，总应该说点什么，我也不能脱离俗套。既然是俗套，或许也的确是必要的。黑格尔不是说过：存在即合理。

既然要说，就得说点有价值的，如果全是废话、大话、空话，就难免遭人唾骂。

在本章里，我想谈谈两个方面的话题：

- (1) 我对 Delphi 的一些认识。
- (2) 这本书的特点。

## 1.1 Delphi 是什么

这一节希望读者朋友对 Delphi 有一个总体的认识，包括其发展历史、特点。无论是在网络论坛、办公室，还是在技术座谈会上，时常有人询问“Delphi 有没有前途”、“我该选择 .NET、Java 还是 Delphi”、“Delphi 的旗帜还能打多久”之类的问题，即使是正在使用 Delphi 的朋友，也常常被这样的问题困扰。本节不指望能正面回答这些问题，但是希望能给出一些理由让困惑的朋友们平静下来，从而将时间和精力放到更重要的事情上去。

### 1.1.1 Delphi 的历史

在 DOS 时代，程序员可选择的开发工具很少：要么是易用但低效的 BASIC 语言，要么是高效但难用的汇编语言。即使到了 Windows 3.X 时代，仍然是两难选择：要么是容易使用但功能十分有限的 Visual Basic，要么是强大但难以使用的 C/C++。

但是到了 1995 年，情况发生了很大变化。那就是 Delphi 1.0 在 Borland 公司诞生了。Delphi 1.0 运行在 Windows 3.X 平台上，以 Object Pascal（面向对象的 Pascal，从 Pascal 发展而来）为开发语言，它提供了一种全新的 Windows 程序开发方法：可视化的开发环境、真编译后的可执行程序、DDL 和支持数据库软件开发。Delphi 1.0 是第一个综合了可视化开发技术、优化的源代码编译器和可扩展的数据库访问引擎的 Windows 平台开发工具。Delphi 1.0 奠定了 RAD（Rapid Application Development，快速应用程序开发）的概念。

尽管在 Delphi 3 开发前后，著名的 Delphi 首席设计师 Anders Hejlsberg（第一个 Turbo Pascal 编译器的作者）、首席技术总裁 Paul Gross 离开 Borland 公司去了 Microsoft 公司，并且在整个的 Delphi 发展过程中，Borland 公司也出现了一些重大决策失误，但是 Delphi 仍然拥有全世界众多的爱好者和使用者。

8年过去了, Delphi 已经进入了 7.0 阶段, 并且推出了 Linux 平台的版本 Kylix, Delphi 8 for .NET 也已经面世。

### 1.1.2 选择 Delphi 意味着什么

虽然很多人和我一样选择了 Delphi, 但是有一点是肯定的, 我们多半同时选择了或迟早要面临选择别的工具。

在 DOS 和 Windows 3.X 时代, 可选择的工具太少, 甚至于没有选择, 所以大家只好安贫乐道, 比如 Anders Hejlsberg、求伯君二位前辈用汇编语言分别完成了 Delphi 1.0 的编译器和 WPS。到了开发工具群雄并起的时候, 面临众多诱惑和困惑, 要作出让自己满意的选择反而成了难事。

我不打算说服你选择 Delphi, 我只能说, 选择自己最适用的工具。

Delphi 对于程序员来说, 应该是一个工具。选择 Delphi, 是因为在一定阶段里, 它能和我们构成最佳匹配, 而再没有别的原因。比如你受到身边 Delphi 高手的鼓动, 经理指定 Delphi 为项目的开发工具, 公司的主流开发工具就是 Delphi, 为了技术研究目的选择 Delphi, 等等。要是我说 Delphi 是最先进、功能最强大的开发工具, 肯定很多人会跳起来说出 Delphi 的一大堆不足和其他工具的许多优势。网上有很多程序来测试 Delphi 和 VC、VB 等开发出来的可执行程序的运行效率, 并指出某某比其他的要快出百分之多少、哪个传送消息要快得多, 我想这些对于软件用户来说都没有多大价值。影响一个人选择的因素很多, 要说哪个因素总是占绝对的优势地位, 是很牵强的。

但是 Delphi 也的确有众多出类拔萃的优势, 我最看重的是以下几点:

- (1) 编译速度非常快。
- (2) IDE 反应速度很快。
- (3) 完全开放的 VCL 源代码和规范简练的帮助系统。

选择开发工具, 和选择恋人是一样的道理。你的选择对于你可能是最适合的, 你认为他/她是全世界最好的, 但是其他人或许就不这么看。因此, 大可不必将精力过分专注于本节开头提到的那些问题。

慎重作出选择, 然后忠实于自己的选择。如果朝三暮四, 最后只能是什么都不能精通, 浪费时间; 当然对于那些一点就通的天才就另当别论。只有乐于此道的人, 才能乐在其中; 上天总是给不安分者更多的痛苦。

在大多数时候, 众多语言都能达到殊途同归的效果, C/C++、Object Pascal、Java、C# 有异曲同工之妙。如果一定要选择最好的, 就得将它们全部精通了, 才能作出绝对公正的决定, 但是世界上恐怕没有这样的人; 即使有, 他也会非常忙, 因为很多人都恳请他帮自己作出选择, 他最终只好偷偷跑到深山里躲起来, 否则他的电话会被打爆或者被提问者的唾沫星子淹死。

最终能作出什么成果, 关键还是在于工具使用者。武林高手发出一声长啸, 就可以让敌人丧失听觉, 但是弱女子手提宝剑、面对身负重伤的敌手时却不见得稳操胜券。

所以, 如果你选择了 Delphi, 那么赶快去掌握它吧, 本书将助你一臂之力; 如果你还在彷徨中虚度光阴, 那就尽早作出选择吧。

### 1.1.3 如何使用 Delphi 编程

Delphi 是一个快速开发工具，“快速”的特点主要体现在两个方面：

#### 1. 提供众多的向导程序

Delphi 向导程序主要包含在对象库（Object Repository）中，选择菜单 File/New/Other 可以看到所有的向导程序，如图 1-1。

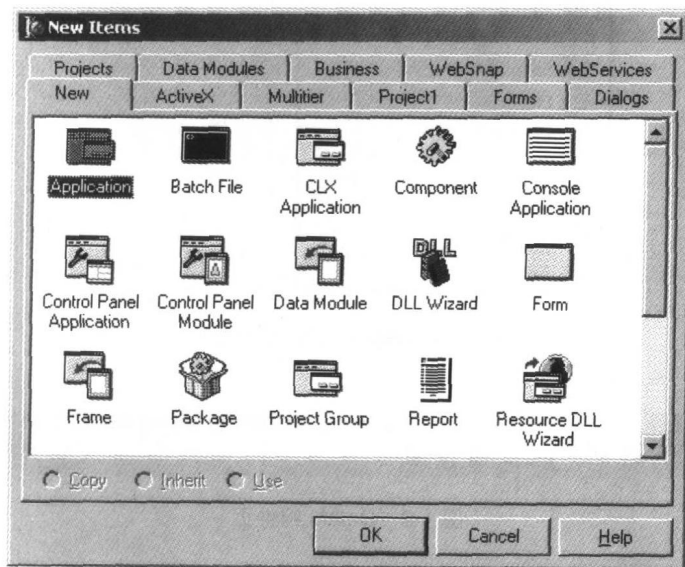


图 1-1 Delphi 全部向导程序（对象库）

如果只须创建普通的应用程序或者应用程序窗体、单元、数据模块等，可以通过菜单 File/New 导入，如图 1-2。

#### 2. 提供了大量的组件

不仅 Delphi 本身附带的组件很多，而且在 Internet 上有非常多的组件包提供下载，可以安装到 Delphi 中直接使用，况且在 Delphi 中自己开发组件也是十分容易的。图 1-3 是 Delphi 的组件面板截图，其中主要是 Delphi 自带的组件，“lxbuaa”是本书光盘（“第 6 章”目录下）附带的组件包，“Data ExControls”是我为公司开发的组件包。

通过向导程序生成一些对象后，如应用程序、窗体、数据模块等，从组件也拖放一些组件到窗体、数据模块中，最后设置必要的属性，再编写一些简单的代码（如处理事件），就可以按 F9 键运行程序了。

比如，我们选择菜单 File/New/Application，Delphi 自动生成包含一个窗体的应用程序，然后从 lxbuaa 组件页（请首先安装光盘的 lxbuaa.dpk 组件包）拖一个托盘组件（即图 1-3 中倒数第 4 个组

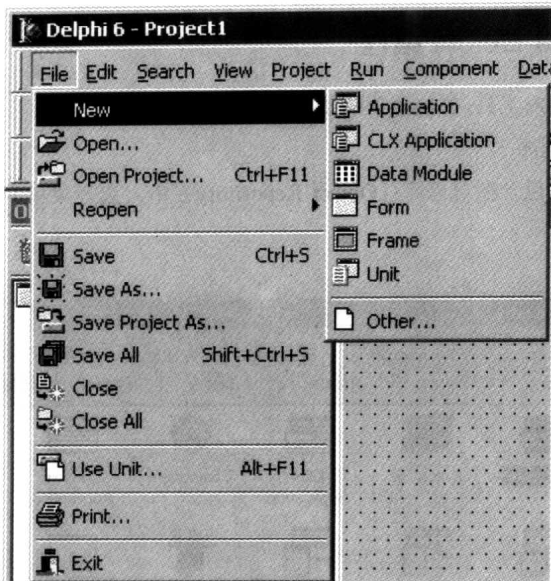


图 1-2 Delphi 常用向导程序

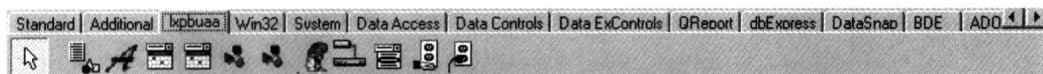


图 1-3 Delphi 组件页

件: TlxpTrayIcon) 到窗体, 设置它的 Hint 属性为“托盘图标!”即可, 不需要写一行代码, 最后按 F9 键运行程序, 我们就看到在 Windows 托盘区出现了该程序的托盘图标(见图 1-4 所示托盘区的第二个图标)。



图 1-4 托盘图标

很简单吧! 如果你还不会安装组件包, 请不要着急, 本书后面章节有详细讲解。直接运行光盘上的“源代码第 1 章”目录下的 Project1.exe 可以看到同样的效果。

## 1.2 本书的特点

本书名为《Delphi 精要》。那什么是 Delphi 的精要呢? 在我看来, Delphi 精要可以分为两个方面:

(1) Object Pascal。

(2) VCL。

所以全书都是围绕着这两个方面来写作的。

### 1.2.1 本书有哪些内容，没有哪些内容

既然是讲精要，所以我没打算将本书写成一本 Delphi 大全。在有限的篇幅内，如果指望将所有问题和内容都包含进来，肯定是不现实的；如果执意这样做，就会主次难分，重点不突出，反而讲不成精要了！

所以，本书重点是在讲原理以及实践这些原理时必须注意的重要技巧，因而没有分章分节讨论从这些原理发展起来的专题，如 COM、数据库、网络程序、多媒体开发。但如果很好地认识了这些原理，那么理解和掌握这些专题知识的问题，也就迎刃而解了！

### 1.2.2 本书是如何写作的

原理非常重要，掌握这些原理是实现 Delphi（乃至使用其他工具）开发能力质变的必要前提，因此你不要以为本书所讲的内容对你的工作学习并没有多大帮助。很多朋友可能会认为开发技巧最为重要，因此“50 开发实例”、“100 实现技巧”之类的内容和书籍大受欢迎；CSDN 上的一些朋友讨论本书内容时，也强烈要求再多写一些实例、技巧。但是你应该认识到，一切技巧都是从原理演变而来的，懂得一个原理可以举一反三地推论出 10 个技巧，而即使掌握了 100 个技巧也未必能推论出第 101 个技巧。

你不用担心本书的内容会非常难以理解。本书力求深入浅出，尽可能用浅显易懂、简洁明快的语言表达作者的意思。书中给出了大量代码片断和完整实例，并作了详尽注释。为了满足初学者和基础较差朋友的需要，在第 3 章、第 5 章讲述 Object Pascal 和 VCL 精要之前，还分别专列一章（第 2 章、第 4 章）介绍 Object Pascal 和 VCL 的入门知识。

对于喜欢开发技巧的朋友，本书也会让你满意。在第 2 章、第 3 章、第 4 章、第 5 章讲述 Object Pascal 和 VCL 知识的过程中，不仅穿插了大量开发技巧的讨论和实现代码，而且第 6 章、第 9 章、第 10 章还是专门讲具体开发的，特别是第 9 章。

如果你希望将本书作为一本 Delphi 的知识手册，那么第 5 章中的“VCL 消息大全”和整个第 8 章也能满足你的要求。

本书的所有源代码和组件全部包含于光盘中，可以在学习和工作中直接使用，但是请不要用于商业目的。

### 1.2.3 如何阅读本书

每个人都有不同的读书方式。有的人喜欢深夜躺在床上阅读，有的人喜欢坐在计算机前边读边实践，有的人喜欢坐在马桶上看书；有的开始时观其大略，需要时再仔细阅读，有的逐字研读，力求全盘吸收。

本书无法满足所有朋友的不同要求。如果你是在没有计算机的地方阅读本书，看到某些地方可能



希望自己动手实践，以验证我是否在故弄玄虚、大放厥词，这时候你可能有点光火——为什么不再花几行文字说详细点？而有些地方又可能让别的朋友觉得没必要讲得那么带劲——婆婆妈妈的，我都知道了，还说那么多！每当这个时候，希望你能消消气，因为我要照顾不同层次的读者。

因为是在讲精要、讲原理，所以一些地方未免显得有些艰深。不过没关系，先放一放往后面看，然后回过头来，很可能就豁然开朗了。Delphi 已经发展了 8 年（这还不包括非可视化的 Pascal 时代），8 年里，计算机技术日新月异。这么多年，Delphi 为全世界众多程序员推崇绝非偶然。经受了时间考验的东西肯定不是肤浅的，因此，希望在一朝一夕掌握整个 Delphi 的精要并不现实。

要完全掌握本书所讲的 Delphi 内容，对于不同基础的读者来说，所需要花费的时间和精力可能有较大的差别。对于高手来说，可能花一点时间翻翻就觉得没什么可看的了，有一定基础的需要几天至数周，而没有编程基础的 Delphi 初学者可能需要花上几月甚至更多的时间。