计算机编程技术丛书

Delphi 5 数据库编程实战与精通

房增华　徐远超　编著

王东辉　审校

清 华 大 学 出 版 社

需要全本请访问 www.ertongbook.com

（京）新登字 158 号

内 容 提 要

Delphi 5 是一个强大的开发工具，在数据库应用开发方面尤为突出。本书是围绕 Delphi 5 开发数据库应用来写的，全书以严谨的结构，详细介绍 Delphi 的发展过程和 Delphi 5 的新特性，数据库应用开发的模式，Delphi 5 的数据库访问组件与数据感知组件。报表与决策方设计，MIDAS、ADO 和 IBX 等新技术，每章都有大量示例，最后以一个综合应用实例作为范例供读者参考。

本书不但详细讲解了 Delphi 5 中与数据库相关的各个组件的属性、方法，还通过例子使读者能深入了解到组件的具体使用技巧。因此，本书对提高开发人员的实践能力很有价值。

本书适用于对 Delphi 有所了解、从事数据库应用开发的人员。也可作为大专院校计算机编程课程的参考书。

版权所有，翻印必究。
本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目（CIP）数据

Delphi 5 数据库编程实战与精通/房增华，徐远超编著. — 北京：清华大学出版社，2000.7
（计算机编程技术丛书）
ISBN 7-302-03930-5

Ⅰ.D... Ⅱ. ①房… ②徐… Ⅲ.Delphi 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP **数据核字**（2000）**第** 30822 **号**

出版者： 清华大学出版社（北京清华大学校内，邮编：100084）
http://www.tup.tsinghua.edu.cn
印刷者： 北京丰华印刷厂
发行者： 新华书店总店北京发行所
开 本： 787×1092 1/16 **印张：** 20.75 **字数：** 488 千字
版 次： 2000 年 7 月第 1 版 2000 年 7 月第 1 次印刷
书 号： ISBN 7-302-03930-5/TP・2298
印 数： 0001~5000
定 价： 30.00 元

《计算机编程技术丛书》出版说明

《计算机编程技术丛书》包含了现今的流行软件，如Authorware 5.0、AutoCAD 2000、Visual C++ 6.0、Visual Basic 6.0、Visual Foxpro 6.0、Delphi 5等，是一套易学易用的指导书。既可循环学习，亦可随查随用，使您学有所依，用有所循，快速便捷地掌握软件的操作方法和编程技术，得心应手地解决实际问题。

为了使本系列书有的放矢，力求使其具有如下特点：

知识速查　我们对各编程软件的基本知识作了简要说明，以备速查。使只要有一点软件开发经验的读者就可以在很短的时间内掌握如何编写基本的应用程序。

循序渐进　在编写本书过程中，尽量按照“由浅入深”的顺序来编写，以便于读者的学习。

深入浅出　作者站的角度较高，能深入探讨编程原理及技巧，因为是国内作者所著，能以流利、规范、通俗易懂的语言来说明。

示例详尽　书中的难点、疑点尽量利用实例来进行说明，以便于理解掌握。同时在每个细节中都尽可能地给出详尽的示例，以避免由于一些细微的错误而导致不能掌握某个知识点。

创新思维　以生动的、具有启发性的示例启发读者，且配有具体的步骤和丰富的画面，以及有关的经验提示，让读者深刻领悟该软件的精华。

内容全面　书中包含的内容丰富而全面（对中级读者而言），可以适应读者开发不同应用程序的需要。

本系列书面向工作在各行各业的计算机编程人员，为他们提供最新的编程技术与编程经验，使他们能尽快跟上技术发展，借鉴他人的经验并结合自身工作，编制出更先进、实用的应用程序。

在本系列书的编著过程中，尽量注意减少冗长无味的说明，代之以具体实用的例子演示。通过例子，引导读者把握精髓所在。丛书注重开发实例、开发经验、开发技巧和高级特性开发，适应读者的急需。

本系列书是集体智慧的结晶，而且每本书都是由最擅长该开发工具的开发人员执笔。

前　言

当今的应用系统愈来愈庞大，愈来愈复杂，因此就需要强大的开发工具来支持。这些开发工具需要具有可视化编程、可复用的组件、集成的开发环境和强有力的辅助工具。Delphi 5 正是生逢其时的一个理想开发工具。和以前的版本相比，Delphi 5 在很多方面都有了很大改进，又增加了很多方便开发者的新特性。MIDAS 是顺应当前多层计算模式、多层体系结构潮流的一项新特性，运用 MIDAS 可以开发出多级分布式应用系统，这可以说是 Delphi 5 的最大特色。Delphi 5 支持 Microsoft 的 ActiveX Data Object（ADO）和 OLE DB 技术，利用 ADO，Delphi 5 能够快速访问关系型和非关系型数据库，其中包括电子邮件和文件系统。开发人员可以对 Internet 信息进行直接访问，以帮助用户更好地进行商业决策。Delphi 5 新增的 InterBase Express 可构架卓越的关键应用系统，它使 Delphi 和 InterBase 更紧密地结合，即将高效率、高性能的开发工具与快速、低维护量和优化配置的中小型关系型数据库等技术融于一体。

凡是复杂的应用，大都是基于数据库开发的，而 Delphi 5 正是以其在数据库开发方面的快速高效而著称。本书正是围绕利用 Delphi 5 开发数据库应用的主题而写的。全书通过翔实的讲解，并辅以恰当的例子，旨在提高开发者的实战能力。本书可以为了解 Delphi 的开发者深入学习数据库编程之用。

本书第 1 章主要讲解 Delphi 的发展以及 Delphi 5 的新特性。了解并运用这些新特性，可以提高编程效率，缩短开发时间。对 Delphi 5 较熟悉的读者可以跳过此章。

第 2 章讲解数据库应用设计。这一章从理论上系统地讲解数据库设计和数据库应用设计的原理，对系统分析员和数据库设计人员有一定的启发意义。

第 3 章讲解数据库工具。这些工具用来维护数据库数据，为应用程序提供访问接口。

第 4 章讲解单层和两层的体系结构，这是以前应用系统普遍采用的结构。我们可以比较其优缺点，对以后的开发具有启发意义。

前面四章可以算是一部分，主要从大的方面讲解数据库开发方面的相关问题。

第 5 章讲解数据感知组件，这些数据感知组件用来设计数据库应用程序的门面，开发者可以用这些组件来设计自己的界面。

第 6 章到第 14 章讲解数据库存取组件，这些组件负责从数据库中存取数据，组织数据集，为数据感知组件提供数据源。

第 15 章讲解如何设计报表，表达和传递数据信息。

第 16 章讲解 TeeChart 报表的设计技巧。

第 17 章讲解决策方的设计。

第 18 章讲解如何使用 ADO 组件，建立对关系型数据和非关系型数据的访问。

第 19 章讲解 IBX 组件的属性与用法及其在快速、低维护量和优化配置的中小型关系型数据库开发方面的优势。

第 20 章介绍了笔者最近用 Delphi 5 开发的一个试题系统，希望对读者有所启发，并对以后的具体项目开发有所帮助。

全书不但详细讲解了 Delphi 5 中与数据库相关的各个组件的属性、方法，还通过例子使读者能深入了解到组件的具体使用技巧。因此，本书对提高开发人员的实战能力是很有价值的。

《Delphi 5 数据库编程实战与精通》是“计算机编程技术丛书”中的一本。书中示例中所涉及的人名、电话号码、通信地址和邮件地址等均为虚构，如有雷同，纯属巧合。

本书由方可燕策划，房增华、徐远超编著，王东辉审校。参与编写工作的还有方可军、王长春、周兆确、毛诗华、方海英、唐家才、段小华、许刚、侯志东、陈斌等。另外，董朝旭、向军、何晓威、王宇、李亨、张双民等提供了许多有益的参考资料和参考意见。

由于作者水平有限，且编写时间仓促，书中难免有疏漏和错误，恳请广大读者提出宝贵意见。

编　者

需要全本请访问 www.ertongbook.com

目　　录

第 1 章　Delphi 5 简介 1
1.1　Delphi 的产生和发展 1
1.1.1　从 Delphi 1 发展到 Delphi 3 1
1.1.2　Delphi 4 3
1.2　Delphi 5 的新特性 7
1.2.1　Internet 上的扩展 7
1.2.2　快速、直接的数据库访问 7
1.2.3　提高团队效率 8
1.2.4　高效的集成开发与集成调试环境 8
1.2.5　Delphi 5 在数据库方面的新特性 10
1.3　升级到 Delphi 5 应注意的问题 11
1.4　选择 Delphi 5 作为数据库应用开发工具 12

第 2 章　数据库应用设计 16
2.1　数据库应用开发策略 16
2.1.1　开发数据库 16
2.1.2　数据建模 17
2.1.3　模型到数据库设计的转换 17
2.1.4　应用开发的一般过程 17
2.2　数据库应用程序的体系结构 18
2.2.1　如何设计体系结构 18
2.2.2　BDE 的结构 19
2.3　用户界面设计 19
2.3.1　窗体的设计 19
2.3.2　分析数据 20
2.3.3　选择要显示的数据 21
2.4　客户/服务器体系结构 21
2.4.1　对客户/服务器结构数据库系统的需求 22
2.4.2　客户/服务器结构的 DBMS 的功能划分 23
2.4.3　客户/服务器结构的数据库系统实现技术 25
2.4.4　客户/服务器结构数据库系统的优点 27
2.5　数据库设计中的问题 29
2.5.1　数据库安全 29
2.5.2　事务 30
2.5.3　数据字典 30
2.5.4　参照完整性、存储过程和触发器 30

第 3 章　数据库工具 31
3.1　BDE 31

3.1.1 BDE 的特点 31
3.1.2 使用 BDE 32
3.2 Database Explorer 35
3.2.1 Database Explorer 环境 35
3.2.2 使用 Database Explorer 维护数据库别名 38
3.3 Data Pump 38

第 4 章 单层和两层数据库的应用设计 43
4.1 单层的数据库应用程序 43
4.2 基于文件的单层数据库应用程序 45
4.2.1 构建数据集 46
4.2.2 数据的装入和保存 48
4.2.3 公文包模式 48
4.3 两层的数据库应用程序 49
4.4 Delphi 访问数据库的方式 50

第 5 章 数据感知(data-aware)组件应用 52
5.1 显示和编辑数据的一般步骤 52
5.2 动态字段和永久字段 53
5.2.1 动态字段对象 53
5.2.2 永久字段对象 53
5.3 数据感知组件简介 55
5.3.1 Delphi 5 中的数据感知组件 55
5.3.2 使用 TDBGrid 57
5.3.3 使用 TDBNavigator 60
5.3.4 使用 TDBText 61
5.3.5 使用 TDBEdit 61
5.3.6 使用 TDBMemo 61
5.3.7 使用 TDBImage 62
5.3.8 使用 TDBListBox 62
5.3.9 使用 TDBComboBox 63
5.3.10 使用 TDBCheckBox 63
5.3.11 使用 TDBRadioGroup 64
5.3.12 使用 TDBLookupListBox 64
5.3.13 使用 TDBLookupComboBox 66
5.3.14 使用 TDBRichEdit 68
5.3.15 使用 TDBCtrlGrid 69
5.4 运用 Database Form Wizard 71
5.5 TDataSource 组件 78
5.5.1 使用 TDataSource 组件 78
5.5.2 TDataSource 组件的属性 78
5.5.3 TDataSource 组件的事件 80

第 6 章 TDatabase 组件及事务 82
6.1 TDatabase 组件概述 82

6.1.1 TDatabase 的关键属性 82
6.1.2 TDatabase 组件的设置 84
6.1.3 TDatabase 组件在事务处理方面的几个方法 86
6.2 连接到数据库服务器 86
6.3 事务 87
6.3.1 使用数据库组件来处理事务 87
6.3.2 使用 Passthrough SQL 方式处理事务 88

第 7 章 数据集 TDataSet 90
7.1 数据集的状态 90
7.1.1 DsInactive 状态 91
7.1.2 DsBrowse 状态 91
7.1.3 DsEdit 状态 92
7.1.4 DsInsert 状态 92
7.1.5 DsSetKey 状态 92
7.1.6 DsCalcFields 状态 92
7.1.7 DsFilter 状态 93
7.1.8 DsCurValue、DsNewValue 和 DsOldValue 93
7.2 数据集的打开和关闭 93
7.3 数据集的导航 94
7.3.1 First 和 Last 95
7.3.2 Prior 和 Next 95
7.3.3 MoveBy 95
7.4 搜索特定记录 96
7.5 筛选 99
7.6 数据集的修改 101
7.7 书签 103
7.8 数据集组件的事件 105

第 8 章 字段对象 106
8.1 永久字段对象 106
8.1.1 创建永久字段对象 107
8.1.2 删除永久字段对象 109
8.2 字段组件的类型 110
8.3 字段组件的重要属性 111
8.4 字段组件的重要方法 112
8.5 字段组件的重要事件 113
8.6 字段组件的访问 114

第 9 章 TTable 组件及应用 116
9.1 TTable 组件常用的属性 116
9.2 使用范围选取部分连续记录 118
9.3 查找数据表中的记录 120
9.3.1 Goto 方式 120
9.3.2 Find 方式 121

9.4 EmptyTable、DeleteTable 和 RenameTable 方法 122
9.5 MasterSource 属性和 MasterFields 属性 124

第 10 章 TQuery 组件及其使用 127
10.1 TQuery 组件与 TTable 组件的区别 127
10.2 TQuery 组件可以查询的数据库 127
10.3 SQL 语言概述 128
10.3.1 SQL 的历史 128
10.3.2 SQL 的优点 129
10.3.3 常用的 SQL 语句及其用法 129
10.3.4 SQL 语言编程 131
10.4 使用 TQuery 组件 131
10.5 设置 TQuery 组件的 SQL 属性 132
10.5.1 字符串列表编辑器 132
10.5.2 SQL Builder 工具 134
10.5.3 直接编写代码 136
10.6 动态 SQL 语句中参数的赋值 138
10.6.1 在设计期利用 Params 属性为参数赋值 139
10.6.2 运行期给参数赋值 140
10.7 执行查询 141
10.8 异构查询 142
10.9 使查询结果集能够编辑 143

第 11 章 存储过程与 TStoredProc 组件 144
11.1 TStoredProc 的重要属性 144
11.2 TStoredProc 的重要方法 145
11.3 使用 TStoredProc 组件 147
11.3.1 在应用程序中使用 TStoredProc 组件 147
11.3.2 给存储过程的输入参数赋值，访问输出参数 147
11.3.3 执行存储过程 148
11.3.4 访问输出参数和结果集 148

第 12 章 批量移动与 TBatchMove 组件 149
12.1 使用 TTable 组件的 BatchMove 函数实现批量移动 149
12.2 使用 TBatchMove 组件 149
12.3 映射字段类型 150
12.4 执行批量移动操作 151
12.5 处理错误 151
12.6 应用举例 152

第 13 章 缓存更新与 TUpdateSQL 组件 161
13.1 应用缓存更新应注意的问题 161
13.2 使用缓存更新 161
13.2.1 允许和禁止缓存更新 162
13.2.2 读取记录 162

13.2.3 申请缓存更新 163
13.2.4 取消未决更新 165
13.2.5 恢复删除的缓存记录 165
13.2.6 指定缓存中的可见记录 166
13.2.7 检验更新状态 167
13.3 TUpdateSQL 组件 168
13.4 应用举例 168

第 14 章 多层客户／服务器应用程序 172
14.1 MIDAS 的相关概念 172
14.1.1 数据库编程的分类 172
14.2 如何创建多层应用程序 173
14.2.1 多层体系结构的优势 173
14.2.2 MIDAS 技术 174
14.2.3 Provider 组件 174
14.2.4 MIDAS 应用程序是怎样工作的 175
14.3 各层结构和连接关系 175
14.3.1 客户程序的结构 175
14.3.2 应用服务器的结构 176
14.3.3 连接方式 176
14.4 创建应用服务器的一般步骤 177
14.5 建立一个示例程序 178
14.5.1 创建一个工程 178
14.5.2 注册应用服务器（Application Server） 180
14.5.3 建立客户应用程序（Client Application） 181
14.5.4 客户/服务器的运行 182
14.6 应用程序服务器的设计 183
14.6.1 应用程序服务器的结构 183
14.6.2 客户应用程序的结构 184
14.6.3 基于 MIDAS 的多层应用程序的执行过程 185

第 15 章 报表设计 187
15.1 QuickReport 简介 187
15.2 QuickReport 组件简介 188
15.2.1 TQuickRep 组件 188
15.2.2 TQRSubDetail 组件 191
15.2.3 TQRBand 组件 192
15.2.4 TQRChildBand 组件 194
15.2.5 TQRGroup 组件 194
15.2.6 TQRLabel 组件 194
15.2.7 TQRDBText 组件 194
15.2.8 TQRExpr 组件 195
15.2.9 TQRSysData 组件 197
15.2.10 TQRMemo 组件 198
15.2.11 TQRRichText 组件 198

15.2.12 TQRDBRichText 组件 198
15.2.13 TQRShape 组件 198
15.2.14 TQRImage 组件 198
15.2.15 TQRDBImage 组件 198
15.2.16 TQRCompositeReport 组件 199
15.2.17 TQRPreview 组件 199
15.2.18 TQRChart 组件 199
15.3 快速生成报表的两种方法 200
15.3.1 QuickReport Wizard 快速报表生成向导 200
15.3.2 Report Templates 模板 203
第 16 章 TeeChart 图表的设计 217
16.1 制作 TeeChart 图表的一般方法 217
16.2 快速设计 TeeChart 图表的两种方法 217
16.2.1 TeeChart 向导 217
16.2.2 利用模板开发 TeeChart 图表 220
16.3 创建真正的数据库图表 220
16.3.1 图表编辑器 220
16.3.2 创建数据库图表的步骤 222
16.4 创建决策图表 224
第 17 章 Decision Cube 组件 226
17.1 使用决策支持组件的一般步骤 226
17.2 Decision Cube 组件组简介 227
17.2.1 Decision Cube 组件 227
17.2.2 Decision Query 组件 230
17.2.3 Decision Grid 组件 232
17.2.4 Decision Graph 组件 238
17.2.5 Decision Pivot 组件 242
17.2.6 Decision Source 组件 244
第 18 章 使用 ADO 组件 245
18.1 连接 ADO 数据源 245
18.1.1 两种连接方法的对比 245
18.1.2 指定连接 246
18.1.3 激活和解除连接 246
18.1.4 查看连接组件的状态 247
18.1.5 有规则地调整连接 247
18.1.6 列出可用的表 250
18.1.7 列出可用的存储过程 251
18.1.8 运用连接组件进行事务处理 251
18.2 ADO 数据集组件的共同特征 252
18.2.1 连接数据源 252
18.2.2 使用批处理更新 253
18.2.3 用批处理更新模式打开数据集 253
18.2.4 查看单行记录的更新状态 254

需要全本请访问 www.ertongbook.com

18.2.5 筛选处于更新状态的多行记录 254
18.2.6 把批处理更新写到基表中 255
18.2.7 取消批处理更新 255
18.2.8 在文件中存取数据 255
18.2.9 在命令中使用参数 256
18.3 使用 ADO 数据集组件和 ADO 命令组件 257
18.3.1 使用 TADODataSet 组件 257
18.3.2 使用 TADOTable 组件 258
18.3.3 使用 TADOQuery 组件 259
18.3.4 使用 TADOStoredProc 组件 261
18.3.5 使用 TADOCommand 组件 262
18.4 实例剖析 264
18.5 TRDSConnection 组件 268
18.5.1 服务器端应用程序分析 269
18.5.2 客户端应用程序分析 270

第 19 章 IBX 组件 273
19.1 IBX 组件简介 273
19.1.1 TIBDatabase 273
19.1.2 TIBTransaction 274
19.1.3 TIBTable 274
19.1.4 TIBQuery 275
19.1.5 TIBDataSet 275
19.1.6 TIBStoredProc 276
19.1.7 TIBDataSource 276
19.1.8 TIBSQL 276
19.1.9 TIBUpdateSQL 277
19.2 IBX 的优势 277

第 20 章 应用实例：考试系统 278
20.1 计算机考试系统概述 278
20.1.1 考试系统实现功能概述 278
20.1.2 与数据库关系密切的模块 279
20.2 数据库结构设计 279
20.2.1 表 tiku 280
20.2.2 表 km 281
20.2.3 表 chapt 281
20.2.4 表 nd 282
20.2.5 表 tx 282
20.2.6 表 exampaper 283
20.2.7 表 student 284
20.2.8 视图 tikubrowse_view 284
20.3 模块介绍及其源代码 284
20.3.1 计算机考试系统主窗口 285
20.3.2 数据模块 287

20.3.3 **添加模块**.....290
20.3.4 **修改模块**.....295
20.3.5 **浏览模块**.....298
20.3.6 **查询模块**.....300
20.3.7 **学生注册模块**.....303
20.3.8 **选择试题模块**.....305
20.3.9 **模拟考试模块**.....306
20.4 **运行效果**.....311
20.4.1 **主窗口**.....311
20.4.2 **添加题**.....312
20.4.3 **修改题**.....312
20.4.4 **浏览题**.....312
20.4.5 **查询题**.....313
20.4.6 **学生注册**.....313
20.4.7 **试卷选择**.....314
20.4.8 **模拟考试**.....314
20.4.9 **考试结果**.....315

第 1 章　Delphi 5 简介

Delphi 5 是 Inprise 公司推出的开发工具 Delphi 4 的升级版本，它在很多方面都有很大的改进，这使开发人员在开发程序时更加方便、快捷。本章简要地回顾一下 Delphi 的发展情况，并着重谈谈 Delphi 5 的新特性。

1.1　Delphi 的产生和发展

Delphi 以它强大的新特性，越来越多地被程序员选用，本节先介绍 Delphi 的发展，以便对该软件的性能有所了解。

1.1.1　从 Delphi 1 发展到 Delphi 3

在 Windows 产生之后，面向 Windows 的应用程序开发变得越来越重要，随之出现了许多运行于 Windows 环境下的开发工具。 BORLAND 公司推出了 Delphi 1.0，它是一个真编译系统，开发出的可执行文件的效率比较高，相当于用 C/C++开发的软件的运行速度，而且打包发布也非常方便。此外，它的数据库功能非常强大，提供的数据库引擎使用户能够快速方便地管理平台和网络数据库。

1996 年 1 月，BORLAND 公司推出了 Windows 应用程序开发工具 Delphi 2.0，仍采用面向对象结构，并增加了 32 位优化的编译器。它针对不同应用，又为个体编程开发人员提供了 Delphi Desktop 版，为基于局域网和进行本地开发的专业人员提供了 Delphi Developer 版，为建立客户机/服务器应用程序的开发人员提供了 Delphi 客户机/服务器版（其中 Client/Server Suite 版功能最全面）。

- 集中式数据字典

在 Delphi 2.0 中提供了一个新的工具：SQL EXPLORE，该工具使其集中式数据字典的管理功能有了显著的提高。在 Delphi 中，使用它自带的驱动程序或 ODBC 驱动程序很容易与后端数据库如 Informix，InterBase，SQL Server，Oracle 或 Sybase 相连。SQL EXPLORE 将后端数据库的定义集中起来，并能检查多服务器定义，能够浏览包括表、域、索引、数据类型和用户在内的大部分数据对象。可以查看、查询和修改数据库中的数据，其并发控制功能很强，不允许多用户同时修改同一数据。Delphi 还将数据字典作为 SQL EXPLORE 的一个函数来提供，数据字典可以将数据库中每一个字段的属性和业务规则集中在一起。

- **对象的可重用性**

在这方面 Delphi 要比 Visual Basic（一般简称为 VB）强得多。其应用程序的重用非常容易，它可以继承一个表单中的所有代码、对象和属性，这样可以保证对父表单进行的修改能立即被其他表单所继承。

- **存取多种数据库**

Delphi 有很强的对多种数据库的存取支持。和 VB 一样，它不仅允许应用程序同时与多个数据库连接，而且在服务器与客户机之间的数据源的位置和连接方法对应用程序也是透明的。一个应用程序能与数据库连接的数目没有限制。除 Delphi 自身提供的数据库驱动程序外，还可通过 BDE 增加第三方的驱动程序。

- **报表**

Delphi 的报表功能极强，因为它有一个与 Delphi 紧密集成的独立报表工具 Report Smith，他可支持各种表格，如简单报表、复杂报表、纵横表和图表等的生成和输出。

- **开发效率**

Delphi 和其他产品一样，可提供源码级的调试，能够检查编码、设置断点和显示执行序列。Delphi 支持 OLE 自动化并能与定制的控件合并。和其他产品相同，Delphi 通过例外处理程序支持错误处理。例外处理程序是通过默认处理程序的定义改善编码健壮性的好办法。Delphi 提供 InterSolv 公司的源码版本管理工具 PVCS（Version Control Manager），开发人员利用 PVCS 可检查项目信息和项目源码状态。

1997 年，BORLAND 公司又推出了 Delphi 3.0，这一版本与前期版本相比，在数据库的体系结构与连接、ActiveX 控件、Web 应用程序包等方面都做了改进，因而功能更强大，使用也更为灵活和方便，大大提高了应用程序的开发效率。

- **集成化的开发环境**

可以在一个集成的环境中完成应用程序的开发、测试和维护。

- **优秀的界面设计功能**

提供了大量用于设计界面的控件，用它们可以设计出各种优秀的用户界面。而且 Delphi 还可以自动生成众多的定制界面，极大地简化了界面设计工作。

- **强大的数据库功能**

Delphi 3 提供了许多与数据库相关的控件，用它们可以快速开发功能强大的数据库应用程序。内置的数据库引擎(BDE)支持访问 Paradox，Access，FoxPro，dBASE，Informix，Oracle，Sybase 和 SQL Server 等数据库。

- **支持 Internet 和 Intranet 开发**

Delphi 3 的 Client/Server 版本提供了几个用于开发 Internet 和 Intranet 应用程序的组件。这些组件可以处理与 Web 服务器的所有通信任务。

- **支持 ActiveX 控件**

Delphi 3 提供了创建标准 ActiveX 控件的向导。所创建的 ActiveX 控件可以用于其他开发工具，如 C++，JAVA，Visual Basic 或 PowerBuilder。

1.1.2　Delphi 4

1998 年推出的 Delphi 4，采用当时流行的分布式和多层应用的编程思想，为不同体系结构的应用开发提供了强大的支持环境。在此，先简要介绍 Delphi 4 新增强的功能。

1. 语言的扩充

Delphi 4 扩展了面向对象的 Pascal 语言，包括动态数组数据类型、方法重载、函数或过程的默认参数、64 位整数类型、32 位无符号整数类型、实数类型位数扩展等等。

- 动态数组数据类型

Delphi 4 不仅定义了静态数组，同时也定义了动态数组。动态数组只指定了数组的数据类型，没有指定元素的个数，如：

```
A: array[1..100] of string;        { 静态数组}
A: array of integer;          {       动态数组}
B: array of array of string;
```

其中，A 是一维整数数组，B 是二维字符串数组。

- 方法、函数、过程重载

所谓重载，是指一个对象有一个以上具有相同名字的构造函数，如方法、函数、过程，应用程序会根据参数表匹配调用相应的重载构造函数。例如：

```
type
 T1 = class （TObject）
  procedure Test （I: Integer）; overload; virtual;
 end;
 T2 = class （T1）
  procedure Test （S: string）; reintroduce; overload;
 end;
   …
 SomeObject := T2.Create;
 SomeObject.Test （'Hello!'）; //  由于参数类型是字符串，因而调用 T2.Test
 SomeObject.Test （7）;       //    由于参数类型是整型，因而调用 T1.Test
```

- 函数或过程的默认参数

在 Delphi 4 中可以在过程或函数中定义默认参数，默认参数只能出现在参数表的最后，形式如下：

```
Name: Type = value
```

当应用程序调用一个具有默认参数的过程或者函数时，默认参数值可以省略。例如有一个具有默认参数的过程 FillArray 声明如下：

```
procedure FillArray（A: array of Integer; Value: Integer = 0  ）;
```

那么，下面的两种调用方法效果是一样的：

```
FillArray（MyArray, 0）;
FillArray（MyArray）;
```

注意：当同时使用方法重载和默认参数时，不要让语义出现二义性。例如，在应用程序中有两个声明：

```
procedure P（I: Integer; J: Integer = 0  ）; overload;
procedure P（Size: Integer）; overload;
```

那么，下面的调用是不合法的，因为编译器无法确定应该调用哪个过程：

```
P（1）;
```

- 64 位整数类型

  为了处理比通用 32 位 Longint 整型更大的值，Delphi 4 增加了 64 位整数类型，它的值域是[-2^{63}， $2^{63}-1$]，相当于十进制[-9.22×10^{18}，9.22×10^{18}]。

- 32 位无符号整数类型

  Delphi 4 增加了真 32 位无符号整数类型，值域是[0，4 294 967 295]。

- 实数类型位数扩展

  原来的实数是 48 位浮点型，现在扩展为 64 位双精度型，由于双精度与 Intel CPU 系列兼容，因此提高了应用程序的性能。

2. IDE 的增强

- 工程管理器的改进

  新的工程管理器能够将多个相互作用的工程组合成为一个项目组，从而大大方便了对多层应用程序或复杂的分布式应用工程的管理。

- 代码浏览器

  代码浏览器使得类的创建、模块导航、代码浏览容易很多，其中包括类自动完成特性。

  由于 Delphi 4 是一个完全面向对象的编程工具，编程过程中经常要声明和实现类（Class），而类的创建是比较麻烦的，即使是 Delphi 编程高手，也会对过分严谨和古板的 Pascal 语法感到枯燥乏味。而使用“类自动完成”向导可以帮助快速生成有关类的代码。

  “类自动完成”功能允许用户只编写对象接口，然后按 Ctrl+Shift+C 键，以自动创建方法的框架、设置属性值以及定义私有数据。

- 模块导航

  把光标放在方法或者过程的全局声明接口（interface）或者实现（implementation）

需要全本请访问 www.ertongbook.com