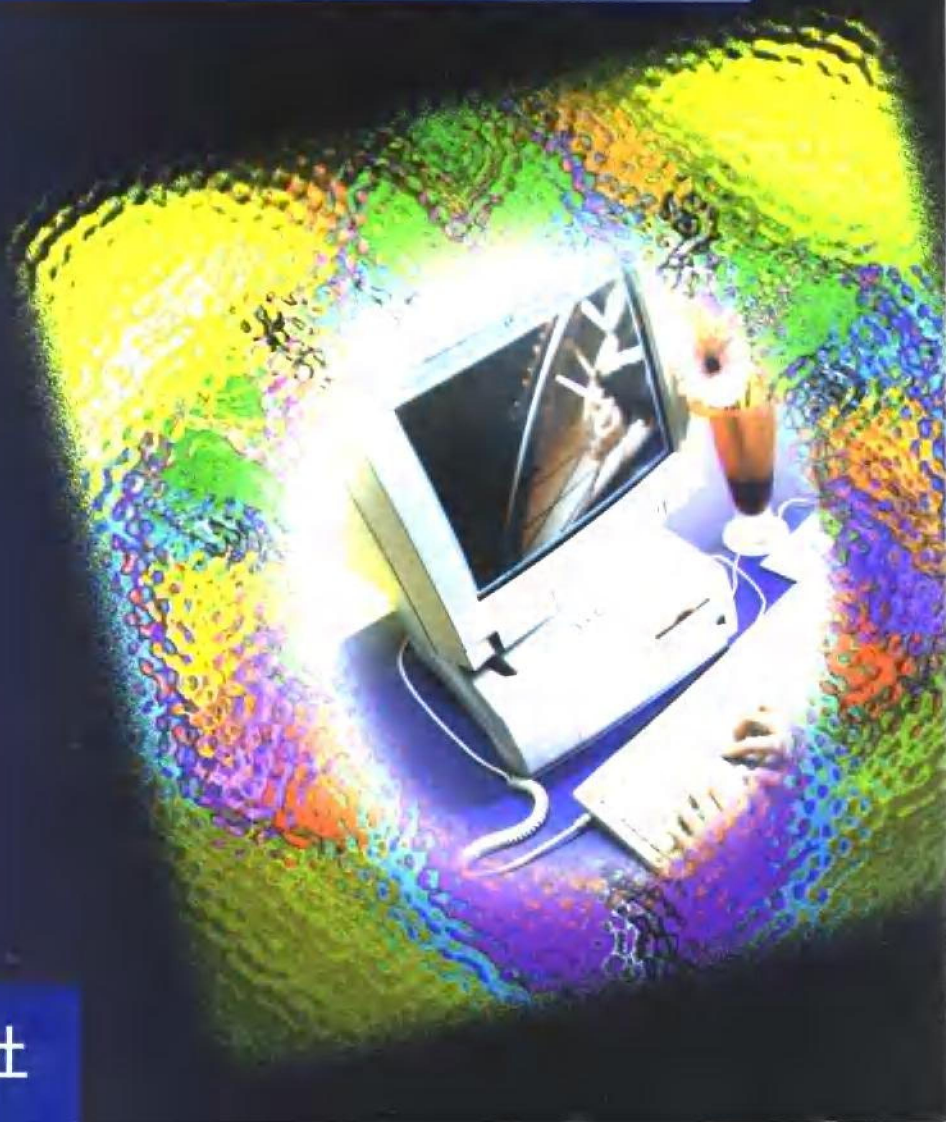


Delphi

数据库编程技术

刘前进 王蒙 张宏林 等编著



人民邮电出版社

Delphi 数据库编程技术

刘前进 王蒙 张宏林 等编著

人民邮电出版社

内容提要

Delphi 是一种优秀、通用的 Windows 环境下的编程工具，可用来编写出具有高效率、高质量的应用程序代码，特别在数据库开发领域中更是性能卓越。本书结合具体实例讲述了 Delphi 开发数据库的经验技术，覆盖了数据库开发的各个方面，包括 Client/Server，分布式 DATASET 等。本书既有基础知识，也包括深层次的实例开发，是一本开发 Delphi 数据库不可多得的参考书。

相信广大编程爱好者，特别是数据库开发领域的读者一定会喜欢这本书。

Delphi 数据库编程技术

- ◆ 编 著 刘前进 王 蒙 张宏林 等
责任编辑 张立科
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本:787×1092 1/16
印张:29
字数:718 千字 1999 年 10 月第 1 版
印数:5 001 - 9 000 册 2000 年 1 月北京第 2 次印刷
- ISBN 7-115-08226-X/TP·1397
-

定价:43.00 元

前 言

面向对象编程（OOP, Object-Oriented Programming）是一项让每个程序员心仪已久的编程技术，Inprise 公司创造性地将 OOP 的概念引入到了 PASCAL 语言中，使这种古老的语言重新焕发出了新的活力。在短短的几年内，Delphi 从众多的开发工具中脱颖而出，迅速成为众多程序员无比青睐的工具，并几度捧得一些计算机权威杂志的大奖，算得上是 Windows 软件开发领域中难得的“重量级”开发工具。

Delphi 的 OOP 技术和控件拖放功能使 Windows 编程的梦想对普通的程序员来说不再遥远，而 PASCAL 的内在潜质又使其在数据库的处理上显示出独特的优越性，甚至包括分布式数据处理这一数据库开发的前沿阵地，Delphi 也占有一席之地。除此之外，Delphi 还可以实现对多媒体及图像进行处理，支持对 INTERNET 的编程，允许多人合作开发以及数据包的发放等。明快简洁的语言风格，强大的整体功能，方便的编译环境，直接可运行的程序，这就是 Delphi！

本书内容主要基于 Delphi 4.0 客户机服务器版本。首先简单介绍 Delphi 的开发环境，让从未使用过 Delphi 的读者快速地掌握 Delphi 的使用。接着是对数据库的介绍，包括数据库的基础知识和开发技巧，以方便那些没有接触过数据库的读者，使他们能快速掌握数据库开发的必要知识。最后，对 Delphi 的数据库开发进行详细的讲解。

本书面向各个层次的读者：从未使用过 Delphi，但有一定编程语言基础的读者可以通过本书完成对 Delphi 的学习；对曾经使用过 Delphi 却没有开发过数据库的用户，更可以通过本书学习，领悟到开发数据库的精髓，成为数据库开发的高手；对于已有 Delphi 开发数据库经验的读者，也可以从本书中获取一些编程的灵感，完全把握 Delphi 的内核，并充实其他相关的知识，从而完成高质量的代码编写。

参加本书编写的有刘前进、王蒙、张宏林、王开、颜学庆、郑旭、高勇、候小龙、刘家森、庞国华、于福军、王建军、马大永、刘永康、刘峰、赵昆、赵炎炎、曹岩、唐虎、张小航、李长顺、刘光、刘亚军等。

由于时间紧迫，作者水平有限，经验不足，书中难免有遗漏、错误和不当之处，敬请读者批评指正。

编者
1999 年 9 月

目 录

第一章 Delphi 基础知识	1
1.1 Delphi 新技术介绍	1
1.1.1 Delphi 的不同版本	1
1.1.2 编辑器性能的提高	2
1.1.3 语言的扩展	2
1.1.4 提高程序设计人员的生产力	3
1.2 对数据库的支持	3
1.2.1 数据分析	3
1.2.2 开放性与灵活性	4
1.2.3 MIDAS	4
1.2.4 其它新的功能	5
1.3 本书内容	5
1.4 小结	6
第二章 Delphi 快速入门	7
2.1 Delphi 集成开发环境 IDE 简介	7
2.1.1 一个简单的示例	8
2.1.2 编写事件处理过程	11
2.2 使用 VCL 控件编程	12
2.2.1 使用常用的文本相关控件	12
2.2.2 使用按钮和检查框控件	13
2.2.3 使用分组、分界控件	14
2.2.4 使用图形、图像控件	14
2.2.5 使用关系图、文件列表控件	15
2.2.6 使用滚动控件	15
2.2.7 使用网格、表格控件	16
2.2.8 使用多媒体(MultiMedia)和 OLE 控件	16
2.3 使用非可视控件编程	17
2.3.1 使用菜单控件	17
2.3.2 使用计时器控件 Timer	20
2.3.3 使用公用对话框控件	20
2.4 使用 Delphi 的项目管理	22

2.4.1 创建一个多窗体项目	22
2.4.2 使用项目管理器 Project Manager	24
2.4.3 使用窗体模板和对话框向导	25
2.4.4 使用项目模板和应用向导	27
2.5 小结	28
第三章 Object Pascal 语言基础	29
3.1 基本标识符的说明与使用	29
3.1.1 变量	29
3.1.2 常量	30
3.2 基本数据类型	30
3.2.1 整型	30
3.2.2 实型	31
3.2.3 布尔型	32
3.2.4 字符型	32
3.2.5 指针型 (Pointer)	32
3.2.6 字符串型	33
3.3 定义新的数据类型	34
3.3.1 枚举类型	34
3.3.2 子界类型	35
3.3.3 数组类型	35
3.3.4 集合类型	37
3.3.5 记录类型	37
3.3.6 文件类型 (FileType)	38
3.3.7 过程类型 (Procedure type)	39
3.3.8 变体类型 (Variant type)	39
3.4 Pascal 的基本语法结构	43
3.4.1 运算元及运算优先次序	44
3.4.2 叙述语法	45
3.5 过程与函数	49
3.5.1 基本结构	49
3.5.2 函数和过程中的类型说明	51
3.5.3 一个调用 Delphi 方法的简单例程	54
3.6 小结	55
第四章 用 Delphi 的对象进行编程	56
4.1 类 (Classes)	56
4.1.1 类的基本特性	56

4.1.2	从一个类中继承数据和方法	58
4.1.3	类的范围	59
4.1.4	类公有域和私有域的说明	60
4.2	对象的操作与访问	61
4.2.1	访问对象的域和方法	61
4.2.2	对象的赋值	62
4.2.3	建立非可视化对象	63
4.3	Object Pascal 程序代码的编写	65
4.3.1	Object Pascal 程序的基本框架结构	65
4.3.2	Object Pascal 的库单元 Unit	69
4.3.3	异常处理	72
4.4	对单元的引用说明	76
4.4.1	单元调用	77
4.4.2	间接单元参考	78
4.4.3	单元间交互参考	78
4.4.4	一个应用程序的示例	79
4.5	小结	82
第五章	数据库系统介绍	84
5.1	关系数据库模型	84
5.1.1	数据库管理系统 (DBMS)	84
5.1.2	数据库应用程序	86
5.2	数据库应用程序的体系结构	87
5.2.1	选择合适的体系结构	87
5.2.2	可伸缩性	88
5.2.3	单层的数据库	88
5.2.4	两层的数据库应用程序	89
5.2.5	多层的数据库应用程序	90
5.2.6	数据库的安全性和完整性	91
5.3	Delphi 数据库系统简介	92
5.3.1	Delphi 的数据库特性	94
5.3.2	Delphi 可以访问的数据源 (DataSource)	95
5.3.3	数据访问控件	95
5.3.4	数据控制控件	98
5.3.5	其它数据库开发工具	100
5.4	开发数据库应用程序	101
5.4.1	概述	101
5.4.2	数据库应用程序的开发步骤	102
5.4.3	交付数据库应用程序	103

5.4.4	安装 BDE.....	103
5.4.5	安装 SQL Link.....	105
5.5	小结.....	108
第六章	使用数据库访问控件.....	109
6.1	TDataSource 控件及其使用.....	109
6.1.1	TDataSource 控件的属性.....	110
6.1.2	TDataSource 控件的事件.....	110
6.2	TTable 控件及其使用.....	112
6.2.1	TTable 控件主要的属性.....	112
6.2.2	TTable 控件的方法及应用.....	113
6.3	字段控件以及字段编辑器的使用.....	127
6.3.1	字段控件.....	128
6.3.2	动态字段与永久字段.....	134
6.3.3	Oracle 8 的对象字段.....	134
6.3.4	字段编辑器的使用.....	138
6.4	TSession 控件及其使用.....	142
6.4.1	TSession 控件的重要属性及作用.....	142
6.4.2	TSession 控件的方法及使用.....	143
6.4.3	TSession 控件应用举例.....	148
6.5	TNestedTable 控件及其使用.....	153
6.6	TBatchMove 控件及其使用.....	153
6.6.1	使用 TBatchMove 控件的一般步骤.....	154
6.6.2	使用 TBatchMove 控件.....	154
6.6.3	一个例子.....	156
6.7	编程实例.....	159
6.7.1	一个捕捉数据库错误的示范程序.....	159
6.7.2	一个动态改变过滤条件的示范程序.....	164
6.8	小结.....	167
第七章	使用数据浏览控件.....	168
7.1	TDBGrid 控件及其使用.....	170
7.1.1	TDBGrid 控件的主要属性及应用.....	171
7.1.2	列对象.....	173
7.1.3	TDBGrid 控件的事件及应用.....	175
7.2	TDBEdit 控件及其使用.....	177
7.3	TDBText 控件及其使用.....	179
7.4	TDBNavigator 控件及其使用.....	180

7.5	TDBImage 控件及其使用	181
7.6	TDBMemo 控件及其使用	182
7.7	TDBComboBox 控件及其使用	182
7.8	TDBListBox 控件及其使用	184
7.9	TDBCheckBox 控件及其使用	185
7.10	TDBRadioGroup 控件及其使用	185
7.11	TDBCtrlGrid 控件及其使用	186
7.11.1	使用 TDBCtrlGrid 控件	186
7.11.2	一个 TDBCtrlGrid 控件的示范程序	187
7.12	TDBChart 控件及其使用	191
7.12.1	TDBChart 的属性	191
7.12.2	使用 TDBChart 控件	191
7.13	编程实例	193
7.13.1	多表格下的 DBNavigator 使用	193
7.13.2	一个综合实例	202
7.14	小结	208
第八章 使用 Delphi 数据库附件		209
8.1	使用 Database Desktop 工具	209
8.1.1	数据库桌面(Database Desktop)新特点	210
8.1.2	关于数据库桌面对象	211
8.1.3	数据库桌面对象的文件扩展名	213
8.1.4	关于数据库桌面窗口	213
8.2	使用 SQL Explorer	217
8.2.1	浏览面板	218
8.2.2	信息面板	219
8.2.3	数据字典	221
8.3	BDE 配置工具	223
8.3.1	数据库引擎配置工具的 Database 页	224
8.3.2	使用“Configuration”页	225
8.3.3	Microsoft ODBC(Open Database Connectivity)配置	228
8.4	SQL Monitor 及其它	230
8.4.1	SQL Monitor	230
8.4.2	使用 Data Migration Wizard	232
8.5	小结	236
第九章 报表制作与统计处理		237
9.1	制作报表入门	237

9.1.1	QuickReport Wizard.....	237
9.1.2	Report 模板	239
9.2	快速报表控件介绍.....	240
9.2.1	TQuickRep	240
9.2.2	TORSubDetail 控件	241
9.2.3	TQRBand	242
9.2.4	TQRChildBand.....	242
9.2.5	TQRGroup.....	243
9.2.6	TQRLabel.....	243
9.2.7	TQRDBText.....	243
9.2.8	TQRExpr.....	243
9.2.9	TQRSysData.....	244
9.2.10	TQRMemo 和 TQRExprMemo	245
9.2.11	TQRRichText 及 TQRDBRichText	245
9.2.12	TQRShape	245
9.2.13	TQRImage 和 TQRDBImage.....	245
9.2.14	TQRCompositeReport.....	245
9.2.15	TQRPreview.....	246
9.2.16	TQRChart	246
9.3	创建报表	246
9.3.1	制作一个简单的报表.....	246
9.3.2	制作带有主从关系的复杂报表	248
9.3.3	给报表增加一些风格.....	249
9.3.4	一个综合的例子	250
9.4	使用 Decision Cube 控件组	254
9.4.1	Decision Cube 控件组	254
9.4.2	使用 Decision Cube 控件组	260
9.5	小结	262

第十章 SQL 语言使用与编程 263

10.1	SQL 入门.....	263
10.1.1	SQL 的历史	263
10.1.2	SQL 的优点	263
10.1.3	Delphi 对 SQL 的支持.....	264
10.2	了解 TQuery 控件	265
10.2.1	使用 TQuery 控件.....	265
10.2.2	SQL 查询.....	266
10.2.3	深入 TQuery 控件.....	269
10.3	使用 SQL 语言	271

10.3.1	SQL 基本语法	271
10.3.2	使用 Delphi 编写 SQL 命令文本	276
10.3.3	SQL 程序的执行	278
10.4	掌握动态 SQL 编程	279
10.4.1	在设计期提供参数	279
10.4.2	在运行过程中, 通过程序为参数赋值	280
10.4.3	从另一个数据集获得参数	281
10.4.5	Prepare 方法的使用	283
10.5	编程实例	284
10.5.1	一个后台查询的示范程序	284
10.5.2	设计一个数据库查询器	290
10.6	小结	299
第十一章 进一步理解数据集控件		300
11.1	TDataSet 控件及其子控件	300
11.1.1	TBDEDataSet	301
11.1.2	TDBDataSet	302
11.1.3	TClientDataSet	302
11.2	单层和两层的应用程序	303
11.2.1	基于 BDE 的应用程序	303
11.2.2	事务	304
11.3	数据集中的操作	306
11.3.1	数据集的状态	306
11.3.2	浏览记录	310
11.3.3	关于书签操作	313
11.3.4	搜索特定的记录	315
11.3.5	使用过滤	316
11.3.6	修改数据	318
11.3.7	事件	322
11.4	编程实例	323
11.4.1	一个对数据集进行过滤的示范程序	323
11.4.2	一个自定义数据集的示范程序	330
11.5	小结	330
第十二章 Delphi 客户/服务器应用开发		331
12.1	Delphi 客户/服务器开发	331
12.1.1	客户/服务器结构	331
12.1.2	Borland Delphi 的客户/服务器解决方案	332

12.2	TDatabase 控件及其使用	334
12.2.1	TDatabase 控件介绍	335
12.2.2	控制数据库的连接	339
12.2.3	TDatabase 与 TSession 的关系	342
12.2.4	事务处理控制	343
12.3	使用存储过程	344
12.3.1	概述	344
12.3.2	TStoredProc 控件介绍	345
12.3.3	怎样使用存储过程	346
12.3.4	创建一个存储过程	347
12.3.5	存储过程的参数	350
12.4	Cached Update 和 UpdateSQL	354
12.4.1	Cached Update	354
12.4.2	UpdateSQL 控件	355
12.5	编程实例	361
12.5.1	数据库环境介绍	361
12.5.2	应用程序分析	363
12.6	小结	379
第十三章	MIDAS 编程	380
13.1	MIDAS 概述	380
13.1.1	走进 MIDAS	380
13.1.2	理解多层分布式体系结构	382
13.2	MIDAS 控件组介绍	384
13.2.1	对象库中的远程数据模块	385
13.2.2	Provider	388
13.2.3	与应用服务器连接	392
13.2.4	TClientDataSet 控件	395
13.3	进一步理解 TClientDataSet	398
13.3.1	浏览和编辑数据	399
13.3.2	索引	400
13.3.3	计算字段	401
13.3.4	统计值	402
13.3.5	数据包	403
13.3.6	与应用服务器通信	404
13.3.7	在文件中存取数据	406
13.4	创建多层应用程序	407
13.4.1	创建应用程序	407
13.4.2	调用服务器上的接口	409

13.4.3	在客户端纠错	410
13.4.4	更新数据	410
13.4.5	自定义应用服务器	412
13.5	采用 ActivexForm 开发 Web 数据库	414
13.5.1	Delphi 开发 Web 数据库	415
13.5.2	把客户程序设计为 ActiveForm	416
13.5.3	一个 ActiveForm 的例子	418
13.6	编程实例	420
13.6.1	一个介绍 TClientDataSet 功能的例子	420
13.6.2	一个演示 Master/Detail 关系的示范程序	437
13.6.3	一个动态传递 SQL 语句的示范程序	442
13.6	小结	448

第一章 Delphi 基础知识

美国 Inprise 公司于去年 6 月发布了最新的 Delphi 版本之后,更加巩固了 Delphi 作为全球公认的快速应用程序开发工具 (RAD) 的领导地位。Delphi 以其友好的可视化开发环境、完全编译执行的高效性而著称,Delphi 是真正能够令开发者感到心跳的开发伴侣!由于 Inprise 公司 32 位 Delphi 编译器采用了独特的本机代码 (native code) 编译器、可视工具、可扩展的数据库技术,因此 Delphi 已获得了众多的世界级大奖,并成为最畅销的十大软件之一。同时, Inprise 公司也赢得了相当数量的第三方厂商的支持。现在,第三方厂商已经开发出上百种程序库和兼容工具,有关 Delphi 的技术参考书已不胜枚举,众多的 Delphi 技术培训机构也应运而生。

因为 Delphi 遵循业界标准来开发分布式多级应用系统,所以它能使应用系统适用于大部分的异构环境。Delphi 同时支持 COM 和 CORBA 接口的最新功能,对于想在对象结构标准的纷争中立足于不败之地的 IT 部门而言,它是最具吸引力的选择。Delphi 确立了高效率 and 快速开发高性能的客户端 / 服务器 (Client/Server) 及 Enterprise 和 Internet 上的多层体系结构应用的标准,其快速应用开发、可伸缩的多层数据库开发、真正的面向对象的可重用性及本地代码优化编译技术,为企业范围内的 Client/Server 开发及 Internet 应用创造了条件。

本章的重点是:

- Delphi 的新特色。
- 对数据库技术的增强。

1.1 新版 Delphi 介绍

新版的 Delphi 为了充分利用 Windows 98 和 Windows NT 提供的功能,在许多方面做了改进工作。新版的 Delphi 全面支持 Windows 98 界面和 API,包括许多新的界面构件,支持全部的 Windows 98 的 API,包括多线程、MAPI,支持 OLE 控件 (OCX)。这些功能都将极大地方便 Windows 98 下的开发人员和提高应用程序的性能。

Delphi 采用了许多新的技术,在优化时不需要猜测,可以直接生成高性能代码。此外,新版的 32 位编译器采用了一系列的自动优化技术,包括寄存器优化、共用子表达式消除等,所有这些优化不改变源代码的语义,保证了应用程序的正确执行。

1.1.1 Delphi 的不同版本

Delphi 有三种版本发行,用于满足不同领域的软件开发人员的需要。

1. 标准版

针对一般程序设计人员的产品,适合普通应用系统的开发。与以前版本相比,Delphi 包括了许多新的特性,如新的控件,Code Insight 功能,新版的 QuickReport,COM 接口支持

和新型的程序包结构等。

2. 专业版

针对专业程序开发人员。它在标准版的基础上，提供了对数据库和 Internet 的支持。

3. 客户机/服务器版

面向的是开发 Client/Server 应用程序的程序员。它包含了专业版的全部功能，并提供了 Decision Cube CrossTab 控件，扩展的 Web 服务器支持，开发式环境的 OLEnterprise 和 SQL Monitor 等。

本书讨论 Delphi 的数据库开发，一些编程技巧是基于 Delphi Client/Server 4.0 版本的，因此，读者应注意不同版本的区别。

1.1.2 编辑器性能的提高

1. 寄存器优化

经常使用的变量和参数将会被自动放在 CPU 的寄存器中，这样便减少了访问该变量而必须执行的指令的数目。由于程序不再从内存中读取变量放入寄存器中，不仅使代码紧凑，而且提高了执行速度。这种优化是由编译器自动完成的，开发人员无需指定把哪一个变量或参数放在寄存器中。编译器会自动分析变量的生命周期，进行寄存器优化。

2. 共用子表达式消除

在编译器翻译复杂的数据表达式时，会自动找出相同的子表达式，以防止运算多次。这样一来，开发人员可以自由地、清晰地书写他们的数据表达式而无需考虑优化，把这一工作留给编译器。

3. 新的优化连接器

Code Insight 是一组代码自动化功能的总称，它能够帮助用户快速生成代码，从而减少语法错误并提高编程效率。Delphi 3 已经有了 Code Insight 功能，但新版的 Delphi 又作了很多改进。作为编译过程的一个部分，Delphi 使用一种新的 32 位连接技术。这种新的 32 位连接技术也将提供若干优化，提高代码运行效率。因为新的智能连接器将消除无用的函数和过程，以及无用的静态和虚方法，这样，Delphi 应用程序不仅代码紧凑，而且执行速度快。

4. 模块窗口

模块窗口能够显示当前所有映射到应用程序地址空间的模块，包括应用程序自身、应用程序显式或隐式调用的 DLL 以及操作系统调用的 DLL，模块窗口可帮助用户优化程序结构。此外，Delphi 还能自动记录在调试过程遇到的事件，如断点、Windows 的消息以及其他调试信息。Delphi 能够监视指针错误，如果某个指针试图非法访问内存的某个地址，程序就会暂时中断运行，由调试器接管控制权，就好像遇到断点一样。

其他还有诸如类自动完成，符号预知等，可以帮助程序员快速生成有关的代码，并减少错误的产生。引入了浮动坞技术，使界面开发体现出 Windows 新的风格；Tool tip Symbol Insight 使程序员更方便地调试程序。

1.1.3 语言的扩展

Delphi 扩展了 Object Pascal 语言，可支持动态数组，参数缺省等。

1. 动态数组

Delphi 中,除了定义静态数组外,还可以定义动态数组。动态数组只需说明数组的类型信息,包括数组的维数和数组元素的类型,但不需要定义元素的个数。例如:

```
Static:array[1..100]of string;
Dynamic:array of integer;
Multi_Dynamic:array of array of string;
```

2. 方法重定义

新增加保留字 **Overload** 标识。方法重定义时,各个同名的方法能够根据参数中不同的类型信息予以区分。例如:

```
Constructor Destroy(Aowner:TComponent);overload;override;
Constructor Destroy(Aowner:TComponent;Text:string);overload;
```

3. 缺省参数

类似 C++, Delphi 中支持为过程与函数定义缺省参数。缺省参数仅仅显示在参数列表的尾部。这样,当调用定义了缺省参数的过程或函数时,缺省参数就可以略去,极大地方便了程序员的开发。

4. 新的数据类型

支持 64 位整数,即 **Int64** 类型,其数值范围可以从 -2^{63} 到 $2^{63}-1$ 。引进了真正的 32 位无符号整数类型 **longword**,其整数范围是 0~4294967295。对实数类型,先前代表 48 位浮点数的实数类型已等价于 64 位的双精度类型。

1.1.4 提高程序设计人员的生产力

在程序开发过程中,语言的语法控件的方法和事件的名称,或已经声明过的变量过程的名称往往很难准确地书写。Delphi 提供了许多辅助方法来解决这些问题,减少程序开发所需要的时间。在 Delphi 有三个 Wizard,分别是 **Code Template Wizard**,**Code Completion Wizard** 和 **Code Parameter Wizard** 用来帮助开发者避免一些在程序开发时不必要的资源和时间浪费。

1.2 对数据库的支持

Delphi 拥有极其强大的数据库操作能力,包括支持对 **Windows NT Services** 的开发,全面支持大型数据库 **Oracle 8.0**;新的灵活的 **Grid** 控件,一步到位的 **COM** 和 **CORBA** 的控件开发等。

1.2.1 数据分析

正确的数据分析,在业务决策分析上是相当的重要的。当大量的数据放在用户面前,用户要搞清这些繁杂数据背后的真实意义可不是件容易的事。如何将现有的数据转成有用的信息,来辅助决策者做出最正确的决择呢?一般来说,运用图形及报表形式是最好不过的了,Delphi 提供了三个重要的控件:**Decision Cube**、**TeeCharts** 和 **Quick Report**。

Decision Cube 让使用者能依据个人需求, 动态查询数据, 显示多维数据和动态图表, 产生不同性质的分析结果, 充分显示数据背后的相互关系, 提供给决策者最快最有用的信息, 帮助其作出最正确的决定; TeeCharts 是一个三维的可视化图表, 使数据更容易被理解; Quick Report 是一个超强的报表生成控件群, 除了提供一般的报表外, 还可以产生 HTML 的报表格式, 使报表得以在 Web 上发布。

1.2.2 开放性与灵活性

Delphi 的开放体系结构支持多个数据库引擎, 任何一个数据库引擎都能与 Delphi4 的数据显示控件如 TDBGrid、TDBEdit 协调工作。

Delphi 可以访问 Oracle 的 BLOB(Binary Large Object)型字段, 实际的数据存储在外部文件中。Delphi4 支持 Oracle8.0 的抽象数据类型(ADT, Abstract Data Types), 允许自己定义数据类型, 自定义的数据类型可以基于真实的类型, 也可以基于已定义的抽象类型。这个功能的意义是非常重大的, 用户可以把任何商业对象(Business Objects)存储到数据库中, 或者把 Oracle 数据库中的商业对象放到 Form 上显示出来。

此外, Delphi 还支持 Microsoft Transaction Server(MTS)。MTS 是 Microsoft 为实现分布式应用而设计的一个中间件, 其基本组成部分是 ActiveX。MTS 中的“两阶段提交”技术使得应用程序能够跨服务器处理事务。MTS 中的“X/OpenXA”技术允许服务器与 MTS 的资源管理器通信, 客户程序能够在异构环境下看到提交的数据包成功还是失败。而对 Multi-Tiered 的支持和 2000 年问题的圆满解决更是解决了软件行业一大难题。

1.2.3 MIDAS

Delphi 是开发多层 Client/Server 应用程序的最佳编程工具, 这与 Inprise 的发展战略有关。MIDAS(Multi-Tiered Distributed Application Services)实际上是一组分布式技术的统称, 它包括以下方面:

Business Broker 是一种分布式的对象名称注册服务, 它能根据当前的负载情况动态地调整远程对象的投入和切除, 从而保证关键性的商业应用永不中断。

OLEnterprise 是一个把传统的 Client/Server 应用程序向三层结构“迁移”的工具, 它能够把应用逻辑和数据访问模块集中在应用服务器上完成。

“Remote Data Broker”是多层 Client/Server 结构的关键技术, 客户端只需要实现用户界面, 而应用逻辑(又称商业规则)则由应用服务器实现, 这样就达到使客户端“减肥”的目的, 并且“瘦”客户端的维护成本很低。“Remote Data Broker”技术的另一个作用在于它能显著地减少网络上的数据流量, 这在带宽问题日益严重的今天具有重要的意义。

MIDAS 除了能传递数据以外, 还能传递表的结构, 对于“瘦”客户来说, 程序员不用写任何代码, 就能保证表与表之间的 Master/Detail 关系和嵌套关系得到维护和更新。为了保证系统的安全性, 鉴别用户身份(登录)是一种最简单而又有效的手段。Delphi4 内建了用户登录功能, 程序员不必为此编写额外的代码。

Delphi 能够使“瘦”客户端显著地加快检索数据的速度, 这是因为 Delphi4 采用了一种 Deferred BLOB 的技术, 它优先下载非 BLOB 型的数据, 等非 BLOB 型的数据下载完后再根据需要下载 BLOB 型数据。