

流行编程软件简明教程丛书

Delphi

简明教程

Delphi 编程快速入门

Windows 通用对话框

菜单程序的开发和设计

Delphi 中的图形绘制

用户窗体的设计和使用

常见 VCL 控件的使用

管理文件、目录和驱动器

数据库开发和管理

多媒体的设计和使用

习题与答案

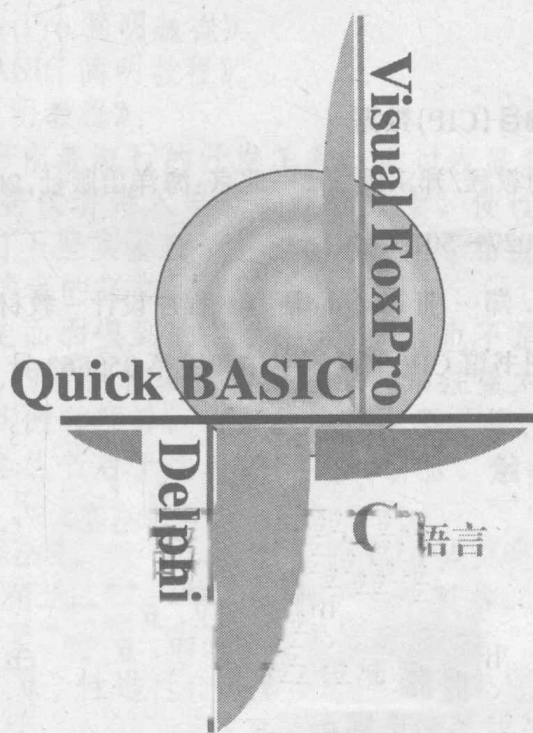


海洋出版社

流行编程软件简明教程丛书

Delphi 简明教程

郑 洪 编著



海洋出版社

2001 年 · 北京

内容提要

Delphi 5 是由软件界的巨头人士、著名的 Inprise 公司开发的新一代可视化程序开发工具, 一直以来受到计算机编程爱好者和各机构专业人士的广泛青睐。为了满足广大读者的愿望, 迅速地掌握 Windows 环境下的 Delphi 编程方法, 编者编写了本书。

全书共分为 9 章, 主要包括 Delphi 5 中 VCL 控件的使用, 文本、图形和图像的处理, 菜单和对话框设计, 文件管理技巧, 多媒体程序制作, 窗体设计方案, 以及 Delphi 中引人入胜的数据库开发应用。书中融合了利用 Delphi 5 开发 Windows 应用程序的理论和实践, 全面深入地介绍了利用 Delphi 5 开发应用的常用方法和技巧, 实用性很强。

本书资料丰富、内容翔实, 具有易学易用、简洁明了的特点, 适合于对 Windows 编程有一定了解的读者使用, 也可以作为广大计算机爱好者的参考读物。

图书在版编目(CIP)数据

Delphi 简明教程/郑洪编著. —北京: 海洋出版社, 2001.7

ISBN 7-5027-5010-X

I. D… II. 郑… III. Delphi 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 050863 号

<http://www.oceanpress.com.cn>

海洋出版社出版发行

(100081 北京市海淀区大慧寺路 8 号)

天津市蓟县宏图印务有限公司印刷 新华书店发行所经销

2001 年 7 月第 1 版 2001 年 7 月北京第 1 次印刷

开本: 787×1092 1/16 印张: 17.75

字数: 450 千字 印数: 1~8000 册

定价: 25.00 元

海洋版图书印、装错误可随时退换

丛书前言

计算机软件技术日新月异，编程软件种类繁多且各具特色。众多的计算机爱好者都希望能成为电脑编程的行家里手，如何才能向“卡乃基殿堂”迈出正确的第一步？

为满足各级 Windows 程序开发人员、大专院校相关专业的师生和业余编程爱好者学习和使用各种流行编程软件的需求，编者在搜集了不同层次读者意见的基础上，经过仔细探讨，精心策划了本套丛书，目的在于引导读者学习 21 世纪真正的编程工具！

本丛书名为“流行编程软件简明教程丛书”，主要包括：

《C++ Builder 简明教程》

《Delphi 简明教程》

《Visual C++ 简明教程》

《Visual Basic 简明教程》

《Visual FoxPro 简明教程》

《Quick BASIC 简明教程》

《C 语言简明教程》

本套丛书针对当前最流行的开发工具，通过浅显易懂、活泼生动的语言，采用逐步引导的方式将读者带入电脑编程的殿堂，使读者轻松掌握编程的基础知识，为进一步深造打下坚实基础。生动友好的例子帮助读者了解、掌握编程技巧，同时还大大激发了读者的兴趣。

本套丛书既不是面面俱到的“功能大全”，也不是单一查询函数的“用户手册”，而是独具特色的操作和编程指导书。丛书注重对软件的主要功能和新增特性进行介绍，结合实例和项目，讲解软件的主要功能、主要原理。

我们希望，本套丛书对于读者能“抛砖引玉、触类旁通”，指引读者踏上通往编程高手之路。

总的来说，本套丛书有以下几个最优越的特点：

- ◇ “精”——精选软件，精选作者，精选图书，是值得用户收藏的精品
- ◇ “准”——紧跟软件版本更新，出版时间准，图书出版市场定位准
- ◇ “简”——教程风格简单朴素，要让读者学习起来感觉简单实用
- ◇ “明”——丛书内容十分明晰，让读者读完之后能够真正明白

我们相信：

如果你是一位初学编程的读者，本套丛书将带领你入门！

如果你是一位对编程有些了解的业余爱好者，本套丛书为你指明前进的方向！

如果你是一位软件编程专业人员或计算机行家里手，本套丛书将为你的程序锦上添花！

编 者

前 言

Delphi是由软件界巨头、著名的 Inprise 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，同时它继承了 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。

经过 Inprise 公司的软件设计师们多年的改进，如今 Delphi 程序开发工具在各个方面都有了不同程度的提高，如数据库的体系结构、VCL 控件的开发、Web 应用程序和安全线程等，从而使得 Delphi 的功能更加的强大，使用起来也更加方便和灵活，提高了应用程序的开发效率。总之，Delphi 是开发 Windows 应用程序最为强大的工具之一，无论是 Windows 程序设计的初学者，还是有经验的 Windows 程序员，利用 Delphi 都可以迅速地开发出自己满意的应用程序。

为了满足广大读者的愿望，迅速地掌握 Windows 环境下的 Delphi 编程方法，编者以最新的 Delphi 5 为背景编写了本书，全书内容安排如下：

第一章介绍了利用 Delphi 编制应用程序的一些基础知识，以及如何利用 Delphi 编制简单的 Windows 应用程序；

第二章和第三章分别介绍了 Delphi 中对话框和菜单程序的设计；

第四章介绍了 Delphi 中常用的图形绘制和图像处理技巧；

第五章集中介绍了 Delphi 中各种窗体的开发技术，以及窗体的设计方案；

第六章介绍了 Delphi 中常用 VCL 控件（可视化控件）的开发应用程序，这是 Delphi 应用领域中非常重要的内容，也是读者应该必备的 Windows 编程本领；

第七章集中介绍了在 Delphi 中进行文件处理操作的方法；

第八章介绍了 Delphi 中非常吸引人数据库管理和开发方法；

最后一章（即第九章）介绍了多媒体程序开发的常用方法，在这一章中还设计了诸如声音播放器和视频播放器等实例程序，它将带领读者进入 Delphi 更高境界。

本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发，适合于各个层次的 Delphi 用户。对初学者来说，可迅速加入 Delphi 高级用户的行列；对有一定使用经验的读者，也可通过本书掌握 Delphi 深层次的开发方法，学会用更巧妙的办法开发出高水平的 Delphi 应用。然而，限于编者水平，书中的错误和不足之处在所难免，竭诚欢迎广大读者对本书提出批评和建议。

最后，书中示例中含有大量的程序代码、实例名称、图形图片、数据信息等，内容如有雷同，纯属巧合。

编 者

目 录

第一章 Delphi 编程快速入门	1
1.1 Delphi 基本概念介绍	1
1.1.1 Delphi 的基本形式	1
1.1.2 面向对象编程的概念	1
1.2 Delphi 新特性简介	2
1.2.1 Object Pascal 语言功能扩展	2
1.2.2 改善 IDE	3
1.2.3 增强 VCL 控件	3
1.2.4 加快复杂工程的开发速度	4
1.3 系统配置和编程环境	4
1.3.1 Delphi 系统配置	4
1.3.2 Delphi 编程环境	5
1.4 第一个示例	7
1.5 小结	11
习题	11
第二章 Windows 通用对话框	12
2.1 对话框函数简介	12
2.1.1 ShowMessage 函数	12
2.1.2 MessageDlg 函数	13
2.1.3 InputBox 函数	14
2.2 系统对话框调用技术	15
2.2.1 文件对话框	15
2.2.2 “颜色”对话框	18
2.2.3 “字体”对话框	19
2.2.4 “查找”对话框	21
2.3 标准对话框的使用	24
2.4 小结	32
习题	32
第三章 菜单程序的开发和设计	34
3.1 主菜单基本概述	35
3.1.1 创建菜单项	35

3.1.2	设置分隔条	37
3.1.3	设置快捷键	38
3.1.4	设置复选标志	39
3.1.5	菜单项有效状态设置	40
3.2	主菜单的应用	41
3.3	主菜单和弹出菜单综合应用	52
3.3.1	项目和设置	53
3.3.2	弹出菜单的设计	54
3.3.3	运行结果和参考程序	58
3.4	小结	62
习题		62
第四章	Delphi 中的图形绘制	63
4.1	基本图形的绘制	63
4.1.1	直线的绘制	63
4.1.2	椭圆的绘制	65
4.1.3	绘制多边形	67
4.1.4	绘制矩形	69
4.1.5	绘制三角曲线	71
4.2	复杂图形的设计	74
4.2.1	设计复杂图案	74
4.2.2	实时绘画程序	79
4.3	手工绘制图形	84
4.4	小结	105
习题		105
第五章	用户窗体的设计和使用	106
5.1	快速使用向导	106
5.2	多页窗体开发	109
5.3	复合窗体开发	117
5.3.1	绘制直线的子窗体设计	117
5.3.2	绘制椭圆的子窗体设计	119
5.3.3	绘制多边形的子窗体设计	121
5.3.4	主窗体的完整设计	123
5.4	多文档界面窗体的开发	127
5.5	小结	136
习题		137
第六章	常见 VCL 控件的使用	138
6.1	Edit 控件和 Button 控件	138

6.2	Listbox 控件和 Combobox 控件	148
6.3	Memo 控件和 Dialog 控件	151
6.4	Radiogroup 控件和 Checkbox 控件	157
6.5	DirectoryOutline 控件和 Animate 控件	166
6.6	小结	173
	习题	173
第七章	管理文件、目录和驱动器	175
7.1	各种控件简介	175
7.1.1	驱动器控件	175
7.1.2	目录控件	177
7.1.3	文件列表控件	177
7.1.4	文件过滤器控件	179
7.2	利用控件管理文件	181
7.3	管理目录	186
7.3.1	ChDir 函数	186
7.3.2	MkDir 函数	187
7.3.3	RmDir 函数	187
7.3.4	应用实例	187
7.4	管理驱动器	192
7.4.1	DiskSize 函数	192
7.4.2	DiskFree 函数	194
7.4.3	GetDir 函数	194
7.5	小结	195
	习题	195
第八章	数据库开发和管理	197
8.1	使用数据库向导	197
8.2	自己定制数据库	202
8.3	数据库管理程序	210
8.4	设计数据库报表	217
8.5	小结	219
	习题	219
第九章	多媒体的设计和使用	220
9.1	认识 MediaPlayer 控件	220
9.1.1	AutoEnable 属性	221
9.1.2	Display 属性	223
9.1.3	Filename 属性	225
9.1.4	DeviceType 属性	226

801	9.1.5 EnabledButtons 属性.....	226
121	9.2 音频播放程序	228
121	9.3 视频播放程序	235
100	9.4 小结	246
171	习题.....	247
171	附录 习题答案.....	248

第一章 Delphi 编程快速入门

Delphi 是全新的可视化编程环境，为我们提供了一种方便、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，采用了弹性可重复利用的完整的面向对象程序语言（Object-Oriented Language）和当今世界上最快的编辑器以及最为领先的数据库技术。

Delphi 使用面向对象的 Pascal（Orient Object Pascal 简称 OOP）作为基本开发语言。由于它继承了 OOP 语言中优越的全功能平台，弥补了可视化编程与全功能平台之间的障碍，以及编程效率高、代码质量好、面向对象等优点，解决了诸如 Visual Basic、Visual C++ 等可视化程序设计语言中的代码执行效率低下等问题。

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，您将会发现编程不再是枯燥无味的工作——“因为 Delphi 的每一个设计细节，都将带给您一份欣喜！”

1.1 Delphi 基本概念介绍

1.1.1 Delphi 的基本形式

Delphi 实际上是 Pascal 语言的一种版本，但它与传统的 Pascal 语言有天壤之别。一个 Delphi 程序首先是应用程序框架，而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西，仍可以严格地按照设计运行。您的工作只是在“骨架”中加入您的程序。缺省的应用程序是一个空白的窗体（Form），您可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以被放大缩小、移动、最大最小化等，但您却没有编写一程序。

因此，可以说应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。Delphi 已经为您做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，只是不处理任何事情。您所需要做的，只是在程序中加入完成您所需功能的代码而已。

在空白窗口的背后，应用程序的框架正在等待用户的输入。由于您并未告诉它接收到用户输入后作何反应，窗口除了响应 Windows 的基本操作（移动、缩放等）外，它只是接受用户的输入，然后再忽略。Delphi 把 Windows 编程的回调、句柄处理等繁复过程都放在一个不可见的 Romulam 覆盖物下面，这样您可以不为它们所困扰，轻松从容地对可视部件进行编程。

1.1.2 面向对象编程的概念

在此，请读者首先记住：面向对象的程序设计（Object-Oriented Programming，简记为

OOP) 是 Delphi 诞生的基础。

OOP 立意于创建软件重用代码, 具备更好地模拟现实世界环境的能力, 这使它被公认为是自上而下编程的优胜者。它通过给程序中加入扩展语句, 把函数“封装”进 Windows 编程所必需的“对象”中。面向对象的编程语言使得复杂的工作条理清晰、编写容易。说它是一场革命, 不是对对象本身而言, 而是对它们处理工作的能力而言。对象并不与传统程序设计和编程方法兼容, 只是部分面向对象反而会使情形更糟。除非整个开发环境都是面向对象的, 否则对象产生的好处还没有带来的麻烦多。而 Delphi 是完全面向对象的, 这就使得 Delphi 成为一种触手可及的促进软件重用的开发工具, 从而具有强大的吸引力。

一些早期的具有 OOP 性能的程序语言如 C++、Pascal、Smalltalk 等, 虽然具有面向对象的特征, 但不能轻松地画出可视化对象, 与用户交互能力较差, 程序员仍然要编写大量的代码。Delphi 的推出, 填补了这项空白。您不必自己建立对象, 只要在提供的程序框架中加入完成功能的代码, 其余的都交给 Delphi 去做。欲生成漂亮的界面和结构良好的程序丝毫不必绞尽脑汁, Delphi 将帮助您轻松地完成。它允许在一个具有真正 OOP 扩展的可视化编程环境中, 使用它的 Object Pascal 语言。这种革命性的组合, 使得可视化编程与面向对象的开发框架紧密地结合起来。

1.2 Delphi 新特性简介

到今天为止, Delphi 程序设计语言已经发展到了 5.0 版。和前期版本相比, Delphi 5 功能更加强大, 使用更加方便。例如, Delphi 5 的集成开发环境界面换成了 Windows95 控制元件, 并且对菜单的安排作了调整。可以建立的对象分类更为详细, 许多部件是 Delphi 4.X 版所没有的; Delphi 5 中 DBE 5.0 的查询引擎经过重新编写后, 不但功能增强了, 速度也比以前的版本快了很多; Delphi 5 提供了 OLE Automation 功能, 可以很方便地创建 OLE Automation Object; Delphi 5 版还提供了新的存储各种资源的工具即所谓的“Object Repository”(对象仓储), 用 Object Repository 可以存储所开发的表格及其他资源, 更能体现软件重用的设计思想。

下面, 我们就对最新版本的 Delphi 新特性作一比较系统的介绍:

1.2.1 Object Pascal 语言功能扩展

Delphi 5 扩展了 Object Pascal 语言的功能, 增加了它的灵活性, 具体包括以下几个方面:

(1) 动态数组

在以前 Object Pascal 语言中, 只能声明静态数组, 也就是在声明数组的同时必须指定数组的长度。如果不能确定数组的长度, 就必须把数组定义的尽量大, 然而这就造成了内存资源的浪费。而使用动态数组, 就可以事先不指明长度的情况下, 让系统随着使用的情况自动的为其分配内存, 提高了资源的使用效率。

(2) 方法重载

在以前的 Object Pascal 语言中, 一个方法对应唯一的一个方法, 然而 Delphi 5 提供了方法重载的机制, 即对象可以有一个或一个以上的同名方法。这些同名方法具有不同的参数类型和数目, 因此编译器可以根据方法的参数类型和数目来确定使用了哪个方法。

(3) 缺省参数

Delphi 的有些函数可能带有很多参数, 在使用这类参数之前需要查阅函数说明, 详细了解各个参数的使用方法和作用, 而其中有些参数并不常用, 往往是一些固定的值, 因此 Delphi 5 提供了缺省参数功能, 将这些不常用的参数设置为缺省参数, 一方面保持了程序的灵活性, 另一方面减少了程序员的工作量。

(4) 64 位整数

Delphi 5 提供了 Int64 类型, 它用于处理比 Longint 还要大的整数, 表示数值的范围是 $-2^{63} \sim 2^{63}-1$ 。

1.2.2 改善 IDE

Delphi 5 中的集成开发环境 (IDE) 有了很大改进, 能极大地提高开发效率, 它主要依靠简化读写和浏览代码的操作来提高开发效率。

代码编辑器让浏览本单元或相关单元的内容变得容易, 工程管理器中的拖放支持使得从已打开的工程中或资源管理器的对话框中选择文件并增加到工程的文件中变得更简单, 使用工程管理器中多个工程管理的功能让用户能同时编辑多个工程的文件, 应用 Delphi 5 的可视化窗口设计让用户能够轻松地模板中选择创建 Internet 程序、分布式计算及 Windows 程序等等。

与 Delphi 4 相比, Delphi 5 的 IDE 新特点主要包括:

- 桌面的用户化设置增强了用户对开发环境的控制;
- 编辑键映射功能可以让编辑器按照用户的习惯来工作;
- 工程浏览可以帮助用户更好地理解代码, 操作 VCL;
- 带有树视图和数据图表视图 (Data Diagram View) 的 DataModule 设计器可以帮助用户充分理解程序中的数据;
- 资源文件与工程管理器 and 编译器集成在一起;
- 动作列表保存了开发计划;
- 控制面板向导可以帮助用户定制应用程序的属性。

1.2.3 增强 VCL 控件

VCL (Visual Component Library) 是 Delphi 实现代码重用的基本工具。在 Delphi 5 中, 用户可以利用面向对象设计的强大功能开发出稳定、可靠、高效的程序, 可以利用现有的面向对象的控件创建自己的控件。

Delphi 5 企业版的 VCL 中一共包含有 200 多个控件, 利用这些控件, 用户可以奇迹般地加快应用程序从开始开发到推向市场的速度。无论是开发 Windows 程序还是 Internet 应用, Delphi 5 基于控件的开发模式都能大大降低开发任务的难度。

Delphi 5 的 VCL 的新特点包括:

- 提供用于创建和重用复合控件的帧架构 (Frames);
- 增加了能将 HTML 浏览功能集成到应用程序的 WebBrowser 控件;
- 增加了 Microsoft Office 自动化组件集, 以便把 Word、Excel 和 Outlook 等 Office 程序快速集成到应用程序中;

- 属性编辑器 (Property Editors) 支持自画 (Owner Draw) 功能, 简化了属性选择;
- 增强了对高级自画 (Advanced Custom Draw) 函数的支持, 以加强对 Windows API 的控制。

1.2.4 加快复杂工程的开发速度

Delphi 5 增加了“模块窗口”。模块窗口能够显示当前所有映射到应用程序地址空间的模块, 包括应用程序自身、应用程序显式或隐式调用的 DLL 以及操作系统调用的 DLL, 模块窗口可帮助您优化程序结构。此外, Delphi 5 还能自动记录在调试过程遇到的事件, 如断点、Windows 的消息以及其他调试信息。Delphi 5 能够监视指针错误, 如果某个指针试图非法访问内存的某个地址, 程序就会暂时中断运行, 由调试器接管控制权, 就好像遇到断点一样。

Delphi 5 进一步增强了调试功能, 即使是非常复杂的工程, 查找并修改错误都变得异常简单。Delphi 5 改进的调试器能帮助用户理解并控制自己编写的代码。用户利用断点可以在需要的地方深入代码进行调试, 而现在, 用户甚至可以在断点处设置触发断点后要采取的动作, 并把断点进行分组集中在一起, 以便快速调试自己感兴趣的代码段。

Delphi 5 的代码调试器包含以下一些新特点:

- 各种调试窗口和编辑器之间紧密集成, 拥有一致的用户界面, 支持剪贴板和拖动操作;
- 增加了断点提示条 (Breakpoint ToolTips), 以加快浏览和定位源代码;
- 新增的断点动作 (Breakpoint Actions) 功能加强了对调试进程的控制;
- 断点集合 (Breakpoint Groups) 方便了对多条断点的控制;
- 在 DCUs 和调试符号查找路径的帮助下, 调试工作变得更简单;
- FPU 调试窗口支持 MMX 指令, 从而增强了底层调试的能力;
- 进行多进程调试时, 增加了临时进程选项, 并支持调试子进程;
- 支持跨越进程边界的调试;
- 能够挂接到正在运行的进程上进行调用。
- Delphi 5 程序调试器支持远程调试、多线程调试, 并且可以观察 CPU 调试窗口。

上面提到的种种新特性引人入胜, 使得 Delphi 5 成为目前在 32 位 Windows 环境下最具有吸引力的开发工具。然而, Delphi 这个神圣殿堂到底会发出多么耀眼的光芒呢? 且让我们拭目以待。

1.3 系统配置和编程环境

1.3.1 Delphi 系统配置

值得注意的是, 要想在计算机上安装 Delphi 5, 必须在计算机上先安装相应的硬件和软件系统, 这些系统要求包括:

- Windows 98/2000、Windows NT 4 或更新版本 (例如 Windows NT 4.X);
- 80586 或更高微处理器;

- 一个 CD-ROM 驱动器;
- Microsoft Windows 支持的 VGA 或分辨率更高的监视器;
- 至少 32 MB 内存 (RAM), 最好使用 64MB 以上;
- 完全安装大约需要 180 兆以上的硬盘剩余空间。

1.3.2 Delphi 编程环境

Delphi 启动后, 在它的缺省画面 (软件开发界面) 中, 包括有主窗口、对象查看器、窗体窗口和代码窗口等, 如图 1-1 所示。

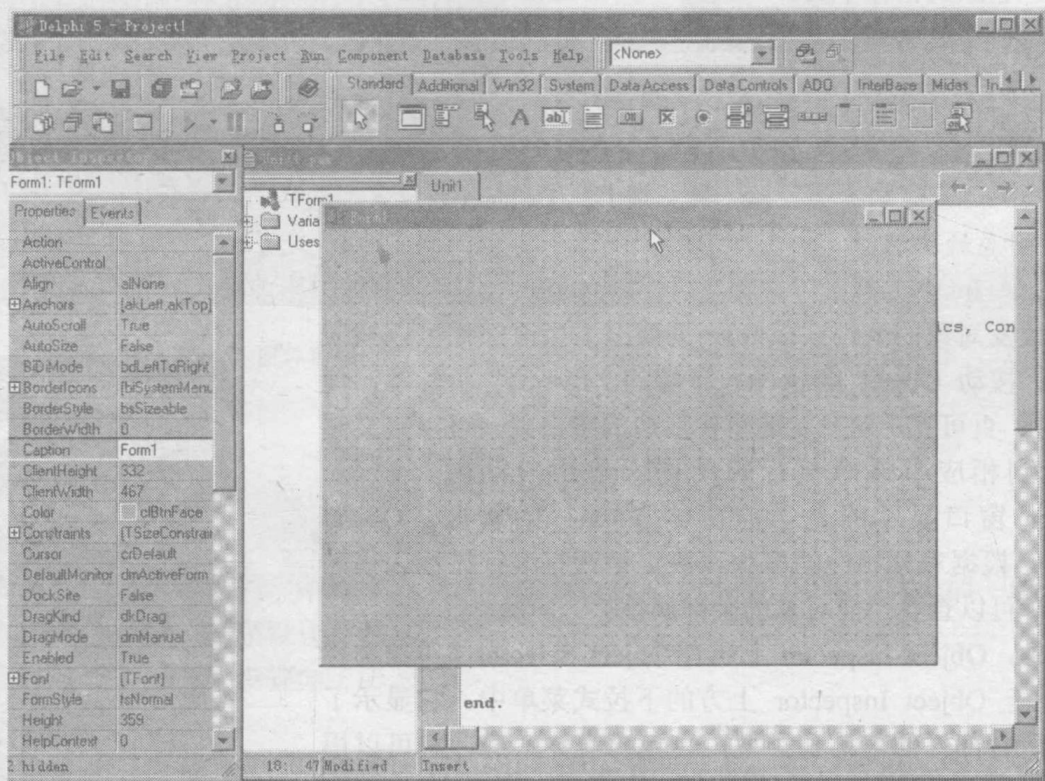


图 1-1 Delphi 开发界面

提示:

为避免隐藏在 Delphi 后的 Program Manager 和曾经运行过的其他程序扰乱版面, 分散您的注意力, 不妨在启动 Delphi 前关掉其他应用程序; 启动 Delphi 后, 再最小化隐藏在后面的 Delphi 5 程序组。这样屏幕上就只留下 Delphi 窗口可见了。

首次加载 Delphi, 屏幕上会出现四个窗口:

- 标题为“Delphi-Project1”的 Delphi 主窗口;
- Object Inspector 窗口;
- 标题为“Form1”的窗体 (Form) 窗口;
- 标题为“Unit1.PAS”的代码编辑窗口。刚启动时这一窗口的大部分被“Form1”窗体所掩盖。将“Form1”窗体移开, 或单击 Form1 窗体下方的状态行, 可以使其全

部可见。在“Form1”窗体的任意可见位置单击鼠标，可以恢复主窗体可见。

1. 主窗口

Delphi 的主窗口位于屏幕的上端，包括 Menu（菜单）、Speed Bar（加速条）和 Component Panel（部件选项板）：

- Menu 是下拉式主菜单。
- Speed Bar 位于主窗口的左下端，由两排共 14 个加速按钮组成。这些按钮是菜单功能的快捷方式，各种图标直观地表示了它能执行的动作。
- Component Panel 由一行、若干页对象按钮所组成，利用它来选择需要的部件并将它放到窗体中去。

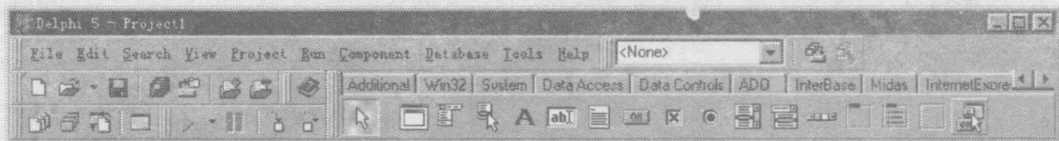


图 1-2 Delphi 主窗口

2. 对象检视器

Object Inspector 窗口含有两页：Properties 页显示窗体中当前被选择部件的属性信息，并允许改变对象的属性；Events 页列出了当前部件可以响应的事件。按动 Object Inspector 下端的“Events”页标签，使得 Events 页可见，这一定事件后边的空白处，可以定义对象接受到相应事件时执行的动作。首次启动时，Object Inspector 窗口显示的是当前窗体 Form1 的属性。Object Inspector 根据对象属性的多少，决定是否有滚动显示。移动滚动行条，可以查看当前对象的全部属性。

此外，Object Inspector 上还有 Object Selector（对象选择器），位于 Object Inspector 上方的下拉式菜单中。它显示了窗体上所有部件的名称和类型，也包含窗体本身。您可以用 Object Selector 很容易地在窗体的各个部件之间切换，也可以快速地回到窗体本身。当窗体中含有较多的对象时，您会发现这是切换对象尤其是回到窗体的最快捷途径。

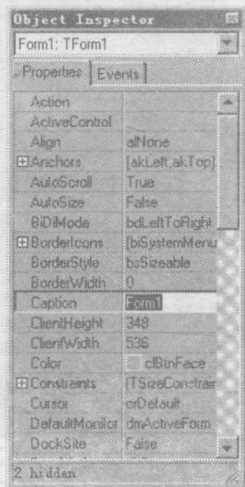


图 1-3 Delphi 对象检测器

提示：

想使 Object Inspector 一直可见，可将鼠标移到 Object Inspector 上，按动右键，以启动 Object Inspector 的弹出式菜单，将其设置为 Stay On Top。这对初学者常是一个很重要的设置方式。

3. 窗体窗口

Forms 窗口是开展大部分设计的工作区域。首次启动 Delphi 5 时显示的是窗体 Form1。可以把部件放在窗体中，通过移动位置、改变尺寸等操作随心所欲地安排它们，以此来开发应用程序的用户界面。您可以把窗体想象成一个可以放置其他部件的容器。窗体上有栅格（Grids），供放置部件时对齐位置用，在程序运行时 Grids 是不可见的。

一个真正的应用程序可能有不止一个窗口，您可以选用不同的窗体进行设计。其他窗体可以是对话框（Dialog Box）、数据录入框等。

4. 代码窗口

代码窗口一开始处于窗体窗口之下。因为在 Delphi 中,设计用户界面直接在窗体中进行,运行结果和设计样板完全一致。当部件被放到窗体上时,Delphi 会自动生成大部分的用户界面代码。您所应做的只是在它为您生成的框架中加入完成所需功能的程序段而已。点动 Form1 的状态行使代码窗口可见。

如图 1-5 所示,显示了空窗体 Form1 的代码窗口。

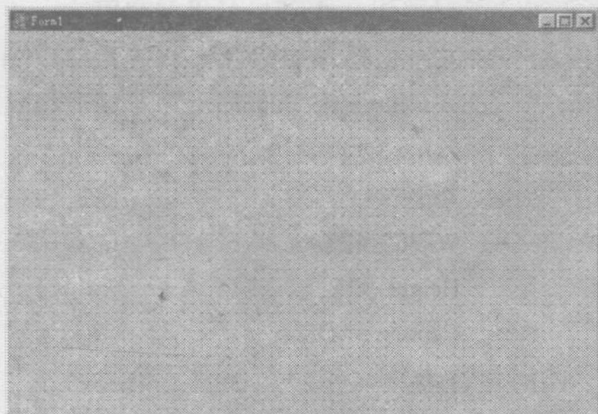


图 1-4 Delphi 窗体窗口

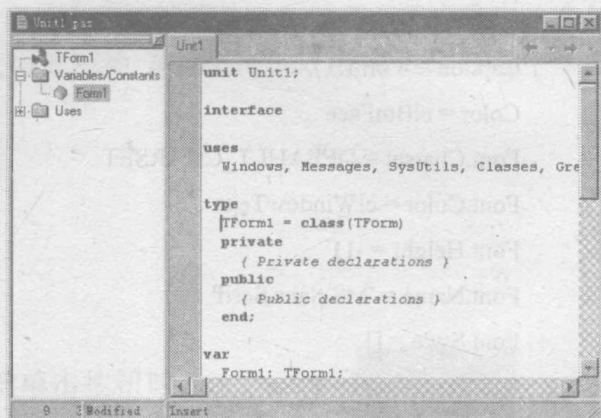


图 1-5 Delphi 代码窗口

1.4 第一个示例

下面我们用一个简单的示例程序,来说明在 Delphi 5 中进行应用程序开发的一般步骤和基本的方法,为以后的程序设计打基础。

编制示例程序的基本步骤如下所示:

1. 创建项目

首先启动一个新的项目,选择菜单 File 中的 New Application 项,在 Delphi 5 的集成开发环境中就会弹出一个新建的窗体,在控件工具栏上选择 Standard 选项后,在 Memo 控件的图标上双击鼠标的左键,这时空白的窗体上就会出现一个 Memo 控件,接着向窗体上添加两个 Button 控件,添加控件后的窗体如图 1-6 所示。

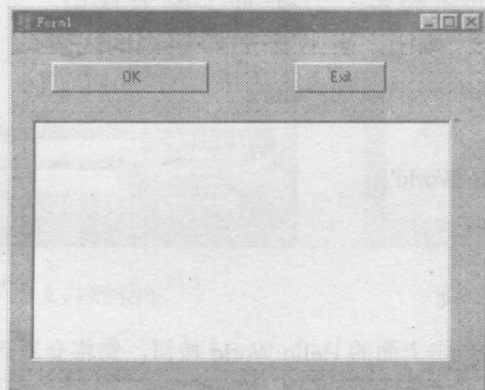


图 1-6 添加控件后的窗体

其中 Memo 控件用于显示文本，而 Button 控件用于显示一个对话框，窗体和控件的属性设置如下所示：

```

object Form1: TForm1
  Left = 225
  Top = 107
  Width = 396
  Height = 323
  Caption = 'Form1'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 13
  object Memo1: TMemo
    Left = 16
    Top = 72
    Width = 345
    Height = 201
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -24
  end
  object Button1: TButton
    Left = 32
    Top = 24
    Width = 129
    Height = 25
    Caption = 'OK'
    TabOrder = 1
    OnClick = Button1Click
  end
  object Button2: TButton
    Left = 232
    Top = 24
    Width = 75
    Height = 25
    Caption = 'Exit'
    TabOrder = 2
    OnClick = Button2Click
  end
end

```

2. 程序初始化

程序的初始化过程，实际上就是对窗体 FormCreate()事件的初始化，在程序设计阶段，用鼠标的左键双击窗体上的空白处，在屏幕上就会弹出一个代码窗口，把光标移动到 FormCreate()事件的过程处理代码中，并且添加如下所示代码：

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  form1.Button1.Caption:='Hello World';
  //设置按钮标题
  form1.Memo1.Clear;
  //清除文本框
  form1.Memo1.Lines.SetText('单击上面的 Hello World 按钮，您将会看到.....');
  //从文本文件中读取信息
end;

```