

金地时代 π

Delphi 4.x

程序设计入门 学得快

张 磊 编著



科学出版社

引 言

Delphi 是 Inprise 公司（原 Borland 公司）推出的 Windows 下快速开发应用程序的工具，它是一个少有的集强大的功能与快速的开发、极好的易用性为一体的 Windows 开发集成环境。我们在本书里要介绍的是 Delphi4 的使用。

在初学 Windows 编程的时候，我是从 Visual Basic 开始的，但是在使用过程中深深的感到 VB 实在不适于进行功能全面的 Windows 开发，于是在 VB 安装的程序又一次使别人的 Windows 崩溃后，我便毅然决然的放弃了 VB，转向 SDK 开发，希望能进行真正全方位的 Windows 程序设计。但是没过多久我又几乎要绝望了，用 SDK 开发 Windows 应用程序不仅要求程序员掌握近千个长长的函数名和各种莫名其妙的类型名称，更要求程序员读无数的技术文档，仔细体会各个概念之间的关系。即便如此，程序也会不时地死机，这种开发真可称得上是令人心力交瘁了。

后来我终于转向了 Delphi，还有 VC++，而且在长期的使用过程中，我越来越喜爱 Delphi 了。在 Delphi 中你可以把精力全部集中在自己程序的功能开发上，不用为每一个窗口的重画、每一个滚动条的翻动而操心。Delphi 提供的大量控件可使你能设计出界面精致的窗口，而这些在 VC 中实现则要花无数的代码调用 API 或者与可怕的 OCX（或 ActiveX）控件打交道。

Delphi 开发 Windows 程序有以下显著的优势：

（1）开发速度远远快于 VC++或 BC++，同时也绝不逊色于 VB 与 PowerBuilder，更别谈 JavaSDK 了，这在一切都高速运转的今天有着重大的意义。

（2）编译与运行速度仅比 C++稍逊一筹，比其他常见的开发工具做出来的程序都不慢，而且你要意识到 Windows 当初可是用大量 C 与少数汇编混合编的，用 C++自然会快一些。在另一方面，Delphi 使用的 Object Pascal 语言结构严谨，模块化好，它的基础是标准的教学语言 Pascal，虽说这样一来开发时写代码可能不够灵活，但是我们学起来会快一点，程序编译链接起来也会快一点，最终也补偿了一点速度，还减少了出错的机会。

（3）以控件为中心的开发充分体现了面向对象开发的思想，让人容易接受类与对象的概念，更快地掌握新的编程方法。而且每次出现新技术后，Inprise 公司可以及时把它体现在新加入的控件中，我们可以不必理会新技术的细节（当然粗略了解还是必须的），只需了解控件的属性、方法与事件响应就可以方便地使用控件来实现新功能了。

（4）Delphi 对底层 API 函数的包装非常好。在用 Delphi 编程序的时候，我们很少被迫直接与 Windows API 函数打交道，绝大部分功能只要调用 Delphi 控件的方法和 Delphi 写好的函数库中的函数就行了。而且在 Delphi 中调用 API 函数也很方便，Delphi 把每个 API 函数都放在了自己的库里，调用它们就像调用其他的库函数一样。

而 VC 编一个稍微要干些活的应用程序就不得不与 API 函数打交道，当然这样做可以使程序功能更强大，速度更快，代码更小，但是对于程序员来说就很不方便了。

(5) Delphi 的开放性好, 它为我们提供了一切源代码, 一直到汇编级别 (如多线程与消息处理中) 与 API 函数级, 包括用来开发与设置 Delphi 集成开发环境的函数库都提供了, 在编程序达到一定程度后, 你可以看看 Delphi 是怎样封装 Windows API 的, 可以去找出 Delphi 的 bug, 去修改 Delphi 控件的源程序, 去给 Delphi 集成开发环境加新的东西……在一般的开发工具中是很难做到这一点的。

(6) Delphi 对数据库开发支持很好。在 Delphi 中, 你可以开发单机数据库如 dBase, Access 等, 也可以开发 Oracle, Informix, Sybase, MS SQLServer 等大型网络数据库, 而且 Delphi 还提供了对新的多端 (Multi-Tier) 数据库编程的强大支持。

(7) 对某些跨平台的程序员来说, Delphi 是一个不错的选择。Delphi 不仅支持 Windows 下的 COM, OLE 与 ActiveX 编程, 同时也支持 CORBA, 而且还有专门的 AS/400 版本。

当然, Delphi 并非完美无缺, 它的 bug 不断, 补丁程序也不断, 它的帮助文件简单, 它的 EXE 文件太大等。但在总体上 Delphi 还是要远好于 VB; 然而比起 VC++ 来, 它的编译与运行速度还是稍微慢了一点, 它的功能还稍微弱了一点, 它对底层开发的支持还稍微少了一点……, 但是尽管如此, 我还是要向你推荐 Delphi——当今最顶尖的 Windows 开发工具之一。

在本书中我们将向你介绍 Delphi 编程的基本概念与方法, 力求把 Delphi 编程中常见的难点与重点讲述清楚, 让你能在最短的时间内快速成长为 Delphi 编程专家, 以最快的速度编制出界面美观、功能强大的 Windows 应用程序来。

本书第 1 章讲述了 Delphi 的历史与性能比较、Windows 编程与面向对象编程的思想以及如何安装与设置 Delphi4。

第 2 章介绍了 Delphi4 的开发环境、Delphi4 中控件的作用与功能, 还有 Delphi4 的文件类型, 并通过开发一个最简单的一个程序介绍 Delphi 程序开发的流程。

第 3 章是 Delphi4 的 Object Pascal 微缩教程。

第 4 章介绍如何利用 Delphi 开发一个 Windows 标准界面的应用程序。

第 5 章介绍在 Delphi 中进行绘图功能的开发, 这一章中讲到的控件是开发一个完整的 Windows 应用出现必用的, 如剪贴板操作、使用打印机、控件的漂泊功能、Owner Draw 控件的编程等等。

第 6 章的内容是 Delphi 最常用、为人们最称道的数据库编程。

第 7 章讲授在 Delphi 中的多媒体编程。

第 8 章介绍如何利用 Delphi 来做多线程程序以及 Delphi 中的异常处理。

第 9 章中介绍如何利用 Delphi 进行强大的网络编程。

附录 A 中有一些大家在开发实用程序时常遇到的问题的解决方法。

附录 B 中是有用的有关 Delphi 的网站地址。

一本学习 Windows 编程工具的书不可能面面俱到, 因为 Windows 的层次已经比较高了, 特别是我们用的是层次比 SDK 更高的 Delphi, 因此我希望读者能满足以下的条件再来读这本书:

(1) 至少要学过一些编程语言, 例如 C, Pascal, Basic 等, 可以是只学过而不精, 但至少要知道什么是变量、什么是函数、什么是函数调用等, 还应该知道源文件要

经过编译与链接后成为 EXE 文件，EXE 文件才能被操作系统调用运行。

(2) 应该用过 Windows 95 或 Windows 98，知道在屏幕上什么是窗口、什么是菜单，以及怎样关闭程序、怎样打开程序与文件。

由于个人水平有限，书中的错误在所难免，希望大家不吝赐教。

作 者

1999 年 4

目 录

第 1 章 预览 Delphi.....	1
1.1 强大快速的开发工具.....	1
1.1.1 Delphi 的历史	1
1.1.2 Delphi 4 性能评估	1
1.2 Windows 编程思想	2
1.2.1 事件驱动思想.....	2
1.2.2 面向对象的编程思想.....	3
1.2.3 面向对象的 Windows 编程	3
1.3 Delphi 的安装与设置	4
1.3.1 Delphi 光盘可安装组件	4
1.3.2 Delphi 组件安装路径设置	5
1.3.3 启动 Delphi 4	5
1.4 小结.....	6
第 2 章 一个最简单的程序	7
2.1 Delphi 集成开发环境	7
2.1.1 主菜单.....	8
2.1.2 工具栏.....	8
2.1.3 控件页.....	9
2.1.4 属性观察窗口.....	9
2.1.5 设计期窗口及其代码窗口	10
2.2 运行你的第一个 Delphi 程序.....	11
2.3 理解控件.....	14
2.3.1 什么是控件.....	14
2.3.2 控件的属性.....	14
2.3.3 控件的方法.....	16
2.3.4 控件的事件响应函数.....	16
2.3.5 控件的几个关键参数与属性.....	16
2.3.6 TForm 控件	19
2.4 Delphi 程序运行流程	20
2.4.1 Windows 的消息处理机制	20
2.4.2 类库开发中的消息处理与传递.....	21
2.4.3 Delphi 中的消息传递与处理	22
2.5 Delphi 文件类型	22

2.5.1	工程的概念	22
2.5.2	DPR 文件	23
2.5.3	DFM 文件	23
2.5.4	PAS 文件	25
2.5.5	RES 文件	26
2.5.6	DOF 文件	26
2.6	小结	27
第 3 章	Object Pascal 语法	28
3.1	常用数据类型	28
3.1.1	简单数据类型	28
3.1.2	字符串数据类型	30
3.1.3	复合数据类型	31
3.1.4	指针类型	33
3.1.5	Variant 类型	34
3.1.6	强制类型转化	35
3.1.7	常量声明	35
3.2	控制流与函数、过程	36
3.2.1	控制流语句	36
3.2.2	函数与过程	38
3.2.3	超前声明	39
3.2.4	过程类型	40
3.3	面向对象编程及类	41
3.3.1	类的声明与继承	41
3.3.2	类的可见性	41
3.3.3	类的成员函数	42
3.3.4	类的引用	43
3.4	Delphi 程序结构	44
3.4.1	了解 Program	44
3.4.2	什么是 Unit	45
3.4.3	Unit 间的引用	47
3.4.4	工程管理	48
3.4.5	调用 API 函数与 DLL 函数	49
3.5	规范的 Delphi 程序	50
3.5.1	前缀命名法	50
3.5.2	良好的程序风格	50
3.6	小结	51
第 4 章	标准 Windows 界面设计	52
4.1	做个标准菜单	52

4.1.1	TMainMenu 控件	53
4.1.2	TMenuItem 控件	53
4.1.3	菜单使用程序演示	55
4.1.4	TPopupMenu 控件	57
4.2	设计工具栏	57
4.2.1	工具栏控件	57
4.2.2	实例演示	58
4.2.3	TPageScroller 控件	60
4.2.4	ToolButton 控件	60
4.3	给工具栏与菜单加上图画	61
4.3.1	ImageList 控件使用方法	62
4.3.2	制作带图画的浮动菜单	62
4.4	什么是状态栏	65
4.4.1	状态栏控件	65
4.4.2	StatusPanel 的主要属性	66
4.4.3	Timer 控件简介	66
4.4.4	状态栏程序演示	66
4.5	让用户输入文字	67
4.5.1	Label 控件	68
4.5.2	Edit 控件	68
4.5.3	Edit 与 Label 控件使用程序演示	69
4.5.4	Memo 控件重要属性	70
4.5.5	Memo 控件程序演示	70
4.6	用对话框来交互	72
4.6.1	对话框控件简介	72
4.6.2	对话框通用使用方法	72
4.6.3	OpenDialog 与 SaveDialog 控件	73
4.6.4	OpenDialog 与 SaveDialog 使用程序演示	75
4.6.5	FontDialog 控件	76
4.6.6	ColorDialog 的主要属性	77
4.6.7	FindDialog 与 ReplaceDialog 的主要属性	77
4.6.8	FindDialog 与 ReplaceDialog 使用演示	78
4.7	可以做 Notepad 了!	80
4.7.1	程序功能简介与界面设计	80
4.7.2	控件初步设置	81
4.7.3	加入 ActionList 控件	82
4.7.4	ActionList 控件与菜单项联合使用	83
4.7.5	利用 Action 控件更新界面	84
4.7.6	系统定制的 Action	84

4.7.7 加上对话框.....	84
4.7.8 加上 About 窗口.....	85
4.8 小结.....	86
第 5 章 高级 Windows 界面设计	88
5.1 使用树形视图与列表视图.....	88
5.1.1 树形视图与列表视图的结构.....	88
5.1.2 树形视图控件.....	89
5.1.3 TTreeNode 控件.....	91
5.1.4 TTreeNode 控件.....	91
5.1.5 树形控件演示程序.....	92
5.1.6 列表视图控件.....	95
5.1.7 列表控件程序演示.....	97
5.2 在 Delphi 中绘图.....	99
5.2.1 认识 TCanvas.....	99
5.2.2 TCanvas 控件.....	99
5.2.3 图像对象 TPicture.....	102
5.2.4 TBitmap 控件.....	103
5.2.5 TImage 控件.....	103
5.2.6 TImage, TBitmap, TPicture 与 TCanvas 程序演示.....	104
5.2.7 操纵 WMF 文件.....	105
5.2.8 TPaintBox 控件.....	106
5.2.9 TPaintBox 程序演示.....	107
5.3 自己画的控件.....	107
5.3.1 什么是 Owner_Draw.....	107
5.3.2 一个 Owner_Draw 的状态栏.....	108
5.3.3 进程条控件.....	109
5.3.4 把进程条加到状态条上.....	109
5.4 让你的工具栏四处飘泊.....	111
5.4.1 什么是“飘泊”.....	111
5.4.2 怎样才能“飘”起来?.....	111
5.4.3 让 Toolbar“飘”起来.....	111
5.4.4 工具箱的飘泊.....	112
5.5 剪贴板的功能及其他.....	113
5.5.1 剪贴板功能.....	113
5.5.2 TClipboard 控件.....	114
5.5.3 剪贴板程序演示.....	115
5.5.4 应用程序实例 Application.....	116
5.5.5 Screen 对象.....	119

5.5.6	为程序添加自己的鼠标图案.....	119
5.5.7	操作打印机.....	120
5.6	让用户组合选择.....	122
5.6.1	TCheckBox 控件.....	122
5.6.2	TRadiobutton 控件.....	123
5.6.3	TGroupbox 与 TRadioGroup 控件.....	123
5.6.4	TCombobox 控件.....	124
5.7	浏览 JPG 图片.....	126
5.7.1	认识 TJPEGImage.....	126
5.7.2	TJPEGImage 控件.....	126
5.7.3	JPG 文件操作程序演示.....	128
5.8	小结.....	131
第 6 章	控制自己的数据库.....	132
6.1	BDE 会帮你攻克难关.....	132
6.1.1	为什么要 BDE.....	132
6.1.2	数据库程序设计步骤.....	133
6.1.3	初识 BDE Administrator.....	133
6.2	建自己的数据库.....	136
6.2.1	数据库基本知识.....	136
6.2.2	使用 Database Desktop.....	137
6.3	怎样联结数据.....	139
6.3.1	Delphi 中数据与程序的连接.....	139
6.3.2	DataSet 与 DataSource.....	139
6.3.3	Data Controls 控件.....	140
6.3.4	最简单的数据库程序.....	140
6.4	建立自己的数据库程序.....	142
6.4.1	限制用户的视野.....	142
6.4.2	操纵记录.....	145
6.4.3	查询数据库.....	148
6.4.4	更高的效率.....	152
6.4.5	使用 CachedUpdates.....	153
6.4.6	使用 TDatabase 控件.....	154
6.4.7	TQuery 控件简介.....	157
6.4.8	TQuery 控件程序演示.....	157
6.4.9	编写主从表程序.....	160
6.5	TTable 与 TQuery, 哪种更好, 哪个更快.....	162
6.5.1	基本性能比较.....	162
6.5.2	单机数据库查询比较.....	162

6.5.3	网络数据库的新增操作.....	165
6.5.4	提高程序效率总结.....	169
6.6	让你的数据变成图表.....	170
6.6.1	程序结构.....	170
6.6.2	决策体使用程序演示.....	171
6.7	完善你的数据库程序.....	172
6.7.1	使用 Datapump.....	172
6.7.2	使用 SQL Monitor.....	174
6.8	小结.....	174
第 7 章	更多的媒体, 更漂亮的媒体.....	176
7.1	自己编动画.....	176
7.1.1	最简单的动画原理.....	176
7.1.2	最简单的动画程序.....	177
7.1.3	改进的动画.....	178
7.1.4	动画特效实现演示.....	180
7.1.5	调色板操作.....	181
7.1.6	调色板程序演示.....	182
7.2	炫耀的封面窗口.....	185
7.2.1	静态炫耀窗口.....	185
7.2.2	动态炫耀窗口.....	186
7.3	强大的 TMediaPlayer.....	187
7.3.1	TAnimate 控件.....	187
7.3.2	TAnimate 控件使用程序演示.....	189
7.3.3	TMediaPlayer 控件简介.....	190
7.3.4	TMediaPlayer 控件.....	191
7.3.5	TMediaPlayer 使用程序演示.....	193
7.4	定做自己的多媒体程序.....	194
7.4.1	利用 TMediaPlayer 定制多媒体程序.....	194
7.4.2	改进的多媒体程序.....	196
7.4.3	使用更强大的函数.....	197
7.4.4	使用 MCI 程序演示.....	198
7.5	小结.....	200
第 8 章	创建多个线程.....	201
8.1	多线程.....	201
8.1.1	为什么要多线程.....	201
8.1.2	什么是多线程.....	201
8.1.3	注意事项.....	202
8.2	Delphi 中的多线程.....	202

8.2.1	TThread 线程对象	202
8.2.2	TThread 的属性	202
8.2.3	TThread 的方法	203
8.2.4	TThread 的事件响应函数	204
8.2.5	多线程程序演示	204
8.2.6	主窗口设计	207
8.3	更高级的线程	208
8.3.1	进一步讨论 Synchronize	208
8.3.2	更完善的同步控制	209
8.3.3	线程间的通信	209
8.3.4	利用事件	210
8.3.5	信号灯与共享内存	211
8.3.6	了解 THandleObject	211
8.4	拖下就放, 简单易行	213
8.4.1	在 Windows 中的拖放事件	213
8.4.2	与拖放相关的属性及方法	214
8.4.3	控件拖动程序演示	214
8.5	输入多个数据	216
8.5.1	TStringGrid 控件	216
8.5.2	格子控件程序演示	218
8.5.3	MPI 控件	220
8.5.4	TTabbedNoteBook 控件	220
8.5.5	TTabControl 与 TPageControl 控件	221
8.5.6	TTabControl 与 TPageControl 程序演示	223
8.6	我们要更健壮的程序	224
8.6.1	“异常”的概念	224
8.6.2	Delphi 中的异常处理语句	225
8.6.3	异常处理程序演示	227
8.7	小结	228
第 9 章	遨游与开发 Internet	230
9.1	你了解 Internet 吗?	230
9.1.1	网络分层	230
9.1.2	Socket 的概念	230
9.2	创建一个标准 Internet 程序	231
9.2.1	创建一个网页浏览器	231
9.2.2	THTML 控件	234
9.3	什么是 Socket 编程	235
9.3.1	Socket 编程的构架	235

9.3.2	非堵塞型 Socket 通信	236
9.3.3	非堵塞通信中的事件响应	238
9.3.4	TCustomWinSocket 控件	240
9.3.5	TServerWinSocket 与 TClientWinSocket 控件	241
9.3.6	堵塞式 Socket 通信	242
9.3.7	客户端线程的编写	243
9.3.8	服务器端线程的编写	244
9.3.9	TWinSocketStream	246
9.4	ActiveForm 编程	246
9.4.1	什么是 ActiveForm	246
9.4.2	创建一个 ActiveForm	247
9.4.3	在网页中发布 ActiveForm	249
9.5	小结	251

第1章 预览 Delphi

在本章中，我们将会讲一讲 Delphi 的特性，Windows 编程与 Delphi 编程的特点，还有我们应该怎么安装与设置 Delphi。

如果你是 Delphi 或是 Windows 编程的初学者，请你一定要把本章好好地读一读。只有你了解了 Delphi 与 Windows 编程的基础后我们才能继续下面关于 Delphi 的学习，不然在以后的学习与使用中你必然会倍感疑惑，乃至对 Windows 编程感到伤心，绝望与失落。即使你见过 Delphi，但是如果只是一知半解的话，也会对以后深入的开发感到困难的。在这一章中我们力争为以后的学习打下一个最底层的基础。

1.1 强大快速的开发工具

1.1.1 Delphi 的历史

Delphi 是古希腊雅典的神庙名称。Inprise 公司取这个名字的原因我们不得而知，可能意在表示 Delphi 的强大吧，因为甚至连著名的学者苏格拉底都曾经在 Delphi 神庙中表白自己的无知并要求神给以指点。

Windows 编程在初期全部是用 Windows SDK(Software Development Kit, 即软件开发包)做的，其困难程度就如同在 DOS 环境下用汇编语言调用 BIOS 与 DOS 中断编程一样，这种方法因此也曾被戏称为 Super Difficult Kit(超级困难包)。

Microsoft 的 Visual Basic 是市面上的第一个快速开发环境 (RAD, 即 Rapid Application Development)，在 VB 中程序员可以用现成的独立的模块来以搭积木的方式进行开发，在开发速度上 RAD 要远远快于传统的代码密集型开发工具如 VC、BC 等，在可维护性上也要好于它们，因为程序员不再需要顾及每一个细节，大部分底层的东西由“积木块”来保障，我们可以把焦点放在程序的独特功能、界面设计和优秀算法上，而不必担心与掌管每一个与 Windows 这个操作系统打交道的地方。而且在 RAD 中编程完全是“所见即所得”的，你在设计期看到的界面与运行时的一模一样，这样一来我们可以进行快速开发界面的任务。

在这种情况下，Inprise 公司推出了 Delphi，在它一开始出现的时候，人们就意识到了它是极强大的开发工具，在几年内 Inprise 公司又以很快的速度将 Delphi 由 1.0 升级到了 4.1，使其功能日趋完善，在开发速度与开发质量上都达到了一个新的水平。

1.1.2 Delphi 4 性能评估

每一种开发工具都有其适用的范围与局限性，为了了解 Delphi 与其他开发工具的差别，我们先来评估一下 Delphi 的性能，如果你发现 Delphi 并不适合你开发的程序，

那么现在后悔还来得及喽。

Delphi 4 有如下强大功能：

- 可以使用动态数组。
- 可以使用智能的动态代码检查，确保代码正确无误，使编程更快捷。
- 支持控件飘泊功能编程。
- 允许用户自定义数据控件。
- 拥有全方位的 ActiveX 编程的支持。

当然 Delphi 也有自己的弱点，我们在下面一一列出：

(1) Delphi 不擅长底层涉及硬件的开发，当然你可以用 Delphi 来开发硬件驱动程序等应用，但是 Delphi 在这上面的帮助是很少的。在因特网上的很多地方都可以找到 Delphi 的并串口通信的控件，但是现在还没有 Delphi 的硬件驱动控件。

(2) Delphi 的运行环节比较多，虽然它的速度远远快于 VB，PowerBuilder 等伪代码生成工具，但是如果你需要时间要求比较高的东西，如编游戏，我劝你还是用其他工具的好。当然，你可以用 Delphi 来编小游戏，速度还是可以的。

还有一点，我想可能要先提醒你一下，那就是 Delphi 的启动很慢，如果你很在乎这一点也可以再考虑一下。

下面我们对 Delphi,VC++与 Visual Basic 来做一个简单的比较与评估。见表 1.1 所示。

表 1.1 Delphi 与 VC, VB 的比较与评估。

	Delphi 4	Visual C++5	Visual Basic5
开发速度	极快	慢	极快
编译速度	较快	快	伪编译，很快
运行速度	快	快	慢
生成应用程序大小	大	很小	大
学习软件的速度	较快	很慢	快
硬件开发	无开发模块，困难	较难	无开发模块，困难
数据库开发	极强大	尚可	较好
应用程序可维护性	版本兼容性好	版本间兼容很不好	版本兼容性好
软件可维护性	可以以极大的自由开发，源代码全部开放	源代码开放少，对软件本身的开发困难	源代码开放少，对软件本身的开发困难
软件大小（带帮助文件）	约 170M	约 400M	约 100M

1.2 Windows 编程思想

1.2.1 事件驱动思想

在 DOS 下面我们的程序是直线型的，每一步我们都要求用户跟着程序走。比如说我们要求用户对某个任务作一些设置时，用户只能一步一个脚印的来，不能对设置错的选项反悔，只要按了回车符就已经定下来了。这种直线型的程序注定是单任务的，是不能允许用户在多个任务中进行切换的。因此在 Windows 中这种编程的思想必须改变

过来。

在 Windows 中程序是事件驱动的，或称是消息驱动的。事件就是指发生在操作系统中的事情，消息是指在一定的事件发生后 Windows 把这个事件告诉了你的应用程序。比如说用户在窗口 A 上面点击了一下鼠标的左键，外部设备于是把这个信息告诉了 Windows 操作系统，然后 Windows 发现在鼠标按下的地方的窗口是窗口 A，它就给窗口 A 发送了一个名为 WM_LBUTTONDOWN(消息的名字一般很容易认，比如这个消息就是 WindowsMessage LeftBUTTONDOWN 的缩写)的消息，窗口 A 所属的应用程序就可以选择是否对这个消息响应以及如何响应。如果 Windows 什么消息都不发给这个程序，此程序就会什么事都不干。

除了用户可以产生事件从而发出消息外，Windows 本身也会发一些消息，比如说 Windows 想关闭一个程序或窗口时就会给此程序或窗口发一条 WM_CLOSE 消息。有时候根据程序的需要我们还会自己定义一些消息在程序内部传递。

在 SDK 编程时代我们要手写代码把我们想要响应的消息与我们响应这个消息的函数联系起来，在 Delphi 中我们就不用担心了，这件事情是 Delphi 帮我们做的。

1.2.2 面向对象的编程思想

现在的软件越编越大，为了便于维护与开发软件，人们想了很多方法。最初我们要求程序模块化，后来软件业曾经作过努力把软件开发人工智能化，但是没有成功，不过在研究软件开发人工智能化的过程中大家发现了一种折衷的方案，就是面向对象的分析、设计与编程思想。

可以这么认为，我们为了把程序更好的模块化，把一组提供相近功能的函数与相关的数据集中在一起就形成了对象。

对象有外部接口与内部机理，我们只要知道了这个对象的接口，就可以调用这个对象的接口函数来完成某个任务，但是我们可以不用管这个对象是怎么实现此功能的，这就是对象的封装概念；除此以外，对象还有继承的概念，比如说“树”这个对象有子类“松树”，松树除了有树的共同特征外还有自己的特有属性，通过继承“树”这个对象，“松树”就拥有了“树”的一切函数与数据，可以少写很多代码，而且它可以改写一些“树”这个对象的函数与数据，并且添加一些函数与数据以形成自己的接口与内部功能。

对象就是一个一个的积木块，我们通过搭积木来编程序不仅快速，而且更容易查找错误，很容易把错误定在某个对象的上面。如果我们想为程序增加功能，我们可以从原来的对象中继承出新对象来实现新增的功能接口而不必从头开始。

在 Delphi 中一个对象的数据类型是类。

1.2.3 面向对象的 Windows 编程

在 Delphi 中我们充分利用了类的概念来编程，窗口中的每一个元素在 Delphi 的源代码中都是一个类，在以后的章节中这些元素我们称之为控件 (Component 或 Control)，比如说一个按钮是 TButton 控件，一个输入文字的框是 TEdit 控件。

每个控件都有三种对外的接口：属性、方法与事件。

- 通过设置控件的属性改变控件的大小、颜色等等，我们可以直接改动程序界面。
- 通过调用控件的方法，我们可以要求控件来实现一些功能。
- 通过响应事件我们可以对 Windows 发送给控件的消息进行响应。

我们在下一章中详细谈控件。

1.3 Delphi 的安装与设置

1.3.1 Delphi 光盘可安装组件

Delphi 的安装与其他的软件基本一样，把光盘放在光驱里面，光驱读盘会自动发现 Autorun 文件然后运行它的。当然了，你也可以直接双击光盘上的 Setup.exe 文件开始安装 Delphi 4。在安装界面中你可以看到光盘带了几个软件包，如图 1.1 所示。

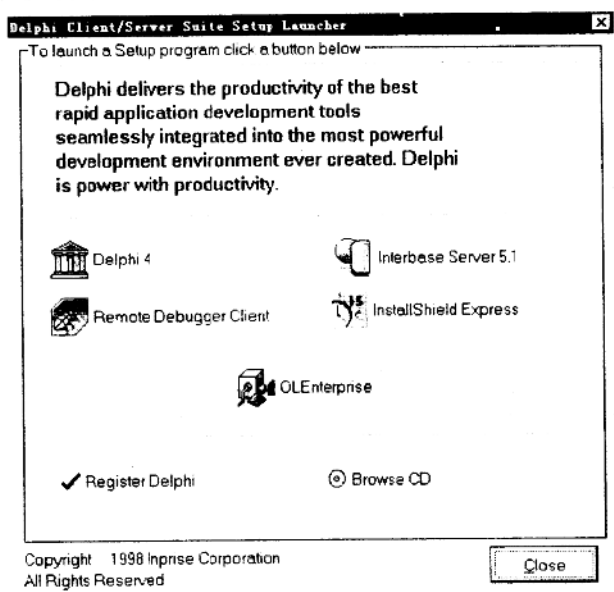


图 1.1 Delphi 安装界面

在上面可安装的五個软件包中 Delphi 4 是不用说了，InterBase Server 5.1 是 Inprise 公司的数据库，它的性能评估是不错的，但是应用不太普遍，Remote Debugger Client 是 Delphi 4 新加的功能，可以通过它进行远程代码调试，InstallShield Express 是 InstallShield 公司为 Delphi 专门做的一个安装 Delphi 程序的安装软件包，这个软件包使用是极其简单的，可以用它来安装你的程序，但是它的安装界面比较朴素，如果你想有一个比较漂亮的安装界面可以考虑使用 InstallShield 出的 InstallShield 的专业版，OLEEnterprise 是 Inprise 在 Delphi 4 中力推的利用 Microsoft 的 DCOM 构筑多端 (Multi Tier) 网络数据库程序的一个构架。

单击 Delphi 4 的图标，让我们直接来安装 Delphi 4。

1.3.2 Delphi 组件安装路径设置

在初始的 InstallShield 的准备工作之后, 接下来你要同意 Delphi 4 的版权协议书, 然后就会要求你确定把 Delphi 4 安装成什么样了, 如图 1.2 所示。

为了使以后能进行各种不同类型的开发, 一般情况下建议你整个 Delphi 都完全安装到硬盘上面, 就是在图 1.2 中选择缺省的 Typical 或是在此界面下直接按 Next 键到下一步去。

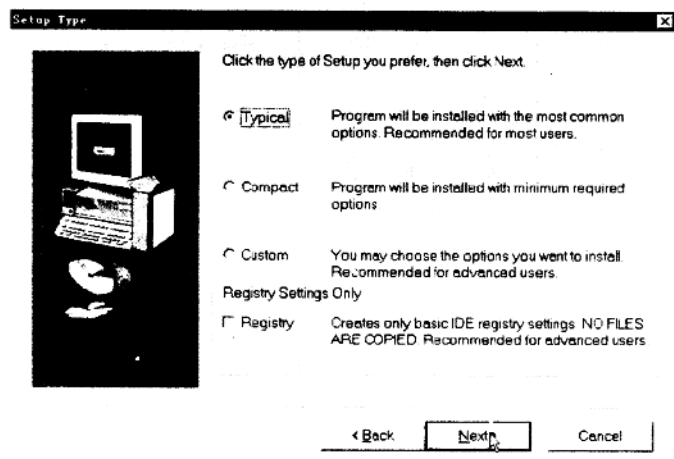


图 1.2 Delphi 安装模式选择

接下来 Delphi 会问你要不要安装 Interbase 数据库, 此数据库我们在前面已经提过, 不过你要注意这个数据库安装有一点问题, 如果你想卸掉它时会卸不干净, 必须要手工把注册表中有关 Interbase 的项目去掉。

再接下来 Delphi 会问你要把各个组件放在什么目录下面。一般软件总喜欢把自己放在系统盘的 \Program Files 目录下, 这使得系统盘的容量急剧减少, 所以一般我们都要把软件的路径改一改的。如果你想把各个组件放在不同的地方, 你可以手工把一个个目录改过来, 但是我劝你还是把它们放在一起好管理。如果这样的话, 你可以单击 Browse 按钮, 在抛出的对话框中选择好路径后所有的组件就都会被安装在新的目录下了。

以后的事情就很简单了, 你可以一直按 Next 按钮向下一步走, 最后就进入了安装 Delphi 的界面。

注意: 如果 Delphi 在安装过程中要你确认更新系统文件, 请你一定要更新, 不然有很多控件如新式的菜单等都会显示不正常的。

1.3.3 启动 Delphi 4

安装完后重新启动系统, 你将在开始菜单里看到 Delphi 4 的程序组, 单击 Delphi 4 快捷方式启动 Delphi 4, 如图 1.3 所示。