



Delphi 5

高级编程

熊 傲 等 编 著

Alt Gr



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



Delphi 5 高级编程

熊 傲 等 编 著

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 提 要

本书融合了利用 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 开发应用程序的方法和技巧。

全书共分 10 章,主要介绍了用 Delphi 5 编制应用程序的技术,如配置 Delphi 5 的集成开发环境,Delphi 5 中 ActiveX 控件的用途和使用方法,文本、图形和图像处理的常用方法,菜单、工具栏和对话框的设计,在 Delphi 5 中进行文件处理的方法,多媒体程序开发的常用方法,如何在 Delphi 5 中进行窗体程序设计,Delphi 5 中的网络与数据库开发方法以及 Delphi 5 中新增加的数据通信技术和多线程技术等。

本书资料丰富、内容详实,列举了大量的技术开发实例,实用性强,具有易学易用、简洁明了的特点。本书既适合于专业软件开发人员使用,也可以作为高等院校师生和科研人员进行软件开发的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Delphi 5 高级编程/熊傲等编著. —北京:清华大学出版社,2000.8
ISBN 7-302-03955-0

I .D... II .熊... III .DELPHI 语言程序设计 IV .TP312

中国版本图书馆 CIP 数据核字 (2000) 第 34498 号

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 世界知识印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 22.75 **字数:** 538 千字

版 次: 2000 年 8 月第 1 版 2000 年 8 月第 1 次印刷

书 号: ISBN 7-302-03955-0/TP·2315

印 数: 0001~8000

定 价: 33.00 元

前 言

Delphi 是 Inprise (原 Borland)公司推出的开发工具,它的基础语言是 Pascal 程序设计语言,因而它继承了 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 1.0 一推出,即迅速得到用户的好评。Inprise 公司以后又相继推出 Delphi 2.0、Delphi 3.0、Delphi 4.0 和 Delphi 5 版。

Delphi 5 版于 1999 年 9 月在中国大陆发行,它一方面保持了良好的向下兼容性,另一方面比以前的版本又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,迅速掌握 Windows 环境下的 Delphi 编程方法,我们结合自己的开发经验编写本书。本书各章的结构安排如下:

第 1 章介绍了利用 Delphi 5 编制应用程序的一些基础知识,以及如何利用 Delphi 5 编制简单的 Windows 应用程序。

第 2 章介绍了 Delphi 5 中基本控件的用途和使用方法,熟练利用 VCL 控件开发应用程序是每一个读者必备的本领。

第 3 章介绍了 Delphi 5 中进行文本、图形和图像处理的常用方法。

第 4 章通过几个示例程序讲解了 Delphi 5 中一些高级 VCL 控件的使用方法。

第 5 章集中介绍了 Delphi 5 中有关窗体和对话框的编程技术。

第 6 章集中介绍了在 Delphi 5 中进行文件处理操作的方法。

第 7 章介绍了 Delphi 5 中非常吸引人的网络与数据库开发方法。

第 8 章介绍了多媒体程序开发的常用方法,并设计了诸如 CD 播放器和动画播放器等实用程序。

第 9 章讲解了在 Delphi 5 中如何利用 Windows 的信息共享机制来实现程序的交互,并重点介绍了 4 种交互技术:剪贴板、OLE、DDE 和 DLL。

第 10 章介绍了 Delphi 5 中非常有用却常常被人忽略的技术——外设操作和多线程,它将带领读者进入 Delphi 更高的境界。

书中融合了利用 Delphi 5 开发 Windows 下应用程序的理论和实践,全面深入地介绍了利用 Delphi 5 开发应用程序的常用方法和技巧。在本书的编写过程中,我们尽量注意减少冗长无味的说明,代之以具体实用的实例演示,通过实例,引导读者把握 Delphi 的精髓所在。

本书注重开发实例、开发经验、开发技巧和 Windows 高级特性,适合于各个层次的

Delphi 用户。对一般读者来说,通过本书可迅速加入 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更高水平的 Delphi 应用程序。

除封面署名作者外,谢文赐、荣厚、方美玲、胡剑箫、李媛、边沿、黄海英、付清源、陈丽洋、王冰、王瑞忠、沈坚、王琳娜、王昕、王少波、贾继承、傅宁、胡佳禾、刘丽娜、高麟等也参与了本书的编写、修改、录排、校对等部分工作,在此对他们表示感谢。限于作者水平,书中的错误和不足之处在所难免,竭诚欢迎广大读者提出批评和建议。

作 者

2000 年 4 月

目 录

第 1 章 Delphi 5 编程基础	1
1.1 Delphi 5 简介	1
1.1.1 Delphi 5 特性介绍	1
1.1.2 系统配置说明	5
1.2 第一个示例	6
1.3 利用向导生成应用程序	10
1.4 小结	21
第 2 章 Delphi 基本控件的应用	22
2.1 列表框与标签控件	22
2.2 Memo 控件	27
2.3 按钮控件	34
2.4 单选按钮和复选框控件	43
2.5 滚动条控件	50
2.6 小结	56
第 3 章 文本、图形和图像处理	57
3.1 文本编辑	57
3.1.1 文本编辑程序	57
3.1.2 绘制字符图案	69
3.2 图形绘制	76
3.2.1 基本图形	76
3.2.2 绘制复杂图案	86
3.2.3 渐进图形的实现	92
3.3 图像处理	96
3.3.1 图像动态效果	96
3.3.2 图像动画效果	102
3.4 小结	106
第 4 章 高级 VCL 控件的使用	107
4.1 时间查询程序	107
4.2 文件查询程序	116
4.3 状态条显示程序	123
4.4 颜色调节程序	129

4.5	屏幕保护程序	133
4.6	小结	140
第 5 章	通用窗体的开发	141
5.1	多页面窗体开发	141
5.2	多窗体开发	149
5.3	单文档界面(SDI)开发	159
5.4	多文档界面(MDI)开发	165
5.5	小结	177
第 6 章	Delphi 5 中的文件操作	178
6.1	文件类控件的使用	178
6.2	文件类函数的调用	184
6.2.1	目录操作	184
6.2.2	驱动器信息显示	189
6.2.3	文件信息显示	190
6.3	对文本文件的处理	192
6.4	文本格式转换	198
6.5	小结	203
第 7 章	数据库开发与网络应用	205
7.1	数据库向导	205
7.2	设计数据库应用程序	210
7.3	复杂的数据库效果	216
7.4	设计数据库报表	224
7.5	简易的网络应用	227
7.6	小结	234
第 8 章	多媒体控件的应用	235
8.1	多媒体简介	235
8.2	声音文件播放器	236
8.3	MIDI 音乐播放器	244
8.4	动画播放器	252
8.5	CD 乐曲播放器	262
8.6	小结	269
第 9 章	数据交换程序的开发	270
9.1	剪贴板应用程序	270
9.2	动态数据交换程序	276
9.3	OLE 程序的开发	285
9.4	动态链接库(DLL)编程	300

9.4.1 利用 DLL 实现数据传输	301
9.4.2 利用 DLL 实现窗体重用	309
9.5 小结	320
第 10 章 Delphi 其他技术应用	321
10.1 鼠标操作	321
10.1.1 改变鼠标的指针	321
10.1.2 确定鼠标的位置	325
10.2 键盘操作	329
10.2.1 确定按键	329
10.2.2 确定换档键	333
10.3 线程技术	337
10.3.1 线程的概念	338
10.3.2 线程的应用	338
10.4 多线程应用程序	347
10.5 小结	355

第 1 章 Delphi 5 编程基础

随着编程概念的更新,可视化编程已经成为人们关注的焦点,由 Borland Pascal 发展而来的 Delphi 程序设计语言就是一种典型的可视化编程语言,它继承了 Borland Pascal 程序设计语言中的优越的全功能平台,弥补了可视化编程与全功能平台之间的障碍;而且 Delphi 继承了编程效率高、代码质量好、面向对象等优点,解决了诸如 Visual Basic 等可视化程序设计语言中的代码执行效率低等问题。

到今天为止,Delphi 程序设计语言已经发展到了 Delphi 5 版,无论是程序设计的初学者,还是大型应用程序的开发人员,利用 Delphi 5 都可以开发出满意的程序,下面就来介绍一下 Delphi 5。

1.1 Delphi 5 简介

1.1.1 Delphi 5 特性介绍

1999 年 9 月 14 日,美国 Inprise 公司(原 Borland 公司)在中国大陆发布了倍受瞩目的企业级开发工具——Borland Delphi 5,它包括 3 个主要的版本——企业版、专业版和标准版。它简化了 Windows 应用程序和浏览器、Web 服务器、中间件以及后端大型数据库系统的集成,并支持 XML 和 HTML 4,加快了 Internet 和分布式计算应用系统的设计开发速度。

Delphi 5 一经推出,就被全美权威杂志《InfoWorld》评为最佳 Web 及 Windows 应用程序开发工具。它在开发 Internet 应用、数据库系统等方面的性能有了很大提高,同时在提高开发人员的效率、方便开发人员的使用方面也做了很多改进。

和前期版本 Delphi 4 相比,Delphi 5 在以下方面的功能得到了进一步的加强。

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

在以前的 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,就必须把数组定义得尽量大,但这会造成内存资源的浪费。而使用动态数组,就可以在事先不指明长度的情况下,让系统随着使用的情况自动地为其分配内存,提高了资源的使用效率。

(2) 方法重载

在以前的 Object Pascal 语言中,一个方法名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) 默认参数

Delphi 的有些函数可能带有很多参数,在使用这些函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 提供了默认参数功能,将这些不常用的参数设置为默认参数,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,Int64 类型表示数值的范围是 $-2^{63} \sim 2^{63} - 1$ 。

2. ActiveX 增强功能

Delphi 5 增加了许多新的 ActiveX 功能,比如:自动化向导可以生成支持各种自动化服务器对象事件的代码,数据绑定 ActiveX 控件可以和 VCL 数据集通信,新的 COM(资源对象模块)对象向导提供了创建简单的 COM 对象的功能,类库编辑器支持 DLL(动态链接库)和 OP(面向对象的 Pascal)。

3. 对 XML 的支持

Delphi 5 提供了对 XML 的支持,从而简化了数据分布,优化了数据交换。用户可以方便地创建高效率的 Internet 程序,来把数据迅速发布到 Internet 上。新版 Delphi 对 XML 的支持使得开发人员能够快速建立起具有可移植性和扩展性的系统,以迎接 Internet 正面临的又一次冲击波——电子商务。

4. 对 HTML 4 的支持

Delphi 5 支持 HTML 4,使用户能在 Web 应用中创建各种类型的动态“瘦”客户端程序。把 HTML 4 和 XML 结合起来使用,从而创建动态的极瘦客户端程序,可满足 Internet 上各种任务的需求。

5. ADO

ADO(ActiveX Data Objects)是 Microsoft 提供的一项技术。通过 ADO,我们可以方便地访问各种类型的数据库,特别是 OLEDB 数据库。ADO 已成为访问数据库的新的标准接口。Delphi 5 增添了对 ADO 的支持,是为了让用户能迅速实现对数据的一致性访问。结合 Delphi 本身的开放式数据元件结构,用户可以很快地建立应用程序,用来把自己的商业数据通过 Internet 发送给客户、最终用户以及整个销售环节。通过 ADO,Delphi 5 能让用户快速访问关系型或非关系型数据库以及 E-mail 和文件系统。

6. InterBase Express

利用 InterBase Express,系统集成商和独立软件商可以快速开发高性能的应用系统。把 Delphi 5 和 InterBase Express 结合起来使用,便可以在只需少量维护工作且只占用少量系统资源的数据库上进行高效率、高性能的开发。有了 InterBase Express 后,用户不需要使用 BDE(Borland Database Engine)便可以创建一个优秀的嵌入式系统。

7. TeamSource

Delphi 5 集成了一个与 Microsoft Visual SourceSafe 功能相似的源代码管理工具—TeamSource,这样,工作组成员之间会合作得更好,工作更有效率。TeamSource 建立在现有的源代码管理工具和版本引擎的基础上,并把它们扩展成一种工作流程的模式,从而简化了工作组开发的源代码管理工作。

8. Borland Translation Suite

这是 Delphi 5 包含的一个国际化和本地化工具,它能帮助用户同时开发多语言版本,管理和维护各种资源的翻译代码,从而加速产品的国际化开发步伐。现在利用 Delphi 5 开发的系统可以在世界范围内应用,以帮助用户开拓全新的世界范围内的市场。

9. MIDAS 3

利用 Delphi 5 中的 MIDAS 3(Multi-tier Distributed Application Services Suite),用户可以创建高效灵活的 Delphi 程序,以满足日益增长的事务量和用户数的需要。MIDAS 3 支持所有的分布式计算标准,如 CORBA、COM 和 MTS 等,可以实现现有系统和电子商务程序之间的无缝集成。它的特点表现为:没有界限的数据请求在移动或窄带宽的环境中可实现更多控制;WebConnection 组件能够在保证安全性的前提下帮助用户把程序移到防火墙外;服务对象库增强了扩展性。

10. 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选择文件并增加到工程中变得更简单,使用工程管理器中多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 from 模板中选择创建 Internet 程序、分布式计算及 Windows 应用程序等。

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 Data Module 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。

11. 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,并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。

12. 加快复杂工程的开发速度

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非常简单。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。用户不仅可以利用断点在需要的地方深入代码进行调试,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试；

- Delphi 5 的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。

上面的介绍仅仅提及了 Delphi 5 的一些主要的新特点,它还有更多的改进有待于我们去挖掘。可以说,Delphi 5 是目前创建 Web 和 Windows 应用程序最快速、最有效的开发工具之一。这颗新星到底会发出多么耀眼的光芒呢?且让我们拭目以待。

1.1.2 系统配置说明

Delphi 5 的安装对用户的计算机性能有更高的要求,包括硬件和软件系统。这些系统要求包括:

- Windows 95/98、Windows NT 4 或更新版本(例如 Windows NT 2000);
- 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 16 MB 内存(RAM),最好使用 32MB 以上;
- 完全安装大约需要 180MB 以上的硬盘空间。

Delphi 5 启动后,它的默认画面(软件开发界面,如图 1-1 所示)包括主窗口、对象查看器、窗体窗口和代码窗口等:

- 主窗口包括标题栏、菜单栏、工具栏和控件栏;
- 对象查看器中包括两个选项卡——Properties 和 Events,其中 Properties 选项卡用于显示选中对象的属性信息,而 Events 选项卡用于显示可以接收的事件列表;
- 用户可以在窗体窗口中对应用程序的界面进行设计,如添加控件等;
- 在程序的设计阶段,当用户在窗体上单击时,就会自动激活程序的代码窗口,用户可以在其中添加程序设计代码。

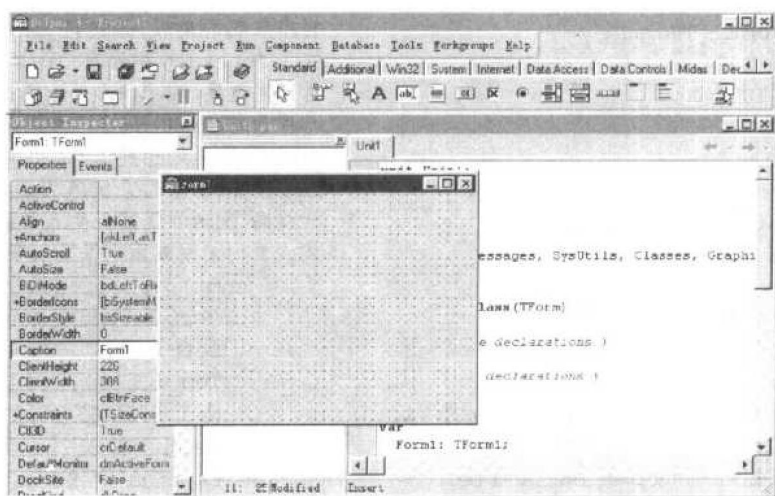


图 1-1 IDE 界面

1.2 第一个示例

下面就以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一个基础,下面介绍编制示例程序的基本步骤。

1. 开始工作

首先启动一个新的项目,选择菜单 File 中的 New Application 选项,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 选项卡后,在 Memo 控件的图标上双击,这时空白的窗体上就会出现一个 Memo 控件,接着向窗体上添加一个 Button 控件,添加控件后的窗体如图 1-2 所示。

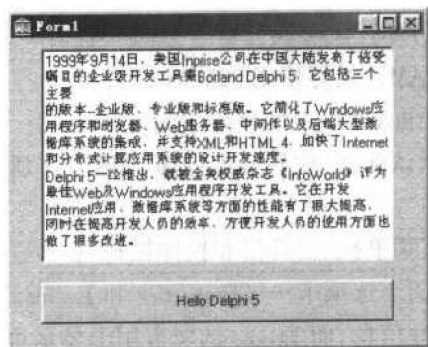


图 1-2 添加控件后的窗体

其中 Memo 控件用于显示文本,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 = 192
  Top = 107
  Width = 328
  Height = 235
  Caption = 'Form1'
  OnCreate = FormCreate
  object Memo1: TMemo
    Left = 16
    Top = 8
    Width = 289
    Height = 153
    Lines.Strings = ('1999 年 9 月 14
    日,美国 Inprise 公司在大陆发布了倍
    受瞩目的企业级开发工具 Borland Delphi
    5,它包括 3 个主要的版本—企业版、专业版
    和标准版。它简化了 Windows 应用程序和
    浏览器、Web 服务器、中间件以及后端大型数
```

据库系统的集成,并支持 XML 和 HTML 4,加快了 Internet 和分布式计算应用系统的设计开发速度。Delphi 5 一经推出,就被全美权威杂志《InfoWorld》评为最佳 Web 及 Windows 应用程序开发工具。它在开发 Internet 应用、数据库系统等方面的性能有了很大提高,同时在提高开发人员的效率、方便开发人员的使用方面也做了很多改进。')

```
end
  object Button1: TButton
    Left = 16
    Top = 176
    Width = 289
    Height = 25
    Caption = 'Hello Delphi 5'
    OnClick = Button1Click
  end
end
```

注意 为节省篇幅,个别地方代码分左右两栏排列,阅读时先看左边栏,再接右边栏。

2. 程序的初始化

程序的初始化过程,实际上就是对窗体 FormCreate 事件的初始化过程。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:= 'Click here';
  //设置按钮标题
  form1.Memo1.Clear;
  //清除文本框
  form1.Memo1.Lines.LoadFromFile('c:\windows\win.ini');
  //从文本文件中读取信息
end;
```

在程序运行的初期,首先执行窗体 FormCreate 事件中的代码,即通过语句 form1.Button1.Caption:= 'Click here' 来设置按钮控件的标题显示为字符串 'Click here', form1.Memo1.Clear 语句用来清空文本框中的显示内容,最后通过一条语句 form1.Memo1.Lines.LoadFromFile('c:\windows\win.ini') 从文本文件 c:\windows\win.ini 中读取文本信息,并且显示在文本控件中,其中 c:\windows\win.ini 中的文本内容如图 1-3 所示。

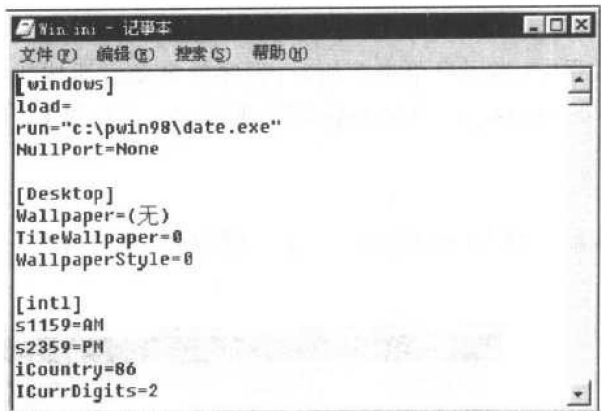


图 1-3 c:\windows\win.ini 文件内容

3. 添加代码

在程序的设计过程中,双击窗体上的按钮控件,在屏幕上就会弹出一个代码窗口。把光标移动到代码窗口的 Button1Click 事件处理过程中,并且添加如下所示的按钮响应代码:

```
unit Hello;
interface
```

```
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TForm1 = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
  {$R *.DFM}

  procedure TForm1.Button1Click(Sender: TObject);
  begin
    showmessage('Hello Delphi 5');
    { 显示一个对话框 }
  end;
end.
```

在程序运行的过程中,当用户单击按钮时,程序就自动激活按钮控件的 Button1Click 事件,然后通过语句 showmessage('Hello Delphi 5')来显示一个如图 1-4 所示的对话框。

4. 运行结果

做完以上的工作后,选择菜单 File 中的 Save All 选项,就会弹出一个如图 1-5 所示的保存文件对话框。

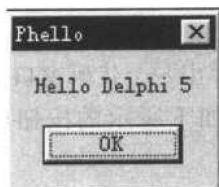


图 1-4 显示对话框信息

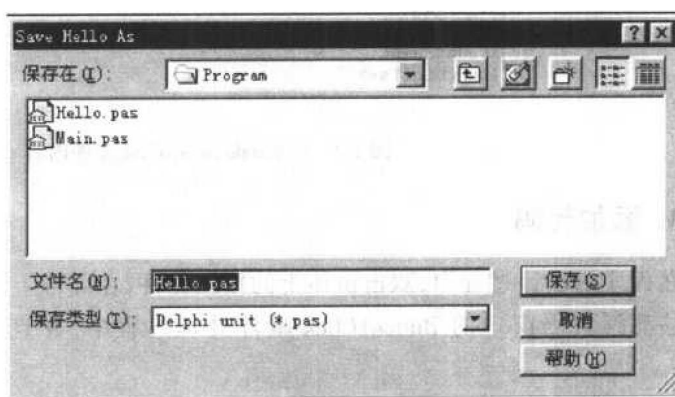


图 1-5 保存文件对话框

在弹出的对话框中选择合适的文件名保存文件。

保存程序文件后,按键盘上的功能键 F9 运行程序,在程序运行的初始画面中,文本控件显示文件 c:\windows\win.ini 中的文本信息,单击 Click here 按钮,在窗体上就会弹出一个对话框,在其中显示一条字符串信息,程序运行的结果如图 1-6 所示。

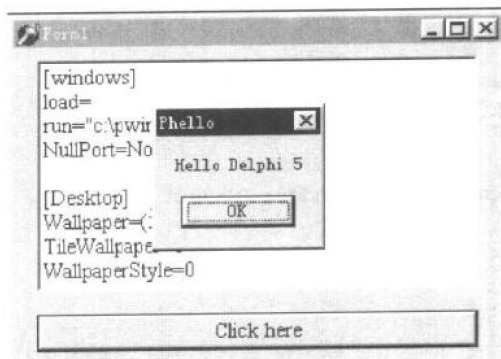


图 1-6 程序运行结果

程序完整源代码如下所示:

程序清单 1-1 Hello Delphi 5

```
unit Hello;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Memo1Change(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
  showmessage('Hello Delphi 5');
  {显示一个对话框}
```