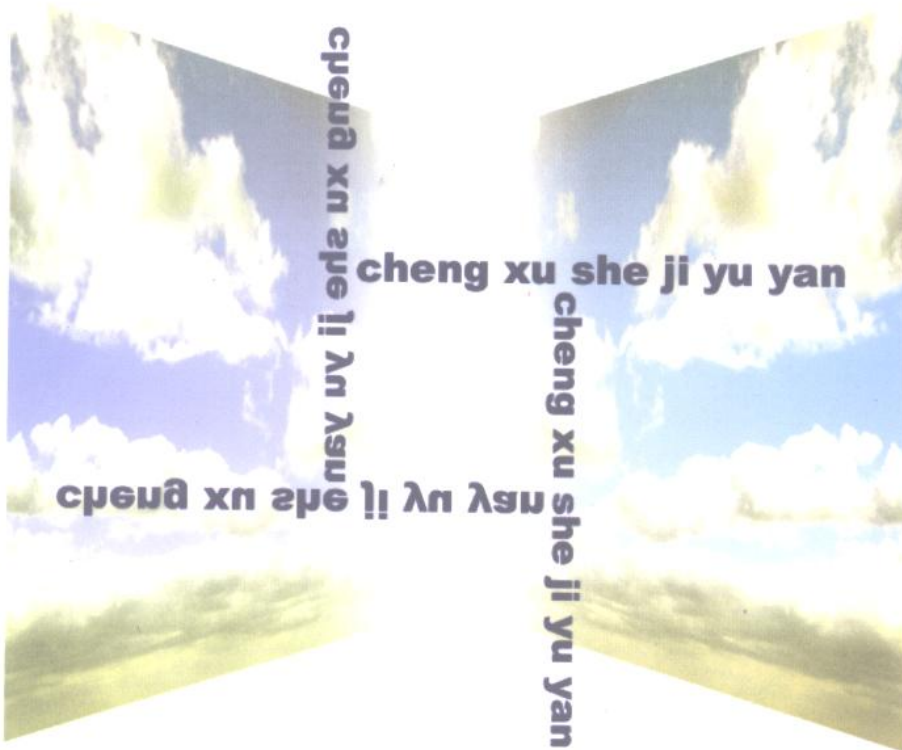


刘 王 宋
保 保 征
东 东 征
编 编 著

Delphi 5.0

自 学 捷 径



T0212
LT/1

电脑自学捷径系列

Easy way

Delphi 5.0 自学捷径

刘韬 王保东 宋征 编著

北京大学出版社
1053477

内 容 简 介

本书是“自学捷径”系列丛书的一本。鉴于 Delphi 的优良特性和顺应广大用户的要求，我们在积累大量开发经验的基础上编写了本书。本书主要讲述了 Delphi 的开发环境以及面向对象的开发方法；Delphi 一般应用程序开发的基本方法；数据库开发的基本方法、使用的基本控件等。附录讲述了 Delphi 辅助工具的应用方法和 SQL 的语法。这本书重点讲述 Delphi 编程中的一般概念和开发方法，突出讲解一般应用程序和数据库应用程序的开发步骤和方法，适合于初学者学习。

图书在版编目 (CIP) 数据

Delphi 5.0 自学捷径/刘韬, 王保东, 宋征编著, 北京: 北京大学出版社, 2000.3

(电脑自学捷径丛书)

ISBN 7-301-00912-7

I.D… II.①刘…②王…③宋… III.Delphi 语言-设计程序 IV.TP312

中国版本图书馆 CIP 数据核字(2000)第 03871 号

书 名: Delphi 5.0 自学捷径

著作责任者: 刘韬 王保东 宋征

责任编辑: 文幼章

标准书号: ISBN 7-301-00912-7/TP·13

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电 话: 出版部 62752015 发行部 62754140 编辑部 62765127

电子信箱: zwxu@mail.263.net.cn

排 版 者: 北京东方人华科技有限公司

印 刷 者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 22 印张 525 千字

2000 年 3 月第一版 2000 年 3 月第一次印刷

定 价: 34.00 元

前 言

Delphi 5.0 简介

Delphi 被称为是第四代编程语言，以它基于窗口和面向对象的编程方法，与 Windows 操作系统紧密的结合、强大的数据库技术支持、快捷的编译速度，同时兼备 Visual C++ 功能强大和 Visual Basic 易学易用的特点，成为程序员们喜爱的编程工具。Delphi 4.0 在 1998 年一经推出就获得巨大成功。在原有 Delphi 版本强大的 Win32 应用程序编程能力的基础上，Inprise 公司着重加强了 Delphi 对网络应用程序的支持，很快获得程序员的青睐。

随着网络日益渗透到人们的日常生活，日常商业活动的网络化、电子化也将成为必然。因此说 21 世纪将是电子商务的世纪并不夸张。就在世纪之交的 1999 年，Inprise 公司隆重推出了面向新世纪的强大应用程序开发平台——Delphi 5.0。Delphi 5.0 正是提供电子商务软件技术支持的利器，这可以从 Delphi 5.0 的新特性中看到：

- 增加了 ADO 数据集，这样使得 BDE（Borland 的数据库引擎）获得了对 Microsoft 的 ADO（活动数据对象）的全面支持，而 ADO 恰恰是网络开发 ASP（Active Server Pages）的核心。
- 增加了数据模块设计器，为快速、简捷地创建数据模块提供了可视化的工具——Data Module Designer。程序员可以更快的创建出数据模块，从而缩短了项目的开发周期。
- 增加了 InterBase Express 控件。这是一种集成了 InterBase 的数据访问控件，可以开发出不需要 BDE 的数据库应用程序。
- 加强了 MIDAS 的功能。

上面只是介绍了 Delphi 5.0 中新增功能中的一部分，通过对 Delphi 更深入的学习就会了解到更多的新功能。当然这些都是建立在 Delphi 原有的强大功能之上，从 Delphi 4.0 可以很快过渡到 Delphi 5.0。

本书阅读指南

本书针对初学者认真设计了由浅入深、从抽象到具体的整体结构。全书分为四篇：第一篇主要介绍利用 Delphi 5.0 进行应用程序开发的集成开发环境，具体包括集成开发环境的组成部分、项目开发和程序调试的基本方法；第二篇主要介绍了 Object Pascal 语法，包括 Object Pascal 的基本概念、类与对象以及对于应用程序中异常的处理；第三篇通过对各种 Win 32 应用程序开发方法的介绍，详细讲解了各种控件的特性及使用方法；第四篇重点介绍了运用 Delphi 5.0 创建数据库应用程序的基本理论、常用控件以及对数据记录各项操作。

在各个章节中注重实例的应用：在对一个基本理论进行具体讲述后，通过一个具有代

表性的例子加深读者的认识。

本书文本约定

本书中，AA|BB 表示 AA 菜单中的 BB 命令；AA|BB|CC 则表示 AA 菜单中 BB 子菜单中的 CC 命令。

本书中如果未特别指出，则“单击”表示用鼠标左键单击，“右击”表示用鼠标右键单击，“双击”表示用鼠标左键双击，“指向”表示移动鼠标指针并停留在屏幕对象上。

作 者

目 录

第 1 章 Delphi 5.0 概述.....	1
1.1 Delphi 产生的背景.....	1
1.2 Delphi 5.0 的主要特点.....	2
1.2.1 可定制的开发环境.....	2
1.2.2 真正的面向对象.....	2
1.2.3 控件库.....	3
1.2.4 完全编译.....	3
1.3 Delphi 5.0 的新特点.....	3
1.3.1 Web 上的 Client/Server 技术.....	3
1.3.2 数据分析.....	4
1.3.3 组件制造中心.....	5
1.3.4 提高程序设计人员的工作效率.....	5
第 2 章 Delphi 5.0 的集成开发环境.....	6
2.1 Delphi 5.0 集成开发环境的运行.....	6
2.2 主窗口介绍.....	7
2.2.1 主窗口中的菜单栏.....	8
2.2.2 主窗口中的工具栏.....	11
2.2.3 主窗口中的控件栏.....	12
2.3 窗体编辑器.....	19
2.3.1 Delphi 5.0 中的窗体.....	19
2.3.2 创建窗体.....	20
2.3.3 窗体中的控件.....	20
2.4 代码编辑器.....	24
2.4.1 代码编辑窗口.....	25
2.4.2 代码浏览窗口.....	27
2.5 对象编辑器.....	28
2.5.1 设置控件对象的属性.....	29
2.5.2 设置控件需要响应的事件.....	30
2.6 其他常用工具.....	31
2.6.1 对齐工具栏.....	31
2.6.2 图像编辑器.....	31
2.7 环境参数设置.....	33

第 3 章 项目管理及程序编译	35
3.1 工程项目的管理	35
3.1.1 工程项目中的各种文件	35
3.1.2 工程管理器	38
3.2 应用程序的编译	39
3.2.1 设置编译环境	39
3.2.2 常用的编译方法	39
3.3 应用程序调试的基本方法	40
3.3.1 设置调试环境参数	41
3.3.2 控制应用程序运行	42
3.3.3 设置断点	43
3.3.4 检查变量和表达式的值	45
3.3.5 检查函数和过程调用	47
3.3.6 监视线程状态	47
3.3.7 监视 CPU 状态	48
第 4 章 Object Pascal 基本概念	50
4.1 数据类型	50
4.1.1 整数类型(Integer)	50
4.1.2 实数类型(Real)	50
4.1.3 字符类型(Character)	51
4.1.4 布尔类型(Boolean)	51
4.2 常量、变量和表达式	52
4.2.1 常量	52
4.2.2 变量	52
4.2.3 表达式	52
4.3 Object Pascal 的语句	53
4.3.1 赋值语句	53
4.3.2 条件语句	54
4.3.3 case 语句	55
4.3.4 while 语句	56
4.3.5 repeat 语句	57
4.3.6 for 语句	57
4.4 自定义数据类型	58
4.4.1 枚举类型	58
4.4.2 子界类型	59
4.4.3 集合类型	60
4.4.4 记录类型	62
4.5 函数和过程	63
4.5.1 函数的声明与调用	63

4.5.2 过程的声明和调用	65
4.6 数 组	66
4.6.1 一维数组	66
4.6.2 二维数组	66
4.7 指 针	67
4.7.1 指针类型和指针变量	67
4.7.2 标准过程 new 和 dispose	68
4.7.3 动态存储单元和动态变量	68
4.7.4 指针操作	69
第 5 章 Object Pascal 中的类和对象	70
5.1 Delphi 中单元文件的结构	70
5.2 类 和 对 象	71
5.2.1 类的基本概念	71
5.2.2 类的定义	72
5.2.3 对象的创建	74
5.3 类的访问控制	75
5.3.1 类成员的可见性	75
5.3.2 私有成员	76
5.3.3 保护成员	77
5.3.4 公有成员	79
5.3.5 发行成员	79
5.3.6 自动成员	79
5.4 类 的 成 员	80
5.4.1 字段成员	80
5.4.2 属性成员	81
5.4.3 方法成员	81
5.5 类 的 继 承	82
第 6 章 异 常 处 理	84
6.1 异常处理的基本概念	84
6.2 Object Pascal 的异常处理	84
6.2.1 Object Pascal 的异常处理机制	84
6.2.2 异常处理语句	85
6.2.3 各种常见的异常情况	87
6.3 使用自定义的异常类	88
6.3.1 定义异常对象类	88
6.3.2 使用自定义的异常类	89
第 7 章 Delphi 5.0 编程方法与控件	91
7.1 创建 Windows 应用程序的基本步骤	91

7.1.1	创建一个简单的 Windows 应用程序.....	91
7.1.2	为应用程序添加菜单.....	92
7.1.3	添加 About 对话框.....	93
7.2	Delphi 应用程序编程基本方法.....	97
7.2.1	应用程序总体设计.....	97
7.2.2	应用程序结构设计.....	97
7.2.3	功能模块的具体实现.....	98
7.3	VCL 基础.....	98
7.3.1	VCL 的结构.....	99
7.3.2	控件的定义及构成.....	100
第 8 章	主窗口设计及其控件.....	101
8.1	窗体控件.....	101
8.1.1	窗体控件的属性.....	102
8.1.2	窗体控件的事件.....	105
8.1.3	窗体控件的方法.....	110
8.2	菜单控件.....	110
8.2.1	菜单概述.....	110
8.2.2	菜单设计器(Menu Designer).....	112
8.2.3	菜单的动态控制.....	116
8.3	工具栏控件.....	119
8.3.1	工具栏控件的重要属性.....	120
8.3.2	工具栏按钮控件.....	121
8.3.3	创建应用程序中的工具栏.....	121
8.4	CoolBar 控件.....	123
8.4.1	CoolBar 控件.....	123
8.4.2	创建应用程序中的 CoolBar 工具栏.....	124
8.5	状态栏控件.....	126
8.5.1	状态栏控件的属性和事件.....	127
8.5.2	显示面板控件(StatusPanel).....	128
8.5.3	在应用程序中创建状态栏.....	128
8.6	创建一个 Web 浏览器.....	134
8.6.1	主窗口设计.....	134
8.6.2	程序代码.....	136
8.6.3	程序运行结果.....	143
第 9 章	标准对话框控件和 RichEdit 控件.....	144
9.1	对话框简介.....	144
9.2	各种对话框控件.....	145
9.2.1	Open 对话框控件(OpenDialog).....	145

9.2.2	Font 对话框控件(FontDialog)	148
9.2.3	Color 对话框控件(ColorDialog)	150
9.2.4	Print 对话框控件(PrintDialog)	151
9.2.5	Find 对话框控件(FindDialog)	152
9.2.6	Replace 对话框控件(ReplaceDialog)	153
9.3	RichEdit 控件	153
9.3.1	RichEdit 控件的属性	154
9.3.2	RichEdit 控件的重要事件	155
9.3.3	RichEdit 控件的常用方法	155
9.4	一个文本编辑器	156
9.4.1	文本编辑器	156
9.4.2	基本步骤	157
9.4.3	应用程序的主窗口	157
9.4.4	应用程序的各个功能模块	158
9.4.5	应用程序源代码	164
第 10 章	Win 32 应用程序常用控件	185
10.1	PageControl 控件	185
10.1.1	PageControl 控件常用属性和方法	185
10.1.2	TabSheet 控件	186
10.1.3	创建一个属性对话框	187
10.2	ImageList 控件	188
10.2.1	ImageList 控件的属性	189
10.2.2	ImageList 控件的方法	189
10.2.3	ImageList 控件的使用方法	190
10.3	TrackBar 控件	192
10.3.1	TrackBar 控件的属性	192
10.3.2	TrackBar 控件的方法	194
10.3.3	TrackBar 控件的使用方法	194
10.4	ProgressBar 控件	196
10.4.1	ProgressBar 控件的属性	197
10.4.2	ProgressBar 控件的方法	197
10.4.3	ProgressBar 控件的使用方法	198
10.5	TreeView 控件	200
10.5.1	TreeView 控件介绍	201
10.5.2	TreeNode 控件介绍	202
10.5.3	TreeNode 控件介绍	203
10.5.4	创建树状结构	204
10.5.5	一个使用 TreeView 控件的例子	205
10.6	ListView 控件	216

10.6.1	ListView 控件介绍	217
10.6.2	ListItems 控件介绍	219
10.6.3	ListItem 控件的属性	220
10.6.4	创建列表结构	220
第 11 章	图形图像应用编程	222
11.1	画布控件(Canvas)	222
11.1.1	画布控件(Canvas)的属性	222
11.1.2	画布控件(Canvas)的方法	223
11.2	图像控件(Image)	225
11.3	几何图形控件(Shape)	226
11.4	一个综合应用程序	228
第 12 章	多线程编程	239
12.1	线程的基础知识	239
12.2	Delphi 中多线程的工作模型	239
12.3	创建线程对象	240
12.4	Thread 对象的属性和方法	242
12.5	控制线程运行	243
12.5.1	线程对象自动退出	243
12.5.2	控制线程对象退出	243
12.6	多线程间的同步与通信	244
12.6.1	多线程同步	244
12.6.2	多线程间通信	246
12.7	一个多线程应用程序的实例	247
第 13 章	Delphi 开发数据库应用程序概述	257
13.1	使用数据库	257
13.1.1	数据访问控件(Data Access)	257
13.1.2	Delphi 5.0 支持的数据库种类	259
13.2	数据库结构	260
13.3	设计用户接口	260
13.4	Delphi 5.0 在数据库方面的新技术	261
13.4.1	Web 上的 Client/Server 技术	261
13.4.2	数据分析	262
13.4.3	组件制造中心	263
13.4.4	提高程序设计人员的工作效率	263
13.5	运用 Delphi 5.0 开发数据库应用程序的步骤	263
13.5.1	Database Form Wizard	263
13.5.2	开发数据库应用程序的基本步骤	264
13.5.3	形成完整的数据库系统	265

第 14 章 数据库应用程序常用控件	266
14.1 数据访问控件 (Data Access)	266
14.1.1 Table 控件	267
14.1.2 DataSource 控件	270
14.1.3 NestedTable 控件	272
14.1.4 Database 控件	272
14.1.5 Session 控件	274
14.1.6 BatchMove 控件	276
14.1.7 UpdateSQL 控件	276
14.2 数据控制控件	277
14.2.1 DBGrid 控件	277
14.2.2 DBNavigator 控件	278
14.2.3 DBMemo 控件	279
14.2.4 DBListBox 控件	280
14.3 Decision Cube 控件	280
14.3.1 Decision Cube 控件	281
14.3.2 Decision Query 控件	283
14.3.3 Decision Grid 控件	284
14.4 QReport 控件	287
14.4.1 QuickRep 控件	288
14.4.2 QRSubDetail 控件	290
14.4.3 QRBand 控件	291
14.4.4 QRChildBand 控件	293
14.4.5 QRGroup 控件	293
14.4.6 QRLabel 控件	294
14.4.7 QRDBText 控件	294
14.4.8 QRExpr 控件	294
14.4.9 QRSysData 控件	294
第 15 章 数据记录的基本操作	296
15.1 一个简单的例子	296
15.1.1 选择合适的控件	296
15.1.2 设置控件的属性	296
15.1.3 运行程序	298
15.1.4 创建数据库应用程序的基本步骤	298
15.1.5 修改成具有更复杂功能的程序	298
15.2 数据库中有关记录的操作	300
15.2.1 查询数据库中的记录	300
15.2.2 修改数据库中的记录	304
15.2.3 插入和删除记录	310

15.2.4	输入数据的合法性验证	314
15.3	字段对象的使用	316
15.3.1	字段对象的类型	317
15.3.2	创建永久的字段对象	317
15.3.3	字段对象的属性及其设置方法	318
15.3.4	字段对象的事件	320
15.3.5	字段对象的类型转换	321
15.3.6	对字段对象的访问	321
15.3.7	自定义字段对象	325
15.3.8	字段编辑器的使用	327
15.4	一个综合数据库应用程序的例子	330
15.4.1	应用程序主窗体	330
15.4.2	应用程序代码	331
15.4.3	运行应用程序	339

第 1 章 Delphi 5.0 概述

Delphi 5.0 是 Inprise 公司在 1998 年推出 4.0 版本以后新开发的面向对象、可视化的快速应用程序开发环境(RAD—Rapid Application Development)。本章首先介绍 Delphi 5.0 产生的背景和主要特点, 然后详细讲解 Delphi 5.0 集成开发环境的组成部分及其用法, 主要包括:

- Delphi 5.0 产生的背景;
- Delphi 5.0 的主要特点;
- Delphi 5.0 的新特性。

1.1 Delphi 产生的背景

信息技术在 20 世纪末的迅猛发展, 人们有理由相信 21 世纪将是“信息的世纪”。作为信息产业重要组成部分的计算机技术已经由初期简单的计算工具渐渐进入人们生活的各个角落, 占据着越来越重要的地位。从制造业的计算机集成制造系统(CIMS, Computer Integrate Manufacture System)到流通领域内电子商务的电子数据交换技术(EDI——Electronic Data Interchange), 无不闪耀着计算机璀璨的身影。

在本世纪末计算机信息技术的广阔的发展前景可以归结为以下几点:

- 随着全球经济一体化进程的加快, 商业竞争日趋激烈。为了在日益激烈的商业竞争中迅速取得优势, 人们需要不断调整自己的商务活动。计算机信息产业要适应这种快速变化的用户需求。
- 计算机技术(如计算机操作系统、软件开发平台和数据库管理系统等)的进步, 给计算机软件开发人员提出了更高的要求。在新的应用程序的开发过程中, 软件开发人员需要迅速掌握和使用相应的新技术。
- 新的软件系统模型[如计算机客户/服务器(Client/Server)、分布式应用程序(Distributed Applications)、Internet 和多层结构的数据库应用程序(Multi-tiered Applications)]的提出, 也要求计算机软件开发人员能够快速把这些模型集成到应用程序中。

正是以上几方面的变化, 软件开发人员在开发过程中需要面对以下几方面的挑战:

- 应用程序的快速开发

为了适应商务活动的迅速变化和日益激烈的竞争, 要求在最短的时间里开发出能够满足各种功能要求的应用程序。很明显, 采用传统的开发工具和开发方法, 应用程序的开发周期远远不能满足市场需求。

- 数据库的集成开发

在当前的应用程序工程项目中, 经常需要管理和处理各种类型的数据, 因此要求应用

程序开发工具能够提供数据库开发环境的支持,以致于更进一步发展为能够同时支持多种格式数据库系统的数据操作。

- 面向 Internet 的应用

Internet 已经逐步渗透到人们生活的各个领域,这要求应用程序的开发也应满足远程数据交换的需求。使用传统工具开发面向 Internet 的应用程序要求开发人员对计算机核心技术有更深入的了解,这就极大的阻碍了 Internet 应用程序的开发。

- 提供良好的用户界面

应用程序工程项目一般是可视化的、面向用户界面的。为了便于用户使用,应用程序必须提供方便、友好的用户界面。

- 复杂应用程序代码的有序管理

计算机硬件技术的迅猛发展使得计算机应用软件的功能越来越强大,软件的规模也越来越庞大。开发应用程序过程需要的源代码也日益增多。因此,在应用程序的开发过程中,需要对各种类型的资源进行有效的管理。

正是在这种背景下,Inprise 公司将可视化界面和面向对象的 Pascal 语言完美地结合起来,推出了最强大和最灵活的应用程序开发工具——Delphi。可视化控件库(VCL——Visual Component Library)提供了大量可以重用的控件。这样可以利用开发环境提供的控件快速地设计和实现 Windows 应用程序,彻底改变了传统 Windows 应用程序的开发方法。从 1993 年 Inprise 公司推出的 Delphi 1.0 到 1999 年的 Delphi 5.0 一共开发了 5 个版本。每一次版本的更新都伴随着功能的加强和新性能的提供。

1.2 Delphi 5.0 的主要特点

Delphi 工作在 Windows 95, Windows NT, 以及即将推出的 Windows 2000 操作系统上,可以开发一般的、基于客户/服务器模式以及多层结构模式的 32 位 Windows 应用程序。Inprise 公司于 1999 年推出的 Delphi 5.0 代表着当前应用程序开发工具的最高水平。这里将介绍 Delphi 5.0 的主要特点。

Delphi 不仅仅是编辑器和编译器,同时也包含各项简明特性的开发环境,令开发工作更加简单。这些特性数量众多足可以写出一本完整的书,这里仅介绍最常用和最有用的几个。

1.2.1 可定制的开发环境

Delphi 的开发环境提供了各种各样的功能,包含从简单的代码编辑、程序编译到复杂的 API 开发工具。然而不同的软件开发人员对开发环境有不同的需要,这就需要对 Delphi 的开发环境进行定制,如软件开发人员可以根据自己的喜好,指定在 Delphi 的开发环境中显示哪些工具栏等。

1.2.2 真正的面向对象

像 Visual Basic 这样的语言是伪面向对象的,它并不支持封装(encapsulation)、继承性

和多态性等基本面向对象的功能。Delphi 是真正的面向对象的语言，它允许将数据和代码合并为一个类(封装)、建立新的继承类(继承性)等。

1.2.3 控件库

各种控件被存放在控件库中。Delphi 的控件库包含了所有用来建立 Win 32 应用程序的对象。用户还可以根据需要扩充功能或者改变某一控件的特性。这只需要从存储的控件库中派生出一个新的控件就可重新编码。

1.2.4 完全编译

许多 Windows 的开发环境采用的是不完全编译或者是生成伪代码。伪代码是机器不可执行的代码，它必须在运行时被翻译成可执行代码，这就大大降低了系统的性能。Delphi 使用的是完全编译器和链接器，产生 100%的本地可执行代码。

1.3 Delphi 5.0 的新特点

Delphi 5.0 的新技术主要表现在数据库方面：Web 上的 Client/Server 技术、数据分析、企业组件和提高程序设计人员的生产力。

有关数据库的处理技术已经发展了很长时间，先后出现过 3 种数据库系统：

第一代数据库系统采用单层结构(1-Tier)。这种结构很浪费计算机资源。第二代数据库采用 Client/Server 结构，也就是所谓的两层结构(2-Tier)。这种结构得到了广泛的应用并获得了极大的成功。但在其结构上也存在一些问题，主要表现在应用程序的伸缩和维护方面。例如，在网络上如何维护数据的统一性和完整性——一旦应用程序有任何改动，维护人员就必须对每个客户端进行修改。

最新一代的数据库系统是在传统的 Client/Server 结构中增加一个应用服务器，这种新结构叫做 n-Tier 或 Multi-Tier。

Delphi 5.0 针对这种新一代的数据库管理系统观念，提出了 3 种 Broker 和新一代的数据库引擎，来适应 n-Tier 的需求。

第一种叫做 Remote Data Broker。其结构的精髓是让每一个客户端不再需要 BDE，取而代之的是一个中央化的 BDE，以集中管理的方式降低每一个客户在 BDE 上调整的开销和复杂度。

第二种叫做 Constraint Broker。顾名思义，它所扮演的角色就是保证所有客户数据的一致性及数据的完整性。

第三种是 Business Object Broker。它的目的是给一些关键性的商业应用程序提供一个快速并且可以信赖的使用环境。为了满足这种高层次的要求，Business Object Broker 自动地将应用程序做适当的划分，并复制重要的业务规则到每一个区间，以达到要求的速度。

1.3.1 Web 上的 Client/Server 技术

Delphi 5.0 的两项新技术——Web Broker 和 ActiveForms，可以使开发人员结合原有的

Client/Server 技术开发出 Web-Enabled 的 Client/Server 应用程序。

先从实际的层面看, 当一个 Client/Server 结构的数据库程序改成 Web-Enabled 的时候, 首先要考虑的是如何把结果显示在不同的机器上, 其次是如何将已经开发出来的程序方便地分发给不同地方的用户。

Delphi 5.0 中 Web Broker 的目的就是让服务器执行的结果以 HTML 格式快速地显示在 Web 浏览器上面。Web Broker 中包含了几个重要的核心成分:

- **WebServer Application:** 建立 ISAPI/NSAPI 或 CGI 的 DLL, 或者建立可执行程序。
- **WebBridge:** 让开发人员开发一些可在 ISAPI/NSAPI 中共用的 API, 使得程序的开发不会因 Netscape 或 Microsoft 的标准改变而受重大的影响。
- **WebModules:** WebModule 所扮演的角色就好像是 WebServer 的信息中心, 它分派每个使用者的要求, 定义 URI 和新建 HTML 网页。
- **Sessions:** Session 可同时处理许多数据库处理的要求, 针对每一个用户请求, 在中央的 BDE 中新建不同的 BDE 区段来应答用户。
- **WebDispatcher:** 一个处理数据库需求的中心, 允许 Delphi 的开发者把传统的 Client/Server 数据库所产生的内容传送到 Web 上去。
- **HTML Producers:** 主要的目的是将数据库信息或其它程序的结果转换为 HTML 的格式。
- **ActiveForm:** 开发 Web-Enabled 的 Client/Server 应用程序的另外一个重要部分, 它针对以 Windows 为基础的 Client/Server 结构的数据库系统。

Delphi 5.0 提供了一种最简易的方式, 将 Delphi 的 Form 转变成 ActiveForm。ActiveForm 是一个 ActiveX 控件, 它用 Delphi 的 Form 为载体来装载其他 Delphi 组件。为了适应网络频宽不足的限制, ActiveForm 配合 Remote Data Broker 可以产生小巧且无需编程的 Multi-Tier 数据库应用程序。

Delphi 5.0 另一个有力的工具叫 Web Deployment。它可以自动地产生 HTML 文件, 传送 ActiveForm 和其他所必需的文件到客户端。Web Deployment 可以支持 CAB 的文件格式, 以增加文件下载的速度。除此之外, 它还提供 Code Signing 机制去提高网络安全性。

Delphi 5.0 还提供了新的编译技术——VCL Package, 它可以把 Delphi 的应用程序缩小到 15KB 左右, 使之适合网络传输。

1.3.2 数据分析

Delphi 5.0 另外值得一提的部分是在客户端的开发应用上, 主要表现在 3 个方面: 数据分析、组件制造中心和减小程序开发的难度。

正确的数据分析, 在业务决策上是相当重要的。当大量的数据放在用户面前时, 用户要搞清这些复杂数据背后的真实意义可不是件容易事。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抉择呢? 一般来说, 运用图形及报表形式是最好不过的了。Delphi 5.0 提供了 3 个重要的组件: 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, 产生不同性质的有用分析结果, 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, 帮助他作出最正确的决定。TeeCharts 是一个三维的可视化图表, 使数据更容易