

Delphi 4 编程技术内幕

(美) Charlie Calvert 著

潇湘工作室 译



机械工业出版社
China Machine Press

本书完整而全面地介绍了Delphi 4。其主要内容有COM和分布式计算、数据库编程、常规编程问题以及图形和DirectX 编程。通过本书的学习,读者不仅可以快速了解Delphi 4编程的要旨,还可以掌握许多新技术,如创建COM自动化服务器和客户程序、使用MIDAS进行多层数据库计算、使用DCOM跨网络分布数据库等。

本书最大的特点是代码丰富,不仅书中详细介绍了代码样例,配套光盘上还包括了许多书中没有介绍的源程序。本书适用于使用Delphi 4 编程的程序员。

Charlie Calvert: Charlie Calvert's Delphi 4 Unleashed

Authorized translation from the English language edition published by Sams Publishing, an imprint of Macmillan Computer Publishing U.S.A.

Copyright 1999 by Sams Publishing.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Chinese Simplified language edition published by China Machine Press.

Copyright 1999 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权所有,翻印必究。

· 本书版权登记号: 图字: 01-99-0034

图书在版编目(CIP) 数据

Delphi 4编程技术内幕 / (美) 卡尔维特 (Calvert, C.) 著; 潇湘工作室译.
— 北京: 机械工业出版社, 1999.6

书名原文: Charlie Calvert's Delphi 4 Unleashed

ISBN 7-111-07216-2

I. D… II. ①卡… ②潇… III. 程序语言, Delphi 4—程序设计 IV. TP312

中国版本图书馆CIP 数据核字(1999) 第10944号

出 版 人: 马九荣 (北京市百万庄大街22号 邮政编码100037)

责任编辑: 陈剑瓯

北京市昌平环球印刷厂印刷 · 新华书店北京发行所发行

1999年6月第1版第1次印刷

787mm × 1092mm 1/16 · 62.5 印张

印数: 0 001-6 000 册

定价: 110.00 元 (附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

编程相当有趣，而且极其富于挑战性。有时，程序员会聚在一起，将自己百思不解的问题和别人讨论，可能突然之间，别人一句不经意的话就让你茅塞顿开：“原来如此！原来我为之苦恼、为之夜不能寐的问题就这样简单地解决了！”

本书就像一个富有经验的高级程序员，他慷慨大方，循循善诱，一点也不保留地把自己常年积累的经验和理论知识告诉你，使你豁然开朗，从此，你的编程生涯又开始了新的篇章。

“阅读一本好书可以给头脑充电”。的确如此。

本书的作者是使用Delphi的行家里手。在本书中，他将Delphi的使用与自己的实践经验紧密地结合起来，这里不仅有可以帮助你提高技术水平的理论知识，还有大量丰富的代码实例。有人说，“这本书太好了，像一个金矿一样”，真的，它的确不虚此名。它介绍有关Delphi编程的中高级内容，主要讨论COM和分布计算、数据库编程、一般编程问题以及图形和DirectX编程。

本书采取了描述性的方式，用作者的话说，“读者可以坐在一张舒服的椅子上，躺在床上，泡在浴池里，或者甚至是在海滩上晒太阳时，安静地进行阅读”。

让我们来简要列一下阅读本书的三大好处：

- 学习和充实自己面向对象编程技术的理论知识。
- 快速而轻松地学会如何进行程序设计。
- 免费获得大量实用性的代码。除了书中给出的以外，在配套光盘中还有许多代码实例。

在看本书时，应当在学习了理论知识之后，在计算机前试一试范例程序，最后根据学到的知识创建自己的程序。也就是把阅读文字和运行程序结合起来。

Delphi是非常优秀的编程语言，尽管它在竞争激烈的编程语言中属于后起之秀，但自从它诞生以来，便逐渐赢得了许多程序员的喜爱。举个例子来说吧，在参与本书翻译的人员中，就有曾经使用Delphi进行课题设计的清华大学的博士生，以及使用它进行项目开发的科技人员。

最后要说明的一点是，原书由于相当厚，所以，有些章节（第3、4、5、8、9章和第11.6节）没有在原书的正文中出现，而是包含在本书附带的光盘中。本书编辑考虑到国内的情况，为了使读者能够更完整地阅读本书，要求我们将上述章节也翻译出来并加到了书的正文中。

由于本书翻译工作量大、时间紧迫，如有错误和疏漏之处，请读者原谅并提出宝贵意见。

潇湘工作室 1999年3月于中关村

与原出版社联系方法

传真： 317-817-7070

Email: prog@mcp.com

通信地址： Brian Gill Executive Editor Programming and Borland Press team
Macmillan Computer Publishing
201 West 103rd Street
Indianapolis, IN 46290 USA

目 录

译者序

第一部分 入门

第1章 程序设计基础	1
1.1 本书的结构	2
1.2 范例程序	4
1.3 创建设计良好的程序	4
1.3.1 设计问题: 书写简单的代码	4
1.3.2 螺旋式上升	5
1.3.3 好OOP与差OOP的对比	5
1.3.4 注意灵活的事件处理程序	6
1.3.5 感到懒于理清程序的环节	7
1.3.6 避免使用隐藏数据交叉链接 对象	8
1.3.7 简短的方法	9
1.3.8 变量、方法以及过程的命名	10
1.3.9 避免特征漂移	10
1.3.10 创建组件	10
1.3.11 使用第三方工具	11
1.4 本书正文的注意事项	12
1.4.1 Unleash和Merc40包	12
1.4.2 CodeBox单元	13
1.4.3 建立网络	13
1.4.4 Borland/Inprise的名称变化	13
1.4.5 注释代码	14
1.5 总结	14
第2章 IDE和VCL增强	15
2.1 可停放的工具栏	15
2.2 使用工具窗口	16
2.3 使用Code Explorer	18
2.4 Delphi 4中的新工具	21
2.4.1 使用Project Manager和调试器	21
2.4.2 调试DLL	22
2.4.3 使用模块视图	23

2.4.4 使用事件日志窗口	23
2.4.5 CPU视图	24
2.4.5 调试器中的新特征	25
2.4.6 Code Browser和Tooltip Symbol Insight	26
2.5 停靠窗口	26
2.5.1 在窗体中停放控件	27
2.5.2 在窗体中停放窗体	28
2.6 动作列表	31
2.6.1 使用动作列表	31
2.6.2 使用标准动作	35
2.7 函数重载	35
2.7.1 方法重载和继承	38
2.7.2 使用方法重载时要避免的错误	40
2.7.3 默认参数	42
2.8 动态数组	43
2.9 implements关键字	46
2.10 总结	48
第3章 多态性	49
3.1 多态性的概念	49
3.1.1 赋值和等于的比较	49
3.1.2 多态性的简单例子	50
3.2 多态性赋值能力	51
3.2.1 设置子对象适合于它们的父 对象	51
3.2.2 多态性的另一面	52
3.3 虚方法和多态性	53
3.3.1 经典的形状演示	56
3.3.2 没有多态性的代码	62
3.4 VCL中的多态性	63
3.5 总结	65
第4章 异常处理	66
4.1 本章如何处理异常	67
4.2 异常的理论	67

4.2.1 异常的功能	67
4.2.2 默认异常处理程序	68
4.2.3 整个程序是一个大try块	69
4.2.4 未处理异常引起的恐慌	69
4.3 异常类	70
4.4 基本异常句法	71
4.4.1 一个简单的try..except模块	74
4.4.2 异常的关联错误字符串	75
4.4.3 找到错误的地址	76
4.4.4 异常处理总结	77
4.5 引发异常	77
4.5.1 引发VCL异常类	79
4.5.2 理解VCL异常类	81
4.5.3 运行Gauntlet: 多异常类	82
4.5.4 在常规except语句中捕获异常	83
4.6 创建和引发自己的异常	83
4.7 再引发异常	87
4.8 try..finally块	88
4.9 流、异常和空闲空间	91
4.10 替换默认的异常处理程序	95
4.11 使用资源跟踪错误字符串	97
4.12 总结	101
第5章 线程	102
5.1 进程和内存	103
5.1.1 线程与多任务	104
5.1.2 抢占式与非抢占式多任务	106
5.2 线程: 一个简单的例子	106
5.3 多线程程序	111
5.3.1 在Thread2程序中使用线程	115
5.3.2 向线程函数传递一个参数	115
5.3.3 临界区: 让多个线程同时工作	117
5.4 处理互斥元	122
5.5 线程和VCL	126
5.6 线程安全数据库访问	127
5.7 VCL线程安全对象	128
5.8 使用TThread对象	128
5.9 设置线程的优先级	137
5.10 重访Delphi线程	142
5.11 总结	143

第二部分 创建组件

第6章 创建组件(一)	145
6.1 组件原理	146
6.2 创建现存组件的派生组件	146
6.2.1 创建简单的组件	146
6.2.2 双向工具	150
6.2.3 注册组件	151
6.2.4 改变组件的默认行为	151
6.2.5 动态创建并检查组件	152
6.3 包: 将组件放入Component Palette 中	153
6.3.1 什么是包	153
6.3.2 创建包	154
6.4 扩展Unleash单元	156
6.4.1 TCCBigEdit 和TCCEmptyPanel 控件	166
6.4.2 创建复合组件	166
6.4.3 创建组件的published属性	169
6.4.4 流属性的注意事项	170
6.5 创建真正有用的组件	171
6.6 总结	175
第7章 创建组件: (二)	177
7.1 从头开始建立组件	178
7.2 时钟组件	180
7.2.1 了解TCCClock	189
7.2.2 TCCColorClock	191
7.2.3 创建Fancy Clock	192
7.3 为组件创建图标	195
7.4 Tools API: Property Editor和 Component Editor	196
7.4.1 五个主要的Tools API	196
7.4.2 Property Editor	196
7.4.3 关于注册你自己的组件以及 Component Editor的更多知识	199
7.4 Component Editor	200
7.5 不可视的组件	201
7.6 FindAllW 程序	202
7.6.1 使用TFileIterator遍历目录	213

7.6.2 什么时候应该把对象做成不可见 组件	215	8.6.1 在运行中控制DBGrid的Options 属性	263
7.6.3 AllDirs单元和FileIter单元	216	8.6.2 运行时显示DBGrid选项	264
7.6.4 遍历目录	216	8.7 使用TDBGrid列	265
7.6.5 使用FindFirst、FindNext和 FindClose	217	8.7.1 改变TDBGrid对象的标题	265
7.6.6 创建事件处理程序	218	8.7.2 改变网格里的整列	266
7.7 CompDirs程序	220	8.7.3 改变网格里一行的颜色	267
7.8 TCCPickDirDlg	226	8.7.4 改变列宽	268
7.9 组件模板	231	8.7.5 在TDBGrid里隐藏列	268
7.10 ActiveX控件	232	8.7.6 显示可见字段的列表	269
7.11 总结	235	8.7.7 运行时移动列	270
第三部分 数据库编程		8.8 查找字段	270
第8章 字段和数据库工具	237	8.9 多记录对象	276
8.1 从关系数据库开始	238	8.10 总结	276
8.1.1 关系数据库模型的优势	239	第9章 平面文件现实世界数据库	278
8.1.2 简单的逻辑设置: 关系数据库的 基础	239	9.1 本章示例程序	279
8.1.3 DBD和Database Explorer中的索引 和关键字	240	9.2 设计应用程序	279
8.2 重点规则: 给每个表建立一个关键字	243	9.2.1 定义数据	280
8.2.1 一对多的关系: 数据和索引	244	9.2.2 程序设计的焦点	282
8.2.2 用主关键字和主索引工作	245	9.3 定义程序的外观	284
8.2.3 用副索引和外部关键字工作	246	9.4 查找: 处理Category字段	286
8.2.4 关键字的重要性	246	9.5 设置程序的命令结构	287
8.3 用数据库工具浏览和操作数据库	247	9.6 查看应用程序的“粗略草稿”	289
8.3.1 浏览DBDEMOS数据库的关键字 和索引	247	9.7 创建自己的对象	290
8.3.2 用Database Desktop创建索引	250	9.8 创建最终的程序	291
8.3.3 创建别名	250	9.9 切换进和切换出只读模式	309
8.3.4 使用Database Desktop创建主关键字 和外部关键字	251	9.10 排序数据	311
8.3.5 探究引用完整性	252	9.11 查询数据	313
8.3.6 关系数据库和联结	253	9.12 筛选数据	314
8.3.7 联结和QBE	254	9.13 标记文件	315
8.4 Fields Editor	255	9.14 设置颜色	317
8.5 计算字段	259	9.15 使用注册表	320
8.6 运行期间的TDBGrid	260	9.16 剪贴板: 剪切、复制和粘贴	323
		9.17 总结	323
		第10章 关系式数据库	324
		10.1 现实世界的数据库	324
		10.2 测试关联地址程序	325
		10.3 kdAdd代码	328
		10.3.1 使用kdAdd程序	338

10.3.2 为kdAdd建立索引	338	第12章 InterBase编程	390
10.3.3 查询记录	339	12.1 关于Music 程序	391
10.3.4 插入数据和取消操作	340	12.2 设计Music 程序	392
10.3.5 删除数据: 程序的数据模块 初览	341	12.2.1 创建数据库方案	393
10.3.6 数据模块: 级联删除	343	12.2.2 修改表: 空或非空	393
10.3.7 数据模块: 总体登记	345	12.2.3 重新命名现有表中的字段	394
10.3.8 在字符串资源中加入错误字符串	345	12.2.4 创建Blob字段	395
10.3.9 About框	346	12.2.5 主关键字和外部关键字	396
10.4 使用kdAddExplore程序	346	12.3 在Music 表创建索引	398
10.4.1 使用TSession	347	12.4 生成器、触发器和存储过程	399
10.4.2 kdAddExplore程序的代码	349	12.4.1 不要在活动索引上使用触发器	400
10.4.3 用TTreeView显示系统数据库	355	12.4.2 使用存储过程	401
10.4.4 展开TTreeView节点	356	12.4.3 来自Delphi端的存储过程	402
10.4.5 为TTreeView加入可用表的 列表	356	12.5 服务器端规则和客户端规则	403
10.4.6 找到索引和字段	358	12.6 Music 程序界面概述	404
10.4.7 显示别名和别名参数	359	12.7 使用子窗体	405
10.5 总结	360	12.8 Music 程序的代码	408
第11章 使用本地InterBase服务器	361	12.8.1 使用Music 程序	408
11.1 从InterBase开始	361	12.8.2 禁止密码: TdataBase对象	408
11.1.1 数据库和工作市场	362	12.8.3 FormCreate事件	408
11.1.2 设置本地InterBase	362	12.9 使用Music程序中的数据	411
11.1.3 设置InterBase别名	365	12.10 向数据库提出一个问题: 简短的 SQL入门	412
11.2 创建数据库	366	12.10.1 不返回数据集合的存储过程	416
11.2.1 使用WISQL	367	12.10.2 查看数据范围	417
11.2.2 创建表	367	12.11 总结	418
11.3 利用WISQL管理数据库	368		
11.4 事务处理	370		
11.4.1 创建TRANSACT程序	370		
11.4.2 使用TRANSACT程序	371		
11.4.3 理解事务处理	372		
11.5 高速缓存更新	373		
11.6 多对多关系	378		
11.7 安全和InterBase Server Manager	387		
11.7.1 定义表的访问权力	388		
11.7.2 利用InterBase Server Manager 备份表	388		
11.8 总结	389		
		第四部分 COM	
		第13章 COM接口和基础	419
		13.1 理解COM接口	419
		13.1.1 COM是什么?	420
		13.1.2 COM和窗口	420
		13.1.3 即插即用应用程序	420
		13.1.4 COM的问题	421
		13.1.5 UNIX上的COM	422
		13.1.6 COM+和COM的未来	423
		13.2 接口介绍	423
		13.2.1 新接口类型	423
		13.2.2 接口不是类	424

13.2.3 不能直接实现接口	424	14.3.4 获取接口信息	494
13.2.4 利用类实现接口	425	14.4 总结	495
13.2.5 调用接口的方法	428	第15章 创建COM自动化服务器和	
13.2.6 破坏接口	429	客户程序	496
13.3 接口背后的理论	432	15.1 理解OLE自动化客户程序和	
13.3.1 使用接口的原因	433	服务器	496
13.3.2 维护或更新接口	435	15.2 建立简单的COM服务器和客户	
13.4 创建和使用COM接口	435	程序	497
13.4.1 理解GUID	436	15.2.1 建造简单的服务器	497
13.4.2 关于注册表和GUID所有内容	437	15.2.2 在服务器上添加方法	498
13.4.3 理解IUnknown	440	15.2.3 查看生成的源代码	499
13.4.4 实现IUnknown	445	15.2.4 添加接口属性	500
13.4.5 为接口指定GUID	447	15.2.5 建立简单客户程序	500
13.4.6 幻数	448	15.2.6 通过TCOMInterface来调用	
13.4.7 UnknownByUser程序的注释	449	方法	504
13.5 高级接口问题	450	15.2.7 通过dispinterface调用方法	505
13.5.1 多个接口	451	15.2.8 通过Variant调用方法	506
13.5.2 方法解析语句	454	15.3 IDispatch、双重接口和	
13.5.3 Implements伪指令	455	dispinterface	507
13.5.4 虚方法表	458	15.3.1 从Word调用SimpleAutoServer	508
13.6 总结	458	15.3.2 VTable和双重接口	509
第14章 TComObject、TTypedComObject		15.4 类型库	510
以及类型库	460	15.4.1 什么是类型库	511
14.1 样例程序	460	15.4.2 创建类型库	511
14.2 TComObject	460	15.4.2 有效的自动化类型	512
14.2.1 在DLL中放置普通接口	461	15.5 用EmptyParam忽略参数	512
14.2.2 IClassFactory	469	15.6 注册	513
14.2.3 ISystemInfo接口	470	15.7 自动编组数据	515
14.2.4 注册DLL	470	15.8 从一个CoClass得到两个或更多	
14.2.5 从可执行文件访问存储在DLL		的接口	515
中的COM对象	471	15.9 总结	526
14.2.6 越过程序边界编组数据	472	第16章 用Delphi完成Word和Excel	
14.2.7 编组数据	473	自动化	527
14.2.8 Variant数组	473	16.1 自动化Office应用程序的系统要求	527
14.2.9 调用ISystemInfo数据	479	16.2 启动Delphi和Excel	528
14.3 类型库和查询对象	482	16.2.1 启动Excel	528
14.3.1 创建类型库	483	16.2.2 使用COMObj	529
14.3.2 创建查询类型库的客户程序	488	16.2.3 了解Excel的简单自动化	530
14.3.3 检索ITypeInfo	491	16.2.4 比较接口和Variant	530

16.2.5 理解CreateOleObject	531	18.3 WebBroker组件	591
16.2.6 Variant和类型	532	18.3.1 TWebModule	592
16.2.7 创建Excel自动对象	532	18.3.2 TWebResponse	593
16.3 了解Excel 自动化对象	533	18.3.3 TWebRequest	593
16.3.1 在Application对象内访问各种 对象	534	18.4 IntraBob	594
16.3.2 使用自动化	535	18.5 TPageProducer	596
16.4 在Excel和Word间共享图表	543	18.6 TDataSetPageProducer	598
16.4.1 创建电子表格	546	18.7 浏览状态	601
16.4.2 创建图表	547	18.7.1 Fat URL	602
16.4.3 使用SeriesCollection对象	548	18.7.2 Cookie	602
16.5 从Excel向Word复制数据	550	18.7.3 隐含字段	602
16.5.1 Word内部的自动化	551	18.8 高级网页的生成	605
16.5.2 发送文档	553	18.9 TDataSetTableProducer	610
16.6 总结	553	18.10 Database Web Application Wizard	613
第五部分 Internet 编程		18.11 TQueryTableProducer	613
第17章 接口和Internet Explorer	555	18.12 总结	617
17.1 Variant与接口的对比	555	第19章 WinINet和FTP	618
17.1.1 补救接口	556	19.1 理解WinINet、FTP和TCP/IP	618
17.1.2 使用类型库	557	19.1.1 所需文件	618
17.2 本章所需文件	557	19.1.2 确保FTP在系统中运行	619
17.3 使用Word和Excel	558	19.1.3 安装TCP/IP应该注意的问题	619
17.3.1 使用Excel 和Word接口	565	19.2 FTP使用WinINet	622
17.3.2 比较Variant和接口技术	566	19.2.1 使用InternetOpen	624
17.4 使用IE和TwebBrowser	568	19.2.2 使用InternetConnect	624
17.4.1 开始: 访问Web Page中的 元素	568	19.2.3 获得当前目录	626
17.4.2 运行时在浏览器上处理 HTML	574	19.2.4 在目录中查找文件(一)	626
17.5 在浏览器内放置ActiveX控件	581	19.2.5 在目录中查找文件(二)	628
17.6 总结	588	19.2.6 检索文件(一)	630
第18章 WebBroker: CGI和ISAPI	589	19.2.7 创建回调	631
18.1 Web Module	589	19.2.8 检索文件(二)	635
18.2 Web Server Application Wizard	589	19.2.9 发送文件到服务器的简单方法	637
18.2.1 CGI	590	19.2.10 删除文件	637
18.2.2 WinCGI	590	19.2.11 创建和删除目录	638
18.2.3 ISAPI/NSAPI	591	19.3 在程序中使用FTP控件	639
18.2.4 CGI与ISAPI	591	19.3.1 连接服务器	647
		19.3.2 响应OnNewDir事件	648
		19.3.3 改变目录	648
		19.3.4 跟踪文件传送	649
		19.4 总结	651

第六部分 分布式编程

第20章 DCOM653

20.1 COM和分布式体系结构653

20.1.1 什么是DCOM653

20.1.2 在Windows 95/98和Windows NT 上运行DCOM服务器654

20.1.3 作为DCOM服务器使用 Windows 98656

20.2 简单的DCOM客户程序和服务器656

20.2.1 建立服务器657

20.2.2 理解Safecall657

20.2.3 机器之间编组数据658

20.2.4 创建DCOM客户程序662

20.2.5 使用远程对象670

20.3 再访问SystemInformation程序670

20.4 轻量远程数据集677

20.4.1 理解DataCOMServer程序685

20.4.2 访问远程数据集的 DataCOMClient689

20.4.3 理解 DataCOMClient694

20.4.4 编辑数据行696

20.5 程序数据和自动化数据697

20.6 测试远程调试系统702

20.7 总结703

第21章 MIDAS704

21.1 多层计算概述704

21.1.1 什么是MIDAS705

21.1.2 分布式计算的目的706

21.1.3 公文包模型概述706

21.1.4 术语707

21.1.5 查找调度程序707

21.2 技术细节: 使用分布式数据集708

21.2.1 建立简单的MIDAS应用程序709

21.2.2 创建 SimpleMidasServer714

21.2.3 理解服务器715

21.2.4 导出TProvider接口715

21.2.5 为客户应用程序准备服务器716

21.2.6 创建简单的MIDAS客户程序717

21.2.7 理解简单的MIDAS客户程序718

21.2.8 远程访问服务器718

21.2.9 使用套接字代替DCOM719

21.3 建立一对多应用程序720

21.4 建立远程数据客户应用程序722

21.4.1 更新和刷新数据725

21.4.2 公文包模型726

21.4.3 PacketRecords727

21.5 错误处理728

21.6 服务器端逻辑与客户端逻辑729

21.7 所需的DLL731

21.8 使用OLEEnterprise732

21.9 总结733

第22章 ActiveForm735

22.1 介绍ActiveForm735

22.2 建立ActiveForm736

22.3 配置在Internet Explorer里使用的 ActiveForm736

22.3.1 为ActiveForm设置选项736

22.3.2 连接到ActiveForm上737

22.3.3 了解OCCACHE738

22.4 使用CAB文件和包739

22.4.1 理解INF文件741

22.4.2 许可证问题742

22.5 在Word中运行ActiveForm742

22.5.1 注册应用程序742

22.5.2 在Microsoft Word里运行窗体743

22.6 使用ActiveForm模板745

22.7 套接字和ActiveForm745

22.7.1 套接字概述745

22.7.2 建立套接字支持的 MIDAS ActiveForm746

22.7.3 创建基于MIDAS的ActiveForm747

22.8 总结749

第23章 MTS750

23.1 什么是MTS750

23.2 如何安装MTS752

23.3 创建简单的MTS服务器752

23.4 创建简单的MTS客户程序758

23.5	调用本地对象	760
23.6	用MTS注册对象	761
23.7	远程调用对象	761
23.8	创建简单的MTS数据库服务器	763
23.9	创建简单的MTS数据库客户程序	770
23.10	处理MTS事务	774
23.11	建立跨越多个服务器的MTS示例	784
23.12	Rocket程序的代码	786
23.13	总结	799
第24章	CORBA	800
24.1	理解CORBA	800
24.1.1	什么是CORBA	800
24.1.2	比较DCOM和CORBA	801
24.1.3	什么是ORB	802
24.1.4	什么是Smart Agent	803
24.1.5	什么是BOA	804
24.1.6	什么是代理、存根和框架	805
24.1.7	理解Delphi和CORBA	807
24.2	两个重要的CORBA存储库	808
24.2.1	什么是Interface Repository	808
24.2.2	什么是Implementation Repository	809
24.3	简单的CORBA对象	809
24.3.1	服务器	810
24.3.2	客户程序	817
24.3.3	理解客户程序	819
24.3.4	理解CORBA存根	822
24.3.5	理解CORBA框架	822
24.4	理解服务器的IDL接口	823
24.5	简单的CORBA动态项目	824
24.6	简单的CORBA数据库对象	831
24.6.1	创建CORBA数据库服务器	831
24.6.2	创建客户程序	840
24.7	利用VisiBroker Manager和OSFind	842
24.8	总结	843
第25章	Delphi、Java和C++	844
25.1	在Delphi和Java中使用CORBA	844
25.2	从Delphi调用Java服务器	851
25.3	从C++Builder调用Delphi CORBA 服务器	856

25.3.1	用C++建立CORBA服务器并从 Delphi调用	866
25.3.2	理解BCBCorbaserver	871
25.4	总结	873

第七部分 游戏编程

第26章	VCL图形类	875
26.1	Windows GDI与VCL TCanvas	876
26.1.1	理解GDI	876
26.1.2	VCL使图形编程更容易	877
26.2	使用TCanvas	877
26.3	研究与使用画刷	878
26.3.1	TBrush.Color属性分析	878
26.3.2	TBrush.Style属性分析	878
26.3.3	TBrush.Assign方法分析	880
26.4	研究与使用画笔	881
26.4.1	TPen.Style属性分析	881
26.4.2	TPen.Width属性分析	881
26.4.3	TPen.Mode属性分析	881
26.5	研究与使用字体	883
26.5.1	TFont.Height属性分析	883
26.5.2	TFont.Size属性分析	883
26.5.3	TFont.Name属性分析	884
26.5.4	TFont.Style属性分析	884
26.6	研究与使用TCanvas.PenPos	884
26.7	研究与使用TCanvas.ClipRect	884
26.8	研究与使用TCanvas.CopyMode	885
26.9	研究与使用TCanvas.Pixels	886
26.10	分析与理解TCanvas方法	886
26.11	MainBrush范例代码	886
26.11.1	OnCreate事件	895
26.11.2	FormShow方法	896
26.11.3	设置画笔模式	897
26.11.4	使用Fonts	898
26.11.5	在画布上画出形状	899
26.11.6	使用CopyMode	902
26.11.7	修改ClipRect	902
26.12	其他名称的线条	903
26.13	深入研究VCL	906

26.13.1 深入了解TColor	906	27.3.5 释放内存	939
26.13.2 Colors样本项目	907	27.4 平滑动画	940
26.13.3 OnChange事件处理程序.....	909	27.5 使用位图	949
26.14 调色板	910	27.6 游戏资源	955
26.15 深入了解Windows的图形文件 格式	910	27.7 总结	956
26.15.1 分析位图和DIB	910	第28章 更多的DirectX技术	957
26.15.2 分析图标	911	28.1 DirectSound	957
26.15.3 分析元文件	912	28.1.1 初始化DirectSound	958
26.16 分析TPaintBox和TImage	915	28.1.2 枚举声音驱动程序	961
26.17 再谈位图	916	28.1.3 使用DirectSoundBuffer.....	962
26.18 总结	923	28.1.4 打开WAV文件	964
第27章 DirectDraw	924	28.1.5 播放声音文件	966
27.1 理解DirectDraw	924	28.1.6 播放大文件	967
27.1.1 什么是DirectDraw.....	925	28.1.7 一次播放两个WAV文件	967
27.1.2 关于双缓冲的几点想法	926	28.2 创建DirectX游戏.....	968
27.1.3 硬件与仿真	927	28.2.1 创建DelphiMan程序	970
27.2 一个简单的DirectDraw程序	928	28.2.2 实现DelphiMan程序的注意 事项	979
27.3 释放接口	934	28.2.3 “机器人物”的控制	982
27.3.1 进入独占模式和改变屏幕 分辨率	935	28.2.4 小结DelphiMan程序	982
27.3.2 DirectDraw表面是什么.....	935	28.3 浏览Direct3D	982
27.3.3 向屏幕写文本	937	28.4 关键的3D技术	983
27.3.4 交换表面	938	28.5 总结	984
		配套光盘内容	986

第一部分 入门

主要内容:

- 程序设计基础
- IDE和VCL增强
- 多态性
- 异常处理
- 线程

第1章 程序设计基础

主要内容:

- 本书结构。
- 范例程序。
- 创建设计良好的程序。
- 本书正文的注意事项。

欢迎使用本书。本书的主要内容是介绍如何使用世界上最优秀的编程语言Object Pascal来进行程序设计。体现Pascal语言环境的就是Delphi，它是当今最优秀的编程工具。

对于Delphi的初学者来说，在看到上面一段话时可能会感到惊讶。当然，C++和Java也相当不错，其他的工具（例如Eiffel和Smalltalk）也非常吸引人。但是Delphi在编程语言的世界中占据了独一无二的位置。自从Delphi的第一个半成品从设计开发人员的桌面进入Borland的网络时起，我就一直对该产品情有独钟。

其他的工具也是令人钦佩的、迷人的，有时甚至是奇妙的。然而，自从Delphi开始发行之日起，它就成为市场上最好的编译系统。

不知本书是否能够使你满意，但我会尽可能写得有趣、生动并富含信息。本书包括中级和高级的编程题目，讨论的主要内容如下：

- COM和分布计算。
- 数据库编程。
- 一般编程问题。
- 图形和DirectX编程。

像往常一样，我的目标是使用清晰、易懂的语言来说明这些主题，以便你容易接受这些困难的题目。只要有可能，我就会尽量使用友好的、交谈式的语调，这样，你在学习这么多技术主题时可能会感到舒适和放松一些。

本章主要是介绍性的内容，描述了本书的主题。另外也简单地涉及了应用程序设计的问题。后者也就是本章中的理论部分，更着重于讨论面向对象技术的合理使用。

在第2章中，将开始分析例子并讲解技术问题。如果你希望立刻开始着手进行编程，那么

可以阅读本章接近结尾处的“本书正文的注意事项”这一部分，然后就可以学习第2章。当时间充裕一些的时候，再回过头来看看那些没有读过的内容。这时就可以轻松地学习，并且花时间仔细地研究更多的本章所讨论的理论问题。

1.1 本书的结构

这本书是用于阅读，而不是用作参考手册。Delphi本身附带具有上下文相关性帮助特征的优秀在线参考。Delphi另外一种新的、令人难以置信的参考形式是在Delphi编辑器中按下Ctrl键，同时把鼠标指向任何一个变量，然后单击鼠标左键，就会直接跳到声明该变量的程序代码处。这种技术可以创造出世界级的参考手册，远远超过我在这本书中所能介绍的任何东西。

所以，这本书不是参考手册。它也不是一本指南，虽然有时会出现一些简短的类似指南的部分，用来帮助你了解向导的操作步骤或者一些相对普通但又必需的算法。不过总的来说，我是不喜欢指南的，因为在阅读指南时，几乎不可能不让人由于厌倦和遭遇挫折而烦躁不安。TrueSpace这个图形程序是我最喜欢的产品之一，它就附带了大量的指南而不是文档文件。这些指南介绍了产品的一切内容，但是对于我来说却几乎是毫无用处的。假如不是Peter Plantec后来写了一本关于这个产品的书——“Caligari TrueSpace2 Bible”，我可能根本不会使用这个工具取得任何成绩。

指南除了令人讨厌之外，同时还难以使用，而且只能在有限范围内使用。你必须坐在计算机前才能从它们之中获益，因为它们本来就是依次操作的。你还必须不断地在指南中的文字和计算机屏幕上的程序之间切换来切换去。最终的结果是你老是找不到自己在这两个媒体中各自阅读到的位置，这使得你很难一次集中精力熬过多于15到20分钟的痛苦时间。

我在写这本书时没有采用指南模式和参考模式，取而代之的是，我选择了一种描述性的风格，这意味着你可以坐在一张舒服的椅子上安静地进行阅读。当然我并不期望读者把本书从头到尾全部看完，因为这并不是技术人员（包括我在内）阅读这种类型的文档所采用的方式。我写这本书是为了让你可以一次阅读一章或者是一章中的一个部分。

注意 通常我可以愉快地连续编程5~6小时。在好的日子里，我也可以愉快地写作这么长的时间。但有时我会感到厌倦。在那些场合下，我就会找到自己喜欢的技术人员写的一本书或一篇文章，然后只阅读看起来像是包含有重要的或有趣信息的一章。看完之后，我可能连着几个月再也不看它了，也有可能下一次休息时又会拾起它去读一下。

有时，我可能会发现其中的一部分吸引了自己的注意力，或者确实想弄明白其中的内容。在这样的情况下，我会连续花几天时间阅读3、4章。但是，我还是很少把一本技术书籍从头到尾全部看完，这太让人感到畏缩了，特别是对于那些长达1000页或者页数更多的书籍来说更是如此。

花1小时左右的时间来阅读一本好书可以给我的头脑充电。读完后，我就又可以继续编程或写作了。而阅读一本参考手册或是一本指南一两个小时，则会使我精疲力尽。我发现读这种书要比自己编写代码或写文章更困难。这意味着我根本就不会去读那些书，它们只会在书架上落满尘土。

在读完本书的某一章之后，你可能会发现也可以把它当作参考手册使用。如果没有事先读过章节内容，你很可能会发现自己难以像使用在线帮助上那种类型的简单参考一样来使用它。因此，我建议把这本书带到一个安静的地方，从某章的开头开始通读那一章，然后再坐到计算

机前试一试范例程序，最后根据学到的知识创建自己的程序。

在我们的社会里，人们总是试图讨好每一个人，以便最大范围地获取听众，同时避免承担会引起某些人义愤的可怕风险。这种做法的问题在于它经常导致产品只能给每个人提供一点点东西，而不是提供某个人所需要的大部分东西。

我对本书最大的愿望是要让它令人愉快，至少让大多数读者感到它是令人舒服满意的。但是，我知道有些人会觉得我罗嗦，有时还会重复，还有些人希望我在组织材料时能更严格地按逻辑方式进行。我对这些合理抱怨的回答是，本书是有意供人阅读的。如果我想让每个人都高兴，最终我会使每个人都不高兴。所以，我诚心诚意地向那些希望我写出另一种风格书籍的人表示歉意。

假如按照一种严格符合逻辑的风格组织材料，那么难以理解的概念应该紧跟在简单概念的后面。但你根本不会从简单的概念建立起复杂的概念，而是宁可按照一个不同的顺序来进行，以便更容易地把本书当作一本参考手册使用，而这种书恐怕读不了几页就得用牙签把眼皮撑起来才能继续看下去。

这本书通过讨论把材料中的概念展现了出来。我希望让你看到它们是如何工作的，它们为什么要以这种方式工作。在明白了技术手段背后所隐藏的理论之后，我相信你就可以领会到这些东西是如何工作的了。例如，在本书中的某个部分，我解释了何时需要使用一种叫作类型库的东西以及如何使用它。如果要写参考手册的话，我只需列出来何时需要使用类型库，以及什么时候不使用类型库就行了。然而，我没有采用这种方式。取而代之的是，我解释了类型库是什么，它为什么会存在，以及它对数据调度这些核心任务有何帮助。弄清楚了这一切，就不再需要一个何时该使用类型库，何时不该使用的列表了。你所要做的就是利用我告诉你的理论知识分析出何时需要使用类型库，而何时不该使用。实际上，你甚至都不必想出这个答案，因为在具有了理论知识以后，这一切对于你来说就都是显而易见的了。

当然，我确实知道，有些读者希望拿起这本书能够找出何时使用类型库和何时不该使用类型库的列表。或者相反，他们也可能希望找到对于一些问题的解释，例如类型库是什么，它有什么作用，以及如何使用它。我知道自己的方式不能使每个人都满意，但我相信，它可以使绝大多数读者感到满意。

注意 那些不顾一切地要跳过章节中理论部分的读者，可能会发现通常在每一章的开头都要花2到6页用于介绍、概述和讲解理论。在这些内容之后，我将用特定的例子来说明我的观点。这意味着几乎总是可以跳过开头几页，然后找到那些说明我的观点的代码和演示，而我的观点是在每一章中开始那平淡无奇的几页中给出的。

要将讨论概念而不是讨论具体事实的书分为许多至多不超过3到4段的小节是不大可能的。有些人把材料分成这样的小段，对此我很理解，但我几乎总是觉得这种书难以阅读。把文章分为小节的问题在于，在这样一种严格的制度下很难形成一个主题，并在几页里展开这个主题。对于一本想使人能够愉快阅读的书来说，它需要有一种连续的感觉、逻辑发展的感觉、从A点流到B点的感觉以及容易理解的方式。我更偏爱这样一种技术：简单地剖析一个题目，然后把它的组成部分保存在“防腐液”中。

当然，我并不是准备创造一种能够讨好大量读者的设计风格。实际上，我写的书都尽可能接近于自己所喜爱阅读的那些技术著作的类型。特别是，我尽量把书写得容易理解。我不想向读者提供关于某个特定问题的最详尽或最完整的分析。恰恰相反，我所希望的是介绍怎样编写程序，并用简单、清楚、易于理解并且让人乐于阅读的语言来解释其中所包含的技术。我希望

写出一本自己在编程或写书休息时乐于阅读的书。最重要的是，我希望写出一本能教给你一些在程序员的日常工作中能够用得上的东西的书。

1.2 范例程序

读完前面的部分之后，你可能意识到我把如何写作作为重点。因为这是一本书，所以这是适当的。不过，它是一本关于编程的书。所以，书中也包括了大量的代码。应该花点时间运行一下这些程序，看一看它们和我在本书中讨论的材料是怎样联系起来的。我在前面已经说过，使用本书的最好方法是把阅读文字和运行程序结合起来。

我编写的这些程序可以按照你的意愿自由使用，但不经我的允许就简单地把它们重新用作培训教材是不可以的。受到认可的学校和大学中的老师如果希望使用这些材料，那么只要他们的学生购买本书，当然也是可以的。程序员也可以把我的这些代码按照他们的需要，用在任何他们想用的地方，但他们不能简单地拿出我的一个程序，做一点修改，甚至不做修改，然后就拿出来卖。

请尽量在程序中利用我的代码。我并不介意是否整段搬用我的代码，只要不是把一个完整的程序都建立在本书光盘上所找到的内容基础上就可以了。

如果阅读了本书中的某一章，并运行了该章的代码，那么就应该已经获得了在某一特定方面进行重要工作所需要的所有知识。为一章的内容编写相应的程序，通常花费的时间就和写这章的内容本身的时间一样长，或者差不多。所以请在代码上花些时间，学习它，并在自己的程序中使用它。本书中的文字只是本书内容的一部分，其余的都在本书附带光盘上。

1.3 创建设计良好的程序

在下面的几章中，将讨论一下如何创建有效的程序。你将会发现这一部分与本书其余章节的主题不同，它的技术性并不特别强。不过，这部分文章是十分重要的，它说明了一系列主题，今后我将不断引用这些主题。

让我明确一下对于每一个程序员来说都很重要的三件事：

- 在编写程序之前先创建一个良好的设计方案。编写程序时可以重复改进这个设计方案，这样，它就可以把个人经验和从测试程序学到的东西结合起来。
- 弄清楚OOP（面向对象编程）的基本概念，特别是封装、接口设计以及信息隐藏。
- 知道怎样编写和使用组件。

简而言之，这三个重要事项就是设计方案、OOP和组件。我在下面这几页里将会分别讨论这几个题目。

1.3.1 设计问题：书写简单的代码

本书的主题来源于George Sand说过的一句话，我一直把它钉在我的书桌上：“简单性是这个世界上最难获得的东西；它是经验的最终界限，也是天才的最终努力目标。”

每本书都需要一个主题，本书的主题看起来可能同它面向的读者相矛盾。我说过这本书是针对中级和高级程序员的，但我的主题是“尽一切所能，让程序尽可能的简单。”

经常会听到这种说法，编程不像是一门科学，而更像是一门艺术。如果编程是一门艺术，那么我们就必须找出它所基于的美学原则。根据我的意见，核心的原则就是简单性。美丽可以通过几种不同的途径获得。我最喜欢的一条途径就是遵循简单性的道路。

在编程世界中，几乎所做的单独的每一步都是简单的。说到底，编程是一个非常简单的职

业。之所以看起来困难，是因为它需要把许多非常简单的过程组合起来。两个简单的事情要比一件简单的事情稍微复杂一点。二十件简单的事情就明显要比一件简单的事情复杂多了。对于大多数人来说，把一万件简单的事情放在一起就很复杂了，当然，有少数人例外。

在一个典型的程序中，成百上千个独立的片段合在一起组成一个整体。如果看得足够透彻的话，即使是一个简单的Delphi窗体也是由上千个复杂的片段组成的。作为程序员，目标是要协调所有这些不同的片段，让它们一起工作。

我相信所有程序都应该努力遵循简单性的道路。而且，我还相信编程的工作也应该简单。这本书致力于找到能够使你更容易地创建出强大程序的方法。我努力让我的目标做到适度，节约使用资源，既符合我的美学选择，又不过分。我努力创造既使用简单又易于理解的程序。

我希望我的程序对于用户和程序员来说都是容易理解的。一个好的应用程序的体系结构应该具有和它的菜单系统同样简单的结构。帕台农神庙的建筑结构是建立在追求简单的建筑学原则的基础上的，我相信Delphi程序也应该以同样的模型作为基础。

这本书不打算让你干到凌晨4点，来挖掘程序中最深入的部分。应该能够用一种简单、容易的方式按时完成工作。这本书偶尔也会集中挖掘Delphi中最深入和最秘密的部分，但我总是推行一种简单、自然和强大有效的方法来完成这些工作。

编程不仅仅是一个技术工作，它也是一种心态。好的程序员表现出一种明智的风格。他们编写代码的方式和他们生活的方式是一样的。如果生活得简单而轻松，那么写的代码也会是简单和容易的。

如果你能够发现有一些小念头，可以把简单的项目变得比初看上去复杂时，就进入了设计阶段。例如，我最初可能不会想到需要报告、保存状态以及创建某种图表。

我经常能够用15分钟或1小时的时间就设计出一个东西，但要执行它却要花费我整整一天或几天的时间。要避免的陷阱是花费几天的时间来编写一个本质上就有缺陷的东西。如果仔细思考15到20分钟，可以为我节省两天编写程序的时间，那么花这点时间是值得的。

1.3.2 螺旋式上升

有时，如果不写一些测试代码，我就无法把设计过程进行下去。在这样的情况下，我会进行一些试验，来测试我的理论的有效性。当我开始看到一个算法的轮廓时，通常我会停下来并把设计方案定下来。然后我可能会再编写一些代码，然后再重新研究一下项目的设计。

这个过程是不断循环直到结束的。写一点代码，停下来看一看目标，然后再停下来研究一下设计问题，然后再写一些代码。

这个过程最后应该像一个螺旋那样结束。从底端的一个大圈开始，然后不断重复编码和设计工作，从而使工作的圈子越来越小。每个循环都应该比上一个更小，直到最终达到顶端。在顶端，圈子可能已经小到了一个点（最佳方案）。于是，就确定了程序的正确实现方法。

1.3.3 好OOP与差OOP的对比

我已经说得非常清楚，我所希望达到的目标就是简单性。我希望我的程序易于使用，并且建立在一个清楚、简单的体系结构的基础上。问题是如何做到设计简单呢？对于我来说，答案就是OOP。但是我认为，使用这个强大的技术一定要特别小心。

OOP是一把双刃剑。它背后的基本概念是非常有价值的。然而，许多OOP程序仅仅是典型的关于如何不用书写代码的例子。

考虑一下下面这两行代码：