

157
北京科海培训中心

►学以致用计算机技术丛书

实用 Delphi 6 教程

陈海波 编著



A1000985

科学出版社

2002

内 容 简 介

Delphi 6 是 Inprise 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。本书通过编写大量实用而有趣的程序,引导读者掌握在 Delphi 环境下开发应用程序的技术。

全书共 12 章。前 3 章是全书的基础,介绍了 Delphi 6 的基本开发环境及语言基础 Object Pascal;第 4 章到第 11 章从基本组件、建立窗体、多媒体技术、网络应用、多线程技术、数据库开发等多方面系统地讲解在 Delphi 环境下开发应用程序的常用方法和技巧;最后在第 12 章介绍了几个综合应用实例(如 FTP 下载工具),供读者提高进阶之用。全书每章之后均附有练习题,供读者巩固所学知识,以便提高实际应用能力。

本书旨在达到学以致用目的,内容叙述由浅入深、循序渐进,适用于 Delphi 初中级用户;同时一些大程序的开发过程及编程技巧对有经验的程序员来说也有很好的参考与借鉴价值。

图书在版编目(CIP)数据

实用 Delphi 6 教程/陈海波 编著.—北京:科学出版社,2002.2

ISBN 7-03-010125-1

I. 实... II. 陈... III. DELPHI 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 008203 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

北京市耀华印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2002 年 3 月第一版

开本: 787×1092 1/16

2002 年 8 月第 2 次印刷

印张: 28.25

印数: 5001-7000

字数: 687 040

定价: 38.00 元

前 言

Delphi 6是Inprise公司（原Borland公司）推出的Delphi编程软件的最新版本，它采用面向对象的程序设计、组件化的编程方式、快速Pascal编译器、OLE自动化、ActiveX编程、数据库以及Internet编程支持，从而使得Delphi 6开发功能更加强大，更加易于学习和使用。

Delphi从一开始出现在市场上，便受到了广大用户的喜爱。Delphi 6更是在旧版本的基础上有了长足的改进：一方面继续保持着原来的优势——例如提供数量众多、功能强大、使用方便的各种组件及非常优秀的数据库开发支持；另一方面弥补了旧版本的不足，大力开拓新的功能和应用——例如提供了更完备更优秀的网络功能组件，使Delphi有了更加强大的网络功能，适应了当今网络迅猛发展的趋势。

可以说，Delphi 6已经具备了作为最优秀的编程开发软件的一切条件：它与Microsoft公司的VC/C++相比，其优势是具有简单易用的IDE开发环境；与VB相比，它的代码更加规范，增强了程序的可移植性，并且开发效率更高。因此，作为一名编程爱好者，Delphi 6应该是不容错过的。为了满足广大读者的愿望和需求——迅速掌握Windows环境下的Delphi编程方法，我们编写了本书。

全书共12章。前3章是全书的基础，介绍了Delphi 6的基本开发环境及语言基础。第4章通过讲述如何制作一个文档处理程序，介绍了Delphi 6中菜单、工具栏以及状态栏等组件的使用；第5章则以多页窗体的设计为重点，讲解了Delphi 6中一般窗体的设计开发；第6章通过制作一个多种格式文件浏览器，介绍了多文档界面的设计；第7章通过开发一个类似于Windows附件“画图”的图像编辑程序，介绍了Delphi 6的图形图像技术；第8章以开发一个CD播放器为主介绍了Delphi 6的多媒体技术；第9章则以两个生动的示例演示了Delphi 6的多线程应用；第10章通过一个网页浏览器和一个发送电子邮件程序的制作，介绍了Delphi 6中的Internet支持；第11章则主要通过设计一个客户资料管理器，介绍了Delphi 6的数据库技术；第12章给出了几个综合应用示例。

通过上面的介绍可以看出，本书在选例时尽量使用大型、实用而又有趣的程序，这样既可以激发初学者对Delphi 6的兴趣，又对中高级用户有实际的指导意义。本书不仅介绍了这些大程序的开发过程，而且还详细地介绍了所用组件的一些重要属性以及设计开发的整体思想，并在完成了程序设计之后，还介绍了与各章主题相关的其他知识内容，力图让读者能够掌握更多的知识。

此外，在每章后面都有练习题，结合本章所开发的程序，给出几个要求很具体的练习，针对前面已介绍的示例程序或进行改进，或进行合并，以督促读者学以致用，更好地掌握所学知识，并尽快使自己步入Delphi 6高手行列。

本书中所有程序源代码可从下列网址查询：<http://www.khp.com.cn>。

作 者
2002年2月

目 录

第1章 Delphi入门	1
1.1 Delphi简介.....	1
1.1.1 Delphi历史.....	1
1.1.2 用Delphi编程的基本概念.....	1
1.2 Delphi 6集成开发环境 (IDE)	3
1.2.1 Delphi 6 主窗口.....	4
1.2.2 代码编辑窗口.....	6
1.2.3 窗体设计窗口.....	9
1.2.4 对象查看窗口.....	10
1.2.5 项目管理窗口.....	11
1.3 Delphi 6的新特性.....	12
1.4 第一个Delphi程序.....	15
1.5 小结.....	19
1.6 练习题.....	20
第2章 Object Pascal语言简介	22
2.1 Object Pascal的面向对象特性.....	22
2.1.1 什么是对象.....	22
2.1.2 从一个对象中继承数据和方法.....	24
2.1.3 对象的范围.....	25
2.1.4 对象公有域和私有域的说明.....	26
2.1.5 访问对象的域和方法.....	26
2.1.6 对象变量的赋值.....	27
2.1.7 建立非可视化对象.....	29
2.2 异常处理机制.....	30
2.2.1 运行时间库异常类(RTL Exception).....	31
2.2.2 对象异常类.....	35
2.2.3 组件异常类.....	37
2.2.4 自定义异常.....	38
2.2.5 利用异常响应编程.....	42
2.3 示例演示.....	43
2.4 小结.....	47
2.5 练习题.....	48

第3章 Delphi 6常用组件	49
3.1 组件及对象	49
3.1.1 组件	49
3.1.2 组件的分类	51
3.1.3 Delphi的组件面板	51
3.1.4 对象	52
3.2 按钮类组件	52
3.3 标签和编辑组件	55
3.3.1 用标签和编辑框设计窗体	55
3.3.2 掩码编辑框	57
3.3.3 Memo组件和RichEdit组件	59
3.4 单选按钮和复选框	63
3.4.1 单选按钮和复选框的使用方法	63
3.4.2 单选按钮组	66
3.5 列表框和组合框	68
3.5.1 列表框	68
3.5.2 自绘列表框	73
3.5.3 组合框	76
3.6 小结	77
3.7 练习题	77
第4章 创建文档处理程序	79
4.1 Delphi菜单设计基础	79
4.1.1 使用MainMenu Designer建立菜单	79
4.2 动态修改菜单	83
4.2.1 菜单项的有效和无效	83
4.2.2 复选标记	84
4.2.3 隐藏菜单	86
4.2.4 单选菜单项	87
4.2.5 动态创建菜单项	89
4.3 设计文档处理程序主菜单	92
4.4 设计文档处理程序对话框	100
4.4.1 无模式对话框	100
4.4.2 模式对话框	103
4.4.3 完善文档处理程序的窗体设计	105
4.5 编写代码	106
4.5.1 MDIFrame父窗口程序设计	106
4.5.2 MDIEdit子窗口程序设计	108
4.5.3 About窗口程序设计	114

4.6 小结.....	116
4.7 练习题.....	116
第5章 设计并建立窗体.....	118
5.1 窗体属性.....	118
5.2 窗体设计.....	120
5.2.1 创建窗体.....	120
5.2.2 修改标题.....	120
5.2.3 选择和放置组件.....	121
5.2.4 编辑组件.....	122
5.2.5 控制Tab顺序和Create顺序.....	123
5.2.6 改变窗体设计环境.....	124
5.2.7 保存窗体.....	124
5.2.8 窗体的文本描述.....	124
5.3 利用模板生成窗体.....	125
5.4 多页窗体.....	127
5.4.1 使用TPageControl建立多页窗体.....	127
5.4.2 使用TTabControl建立多页窗体.....	136
5.4.3 使用TNoteBook和TTabSet建立多页窗体.....	137
5.4.4 使用TTabbedNoteBook建立多页窗体.....	143
5.5 分割窗体.....	143
5.6 小结.....	145
5.7 练习题.....	145
第6章 多文档界面设计.....	147
6.1 MDI的组成.....	147
6.1.1 父窗口.....	147
6.1.2 子窗口.....	147
6.2 多种格式文件浏览器.....	149
6.2.1 项目的建立.....	150
6.2.2 窗口属性的设置.....	150
6.2.3 向父窗口中添加菜单.....	151
6.2.4 向父窗口中添加工具栏.....	155
6.2.5 向父窗口中添加状态栏.....	158
6.2.6 向父窗口中添加定时器.....	160
6.2.7 创建显示和编辑文本文件的子窗口.....	161
6.2.8 建立显示和编辑RTF文件的子窗口.....	162
6.2.9 建立显示图像文件的子窗口.....	163
6.2.10 创建显示系统信息窗口.....	165
6.2.11 在父窗体中加入标准对话框组件.....	168

6.2.12 在父窗口中加入动作列表组件对象.....	169
6.2.13 编写代码.....	174
6.2.14 改进程序.....	182
6.3 MDI模板.....	183
6.4 小结.....	187
6.5 练习题.....	187
第7章 开发图形/图像编辑程序.....	188
7.1 图形设计综述.....	188
7.2 基本图形绘制.....	189
7.2.1 直线的绘制.....	189
7.2.2 椭圆的绘制.....	191
7.2.3 多边形绘制.....	193
7.2.4 绘制矩形.....	195
7.3 图像编辑工具的开发.....	198
7.3.1 窗体属性的设置.....	199
7.3.2 添加菜单.....	200
7.3.3 创建工具栏.....	204
7.3.4 添加状态栏.....	210
7.3.5 建立图像编辑区域.....	211
7.3.6 其他组件.....	212
7.3.7 信息显示窗口.....	213
7.3.8 图像属性设置窗口.....	216
7.3.9 画笔宽度设置窗口.....	217
7.3.10 编写事件响应程序.....	217
7.4 图像特效.....	231
7.4.1 镜像效果.....	231
7.4.2 推进、交错、百叶窗等效果.....	237
7.5 小结.....	244
7.6 练习题.....	245
第8章 多媒体制作.....	246
8.1 多媒体组件.....	246
8.1.1 TMediaPlayer组件的主要属性.....	247
8.1.2 使用TMediaPlayer组件示例.....	252
8.2 简单的CD播放器.....	254
8.3 CD播放器的改进.....	259
8.3.1 新建项目.....	259
8.3.2 设置窗口属性.....	259
8.3.3 添加菜单.....	261

8.3.4 添加工具栏	262
8.3.5 添加状态栏	266
8.3.6 添加标签	267
8.3.7 添加媒体播放器组件	268
8.3.8 添加其他组件	269
8.3.9 系统信息显示窗口	271
8.3.10 编写代码	271
8.4 多媒体动画	279
8.4.1 位图动画	279
8.4.2 图标动画	285
8.5 小结	288
8.6 练习题	288
第9章 多线程应用	289
9.1 线程的基本概念	289
9.1.1 线程的优先级	290
9.1.2 线程的同步	291
9.1.3 线程的局部存储 (TLS)	291
9.2 定义线程对象	291
9.2.1 创建线程对象	292
9.2.2 初始化线程对象	293
9.2.3 编写线程函数	294
9.2.4 编写线程的清除代码	296
9.3 使用线程对象	296
9.3.1 线程的同步	296
9.3.2 执行线程对象	297
9.4 利用多线程排序	298
9.5 动态添加线程示例	308
9.6 小结	318
9.7 练习题	319
第10章 Internet编程	320
10.1 Socket编程	321
10.1.1 Socket基本概念	321
10.1.2 Socket组件	322
10.2 制作浏览器	333
10.3 制作网页	337
10.4 发送电子邮件	344
10.5 小结	359
10.6 练习题	359

第11章 数据库应用	361
11.1 数据库系统概述	361
11.1.1 数据库管理系统 (DBMS)	362
11.1.2 数据库应用程序	362
11.2 Delphi数据库原理	364
11.3 开发一个简单的数据库应用程序	365
11.3.1 使用BDE Administrator配置数据库驱动程序参数	365
11.3.2 使用Database Desktop建立一个雇员表	366
11.3.3 使用Database Form Wizard生成数据库应用程序	368
11.4 Delphi 6数据库基本组件	369
11.4.1 TTable和TQuery组件	369
11.4.2 TDataSource组件	375
11.4.3 TDataBase组件	376
11.4.4 TStoredProc组件	377
11.4.5 TSession组件	377
11.4.6 TDBText组件	378
11.4.7 TDBEdit组件	378
11.4.8 TDBGrid组件	378
11.4.9 TDBNavigator组件	380
11.4.10 TField组件	381
11.5 设计客户资料管理器	387
11.5.1 新建项目	388
11.5.2 设计数据库模块	389
11.5.3 设计主窗口	391
11.5.4 创建和设计过滤窗口	396
11.5.5 “关于”窗口设计	400
11.5.6 数据库模块的程序代码	401
11.5.7 主窗口程序设计	402
11.5.8 过滤窗口程序设计	406
11.5.9 其他工作	409
11.6 数据库专家工具	411
11.7 小结	414
11.8 练习题	414
第12章 综合练习	415
12.1 练习一 利用API创建多线程	415
12.2 练习二 FTP下载工具	418
12.3 练习三 使用TDBText显示数据库数据	435
附录 练习题答案	438

第1章 Delphi入门

Delphi是全新的可视化编程环境，它是一种方便、快捷的Windows应用程序开发工具。它使用了Microsoft Windows图形用户界面的许多先进特性和设计思想，采用了可重复利用的完整的面向对象语言（Object-Oriented Language），以及当今世界上最快的编辑器、最为领先的数据库技术。对于广大的程序开发人员来讲，使用Delphi开发应用软件，无疑会大大地提高编程效率，而且随着应用的深入，你将会发现编程不再是枯燥无味的工作——Delphi的每一个设计细节，都将带给你一份欣喜。

Delphi 6的出现，无疑给广大的Delphi用户带来了福音，因为它在性能上更加完善、组件更加丰富，特别是在Web服务功能方面，给网络编程带来了更大的便利。下面我们将从Delphi的最基本概念讲起，带领Delphi爱好者进入Delphi 6的殿堂。

1.1 Delphi简介

Delphi是由Inprise公司（原Borland公司）开发的一个可视化的软件开发工具，它具有开发速度快、功能强大以及容易使用的可视化开发环境。下面将向读者简单介绍有关Delphi系列的基本情况。

1.1.1 Delphi历史

随着面向用户、界面友好的Windows操作系统的出现，这种图形化操作系统（GUI）立刻风靡全球。用户仅凭鼠标就可以在各个窗口之间完成各种操作。但是，由于窗口的程序设计要比传统的程序设计复杂得多，因此对于一个程序设计者来说，如何开发一个窗口应用程序，就成为一件令人头疼的事。随着面向对象、可视化等技术的进一步发展，不少人提出，能否利用Windows系统可视化好的特点来方便地进行Windows下的程序设计呢？

Delphi就是在这个时候应运而生的。它是由以制作编译器而闻名于世的Inprise公司推出的，采用Object Pascal语言作为开发语言。其最大的优点就在于它提供了几乎完美的可视化特性、完全的面向对象、强大的数据库开发功能和高效率的编译器。

Delphi一经推出，就受到了广大用户的好评，迅速成为开发Windows应用程序最主要的工具之一。随着计算机技术的不断发展，Delphi也不断的推出新版本，最近推出的Delphi 6版本，又给Delphi系列注入了新的活力，其新特性我们在后面将会详细介绍。

1.1.2 用Delphi编程的基本概念

Delphi作为可视化、面向对象的编程语言，其编程思想自然也跟我们传统的基于过程

的编程思想有很大不同，编写Windows下的应用程序又跟以前DOS下的应用程序完全不同。因此，在这里我们首先要弄清楚有关使用Delphi进行Windows程序设计的一些最基本的概念。在后面的章节还会对这些基本概念的应用进行详细介绍。

面向对象

Delphi作为一个可视化的软件开发包，其最根本的思想就是“面向对象”（Object Oriented）。所谓“面向对象”就是把一切事物都看成对象，无论是Delphi开发环境直接提供给你的各种组件、窗体，还是程序中自己定义的各种对象。一个对象是一个数据类型，它包括了数据以及程序代码；而在面向对象的程序设计出现之前，程序代码和数据通常被视为独立的部分。

为了方便理解，不妨将对象想象成一种特殊的变量，它保存着数据，并包括对自身的操作。每个对象都有自己的存储空间，还可以容纳其他对象。程序就是一大堆对象的组合，对象之间只能通过传递消息来互相联系、互相作用。一组具有相同特性的对象称作“类”，每个对象都是某个类的实例。一般认为，面向对象具有以下三个基本特性：

- 封装（Encapsulation） 封装就是将代码和它操作的数据打包在一起，形成一个黑盒子，外界只能通过定义好的接口来访问它，这个黑盒子就叫做对象。通过封装，保证了数据和代码的隐藏性和安全性，从而保证了系统的模块化程度，这也是符合软件项目的观点的。
- 继承（Inheritance） 就像遗产的继承一样，一个类也可以“继承”另外一个类的特性，同时还可以有自己独有的特征，这样就得到了一个新的类。原来那个被继承的类叫做“父类”，或“祖先类”，新产生的那个类叫做“子类”，或“子孙类”。
- 多态（Polymorphism） 多态是“一个接口，多种方法”的思想，它可以为一组相关但不相同的操作定义同一个接口名称，具体执行哪个特定动作将由数据来确定。

有关Object Pascal语言的面向对象特性，我们在下一章中进行详细介绍。

Windows 程序设计

在Windows环境下的程序设计和以往DOS下的编程有很大的不同，下面介绍用Delphi开发Windows程序时涉及到的几个基本概念：

窗体（Form） 窗体就是我们在Windows操作系统中随处可见的窗口，在用Delphi进行程序设计时，我们称它们为窗体。它是程序的基本表现形式和最常见的组成单元。

消息（Message） 消息可以是在某个按钮上单击了鼠标左键，也可以是键盘上的某个键被按下；可以是系统的时钟计时到了某个时间，也可以是某个函数传回一个值。各个对象之间就是通过消息来互相通信，互相调用某些操作。整个Windows下的程序都是基于消息驱动的。

组件（Component）、属性（Property）和事件（Event） 前面已经反复强调过，Delphi结合了可视化和面向对象两大特点，由于它是可视化的，因此在设计时，Delphi会提供给我们一些可以直接看到、直接使用的组件，比如：窗体、按钮、文本框等等，它们都是组

件。由于它是面向对象的，自然就都具有自己的属性和事件。一般而言，属性就是这个组件的静态特性（通常是一些数值设置），而事件则是组件的一些动态的动作（通常是需要用户填写代码）。

1.2 Delphi 6集成开发环境（IDE）

Delphi 6作为一个优秀的程序开发工具，为用户提供了一种方便、快捷的应用程序开发环境。对于广大的程序开发人员来说，使用Delphi 6开发应用程序，将会极大的提高编程效率。随着应用的深入，用户将会进一步体会到它的众多优点。

Delphi 6是一个容易上手的开发系统，下面我们将为初学Delphi的入门者详细描述Delphi 6 IDE的每个主要组件，介绍用户常用工具、窗口及功能。要想成为一个优秀的Delphi程序开发人员，必须深入了解IDE的所有功能，并能熟练地运用它们。

在开始菜单中找到Delphi 6程序启动图标，如图1-1所示。

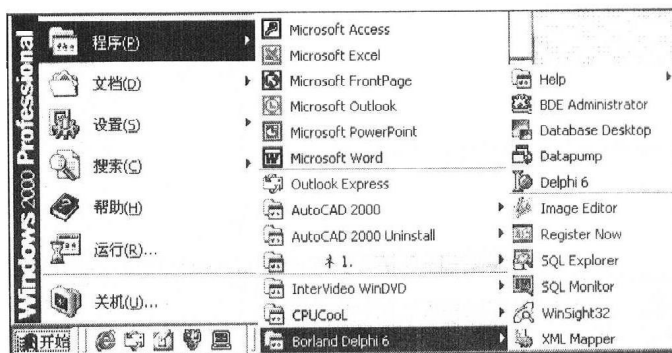
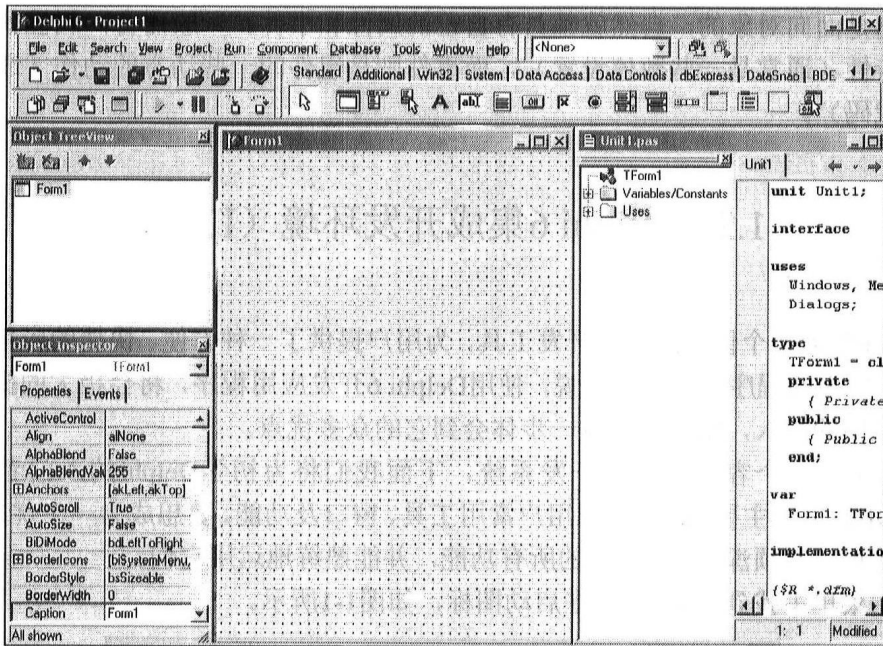


图 1-1 Borland Delphi 6 程序组

单击该图标，出现Delphi 6的启动画面，最后出现了Delphi 6的运行界面，也就是它的开发环境，如图1-2所示。

下面我们将介绍此开发环境中的各个窗口。这些窗口包括：

- Delphi 6主窗口
- 代码编辑窗口
- 窗体设计窗口
- 对象查看窗口



1.2.1 Delphi 6 主窗口

Delphi 6主窗口位于整个IDE的顶端，如图1-3所示。

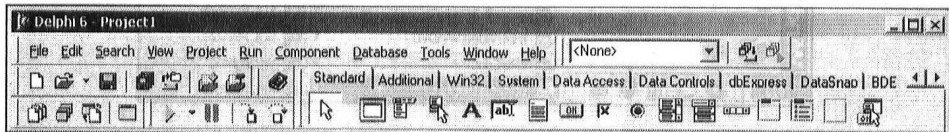


图 1-3 Delphi 主窗口

主窗口是Delphi IDE的核心，通过主窗口，开发人员可以控制包括从项目创建、程序编制、调试到运行的整个开发过程。开发人员还可以通过主窗口控制它的从属窗口、访问帮助系统、维护应用程序等功能。

主窗口包含了很多菜单命令、工具栏命令和组件按钮。通过这些工具，用户可以访问Delphi提供的许多优秀特性和功能。主窗口包括以下几部分：

Menu（菜单栏）

Windows中的大部分应用程序都包括菜单栏，用户通过选择菜单栏中的命令，可以完成应用程序提供的特定功能。Delphi 6的菜单栏提供了Delphi主要应用中的大部分功能。

Speed Bar（快捷工具栏）

快捷工具栏如图1-4所示。它实际上由一组工具栏组成，每个工具栏上面包括一系列的图标按钮，按钮上的图标直观的显示了该按钮所实现的功能。当用户将鼠标移动到某个可选按钮时，该按钮将凸起。按钮变灰表示该按钮对应的命令当前不可用，通过快捷工具栏

用户可以快速实现某些命令，而不用选择相应的菜单命令。

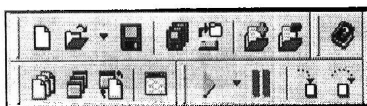


图 1-4 快捷工具栏

在快捷工具栏的非按钮区按住鼠标左键拖动鼠标可以移动该工具栏。可以将其移动到主窗口的其他位置或使其脱离主窗口而变为浮动。

在快捷工具栏上单击鼠标将弹出快捷菜单，用户可以选择显示或关闭某个工具栏，或选择Customize选项弹出对话框。

Component Palette (组件模板)

标准的组件模板如图1-5所示。

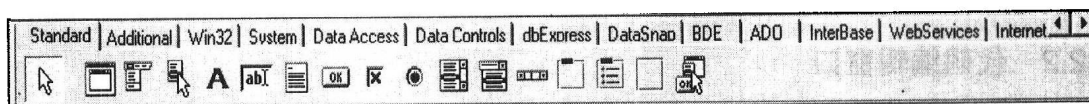


图 1-5 组件模板

组件模板包含一系列页面，每个页面显示了Delphi组件中的一类组件，用户可以选择不同页面的组件加入到自己的窗体中。

每个组件就是一个功能集合，用户窗体中的编辑框、按钮或标签等都是组件，用户通过在窗体中添加组件，可以使一个窗体实现用户希望的功能。

虽然Delphi为用户提供了默认的组件模板，但用户也可以配置个性化的组件模板，按照需要可增加或删除页面，或者删除或增加页面中的组件。

可以使用如下步骤定制组件模板：

(1) 在组件模板上单击鼠标左键，然后在弹出菜单中单击Properties选项，将弹出Palette Properties对话框，如图1-6所示。

(2) 加入新页面 将焦点移到左边的Pages列表框，单击Add按钮，在弹出的Add Page对话框中输入名称，然后单击OK按钮，则新页面加入到Pages列表框中。

(3) 移动页面位置 选中Pages列表框中的某页面，然后单击Move Up或Move Down按钮将移动页面在组件模板中的位置。

(4) 修改页名 选中Pages列表框中的某页面，然后单击Rename按钮，在弹出的Rename Page对话框中输入新的名称，然后选择OK按钮。

(5) 隐藏组件 选中Components列表中的某个组件，然后单击Hide按钮。

(6) 在页面中加入新组件模板 选中窗体中的某个组件，设置它的属性，然后单击Component菜单中的Create Component Template命令。将弹出Component Template Information对话框。在Component name中输入组件名称，然后在Palette Pages列表框中选一个页面，选择Change按钮可以改变组件按钮的图标。

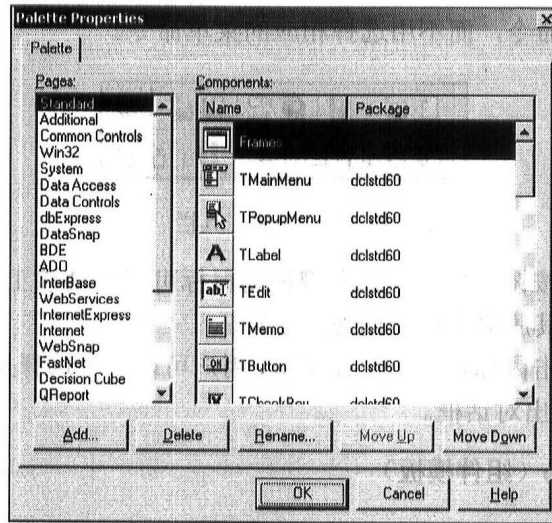


图 1-6 Palette Properties 对话框

1.2.2 代码编辑窗口

在Delphi中，代码编辑工作是在如图1-7所示的代码编辑窗口完成的，代码编辑器是一个高性能的编辑器，它提供了许多强大的功能，例如以不同颜色高亮显示各代码块、支持多步Undo功能及代码管理等。

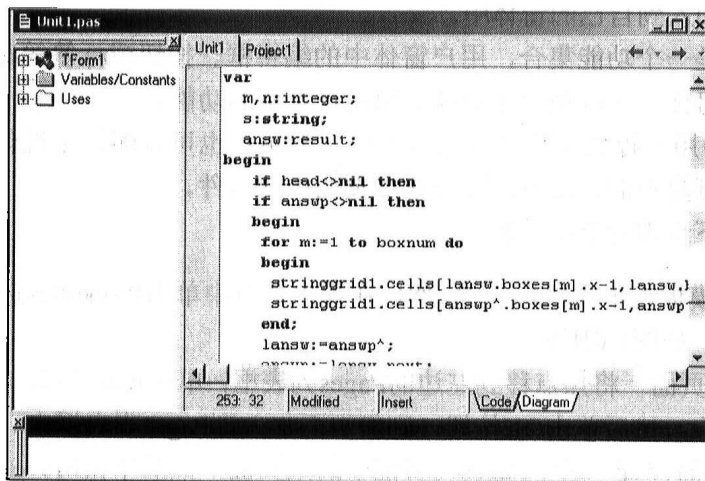


图 1-7 代码编辑窗口

下面我们来介绍关于代码编辑器的某些重要功能。

弹出菜单

在代码编辑窗口单击鼠标右键，将出现弹出菜单，菜单中包含了许多与编辑窗口相关的命令。这些命令包括：

- **New Editor Window**命令 使用该命令将新打开一个代码编辑窗口，使用户可以同

时查看两个窗口中的代码，这样有利于对照两个窗口中的代码，进行窗口间代码的拷贝和对比。

- **Complete Class At Cursor**命令 在类定义的内部单击鼠标右键并选择该命令，该命令将通过生成框架代码来自动完成该类的定义。例如，下面为一个类的定义：

```
type TMyButton=class(Tbutton)
  property Size:Integer;
  procedure DoTest;
end;
```

然后选择该命令，将自动生成如下代码：

```
type TMyButton=class(TButton)
  published
  property Size:integer read FSize write SetSize;
  Procedure DoTest;
  private
    FSize: integer;
    procedure SetSize(const Value: integer);
end;
implementation
{ TMyButton }

procedure TMyButton.DoTest;
begin

end;

procedure TMyButton.SetSize(const Value: integer);
begin
  FSize := Value;
end;

end.
```

- **Toggle Bookmarks** 以及 **Goto Bookmarks** 命令 在代码编辑器中使用 **Toggle Bookmark**命令可以定义或取消书签，而使用**Goto Bookmarks**命令可以转到某个定义的书签处。使用“Ctrl+书签编号”组合键同样可以转到相应的书签处。
- **Read Only**命令 选择该命令将使编辑器中的代码变为不可编辑，这样可以防止用户错误操作而使得程序遭到破坏。
- **Open File At Cursor**命令 如果用户将鼠标指到代码中的一个单元文件名处，单击右键并选择该命令，将会打开该文件。
- **Properties**命令 选择**Properties**命令将激活**Editor Properties**对话框，如图1-8所示。使

用该对话框用户可以设置代码编辑器的属性，包括设置编辑器的一般属性、代码格式、代码字体以及颜色等。

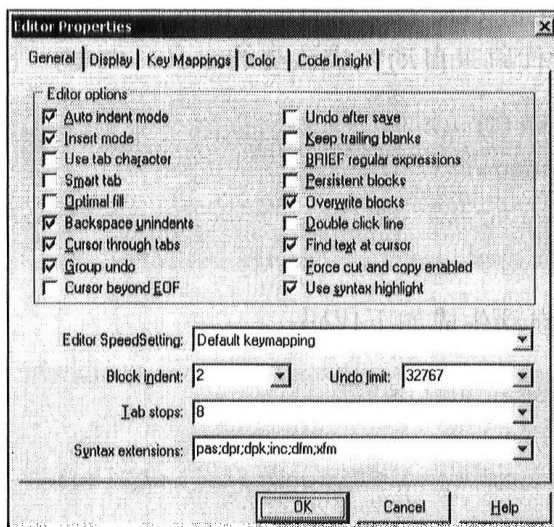


图 1-8 Editor Properties 对话框

代码管理器使用户可以很容易的操作单元文件并将类的创建自动化，默认情况下它位于代码编辑器的左边，如图1-7所示。

单击代码编辑器右上角的小按钮可以关闭代码编辑器。如果要重新打开，可以参考前面介绍的菜单命令进行操作。

代码编辑窗口包括了一个树形视图，它列出了用户单元中所有的类型、类、属性、方法、全局变量以及过程变量，同时还列出了在uses子句中给出的其他单元。

在代码编辑器和代码管理器之间切换可以使用Ctrl+Shift+E组合键。

如果用户在代码管理器中选中了某项，代码编辑器中的光标将移动到该项对应成员的实现代码处。若想增加或者重新命名代码管理器中的某项，在代码管理器的相应节点处单击右键，然后选择New或Rename选项。

要调整代码管理器的设置，选择Tools菜单中的Environment Options命令，然后选择Explorer页面。如图1-9所示。

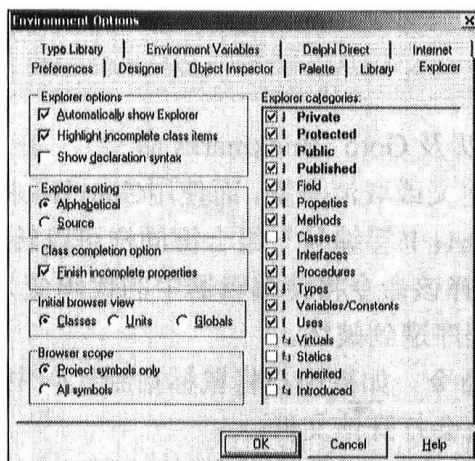


图 1-9 设置代码管理器