Delphi
数据库开发与应用

暴风雪科技　主编

上海科学普及出版社

需要全本请访问 www.ertongbook.com

内 容 提 要

Delphi 7.0 是 Borland 公司最新推出的新一代可视化开发工具，它具有功能强大、简便易用和代码执行速度快等优点。业界人士一致认为：真正的程序员用 C，聪明的程序员用 Delphi。

本书注重开发实例、开发经验和 Windows 高级特性开发的介绍，适合于各个层次的用户。对初学者来说，可迅速掌握 Delphi 7.0 的数据库开发过程和方法；对有一定使用经验的读者，也可通过本书掌握 Delphi7.0 深层次的开发方法，开发出更优秀的数据库应用程序。

本书同样也适合各院校作为教材使用。

图书在版编目（CIP）数据

Delphi 数据库开发与应用/暴风雪科技主编.—上海：
上海科学普及出版社，2003.11
ISBN 7-5427-2600-5

Ⅰ．D… Ⅱ．暴… 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 097850 号

策　　划　铭　政
责任编辑　徐丽萍

Delphi 数据库开发与应用
暴风雪科技　主编
上海科学普及出版社出版发行
（上海中山北路 832 号　邮政编码 200070）

各地新华书店经销			北京富生印刷厂印刷
开本 787×1092	1/16	印张 25.5	字数 688000
2004 年 3 月第 1 版			2004 年 3 月第 1 次印刷

ISBN 7-5427-2600-5/TP・492　　定价：39.00 元

前　言

随着计算机技术的发展，计算机软件行业也随之迅速崛起，计算机软件的开发也迅速得到普及推广。自从面向对象的编程概念提出之后，可视化编程的技术应运而生，相继出现了 Visual C++，Visual Basic，C++ Builder 和 Delphi 等可视化应用程序开发工具。

Delphi 是由软件世界巨头、著名的 Borland 公司开发的“可视化”程序开发工具，它的基础语言为 Pascal 程序设计语言，它继承了 Pascal 语言的优点，如代码结构清晰、可读性好和代码执行效率高等，所以 Delphi 一经推出，迅速得到用户的好评。“真正的程序员用 C，聪明的程序员用 Delphi”，这句话是对 Delphi 最经典、最实在的描述。

数据库系统为我们提供了一种把与我们的工作和生活紧密相关的信息集合在一起的方法，它还提供了在某个集中的地方存储和维护这些信息的方法。数据库系统主要由三大部分组成：数据库管理系统（DBMS 是专门负责组织和管理数据信息的程序）、数据库应用程序（它使我们能够获取、显示和更新由 DBMS 存储的数据）、数据库（按一定结构组织在一起的相关数据的集合）。

一般来说，DBMS 和数据库应用程序都驻留在同一台计算机上并在同一台计算机上运行，很多情况下两者甚至结合在同一个程序中，以前使用的大多数数据库系统都是用这种方法设计的。但是随着 DBMS 技术的发展，目前的数据库系统正向客户/服务器模式发展。客户/服务器数据库将 DBMS 和数据库应用程序分开，从而提高了数据库系统的处理能力。数据库应用程序运行在一个或多个用户工作站（客户机）上，并且通过网络与运行在其他计算机上（服务器）的一个或多个 DBMS 进行通信。

目前，数据库应用已经涉及到社会的各个领域，编写数据库应用程序已成为面向对象语言的重要用途之一，为了满足广大读者的愿望，迅速掌握 Windows 环境下的 Delphi 数据库编程方法，我们编写了这本《Delphi 数据库开发与应用》，本书以 14 章的内容从各个方面讲述了 Delphi 在数据库方面的强大功能。

关于本书

全书共分十四章，各章的内容如下：

第一章：Delphi 7.0 概述，主要介绍了 Delphi 7.0 的新特性及其集成开发环境，以及基本的 Pascal 语言知识和结构化异常处理。

第二章：Delphi 7.0 数据库开发基础，主要介绍了数据库的基本概念、数据的存取、数据库管理系统的特点以及如何开发一个数据库应用程序。

第三章：Delphi 7.0 的数据库环境，主要介绍了 Delphi 7.0 的数据库工具，包括 Database Desktop、BDE、SQL Explorer、SQL Monitor 和 Datapump。

第四章：Delphi 7.0 的 BDE 访问组件，主要介绍了 Delphi 7.0 的 BDE 数据访问组件的方法、属性和事件的使用方法，同时对 TTable 组件、TQuery 组件和 TStoredProc 组件的使用做了具体的说明。

第五章：ADO 数据访问组件，主要介绍了 Delphi 7.0 中 ADO 组件的特点及其使用方

法，主要包括 TADOConnection 组件和 TADOCommand 组件。

第六章：数据浏览组件，主要介绍了 Delphi 7.0 的 DataControls 组件的属性和用法，主要包括 DBGrid、DBNavigator、DBText、DBEdit 等组件。

第七章：Decision Cube 组件，主要介绍了 Delphi 7.0 Decision Cube 组件的应用。

第八章：QuickReport 的报表制作，主要介绍了如何用 QuickReport 组件制作报表。

第九章：SQL 结构化查询语句，主要介绍了 SQL 的基础知识以及它在 Delphi 7.0 中的应用。

第十章：Delphi 7.0 的 InterBase 应用，主要介绍了 Delphi 7.0 的 InterBase 数据库知识和应用。

第十一章：Delphi 7.0 的客户/服务器开发，主要介绍了 Delphi 7.0 客户/服务器结构的基本原理和应用。

第十二章：Delphi 7.0 的多层应用程序，主要介绍了 Delphi 7.0 多层数据库结构的优势、原理以及如何利用 Delphi 7.0 来构建一个有效的多层应用程序。

第十三章：综合实例 1，比较简单的数据库管理系统——商店管理系统，实现了商品的出租和出售、商品信息查询、客户和商品信息管理、统计报表生成等功能。

第十四章：综合实例 2，主要通过一个较为完整的数据库管理系统——进销存管理系统，来进一步让读者掌握 Delphi 7.0 的数据库开发。

关于源代码

为了配合读者更好的领会书中各个开发实例，我们将书中讲解的各个例子的源程序和编译执行后的可执行文件放在http://www.baofengxue.com网站，欢迎读者下载。

关于作者

本书由暴风雪科技公司主编，本书作者有着多年的 Delphi 使用及编程经验，希望这本书能够使无论 Delphi 的编程新手，还是有编程经验的用户，都有不同的收益。虽然作者投入了大量的时间和精力进行编写、创作，但百密难免一疏，其中不免有许多疏漏之处，还恳请广大的热心读者批评指正。

编者

E-mail：leeworks@263.net

目　录

第 1 章　Delphi 7.0 概述……1
1.1　Delphi 7.0 的新特性……2
1.2　Delphi 7.0 的集成开发环境（IDE）…3
1.2.1　主窗口……4
1.2.2　组件面板……4
1.2.3　代码浏览器……4
1.2.4　代码编辑器……4
1.2.5　窗体设计器……5
1.2.6　工程管理器……5
1.2.7　对象观察器……6
1.3　面向对象的 Pascal 语言……6
1.3.1　Object Pascal 面向对象编程……6
1.3.2　Object Pascal 的数据类型……7
1.3.3　Object Pascal 操作符及表达式…8
1.3.4　Object Pascal 语句……9
1.3.5　Object Pascal 过程与函数……12
1.4　结构化异常处理……13
第 2 章　Delphi 7.0 数据库开发基础……15
2.1　数据库的基本概念……15
2.1.1　数据库的建立……15
2.1.2　数据表格的结构……16
2.2　数据存取……16
2.2.1　TDataSource 组件……16
2.2.2　TClientDataSet 组件……20
2.3　数据库应用程序的开发……20
2.3.1　初步设计……20
2.3.2　功能实现……20
2.3.3　运行和维护程序……21
2.4　Delphi 7.0 数据库管理系统的特点…21
2.4.1　将 Object Pascal 语言与关系数据库相结合……21
2.4.2　通过对象管理数据……21
2.4.3　吸取了当前最先进的同数据库的连接技术……22
2.4.4　采用开放式平台共享数据库……22
2.4.5　强大的网络数据库功能……22
第 3 章　Delphi 7.0 的数据库环境……23
3.1　数据库工作平台（Database Desktop）……23
3.1.1　设置工作目录……24
3.1.2　创建数据表格……24
3.1.3　数据表格的属性……26
3.2　数据库引擎（BDE）…… 29
3.2.1　BDE 操作界面……29
3.2.2　维护数据库别名……30
3.3　SQL 资源管理器（SQL Explorer）…32
3.3.1　数据库资源管理器（Database Explorer）……32
3.3.2　数据字典（Data dictionary）35
3.4　SQL 监视器（SQL Monitor）……36
3.5　数据转移工具（Datapump）……36
第 4 章　Delphi 7.0 的 BDE 组件……39
4.1　BDE 组件页的简介……40
4.2　TTable 组件介绍……40
4.2.1　数据集组件基本知识……41
4.2.2　TTable 主要的属性……41
4.2.3　TTable 组件的重要方法……47
4.2.4　TTable 组件的重要事件……58
4.3　TQuery 组件介绍……59
4.3.1　TQuery 组件和 TTable 组件的比较……59
4.3.2　TQuery 组件的重要属性……59
4.3.3　TQuery 组件的重要方法……62
4.4　TStoredProc 组件介绍……64
4.4.1　TStoredProc 组件的重要属性…64
4.4.2　TStoredProc 组件的重要方法…65
4.4.3　利用 TStoredProc 组件实现存储过程……67
4.5　TDatabase 组件介绍……67
4.5.1　TDatabase 组件的重要属性……68
4.5.2　TDatabase 组件的重要方法……70
4.5.3　TDatabase 组件的重要事件……71
4.6　TSession 组件介绍……71
4.6.1　TSession 的重要属性……71
4.6.2　TSession 的重要方法……72

需要全本请访问 www.ertongbook.com

4.7 其他组件介绍……73
4.7.1 TBatchMove 组件……73
4.7.2 TUpdateSQL 组件……73
4.7.3 TNestedTable 组件……73
4.8 BDE 主要访问组件的应用……74
4.8.1 利用 TTable 访问表格……74
4.8.2 利用 TQuery 组件查询数据库…81
4.8.3 利用 TstoredProc 组件执行存储过程……85
第 5 章 ADO 数据访问组件……89
5.1 Microsoft 数据访问简介……89
5.2 ADO 组件页的介绍……90
5.2.1 TADOConnection 组件……91
5.2.2 TADOCommand 组件……92
5.2.3 TADODataSet 组件……95
5.2.4 TADOTable 组件……96
5.2.5 TADOQurey 组件……98
5.2.6 TADOStoredProc 组件……101
5.2.7 TRDSConnection 组件……101
5.3 应用 ADO 操作数据……101
第 6 章 数据浏览组件……105
6.1 数据浏览组件概述……105
6.2 数据浏览组件的基本特征……107
6.3 DBGrid 组件……108
6.3.1 应用……108
6.3.2 基本属性……110
6.3.3 基本事件……113
6.4 DBNavigator 组件……114
6.4.1 应用……114
6.4.2 基本属性……115
6.4.3 基本方法和事件……116
6.5 DBText 组件……117
6.5.1 应用……117
6.5.2 基本属性……119
6.6 DBEdit 组件……120
6.6.1 应用……120
6.6.2 主要属性……122
6.7 DBMemo 组件……122
6.7.1 应用……122
6.7.2 主要属性及方法……123
6.8 DBImage 组件……124
6.8.1 应用……124
6.8.2 主要属性及方法……126
6.9 DBComboBox 组件……126
6.9.1 应用……126
6.9.2 基本属性……127
6.10 DBListBox 组件……128
6.10.1 应用……128
6.10.2 基本属性……131
6.11 TDBCheckBox 组件……131
6.12 TDBRadioGroup 组件……134
6.13 TDBLookupListBox 组件……137
6.13.1 应用……137
6.13.2 主要属性……139
6.14 TDBLookupComboBox 组件……140
6.15 TDBCtrlGrid 组件……142
6.16 TDBRichEdit 组件……146
6.17 TDBChart 组件……146
第 7 章 Delphi 7.0 的数据仓库应用……147
7.1 数据仓库概述……147
7.1.1 数据仓库的产生……147
7.1.2 数据仓库技术……148
7.2 DecisionCube 组件……149
7.2.1 DecisionCube 组件的属性……150
7.2.2 DecisionCube 组件的事件……151
7.2.3 DecisionCube 组件的方法……151
7.3 DecisionQuery 组件……152
7.3.1 DecisionQuery 组件的属性……153
7.3.2 DecisionQuery 组件的方法……153
7.3.3 DecisionQuery 组件的事件……154
7.4 Decision Source 组件……154
7.5 Decision Pivot 组件……154
7.6 Decision Grid 组件……154
7.6.1 Decision Grid 组件的属性……154
7.6.2 Decision Grid 组件的方法……155
7.6.3 Decision Grid 组件的事件……155
7.7 DecisionGraph 组件……156
7.8 一个使用 Decision Cube 组件的例子……157
第 8 章 QuickReport 的报表制作……163
8.1 QuickReport 组件的概述……163
8.2 QuickReport 组件页的介绍……164
8.2.1 TQuickRepr 组件……164
8.2.2 TQRSubDetail 组件……167

8.2.3 TQRBand 组件……168
8.2.4 TQRChildBand 组件……169
8.2.5 TQRExpr 组件……169
8.2.6 TQRSysData 组件……171
8.2.7 TQRGroup 组件……172
8.2.8 TQRLabel 组件……172
8.2.9 TQRDBText 组件……172
8.2.10 TQRMemo 组件……172
8.2.11 TQRRichText 组件……172
8.2.12 TQRDBRichText 组件……172
8.2.13 TQRShape 组件……173
8.2.14 TQRImage 组件……173
8.2.15 TQRDBImage 组件……173
8.2.16 TQRCompositeReport 组件…173
8.2.17 TQRPreview 组件……173
8.2.18 TQRChart 组件……173
8.3 使用 QuickReport 组件编程……173
8.3.1 设计报表实例……173
8.3.2 应用向导快速生成报表……192
第 9 章 Delphi 7.0 的 SQL 编程……195
9.1 SQL 的历史……195
9.2 SQL 语句简介……196
9.2.1 数据定义……196
9.2.2 数据查询……198
9.2.3 数据更新……199
9.3 SQL 的优点……200
9.4 SQL 在 Delphi 7.0 中的应用……200
9.4.1 TQuery 组件的使用……201
9.4.2 在 Tquery 组件中编写简单的 SQL 查询命令……202
9.5 SQL 语言编程概述……204
9.5.1 SQL 命令文本的编写……205
9.5.2 SQL 程序的执行……207
9.5.3 通过 TQuery 组件如何获得活动的数据……208
9.6 动态 SQL 语句的编程……209
9.6.1 使用 Params 属性赋值……209
9.6.2 使用 ParamByName 方法赋值 210
9.6.3 使用 Datasource 属性赋值……210
9.6.4 Prepare 方法的使用……213
9.7 服务器端 SQL 编程……214
9.7.1 存储过程编程……214
9.7.2 触发器编程……215
9.8 SQL 编程实例……215
第 10 章 Delphi 7.0 的 InterBase 应用……219
10.1 InterBase 服务器……219
10.1.1 客户与服务器的定义……219
10.1.2 Borland 数据库引擎（BDE）…220
10.1.3 InterBase 数据库的管理和维护……220
10.1.4 服务器端程序……224
10.1.5 本地缓存……226
10.2 Delphi 7.0 的 InterBase 组件页……226
10.2.1 IBTable 组件……226
10.2.2 IBQuery 组件……226
10.2.3 IBStoredProc 组件……227
10.2.4 IBDataBase 组件……227
10.2.5 IBTransaction 组件……227
10.2.6 IBUpdateSQL 组件……227
10.2.7 IBDataSet 组件……227
10.2.8 IBSQL 组件……227
10.2.9 IBDataBaseInfo 组件……228
10.2.10 IBSQLMonitor 组件……228
10.3 InterBase 数据库……228
10.3.1 Local InterBase……228
10.3.2 InterBase Server……228
10.3.3 InterBase 的特性简介……229
10.3.4 InterBase 的约束……230
10.3.5 InterBase 语法规则……231
10.3.6 数据库对象命名规则……231
10.3.7 文件命名规则……231
10.3.8 文件说明……232
10.4 一个应用 InterBase 组件的例子…232
第 11 章 客户/服务器开发……239
11.1 Delphi 7.0 客户/服务器开发原理 239
11.1.1 客户/服务器体系结构……239
11.1.2 关系数据库体系结构与客户/服务器模式……240
11.1.3 各种数据库服务器功能介绍? 242
11.1.4 IDAPI 结构原理……245
11.2 Delphi 7.0 客户/服务器开发环境的构造……248
11.2.1 Borland SQL Links 的安装…248
11.2.2 配置 SQL 环境……249

11.3 Delphi 7.0 的 Client/Server 编程…252
11.3.1 定制数据库服务器的注册参数 252
11.3.2 建立应用程序特定的别名……252
11.3.3 控制数据库的联接……252
11.3.4 处理 Client/Server 事务控制…253
11.3.5 使用存储过程……255
11.3.6 从开发平台到服务器的向上适化……258
11.3.7 Delphi 7.0 的客户/服务器应用实例分析……259
第 12 章 Delphi 7.0 的多层应用程序……275
12.1 多层体系结构的优势……275
12.1.1 在共享的中间层封装了商业规则……276
12.1.2 瘦客户端……276
12.1.3 分布式数据处理……276
12.1.4 提高安全性……276
12.2 理解 DATASNAP 技术……276
12.2.1 多层应用程序的工作原理…277
12.2.2 应用程序服务器的结构……278
12.3 创建多层应用程序……281
12.3.1 创建应用程序服务器……281
12.3.2 创建客户端应用程序……290
12.4 自定义应用程序服务器……296
12.4.1 扩展应用程序服务器的接口 296
12.4.2 使用自定义的数据表……298
12.4.3 与状态无关的远程数据模块（Remote Data Module）支持……298
12.5 多层应用程序的事务处理管理…298
12.6 支持主从关系数据表……299
12.7 把客户端应用程序作为 ActiveX 控件来发布……299
12.7.1 把客户端应用程序作为 ActiveX 控件来发布……299
12.7.2 为客户端应用程序创建 ActiveForm……299
12.8 一个简单的分布式应用……301
第 13 章 综合实例 1——商店管理系统……307
13.1 系统简介……307
13.2 系统主要的功能模块……309
13.2.1 系统首页……309
13.2.2 查询功能……309
13.2.3 日常维护……309
13.2.4 信息管理……310
13.2.5 报表查看……310
13.3 系统各模块窗体分析……310
13.4 系统数据库表设计……317
13.4.1 经理表……318
13.4.2 员工表……318
13.4.3 顾客表……318
13.4.4 商品表……319
13.4.5 出租记录表……319
13.4.6 销售表……319
13.5 系统代码分析……319
13.5.1 功能选择页面……319
13.5.2 商品查询页面……321
13.5.3 日常管理的登录页面……333
13.5.4 日常维护页面……336
13.5.5 管理员登录页面……343
13.5.6 商品和顾客信息管理页面…345
13.5.7 商品出售页面……346
13.5.8 查看统计报表页面……348
第 14 章 综合实例 2——进销存管理系统……350
14.1 系统简介……350
14.2 系统主要的功能模块……350
14.2.1 系统设置……350
14.2.2 账簿登记……350
14.2.3 查看明细……351
14.2.4 账簿统计……351
14.2.5 利润分析……351
14.2.6 报表打印……351
14.2.7 帮助……352
14.3 系统各模块窗体分析……352
14.4 系统数据库表设计……356
14.4.1 客户表……357
14.4.2 员工表……357
14.4.3 进货商表……357
14.4.4 商品表……358
14.4.5 单号表……358
14.4.6 结款表……358
14.4.7 费用支出表……359
14.5 系统代码分析……359

第 1 章　Delphi 7.0 概述

Delphi 被称为是第四代编程语言，它具有简单、高效、功能强大的特点。和 VC 相比，Delphi 更简单、更易于掌握，而在功能上却丝毫不逊色；和 VB 相比，Delphi 则功能更强大、更实用。可以说 Delphi 同时兼备了 VC 功能强大和 VB 简单易学的特点。所以，它一直是程序员至爱的编程工具。

Delphi 具有以下的特性：基于窗体和面向对象的方法，高速的编译器，强大的数据库支持，与 Windows 编程紧密结合，强大而成熟的组件技术。但最重要的还是 Object Pascal 语言，它才是一切的根本。Object Pascal 语言是在 Pascal 语言的基础上发展起来的，简单易学。

Delphi 提供了各种开发工具，包括集成环境、图像编辑（Image Editor），以及各种开发数据库的应用程序，如 DesktopDataBase Expert 等。除此之外，还允许用户挂接其他的应用程序开发工具，如 Borland 公司的资源编辑器（Resourse Workshop）。

在 Delphi 众多的优势当中，它在数据库方面的特长显得尤为突出：适应于多种数据库结构，从客户机/服务机模式到多层数据结构模式；高效率的数据库管理系统和新一代更先进的数据库引擎；最新的数据分析手段和提供大量的企业组件。

Delphi 发展至今，从 Delphi 1.0、Delphi 2.0 到现在的 Delphi 7.0，Delphi 7.0 是当前 Windows 平台上全面支持最新 Web 服务的快速开发工具之一。无论是企业级用户，还是个人开发者，都能够利用 Delphi 7.0 轻松、快捷地构建新一代电子商务应用。Delphi 7.0 有着所有工具中最为灵活的数据库结构。对大多数基于本地、客户/服务器和 ODBC 数据库平台的应用程序来说，BDE 的功能都非常强大。如果你觉得这个仍然满足不了你的需求，那么同样你可以使用 Delphi 支持的 ADO 组件。因为 Dlephi 对 ADO 组件的支持来操作数据库一样的方便。大大地提高了应用程序的开发效率。



本章将就以下几个内容进行介绍：

- Delphi 7.0 的新特性
- Delphi 7.0 的集成开发环境（IDE）
- 面向对象的 Pascal 语言
- 结构化异常处理



本书假设读者已经使用过 Delphi 7.0 以前的版本，有一定的编程经验，对于一些基本的知识，本书不做具体介绍。

1.1 Delphi 7.0 的新特性

- **企业应用** MDA **开发**

让开发者在根本上减少从设计到开发所需的编码数量和时间，从而加速应用开发过程。

- RAD **可视** Web **开发**

在 Delphi 7 Studio **环境下，开发者能够进行可视** Web **应用开发，它的应用模式（**Application Mode**）框架能够在切断与服务器连接情况下进行透明的对话（**session**）管理。**

- **内建** Linux **跨平台支持器**

Delphi 7 Studio **装配** Delphi **语言版的** Borland Kylix 3**。第一个高性能的可视集成开发环境（**IDE**），可迅速创建数据库，图形用户接口（**GUI**），**Web **和** Linux **操作系统的** Web **服务应用。**

- **企业型报告**

可创建跨平台报告，帮助分析应用执行的效率。

- **免版税的** DataSnap**（前称多层应用开发（**Multi-tier Application Deployment**，**MIDAS**））**

新版的 Delphi 7 Studio DataSnap **许可证允许免除从单层和客户/服务模式无缝过渡到多层应用模式的附加费。**DataSnap **主要是多了** Connection Pooling.**连接池，本来在应用程序中通过代码实现，现在成为一个独立的控件，方便了用户的操作，**DataSnap **创建从客户端或服务连结到主数据库的中间组件，**DataSnap **支持** Oracle、MS-SQL Server、Informix、IBM DB2、SyBase **和** InterBase **等主流数据库。它遵循业界标准的** SOAP/XML HTTP **协议，使客户端无需数据驱动和复杂配置，即可直接与其相连接。**DataSnap **还支持** DCOM、CORBA、TCP/IP**。**

- Windows XP **应用**

支持 Windows XP **主题（**Theme**），开发者可以利用** Windows XP **用户接口主题（**User Interface themes**）来开发应用。**

- BizSnap **轻松创建基于** SOAP/XML **的** Web **服务**

Delphi 7.0 **无缝集成了基于** SOAP **的** Web **服务和** XML **数据交换技术，简化新一代电子商务的开发，是当前唯一在** Internet **上集成** Web **服务、**B2B、B2C **和** P2P **的快速开发工具，**Delphi 7.0 **中的** BizSnap **用来创建基于** XML **和** SOAP **的** Web Services **及连结，**WebSnap **基于组件的** Web **开发平台，对应当前火爆的电子商务，**Delphi 7.0 **也提供了更为强劲的支持，利用它可以开发出安全、稳定、高效的网络数据库应用程序。在基于** Web **的平台方面，除了** Microsoft **的**.NET **平台，**Delphi 7.0 **还支持** SUN **公司的** ONE **平台。**

- WebSnap **支持** Apache **和微软** IIS **等主流** Web **应用服务器**

InternetExpress **现在变成了** WebSnap **了，主要就是多了对** Snap **和** WebService **的支持。这对开发网络数据库提供了更多的方便及更高的效率，**WebSnap **将** Delphi 7.0 **应用及以当前流行的** HTML **开发环境如** Dreamweaver、Frontpage、VBScript **和** JavaScript**，所开发的网站无缝集成在一起。**

- **构建单一代码的** Windows/Linux **应用**

Delphi 7.0 **与** Kylix **兼容。使用** Kylix**，可在** Linux **平台上重新编译基于** Windows **平台的** CLX **应用；而利用** Delphi 7.0**，即可在** Windows **上重新编译基于** CLX **组件的** Linux **应用。**Delphi 7.0 **包含** BaseCLX、VisualCLX、DataCLX **和** NetCLX **四个组件。**

- Delphi 7.0 **与** AppServer **集成**

Delphi 7.0 **通过最新** SIDL **与** AppServer **连接。为** AppServer **应用开发出高性能、具有丰富** GUI **环境的客户端应用，通过** Internet **将** AppServer **的** EJB **功能作为遵循业界标准的** SOAP/XML Web **服务发布到全球。**

- Borland VisiBroker for Delphi 7.0 支持客户端与服务器端的完整开发

Delphi 7.0 为 AppServer 和 VisiBroker 应用开发出高性能、具有丰富 GUI 环境的客户端和 Web 浏览器。在复杂的 IT 环境下，Delphi7.0 所建立的 VisiBroker CORBA 对象能与任意 CORBA 客户端或对象进行互操作。

- 支持 Windows 2000/ME 和 Office 2000

通过 ActionBands、ActionManagers 和 Shell Controls 创建的用户界面支持微软最新平台， 使用户无需在繁琐的用户界面上耗费精力，便能轻松定制属于自己的 UI 应用。

Delphi 7.0 的新特性主要体现在对网络数据库应用的支持，以及对今后应用程序实现跨平台迈进了重要的一步。

1.2 Delphi 7.0 的集成开发环境（IDE）

Delphi 7.0 的 IDE 与 Delphi 以前的版本一样，使用一组窗口、菜单和应用程序来设计可视化界面，将代码以事件的形式与界面的每一元素建立联系，并可以对整个应用程序进行调试。Delphi 7.0 的 IDE 主要包括七部分：

- 主窗口
- 组件面板
- 代码浏览器
- 代码编辑器
- 窗体设计器
- 工程管理器
- 对象观察器

下面的图 1-1 是 Delphi 7.0 的运行界面，关于每个面板的功能下面还会详细介绍。

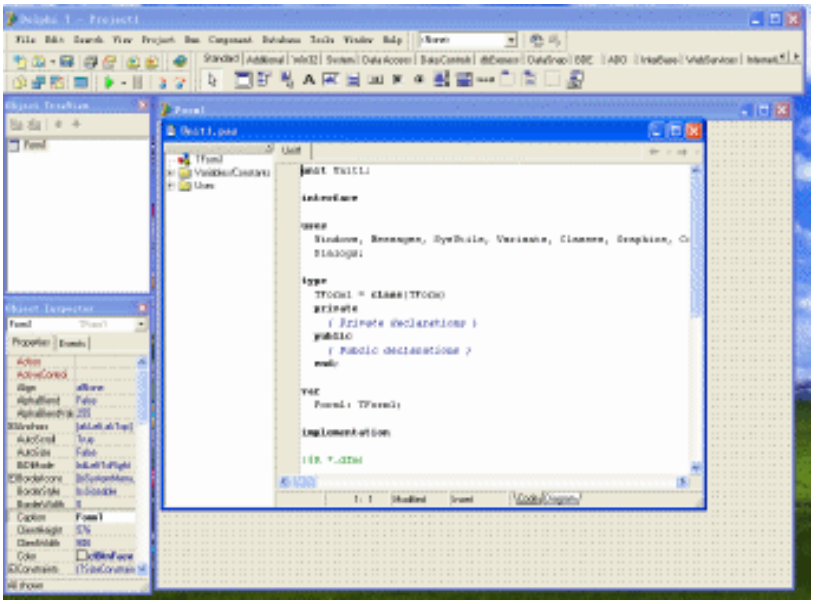


图 1-1 Delphi7.0 的运行界面

需要全本请访问 www.ertongbook.com

1.2.1 主窗口

主窗口可以认为是 Delphi 7.0 IDE 的核心，具有其他 Windows 应用程序的主窗口所具有的一切功能，如图 1-2 所示。



图 1-2 主窗口

1.2.2 组件面板

组件面板是一个多层工具栏，包含了在 IDE 中所有的 VCL 组件和 ActiveX 组件，如图 1-3 所示。



图 1-3 组件面板

1.2.3 代码浏览器

代码浏览器显示的是代码编辑的接口部分的结构组成，包括类定义部分、变量和常量的定义部分以及引用声明部分，如图 1-4 所示。

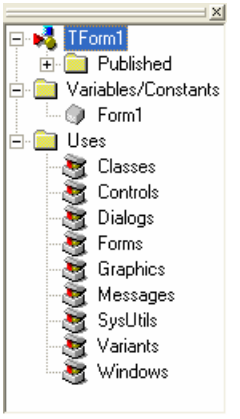


图 1-4 代码浏览器

1.2.4 代码编辑器

代码编辑器又称作单元窗口，是编写程序代码的地方。它一般隐藏在窗体设计器的下面。

我们可以通过以下几种方法来显示它：选择【View】⇨【Code Explorer】命令或者快捷键【F11】键或单击快捷工具栏中的 Form / Unit 转换快捷键，如图 1-5 所示。

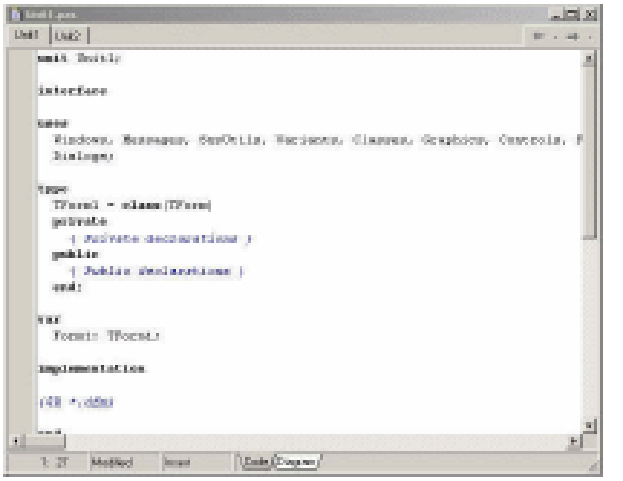


图 1-5 代码编辑器

1.2.5 窗体设计器

应用程序的用户界面是由窗体实现的，可以通过 Object Inspector 和代码编辑器来控制窗体以及窗体上组件的外观和行为，如图 1-6 所示。

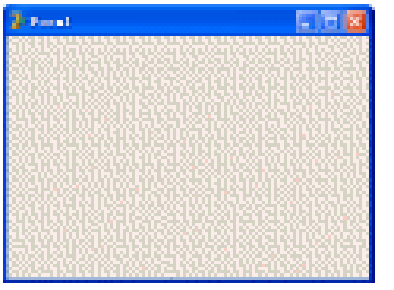


图 1-6 窗体编辑器

1.2.6 工程管理器

用于管理组成应用程序（工程）的多个文件，使各个文件、可执行文件间协调，如图 1-7 所示。

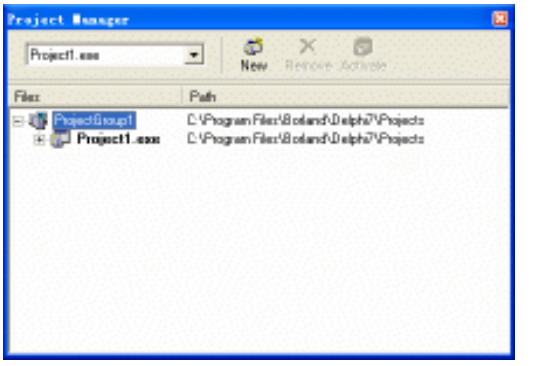


图 1-7 工程管理器

1.2.7 对象观察器

对象观察器是用来描述组件对象及窗体对象的属性特征和行为事件的。它是应用程序设计过程中最重要的一个工具。正是由于它将生动友好的可视化窗体与原本枯燥无味的程序代码联系到了一起，使得 Windows 应用程序的开发变得如此轻松。我们无需探究乏味的编程语言就可以开发出可视化的 Windows 应用程序，如图 1-8 所示。

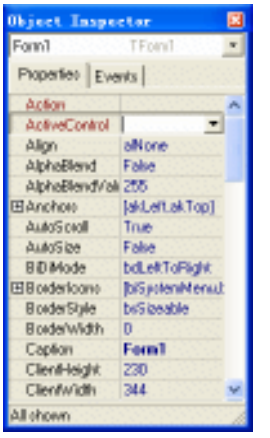


图 1-8 对象浏览器

1.3 面向对象的 Pascal 语言

Delphi 7.0 采用面向对象的 Pascal 语言作为开发应用程序的语言。Pascal 语言是 20 世纪 70 年代由 Nicholas Wirth 开发出的一种结构化程序设计语言。它具有思路清晰、结构紧凑等特点，因此被广泛使用。Pascal 是一种编译性语言，它可以高效地执行代码，其编译速度是所有编程语言中最快的。Object Pascal 是在 Pascal 语言基础上通过引入面向对象的特征发展而来的。所以，Object Pascal 不但保留了 Pascal 的所有优点，还具有便于应用程序开发的面向对象的特征，本节将简要讲述 Pascal 语言的应用：

- Object Pascal 面向对象编程
- Object Pascal 数据类型
- Object Pascal 操作符及表达式
- Object Pascal 基本语句
- Object Pascal 过程与函数



这里只是对 Pascal 语言做一个简单的介绍，需要对 Pascal 语言有更多了解的读者请参考相关的 Pascal 书籍。

1.3.1 Object Pascal 面向对象编程

面向对象编程虽然不能使得代码容易编写，但是它使代码更容易维护，同时将代码与数据结合在一起，使的代码的调试和定位变得简单，从而提高了程序的性能。一般的面向对象要实现以下三个概念：

1．封装：把相关的数据和代码结合在一起，并隐藏了实现的细节，它的好处就是有利于程序的模块化。

2．继承：是指一个新对象从父对象中获取属性和方法，同时能在父对象上建立独立于父对象的属性

与方法，它的好处是能共享代码。

3．多态性：指调用一个对象变量的方法时，实际被调用的代码与实际在变量中的对象的实例有关。

1.3.2 Object Pascal 的数据类型

1.3.2.1 基本的数据类型

Object Pascal 的最大的特点是，它的数据特别严谨，在传递给过程或函数的实参与定义过程或函数的行参的类型必须一致，以下是在 Delphi 7.0 的开发过程中常用的一些数据类型，表 1-1 列出了 Delphi 7.0 中常用的一些数据类型：

表 1-1 Pascal 常用的数据类型

变 量	含 义
shortInt	8 位有符号整数
Byte	8 位无符号整数
SmallInt	16 位有符号整数
Word	16 位无符号整数
integer	32 位有符号整数
single	4 字节浮点数
double	8 字节浮点数
bool	4 字节布尔值
Pchar	以 NULL 结束的字符串
Pwidechar	动态 2 字节字符串
char	1 字节字符
boolean	1 字节的布尔值
extended	10 字节浮点数

1.3.2.2 子界类型

子界类型是 Pascal 允许用户定义的一个结构数据类型。如果我们预先知道一个变量的范围，我们可以通过定义子界类型和子界类型变量来实现，如：

```
type
 TDay=1..7 ; //自定义的一个数据类型
var
Day:TDay;
```

1.3.2.3 数组类型

Object Pascal 允许你建立各种类型变量的数组（除文件类型外），如：

```
var
 A:Array[1..5] of integer; //静态数组
 B:Array of string; //动态数组
```

1.3.2.4 记录类型

在 Object Pascal 中用户自定义的结构称为记录，它跟 C 语言中的 Struct 的功能是一样的，如：

```
type
Tstudent=Record;
Name: String;
 Age: Integer;
end;
var
Student1:TStudent;
begin
Student1.Name :='Jeronge';
Student1.Age := 20;
end;
```

1.3.2.5 集合类型

集合是 Pascal 特有的数据类型，它采用一种有效的手段来表示一组有序的数、字符或是枚举值。声明一个集合用关键字 Set of，如：

```
TCharSet=Set of char;
```

1.3.2.6 对象类型

可以把对象类型看成是记录类型，不过它包括的函数与过程，如：

```
type
 TSubClass:Class;
  i:integer;
  procedure smpProc;
en;
```

1.3.2.7 指针类型

指针变量指示了内存的位置，在 Pascal 中用 Pointer 表示指针类型。有时 Pointer 又被称为无类型指针，因为它只指向内存地址，而不判断数据类型，如：

```
var
 P:Pointer;//无类型指针
 Pint:^integer;//指向有类型的指针
```



指针是一种高级技术，在编程过程中不是必需的。但在你熟练使用它以后，它是你程序编写的一个好工具。

1.3.3 Object Pascal 操作符及表达式

1.3.3.1 操作符

操作符包括普通字符、Pascal 保留字和标识符，其中 Pascal 保留字包括下面一些：

and，array，as，asm，begin，case，class，const，constructor，destructer，div，do，downto， else，

end，excep，export，file，finally，for，function，goto,，if，inherited，initialization，inline，interface，is，label，library，mod，nil，not，object，of，on，or，packed，procedure，program，property，raise，record，repeat，set，shl，shr，string，then，to，try，type，unit，until，uses，var，while，with，xor。

标识符必须由一个字母或下划线开始，后面是具有任意长度的字符串（不能包含空格字符）。例如以下几个标准的标识符：

1．**变量类型：**integer，real，single，char。

2．**过程：**read，write，reset，eof。

3．**函数：**random，upper，date，succ。

1.3.3.2 表达式

一个表达式由运算对象和运算符两部分组成。运算符可以分为算术运算符、逻辑运算符、串运算符、字符指针运算符、集合运算符、关系运算符以及@运算符。其中赋值、比较和逻辑运算符如表 1-2 **所示。**

表 1-2 **赋值、比较和逻辑运算符**

运 算 符	Pascal
赋值	:=
比较	=
不等于	<>
小于	<
大于	>
小等于	<=
大等于	>=
逻辑与	and
逻辑或	or
逻辑非	not

1.3.4 Object Pascal 语句

1.3.4.1 赋值语句

给变量赋值，如：

```
var
 i:integer;
begin
 i:=10;
end;
```

1.3.4.2 条件语句

（1**）**if **语句：**

主要功能：作条件判断控制流程。

语法形式：

```
if<布尔型表达式>then
<语句>；
```

需要全本请访问 www.ertongbook.com