



国家示范性软件职业技术学院系列教材

# Delphi 数据库编程

陈秋劲 编著

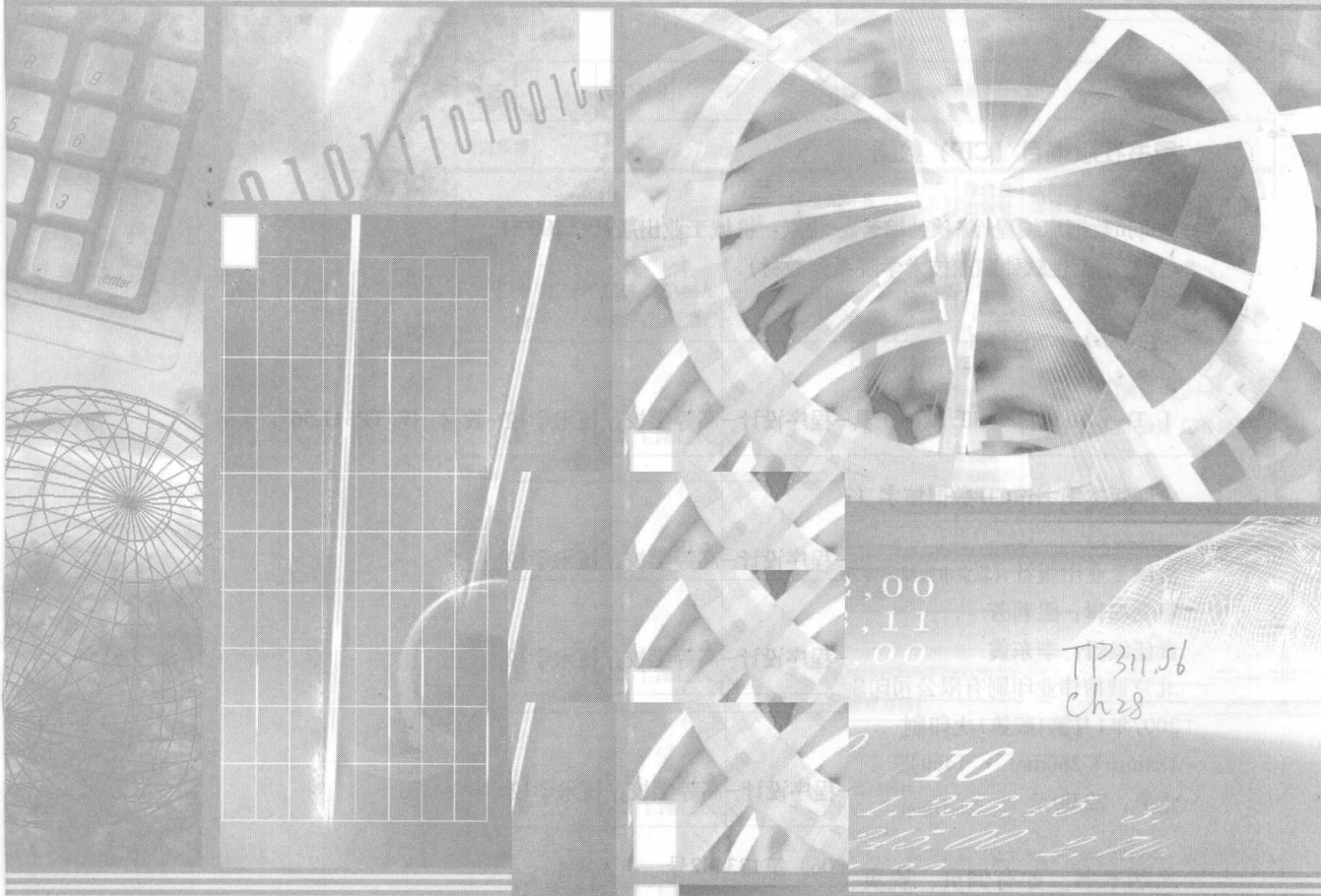


机械工业出版社  
China Machine Press

国家示范性软件职业技术学院系列教材

# Delphi 数据库编程

陈秋劲 编著



机械工业出版社  
China Machine Press

本书系统地介绍Delphi数据库编程的基础知识，主要包括：Object Pascal基础、Delphi数据库工具、数据库连接访问组件、ADO组件、数据控件、报表设计等。本书的内容经过精心安排，以通俗实用为主，并安排了实训及其详细的实现步骤和课后强化训练，使初学者能够快速入门。另外，第9章介绍了一个完整的项目开发实例，以指导课程设计。

本书可以作为大专、高职高专院校学生的教材和教学参考书，也可以供各类从事数据库系统开发的人员参考。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

### 图书在版编目 (CIP) 数据

Delphi数据库编程/陈秋劲编著. —北京：机械工业出版社，2007.1  
(国家示范性软件职业技术学院系列教材)

ISBN 7-111-20327-5

I. D… II. 陈… III. 软件工具—程序设计—高等学校：技术学校—教材 IV. TP311.56

中国版本图书馆CIP数据核字 (2006) 第133482号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

策划编辑：温莉芳

责任编辑：李东震

北京诚信伟业印刷有限公司印刷 · 新华书店北京发行所发行

2007年1月第1版第1次印刷

184mm × 260mm · 14印张

定价：22.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换  
本社购书热线：(010) 68326294



# 前 言

随着信息化社会的发展、计算机网络的延伸,数据库应用已遍及社会的各个领域。根据国家教育部、信息产业部启动的“国家技能型紧缺人才培养项目”的要求,数据库编程作为学习和掌握数据库专业知识和应用能力的重要课程也越来越受到重视。

Delphi用于开发数据库应用程序有着得天独厚的优势,加之高等职业教育以实践应用能力为主。所以如何规划数据库编程课程对于我们来说是个新的课题,急需进行改革和探讨。编者长期从事Delphi数据库编程的一线教学,在教学中摸索并积累了一些经验。现通过这本教材和大家广泛交流,可以根据教材内容选择灵活的教学方法,充分利用多个案例、课件和师生协同的方式,努力提高数据库编程课程的教学水平。

本书主要内容包括:Delphi基础、Object Pascal基础、数据库编程基础、Delphi数据库工具、数据库连接访问组件、ADO组件、数据控制控件、报表设计等内容。本书的内容经过精心挑选,都是最实用的技术。第9章介绍了一个完整的项目开发实例,以指导课程设计。

本书遵照国家技能型紧缺人才培养培训指导方案,根据教育部高等职业教育的培养目标和规格要求,并紧密结合当前计算机的最新应用来编写,内容新颖,语言精炼、准确,运用了丰富的图片,配有大量的实训和练习,力求淡化理论性阐述,突出实践性指导和实践,循序渐进、深入浅出。

本书不仅有助于数据库编程入门,而且有助于学会实用技能、解决实际问题,对企业信息化进程有一定帮助。既可作为高等职业技术学院计算机及相关专业的数据库编程核心课程教材,也适合作为社会各界自学数据库编程的教材和参考书。

本书由武汉软件职业学院陈秋劲编著。在此,编者对在本书的编写、整理、排版、出书等方面给予很大帮助和支持的全体人员表示衷心的感谢。由于编写时间仓促且作者水平有限,书中难免有疏漏和不足之处,恳请广大读者提出宝贵意见。

编者的E-mail: czy57@163.com。

编 者

# 目 录

前 言	第4章 Delphi数据库工具.....52
第1章 Delphi基础知识 .....1	4.1 Delphi数据库编程简介 .....52
1.1 概述 .....1	4.2 Database Desktop的使用 .....52
1.1.1 集成开发环境 .....2	4.2.1 简介 .....52
1.1.2 IDE界面操作 .....5	4.2.2 使用方法 .....54
1.2 可视化编程基础 .....7	4.3 SQL Explorer的使用 .....59
1.2.1 第一个Delphi应用程序 .....7	4.3.1 SQL Explorer 主界面 .....60
1.2.2 Delphi应用程序结构 .....11	4.3.2 利用SQL Explorer注册别名 .....61
课后强化训练 .....13	4.3.3 数据字典 .....62
第2章 Object Pascal基础知识 .....14	4.4 Datapump的使用.....63
2.1 Object Pascal的语言基础 .....14	4.5 SQL监视器的使用 .....65
2.1.1 数据类型 .....14	实训1 熟悉数据库工具 .....65
2.1.2 变量和常量 .....16	课后强化训练 .....66
2.1.3 操作符、运算符和表达式 .....18	第5章 数据库连接访问组件.....67
2.1.4 过程与函数 .....19	5.1 基础知识 .....67
2.1.5 常用函数 .....22	5.1.1 组件的基本常识 .....67
2.2 Object Pascal的程序设计 .....22	5.1.2 数据模块 .....68
2.2.1 程序结构 .....22	5.1.3 字段和字段编辑器的使用 .....69
2.2.2 面向对象程序设计 .....26	5.2 BDE数据库连接访问组件基础 .....71
2.3 Delphi的编程规范 .....30	5.2.1 简介 .....71
2.3.1 源程序书写规范 .....30	5.2.2 BDE数据集组件通用的事件 .....72
2.3.2 命名规范 .....34	5.3 BDE数据库连接访问组件 .....72
2.3.3 文件 .....37	5.3.1 TDataSource组件 .....72
课后强化训练 .....38	5.3.2 TTable组件 .....74
第3章 数据库编程基础.....42	实训2 过滤和索引 .....76
3.1 数据库基础知识 .....42	实训3 TTable组件查询方法的应用 .....81
3.2 Paradox、Access、SQL Server 2000	实训4 TTable组件数据编辑的重要
关系型数据库 .....45	方法 .....87
3.3 数据库访问 .....45	5.3.3 TQuery组件 .....90
3.3.1 数据库引擎 .....46	实训5 TQuery组件过滤和索引 .....93
3.3.2 数据库应用程序 .....49	5.3.4 TStoredProc组件 .....97
课后强化训练 .....51	5.3.5 TDatabase组件 .....99
	5.3.6 TSession组件 .....99

5.3.7 其他BDE组件介绍 .....	100	7.2.8 TDBLookupListBox和	129
课后强化训练 .....	101	TDBLookupComboBox组件 .....	129
第6章 ADO 组件 .....	102	7.2.9 TDBRichEdit组件 .....	130
6.1 ADO组件面板 .....	102	7.2.10 TDBCtrlGrid组件 .....	130
6.1.1 概述 .....	102	7.2.11 TDBChart组件 .....	131
6.1.2 TADOConnection组件 .....	103	7.3 使用表单向导 .....	132
6.1.3 TADOCommand组件 .....	103	7.3.1 单窗体应用程序 .....	133
6.1.4 TADODataSet组件 .....	104	7.3.2 主/明细窗体应用程序 .....	135
6.1.5 TADOTable组件 .....	105	实训9 简单人事管理系统的实现 .....	140
6.2 利用ADO组件访问数据库 .....	105	实训10 利用非数据库相关的组件来	
6.2.1 数据集为ADOTable .....	105	实现数据库的操作 .....	148
实训6 登录窗口的实现 .....	108	7.4 Decision Cube组件面板 .....	151
6.2.2 数据集为ADOQuery .....	112	7.4.1 TDecisionCube组件 .....	152
实训7 BDE与ADO的变换 .....	113	7.4.2 TDecisionQuery组件 .....	153
实训8 ADO的综合运用: 查询程序 .....	113	7.4.3 TDecisionSource组件 .....	154
6.2.3 用ADO连接带密码的Access		7.4.4 TDecisionPivot组件 .....	154
数据库 .....	118	7.4.5 TDecisionGrid组件 .....	154
6.2.4 实现动态连接数据库 .....	119	7.4.6 TDecisionGraph组件 .....	155
6.3 利用dbExpress组件访问数据库 .....	120	实训11 利用Decision Cube组件	
6.3.1 TSQLConnection组件 .....	120	面板实现数据分析 .....	155
6.3.2 TSQLDataSet、TSQLTable、		课后强化训练 .....	158
TSQLQuery和TSQLStoredProc		第8章 数据报表的设计 .....	159
组件 .....	121	8.1 Rave Reports报表设计 .....	159
6.3.3 TSQLMonitor和TSimpleDataSet		8.1.1 Rave Reports概述 .....	159
组件 .....	121	8.1.2 Rave Reports界面 .....	160
课后强化训练 .....	121	8.1.3 Rave Reports组件页 .....	161
第7章 数据控制组件 .....	123	8.1.4 Rave组件页 .....	164
7.1 概述 .....	123	8.1.5 制作Rave报表的基本步骤 .....	166
7.2 数据控制组件介绍 .....	124	实训12 应用Rave报表 .....	167
7.2.1 TDBGrid组件 .....	124	8.2 Quick Report报表设计 .....	171
7.2.2 TDBNavigator组件 .....	125	8.2.1 Quick Report概述 .....	171
7.2.3 TDBText、TDBEdit和TDBMemo		8.2.2 QReport组件页 .....	173
组件 .....	127	8.2.3 制作Quick Report报表的	
7.2.4 TDBImage组件 .....	127	基本步骤 .....	175
7.2.5 TDBListBox和TDBComboBox		课后强化训练 .....	178
组件 .....	128	第9章 综合课程设计 .....	179
7.2.6 TDBCheckBox组件 .....	129	9.1 软件开发概述 .....	179
7.2.7 TDBRadioGroup组件 .....	129	9.1.1 软件生存周期 .....	179

9.1.2 软件开发文件 .....180

9.2 课程设计说明 .....181

9.2.1 实践目的 .....181

9.2.2 组织形式 .....181

9.2.3 基本流程及要点 .....181

9.3 参考课题 .....181

9.4 课程设计的评价 .....182

9.5 课程设计案例：教务管理信息系统 .....182

9.5.1 功能说明 .....182

9.5.2 功能模块 .....183

9.5.3 系统功能结构图 .....184

9.5.4 数据库设计 .....185

9.5.5 程序代码 .....188

附录A Delphi的常用函数、事件、属性参考 .....189

附录B 计算机软件产品开发文件编制指南 .....196

参考文献 .....215

实训10 利用非数据库的组件 .....148

7.4 Decision Cube组件 .....151

7.4.1 TDecisionCube组件 .....152

7.4.2 TDecisionQuery组件 .....153

7.4.3 TDecisionSource组件 .....154

7.4.4 TDecisionPivot组件 .....154

7.4.5 TDecisionGrid组件 .....154

7.4.6 TDecisionGraph组件 .....155

实训11 利用Decision Cube组件 .....157

界面实现数据库分析 .....158

课后强化训练 .....158

第8章 数据库报表的设计 .....159

8.1 Rave Reports报表设计 .....159

8.1.1 Rave Reports概述 .....159

8.1.2 Rave Reports界面 .....160

8.1.3 Rave Reports组件页 .....161

8.1.4 Rave组件页 .....164

8.1.5 制作Rave报表的基本步骤 .....166

实训12 应用Rave报表 .....167

8.2 Quick Report报表设计 .....171

8.2.1 Quick Report概述 .....171

8.2.2 QR组件页 .....173

8.2.3 制作Quick Report报表的基本步骤 .....173

课后强化训练 .....178

第9章 综合报表设计 .....179

9.1 软件开发概述 .....179

9.1.1 软件生存周期 .....179

6.2 利用ADO组件访问数据库 .....102

6.2.1 数据库为ADOTable .....102

实训6 登录窗口中的实践 .....108

6.2.2 数据库为ADOQuery .....112

实训7 BDE与ADO的变换 .....113

实训8 ADO的综合运用：查询组件 .....113

6.2.3 用ADO连接带密码的Access数据库 .....118

6.2.4 实践动态连接数据库 .....119

6.3 利用dbxpress组件访问数据库 .....120

6.3.1 TSQLConnection组件 .....120

6.3.2 TSQLDataSet, TSQLTable, TSQLQuery和TSQLStorageProc组件 .....121

6.3.3 TSQLMonitor和TSimpleDataSet组件 .....121

课后强化训练 .....121

第7章 数据库控制组件 .....123

7.1 概述 .....123

7.2 数据库控制组件介绍 .....124

7.2.1 TDBCGrid组件 .....124

7.2.2 TDBNavigator组件 .....125

7.2.3 TDBText, TDBEdit和TDBMemo组件 .....127

7.2.4 TDBImage组件 .....127

7.2.5 TDBListBox和TDBCComboBox组件 .....128

7.2.6 TDBCCheckBox组件 .....129

7.2.7 TDBRadioGroup组件 .....129

# 第1章 Delphi基础知识

本章重点：

- Delphi 7概述及其集成开发环境；
- Delphi 7的可视化编程的概念和用它来编写一个程序的基本步骤；
- Delphi应用程序结构。

## 1.1 概述

传说Delphi是世界的中心：宙斯（Zeus）放了两只鹰，一只从东飞，一只从西飞。在两只鹰相遇的地方，宙斯投下了一颗圣石，他宣布这里就是世界的中心。圣石落下的城市就叫Delphi，是古希腊中部的一个小城市，于是Delphi就成了古希腊人心目中的世界中心。

Delphi是Inprise公司的旗舰软件产品之一。作为可视化编程环境，人们希望Delphi能成为世人瞩目的焦点，事实上Delphi已成为和微软Visual C++齐名的软件开发平台。发展至今，从Delphi 1、Delphi 2……Delphi 7、Delphi 8到Delphi 2005，经过不断添加和改进各种特性，Delphi的功能越来越强大。它不但扩展了数据库支持（ADO和InterBase数据库），更能方便地支持Internet、XML、框架概念，具备强大的异常处理功能以及更多的新组件与新特性。Delphi 8、Delphi 2005则支持.NET平台。

“执著的程序员使用C++，聪明的程序员使用Delphi”，人们常常将Delphi和Visual C++这两大主流开发平台相比较，比较它们的技术水平、功能、性能、易用性、稳定性，讨论它们的发展历程和发展前景。它们都是非常优秀的开发平台，各有侧重，各有特点，在许多领域互补发展。

Delphi使用Object Pascal语言作为程序开发语言，Visual C++使用C++语言。对于面向对象的程序设计理论来说，连接、编译得到应用程序，无论是C++、Object Pascal或是Java乃至C#，其结果会使你感觉到它们具有异曲同工之妙，到底选用哪一个作为开发平台，得根据项目特性来决定。

Delphi目前最流行的版本是Delphi 7。Delphi 7继承了Delphi紧凑、快速、高效的开发特点，各方面速度更快，同时提供了一个VCL（Visual Component Library）的应用框架，高效可重用的VCL控件以实现RAD（Rapid Application Development）快速应用开发。

Delphi 7提供了各种开发工具，包括高效的IDE集成环境、图像编辑器（Image Editor）以及各种开发数据库的应用程序（DataBase Desktop、BDE Administrator、SQL Explorer、DataDump等）。

Delphi 7在数据库开发方面具有一定优势，适用于多种数据库结构，从客户机/服务器模式到多层数据结构模式；拥有高效率的数据库管理系统和先进的数据库引擎；拥有大量的数据分析和大量的企业组件，使其成为目前最流行的32位数据库应用开发平台。



### 1.1.1 集成开发环境

Delphi 7有着高效的可视化集成开发环境 (Integrated Development Environment, IDE), 用于快速Windows应用程序开发 (RAD), 整个界面可以根据程序员的喜好自行定制。

Delphi 7默认的IDE主要由主窗口、代码编辑窗口 (Code Editor, Code Explorer)、对象树视图 (Object TreeView)、对象观察器 (Object Inspector)、窗体设计器 (Form) 五部分组成。五个部分均为独立窗体, 除主窗口和代码编辑窗口外, 其余部分均可以临时关闭。

运行Delphi 7后的IDE界面如图1-1所示。

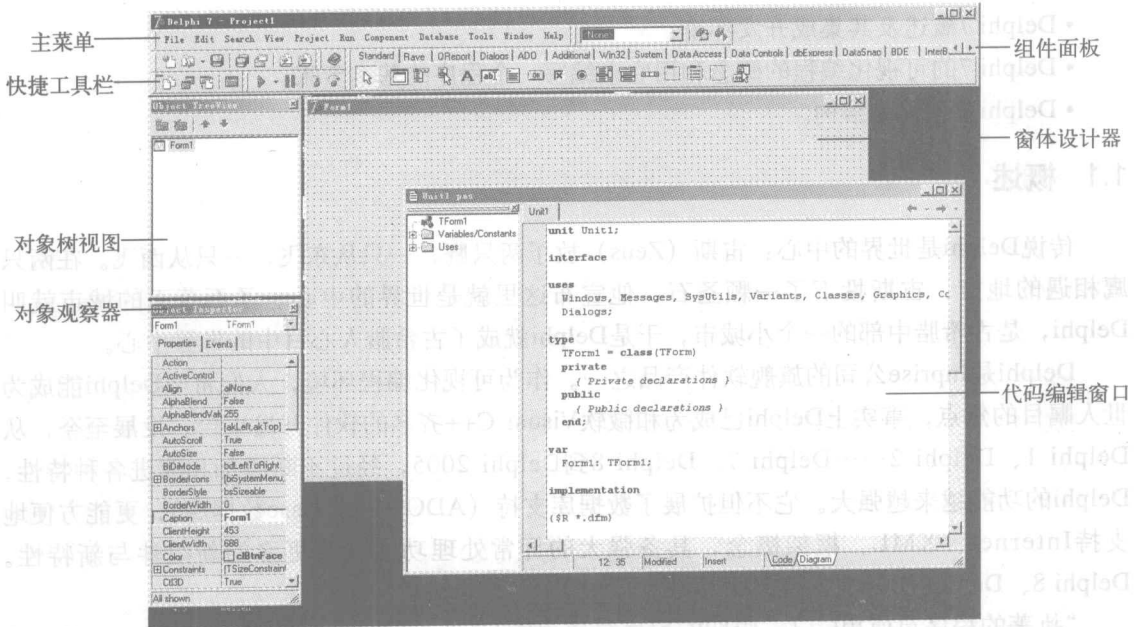


图1-1 Delphi的IDE

#### 1. 主窗口

主窗口由主菜单 (Menu)、快捷工具栏 (Speed Bar)、组件面板 (Component Palette) 三部分组成, 其摆放的位置可以通过鼠标的拖曳而任意变动。

##### (1) 主菜单 (Menu)

利用主菜单, 可以完成Delphi 7的IDE提供的所有功能。包括文件 (File)、编辑 (Edit)、查找 (Search)、视图 (View)、工程 (Project)、运行 (Run)、组件 (Component)、数据库 (Database)、工具 (Tools)、窗口 (Window)、帮助 (Help) 等菜单项, 如图1-2所示。

##### (2) 快捷工具栏 (Speed Bar)

快捷工具栏由两排按钮组成。利用快捷工具栏, 用户可以完成工程的打开、文件的保存、向工程中添加文件、文件的打开、文件的删除、窗体设计器和代码编辑器之间的切换、创建新的窗体以及一些基本的调试功能, 如图1-3所示。



图1-2 主菜单



图1-3 快捷工具栏

### (3) 组件面板 (Component Palette)

组件面板里存放了Delphi预设的各种组件，按各自类型分别放在不同的组件页中。这些组件是Delphi应用程序的组成要素，如图1-4所示。



图1-4 组件面板

Delphi为用户提供了丰富的组件库，既有可视的组件（如编辑框、按钮）等，也有非可视的组件（如定时器、数据源等）。

组件面板上的组件可以根据喜好任意调整。在组件面板的区域单击鼠标右键，出现如图1-5所示的弹出菜单，然后选择属性（Properties）菜单项，出现面板属性对话框，如图1-6所示，在其中可以对面板进行添加、删除、更名、移动等操作，也可以对其中的组件进行隐藏和移动操作。

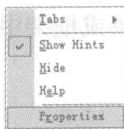


图1-5 调用组件面板属性

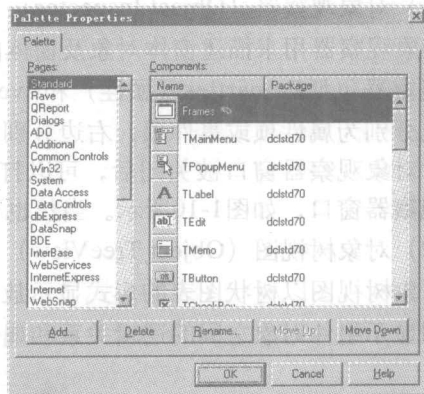


图1-6 面板属性对话框

## 2. 代码编辑窗口

代码编辑窗口用于在其中输入和编辑程序代码。代码编辑窗口由代码编辑器（Code Editor）、代码浏览器（Code Explorer）和信息视图（MessageView）三个窗口组成，如图1-7所示。

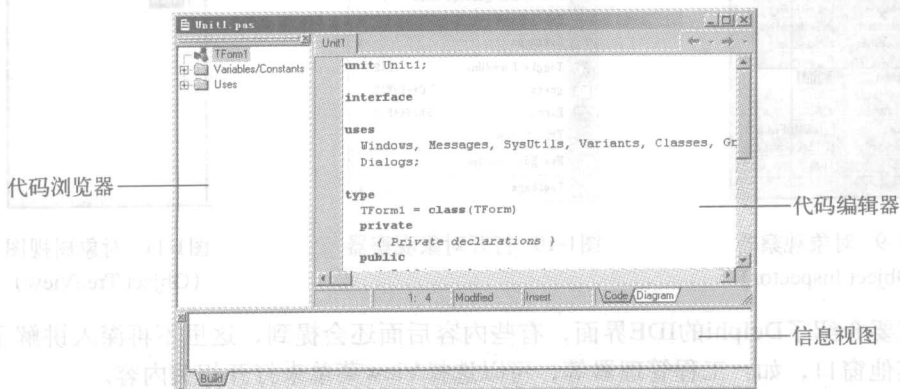


图1-7 代码编辑窗口

代码编辑器不能够单独关闭，而代码浏览器和信息视图两部分可以根据需要单独关闭和打开。如果关闭代码编辑窗口，就关闭了整个项目文件。

通常代码编辑器可以编辑多个文件，分别由多个页面打开。每页的代码对应一个独立文件，可以是单元文件（.pas），也可以是项目文件（.Dpr）以及其他代码文件。

### 3. 窗体设计器

窗体设计器体现了可视化编程环境，使程序设计所见即所得，设计与运行的完全统一使开发人员能轻松把握面向对象的程序设计，如图1-8所示。窗体也是一个单独的文件，其扩展名为.dfm（Delphi Form）。

一个窗体设计器用来设计一个窗体，每个窗体就像一个容器，通过向窗体中添加组件以及更改窗体及组件的属性来完成窗体界面的直观设计。

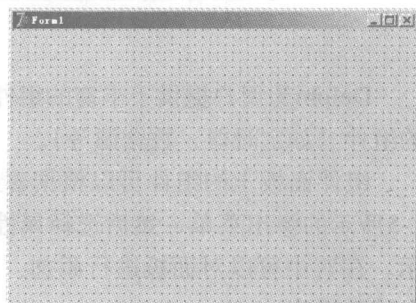


图1-8 窗体设计器

### 4. 对象观察器和对象树视图

#### (1) 对象观察器（Object Inspector）

对象观察器用来描述窗体对象及窗体内组件对象的属性和事件的设计窗体，如图1-9所示。对象观察器包括Properties（属性）和Events（事件）两个标签页。每一页分左右两部分，其中左边分别为属性项或事件项，右边分别为属性值和事件过程名。

当对象观察器窗口被关闭后，可以打开View菜单，然后点击Object Inspector项再次打开对象观察器窗口，如图1-10所示。当然也可以直接按F11功能键来显示对象观察器窗口。

#### (2) 对象树视图（Object TreeView）

对象树视图以树状图表的形式显示放置在一个表单、数据模块或框架中的所有组件对象。树视图显示组件对象之间的逻辑关系和属性联系，如图1-11所示。

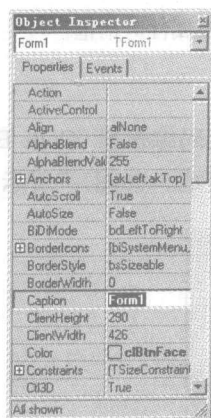


图1-9 对象观察器  
(Object Inspector)

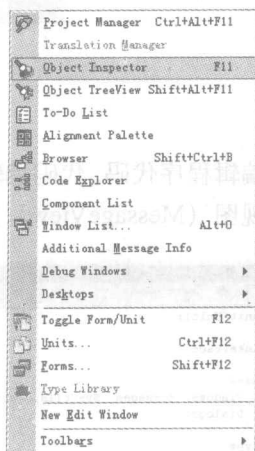


图1-10 打开对象观察器

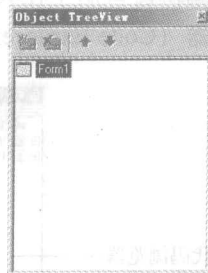


图1-11 对象树视图  
(Object TreeView)

以上简要介绍了Delphi的IDE界面，有些内容后面还会提到，这里不再深入讲解了。其实还有很多其他窗口，如：工程管理等，可以选择View菜单来打开相关内容。

## 1.1.2 IDE界面操作

### 1. 组件操作

#### (1) 选定组件

在组件对象上单击鼠标左键可以选中该组件。为了选中多个组件，可以在窗体空白处按住鼠标左键不放，然后在多个组件所在区域拖曳鼠标，同时可以辅助Shift按键来完成多个组件的选择。

#### (2) 移动组件

单击选中该组件，按住左键不放并拖曳组件到目标位置，释放左键，组件即被移到最接近窗口栅格的位置。如果在拖曳鼠标时按下Alt键，则可以脱离栅格的束缚。

#### (3) 组件的尺寸大小调整与对齐

通常，设计窗体内的组件需要排列整齐，同时除了非可视组件外，组件的尺寸大小和显示字样也需要统一。Delphi 7可以很方便地做这个工作，它提供了窗口的栅格和对齐工具。

1) 手动调整组件尺寸大小和组件对齐。单击选中该组件，此时在该组件范围内会出现8个黑色小方块。用鼠标拖曳任意黑方块，可以自如地伸缩组件。

如果要精确地设置组件的尺寸，可以修改组件的相关属性的值：

- Left: 组件左边缘距离窗体左边框的点数；
- Top: 组件上边缘到窗体上边框的点数；
- Width: 组件的宽度；
- Height: 组件的高度；

利用窗口的栅格可以很方便地依据栅格来调整组件尺寸大小和组件对齐。按下Alt键，同时调整组件尺寸大小和组件对齐就可以脱离栅格的束缚。

2) 利用工具调整组件尺寸大小和组件对齐。可分别利用菜单和对齐面板来调整组件尺寸大小和组件对齐。

• 利用菜单。单击选中该组件，通过主菜单上的Edit菜单作相应的操作，也可以在组件上面单击右键，弹出如图1-12所示菜单，然后单击其中的Position命令即可作相应操作。

Align to Grid: 对齐到栅格。

Align: 出现对齐对话框，如图1-13所示。

Size: 出现尺寸对话框，如图1-14所示。

Scale: 按一定比例缩放。出现比例对话框，如图1-15所示。

• 利用对齐面板。选择主菜单上的View→Alignment Palette菜单项打开Align对话框，如图1-16所示。可以利用此对话框提供的对齐模板直观地调整组件的对齐方式。

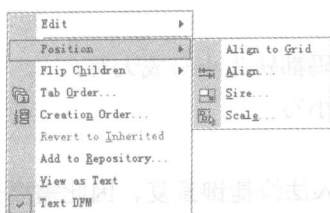


图1-12 调整组件尺寸和组件对齐

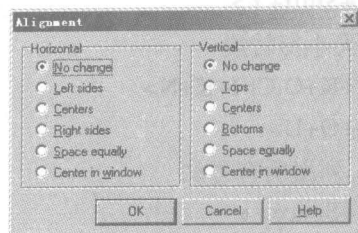


图1-13 Alignment 对话框



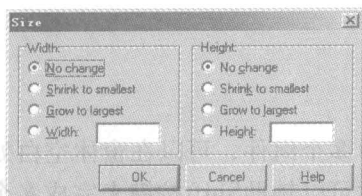


图1-14 Size对话框

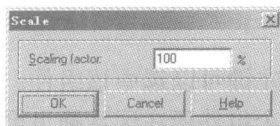


图1-15 Scale对话框



图1-16 Align对话框

#### (4) 锁定组件

为防止已定位的组件任意移动，可以将组件锁定。选择主菜单上的Edit→Lock Controls菜单项，使得组件不能进行移动操作，同时其尺寸也不能改变。

组件的解锁只需再次作上述操作即可。

### 2. 代码编辑时的快捷键

#### (1) 块选代码

<Alt+鼠标左键> (或者 <Shift+Alt+方向键>)

#### (2) 代码模板

<Ctrl+J>

#### (3) 代码整块移动

<Ctrl+Shift+I> (右移) <Ctrl+Shift+U> (左移)

#### (4) 源码定位

<Ctrl+左键>将鼠标指向变量名、单元名、类名，再用该热键可定位到相应的说明

#### (5) 定义与实现切换

<Shift+Ctrl+向上(或向下)的方向键>可与相应的过程、函数、事件的定义切换光标位置

#### (6) 补上声明或函数

<Ctrl+ Shift + C>

#### (7) 书签

<Ctrl + Shift + n (n=1, 2, 3, 4...)> 定义第n个书签

<Ctrl + n (n=1, 2, 3, 4...)> 跳到书签n

#### (8) 删除

<Ctrl+T> 删除光标右边的一个单词

<Ctrl+Y> 删除光标所在行

<Ctrl+Backspace> 后退删除一个词

<Ctrl+Shift+Y> 删除当前光标之后至本行末尾之间的文本

#### (9) 大小写转换

<Ctrl+K+O, Ctrl+K+N> 将选中的整块代码都转为小写或大写

<Ctrl+O+U> 切换选择块的大小写

#### (10) 列出所有方法

<Ctrl+Space>

与Windows的输入法快捷键重复，因此需要先关闭输入法的快捷键才能具有列出所有方法的功能

## (11) 组件调整

<Ctrl+UP>	向上移动当前组件 (精确)
<Ctrl+Left>	向左移动当前组件 (精确)
<Ctrl+Down>	向下移动当前组件 (精确)
<Ctrl+Right>	向右移动当前组件 (精确)
<Shift+UP>	减小当前组件的高度
<Shift+Left>	减小当前组件的宽度
<Shift+Down>	增加当前组件的高度
<Shift+Right>	增加当前组件的宽度
<Esc>	层层往后选, 直到被选为当前组件窗口

## (12) 快速选择

<Ctrl+E>	快速选择
----------	------

## (13) 插入GUID

<Ctrl+Shift+G>	
----------------	--

## (14) 回车换行

<Ctrl+M>	同Enter键
<Ctrl+N>	同Enter键, 但光标位置保持不变

## 3. 获得帮助

Delphi 7提供了强大的帮助系统。在Delphi IDE环境中按下F1键就会显示与当前对象相关的帮助信息, 同时也可以直接通过Help菜单获得帮助。

在Delphi 7的代码编辑器中, 在对象后加“.”(一个小圆点)即可获得该对象及其所有子对象的所有属性和方法, 在输入每个方法名或函数名后加上“(”即可获得此方法或函数的参数列表, 从而极大地提高了编程效率。

## 1.2 可视化编程基础

## 1.2.1 第一个Delphi应用程序

编写任何一个Delphi应用程序, 其基本步骤几乎相同, 都要经历建立工程项目文件、窗体设计、运行调试、保存退出等步骤。下面通过一个简单的实例来介绍整个编程过程。

## 1. 建立工程项目文件

工程项目文件是整个应用程序开发的组织者和管理者, 它能将一个个相对独立的源文件链接成一个整体, 同时对相关文件的信息加以存储和控制, 也就是说它包含了与该工程有关的全部文件和对象的信息。

新建应用程序其实就是要新建一个工程项目文件, 这是应用程序的起点:

启动Delphi, 打开File菜单, 选择New菜单项, 然后选择Application菜单项, 新建一个应用程序, 如图1-17所示。

其实当Delphi刚启动时, 就默认新建了一个应用程序。而应用程序本身就是一个工程项目文件(Project), 以后要打开应用程序, 也就是要打开一个工程项目文件。可以选择File菜单下的Open Project...菜单项。

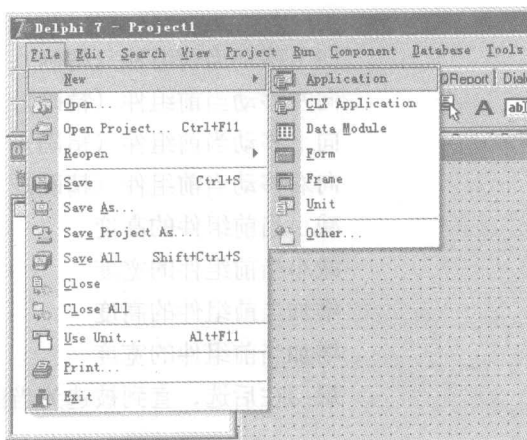


图1-17 新建应用程序

对工程项目文件必须进行简单的设置。打开Project菜单，单击Options... 菜单项，会弹出一个工程选项（Project Options）对话框，然后选择应用程序（Application）标签页，如图1-18所示。在标题文本框（Title）中输入：“我的第一个程序”，这将是运行这个应用程序的任务栏标题。

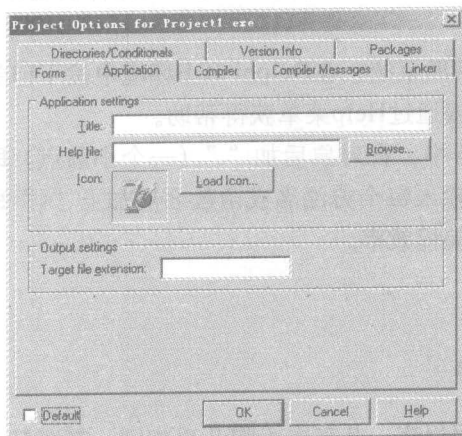


图1-18 工程选项对话框

这时可以看到程序的图标是Delphi 7默认的图标，单击Load Icon ...可以更换该图标。注意通常图标文件的格式为\*.ico。

## 2. 窗体设计

当新建了应用程序后，会出现一个标题为Form1的窗体。窗体设计主要步骤如下。

### (1) 修改窗体属性

选中Form1，在对象观察器中就会产生Form1的属性列表和与之相关的事件列表。如图1-19所示。调整下列属性：

- Caption: Form1窗体的标题。修改其属性值为：“我的程序”。
- Name: Form1窗体对象的名字，注意与Caption的区别。Name的值不能为空，而且在整个工程项目文件中是唯一的。而Caption可以为空，内容可以任意。将Name属性值改

为frmTest。

- Position: 当程序运行时,窗体所在的位置。默认为设计时的位置 (poDesigned), 现改为屏幕的中心 (poScreenCenter)。

(2) 向窗体中添加可以完成应用程序任务的组件

在组件面板的Standard组件页中找到 **A**, 单击它, 然后再在Form1上单击一下, 出现一个Label1标签, 或双击 **A** 组件也可, 如图1-20所示。

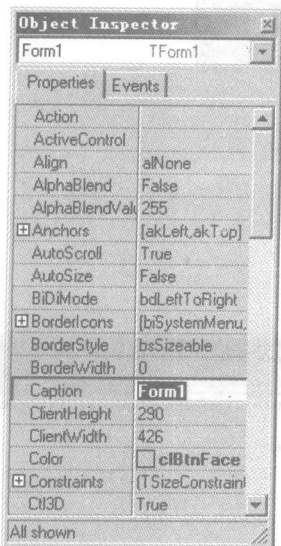


图1-19 Form1的属性

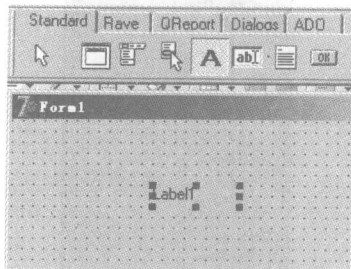


图1-20 添加一个标签对象Label1

使用相同的方法, 在Standard组件页中选择 **ON** 来添加一个按钮对象Button1。

(3) 对选中的对象设置或调整属性值

当选中Label1对象后, 在对象观察器中就会产生这一对象的属性列表和与这一组件对象相关的事件列表。调整下列属性:

- Caption: 在Form1中显示的内容。修改其属性值为: “Hello, World, 这是我的第一个Delphi应用程序!”。
- Name: 对象的名字, 注意Name的值在整个窗体中应是唯一的。将Name属性值改为lblTest。
- Color: 标签的背景颜色, 将Color属性值改为clYellow (黄色)。
- Font: 标签的字体。打开Font属性, 将字体名称Name的属性值改为黑体, 将字体大小Size属性值改为12。

同样, 将Button1的Caption属性改为: 退出, 将Name属性值改为btnTest, 如图1-21所示。

(4) 将组件的事件或方法与程序代码相联系, 用程序代码来实现组件的功能

确认选中的对象为按钮btnTest, 然后切换对象观察器为Events标签页。找到OnClick, 我们要通过OnClick事件来实现单击按钮时退出程序的操作。

双击右边空白处, 会激活代码编辑窗口。如图1-22所示, 在btnTestClick过程的begin和end;之间输入close;。



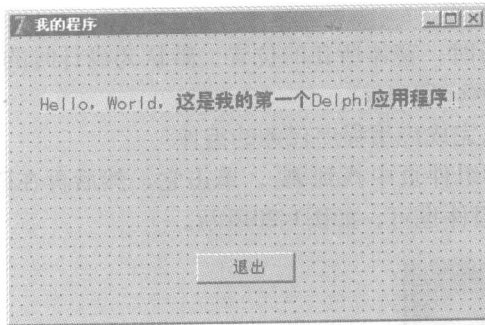


图1-21 窗体设计的结果

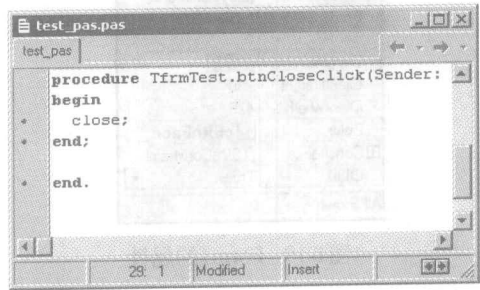
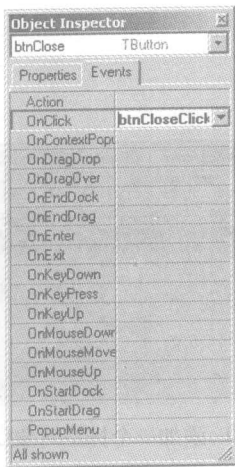






图1-22 编辑程序代码

### 3. 保存项目所属文件

在File菜单中单击Save菜单项，保存当前窗体的窗体文件和代码文件；也可以利用快捷工具栏上的  (Save) 和  (Save As ...) 按钮来保存。 是保存所有的文件 (Save all)，包括工程项目文件及其他窗体的所有文件。建议每隔一段时间就单击一下  按钮。

第一次存盘，会弹出Save...as对话框，共需要保存两类文件：\*.pas和\*.dpr，前者是代码文件，后者是工程项目文件。这里分别取名为test\_pas.pas和test.dpr。

观察一下存盘文件夹，会发现整个工程主要包括如下文件：

- 工程项目文件.dpr；
- 工程选项文件.dof；
- 配置文件.cfg；
- 资源文件.res；
- 表单文件.dfm；
- 单元文件.pas。

其中前4个文件与工程项目文件同名，都是以test作为文件名，后2个文件使用单元的名字，都是Test\_pas。

扩展名前带~的文件为临时文件。