



北京希望电子出版社
Beijing Hope Electronic Press
www.bhpe.com.cn

21 世纪计算机编程指南系列



本书配套光盘内容包括：

1. 本书电子图书
2. 书中相关程序源代码
3. 送“Windows NT 4.0速成”多媒体教学软件



北京希望电脑公司 总策划
网胜工作室 编 著

Delphi 5

从入门到精通



北京希望电子出版社
Beijing Hope Electronic Press
www.bhpe.com.cn

21 世纪计算机编程指南系列



本书配套光盘内容包括：

1. 本书电子图书
2. 书中相关程序源代码
3. 送“Windows NT 4.0速成”多媒体教学软件



北京希望电脑公司 总策划
网胜工作室 编 著

Delpni 5

从入门到精通

内 容 简 介

本书是为从事 Delphi 开发和应用的开发人员而编写的。Delphi 作为一种以 Pascal 为基础的快速应用开发工具 (RAD), 其容易使用的特性和强劲的开发能力征服了许多程序员。本书是 Delphi 资深开发人员根据多年开发经验依据最新版本 5.0 编写。全书由 21 章和一个附录构成, 内容涵盖 Delphi 程序快速开发入门, 窗体介绍, 按钮、成组组件与列表, 文本控件, 菜单的构造与使用, 键盘与鼠标, 工具栏、Coolbar 与状态面板, MDI 程序与页面组件, 输入、输出与打印, 图形处理, 多媒体与动画, 系统对象和常用函数, Delphi 程序调试与异常处理, 数据库编程, 图表与报表, 构造定制组件, COM 对象和 ActiveX 控件, Internet 编程, 线程和 COBAR 编程。附录为 Pascal Object 语言基础, 为查询相关语法和结构类型提供了方便。

本书注重理论与实际的结合, 原理讲解深入浅出, 程序实例步骤详细, 代码剖析精辟深刻, 不但是所有 Delphi 新老程序员重要的参考书, 同时也是高校相关专业师生自学或教学的参考书和社会相关领域培训的推荐教材。

本书配套光盘包括本书的电子图书, 书中相关程序源代码和“Windows NT 4.0 速成”多媒体教学软件。

系 列 书 : 21 世纪计算机编程指南系列

总 策 划 : 北京希望电脑公司

书 名 : Delphi 5.0 从入门到精通

文 本 著 者 : 网胜工作室

责任编辑/审校 : 苏 静

C D 制 作 者 : 希望多媒体制作中心

C D 测 试 者 : 希望多媒体测试部

出版/发 行 者 : 北京希望电子出版社

地 址 : 北京海淀路 82 号 (100080)

网址: www.bhp.com.cn

E-mail: lwm@hope.com.cn

电话: 010-62562329, 62541992, 62637101, 62637102 (图书发行, 技术支持)

010-62633308, 62633309 (多媒体发行, 技术支持)

010-62613322-215 (门市) 010-62531267 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 北京希望图书照排中心

C D 生 产 者 : 文录激光科技有限公司

文 本 印 刷 者 : 北京市媛明印刷厂

规 格 / 开 本 : 787 毫米×1092 毫米 16 开本 25 印张 580 千字

版 次 / 印 次 : 2000 年 1 月第 1 版 2000 年 4 月第 2 次印刷

印 数 : 5000 → 10000 册

本 版 号 : ISBN 7-900031-19-7/TP·19

定 价 : 39.00 元 (1CD, 含配套书)

说明: 凡我社光盘配套图书若有缺页、倒页、脱页、自然破损, 本社发行部负责调换。

前 言

Delphi 是一个理想的 Windows 编程工具,适用于初学者和专业程序员。使用它可以在很短的时间内创建自包含式的、用户界面友好的 Windows 应用程序。而且所创建程序的窗口是在屏幕上“绘制”出来的,这样始终可以看到程序最终的执行效果。完成这种设计工作不需要编写任何代码,也不需要以前的 Windows 编程语言所需要的传统的调试和错误处理。

设计完用户界面后,当然还有很多的工作要做。由于 Delphi 编程语言不仅功能强大而且容易掌握,所以即使是很复杂的任务也能以很快的速度来完成。自从 1993 年 Delphi 1.0 推出以来,相继又出现了 Delphi 2.0、Delphi 3.0、Delphi 4.0。1999 年 9 月,Inprise 公司推了 Delphi 的最新版本 5.0。各个版本虽然保持了向后兼容,但随着版本的提高,在许多的方面都有了新的改进。Delphi 5 在界面上和 Delphi 4 的界面虽然似曾相识,但它在开发 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。

本书的一个主要特点是内容新颖,那些由浅入深的阐述、抛砖引玉的例子使本书既通俗易懂又信息量丰富。考虑到国内的实际情况,在本书的编写过程中,克服了 Delphi 中译本常有的语言罗嗦通病。针对国内同类书籍中基本不涉及过深内容的特点,添加了有关“Com 对象”、“分布式多层结构编程”等内容,特别是 Delphi 5 中新出现的 InternetExpress 编程。

本书在内容安排上遵循由浅入深的原则,适用于 Delphi 初学者和中级用户。对于 Delphi 初学者,建议按部就班地顺序阅读。对于中级用户,如果对 Delphi 的界面设计比较熟悉了,可以直接从后面的专题章节开始阅读。如果是一位高级用户,笔者相信在本书的最后几章你可以发现所需要的知识,因为它们是为中高级用户准备的。

本书由李真文组织编写,张奇支编写了全书的主要部分。另外,参与本书编著的还有吴海薇、张豫、金春林、苏效乐、张津、李海军等同志。由于时间仓促,加之作者水平有限,错误之处在所难免,恳请广大读者批评指正。

目 录

第 1 章 Delphi 程序快速开发入门.....1	
1.1 编写第一个程序.....1	
1.2 Delphi 程序的组成.....5	
1.3 编写第二个程序.....10	
1.4 IDE 简介.....12	
1.5 Delphi 中的命名习惯.....17	
第 2 章 窗体介绍.....19	
2.1 窗体概貌.....19	
2.2 窗体模板.....23	
2.3 动态建立窗体和 Splash 屏幕.....27	
第 3 章 按钮、成组组件与列表.....32	
3.1 基本按钮.....32	
3.2 按钮组.....36	
3.3 位图按钮.....41	
3.4 其它按钮.....51	
第 4 章 文本控件.....52	
4.1 标签.....52	
4.2 静态文本.....54	
4.3 文本编辑框.....56	
4.4 掩码编辑框.....60	
4.5 Memo 控件.....62	
4.6 列表框.....63	
4.7 组合框.....64	
4.8 几个常用文本处理过程和函数.....66	
第 5 章 菜单的构造与使用.....71	
5.1 主菜单和弹出菜单.....71	
5.2 使用菜单设计器.....72	
5.3 动态菜单设计.....76	
5.4 运行时菜单管理.....81	
5.5 NotePad 程序的菜单事件.....84	
第 6 章 键盘与鼠标.....89	

6.1 键盘.....89	
6.2 鼠标.....97	
第 7 章 工具栏、Coolbar 与状态面板.....105	
7.1 工具栏.....105	
7.2 Coolbar 组件.....114	
7.3 状态面板.....116	
7.4 ControlBar 组件.....123	
第 8 章 MDI 程序与页面组件.....125	
8.1 MDI 程序开发.....125	
8.2 页面组件.....137	
第 9 章 输入、输出与打印.....143	
9.1 文件输入与输出.....143	
9.2 打印.....155	
第 10 章 图形处理.....165	
10.1 TCanvas 类.....165	
10.2 绘图与着色.....167	
10.3 图像组件.....174	
10.4 图片对话框.....185	
第 11 章 多媒体与动画.....189	
11.1 Animate 组件.....189	
11.2 MediaPlayer 组件.....195	
第 12 章 系统对象和常用函数.....202	
12.1 剪贴板.....202	
12.2 Screen 对象.....207	
12.3 Application 对象.....208	
12.4 Delphi 中的常用函数.....210	
第 13 章 Delphi 程序调试与异常处理.....218	
13.1 为什么使用异常.....218	
13.2 Delphi 异常处理机制.....218	
13.3 异常响应.....219	



13.4 异常保护.....	226	19.1 进程与线程介绍.....	328
13.5 用异常响应编程.....	228	19.2 用 Windows 函数编写线程.....	330
13.6 程序调试简介.....	230	19.3 使用 Delphi 的 TThread 类.....	333
第 14 章 数据库编程.....	239	19.4 线程的优先级.....	336
14.1 学习使用 Form Wizard.....	239	19.5 线程同步技术.....	339
14.2 如何访问数据库表.....	244	第 20 章 COBAR 编程.....	347
14.3 利用代码访问数据库.....	249	20.1 COBAR 简介.....	347
14.4 使用结构化查询语言.....	258	20.2 第一个 COBAR 程序.....	348
14.5 主从数据库.....	261	20.3 CORBA 数据库编程.....	355
第 15 章 图表与报表.....	264	第 21 章 多层结构数据库程序.....	360
15.1 利用 QReport 创建报表.....	264	21.1 多层结构简介.....	360
15.2 利用 TeeChar 创建图表.....	272	21.2 MIDAS 技术简介.....	361
第 16 章 构造定制组件.....	277	21.3 创建 MIDADS Web 应用 程序.....	364
16.1 组件包简述.....	277	21.4 编程示例.....	371
16.2 创建第一个组件.....	278	附录 I Object Pascal 语言简介.....	382
16.3 安装包和组件.....	282	I.1 程序结构.....	382
16.4 非可视化对话框组件.....	285	I.2 常量和变量.....	382
第 17 章 COM 对象和 ActiveX 控件.....	289	I.3 数据类型.....	384
17.1 COM 对象.....	289	I.4 运算符.....	386
17.2 ActiveX 控件.....	299	I.5 复合语句和赋值语句.....	387
第 18 章 Internet 编程.....	307	I.6 流程控制语句.....	387
18.1 TCP/IP 编程.....	307	I.7 循环语句.....	388
18.2 服务器 Web 编程.....	315	I.8 过程和函数.....	389
第 19 章 线程.....	328	I.9 单元.....	390

第 1 章 Delphi 程序快速开发入门

Delphi 是一种 RAD (快速应用程序开发) 编程语言, Delphi 编程是一种可视化的编程。我们先以一个简单的“Hello,World!”程序作为引子,介绍怎样在 Delphi 环境下进行快速程序开发。然后再简单介绍 Delphi 开发环境的一些相关知识,使读者对 Delphi 编程有一个总体认识。

本章的主要内容包括:

- 创建第一个“Hello,World!”程序
- Delphi 程序的组成结构
- 创建电子钟程序
- IDE 编程环境介绍
- 代码编辑器中的快捷键操作
- Delphi 程序命名习惯

1.1 编写第一个程序

我们假设读者的计算机上已经安装了 Delphi 5.0。如果读者的计算机上还没有安装它,则请参照 Delphi 5.0 的安装提示安装,具体安装过程本书不作介绍。

1.1.1 启动 Delphi

读者如果还没有启动 Delphi,请马上启动它。方法是在 Windows 任务栏上的“开始”按钮上单击鼠标左键,选择“程序”项,找到 Borland Delphi 5,再单击 Delphi 5 (约定:以后所有单击均指鼠标左键的单击动作)。如果一切顺利,屏幕的画面如图 1.1 所示。

Delphi 的编程环境分为三个部分,最上面的是菜单栏、加速栏和 VCL (可视化组件) 面板。左下部的窗口通常叫对象观察器 (Object Inspector),右下部的窗口叫做设计窗体或设计面板,在本章的第三节,我们将详细介绍它们。

没有使用过 Delphi 的读者也许不知道,Delphi 在启动后,已经生成了一个最简单的程序。按 F9 键或用单击 Run 菜单下的 Run 菜单项,可在 Delphi 的编程环境下运行这个程序。程序运行时,Windows 桌面上出现一个空白窗体。窗体中什么也没有,它的边框上却有 Windows 中一般窗口所共有的最大化、最小化和关闭按钮。用鼠标单击它们,可以实现窗体的最大化、最小化和关闭操作。窗体左上角还有一个系统菜单,菜单中包含了一些常见的菜单项。此外,当我们使鼠标靠近窗体的边框时,还可以通常拖放调整窗体的大小。窗体的样子如图 1.2 所示。

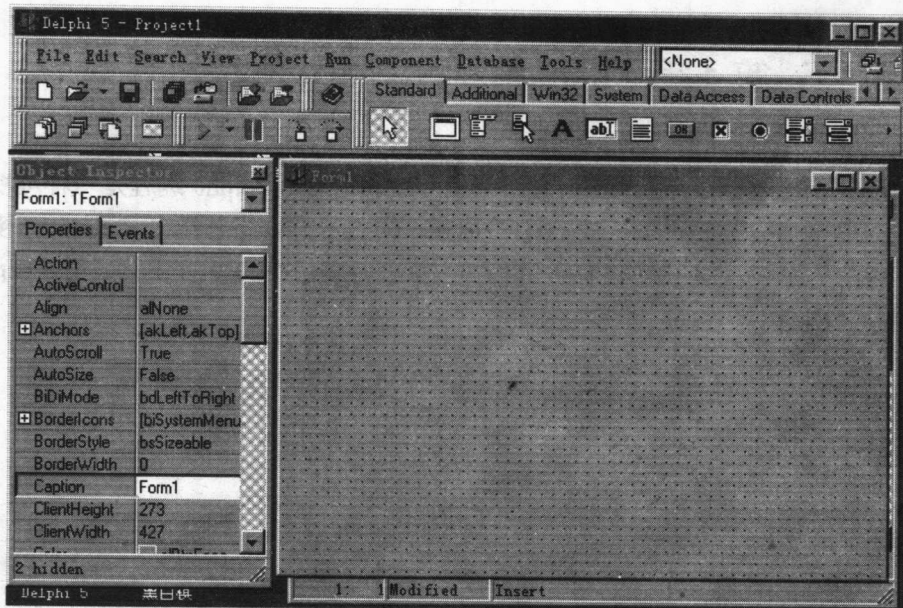


图 1.1 Delphi 5.0 的启动界面

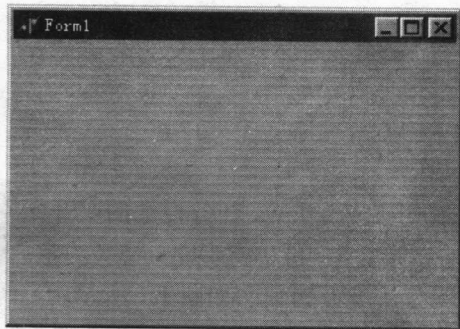


图 1.2 第一个 Delphi 程序

这个程序当然不是我们所希望的程序，因为它不能为我们作任何其它的事件。我们还必须对它做进一步的改进。单击窗体的关闭按钮（或系统菜单下的 Close 菜单项）关闭该程序，重新回到 Delphi 的编程环境。我们希望该程序在运行时，窗体上有一个按钮，当用户单击该按钮时程序会显示一则“Hello,World!”的消息框。为此我们还必须进行以下的编程工作。

1.1.2 给窗体添加按钮

单击控制面板中 Standard 页上的 Button 组件，然后在设计窗体上的合适位置单击鼠标，这时一个按钮出现在窗体上。

还有几个细致的工作必须做。先在按钮外的窗体部分用鼠标点一下，然后查看窗体左边的对象观察器。它有两个页面，一个是属性页，另一个是事件页。属性页上有关于设计窗体（下拉框显示内容为“Form1”）的很多属性，这些属性按字母顺序排列。在对象观察器中找到窗体的 Caption 属性，将默认值“Form1”改为“主窗体”。窗体的 Caption 属性决定着窗体的标题。然后找到窗体的 Name 属性，将它设置为 MainForm。在 Delphi 中，所有的窗体或组件都有一个 Name 属性，它是它们唯一的标识符，在程序中我们就可

通过它来调用窗体或组件的属性、方法的。由此可知,不同的窗体或组件应该有不同的 Name 属性值。再找到窗体的 Height 属性和 Width 属性,分别将它们改为 150 和 300。这两个属性控制窗体的高度和宽度。通常所有属性的含义都可以通过它的英文意思来得知。有时,窗体上的按钮有可能会“不见了”,但出现了一个滚动条。拖动滚动条,即可找到按钮。

在按钮上单击鼠标,选中按钮。如果按钮被选中,它的周围会出现几个小黑点。然后将它拖到窗体的适当部位,使滚动条消失。这时对象观察器变为按钮的属性设置。将它的 Caption 属性改为“显示(&S)”, Height 和 Width 属性可以不修改。符号&的作用是使 S 成为按钮的加速键,使得可用 ALT+S 来选取该按钮。然后将 Name 属性改为“ButtonShow”。这样处理之后,设计窗体应该是如图 1.3 所示的样子。

要使单击了按钮后窗体立即显示一则消息框,还必须做一些编程工作(至此我们还没有写入一行代码)。双击按钮,光标切换到 Delphi 的代码编辑窗口中,在光标出现的位置输入以下代码:

```
showmessage('Hello,World!');
```

这一步的实质是在编写 ButtonShow 按钮

的 OnClick 事件的实现代码。在 Delphi 中,窗体和组件都有一些对应的事件,事件的名称以 On 为前缀。按钮的 OnClick 事件为:当用户单击或通过加速键选择它时,OnClick 事件被触发,Delphi 调用用户在 OnClick 事件中编写的程序代码。

OK! 我们的程序完成了,不要急于运行,先将程序存盘!

1.1.3 保存文件

单击控制面板加速栏中的 Save All 图标或 File 菜单下的 Save All 菜单项。由此引出 Save Until As 对话框,如图 1.4 所示:

首先保存的是 Delphi 的单元文件,在“文件名”文本框中输入“Main”,如果必要,可以在文件名前面加上路径名,单击“保存”按钮即可保存。接下来的是保存 Delphi 的项目文件,在“文件名”文本框中输入“HelloWorld”,再单击“保存”按钮。注意,Delphi 的单元文件名不能和窗体的 Name 属性值相同,也不能和项目文件同名。

作为一名程序开发者,时刻记得存盘是一件很重要的事,因为我们无法保证计算机永远会按我们的思维去正常工作。一旦出现诸如死机之类的问题,我们的很多工作就会白白损失。而经常存盘是减少损失的一个最好的办法。

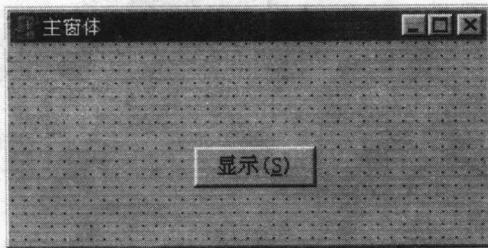


图 1.3 修改了属性后的窗体

0255 8265211

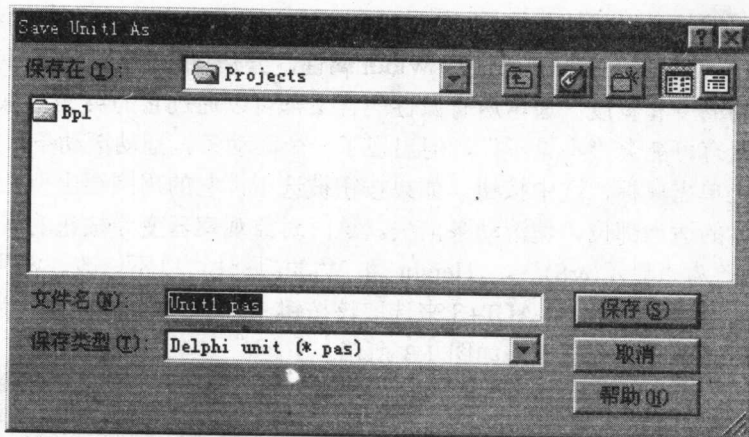


图 1.4 Save Until As 对话框

1.1.4 运行程序

激动人心的时刻终于到了，我们马上就可看到自己的第一个程序了。按 F9 键，程序运行，单击“显示”按钮，出现“Hello,World!”的消息框。大致如图 1.5:

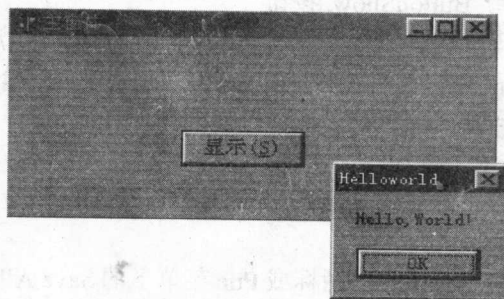


图 1.5 运行第一个程序

1.1.5 关闭项目和退出 Delphi

在程序编写完成以后，我们可以将其关闭了。单击 File 菜单上的 Close All 选项，如果在项目最后一次保存前对其进行了改动，则 Delphi 会出现 Confirm 对话框提示我们是否保存所做的改动，选择“Yes”按钮则将项目所做的改动进行保存，选择“No”按钮则不进行保存，“Cancel”则取消关闭命令，重新回到 Delphi 编程环境。

要退出 Delphi，只需要选择 File 菜单上的 Exit 选项或单击主菜单上的关闭按钮。如果关闭 Delphi 前还有项目没关闭，则它先自动对项目进行关闭。

说明：该程序位于配套光盘的 HelloWorld 子目录下。

以上是我们开发的第一个程序，虽然简单，但它的确是一个可行的程序。Delphi 的快速开发能力，使我们这个程序的开发过程不到 5 分钟。如果用纯 C 或纯 C++ 语句在 Windows 下编写类似的程序，恐怕并没有这么简单。

1.2 Delphi 程序的组成

下面看一下我们刚才保存的文件。单击“我的电脑”，进入我们刚才所保存的 Delphi 文件所在目录，读者会发现有这么几个文件：HelloWorld.exe、HelloWorld.dpr、Main.pas、Main.dfm、Main.dcu、HelloWorld.dof、HelloWord.cfg、HelloWorld.res 和几个临时文件。对于一般的程序，其中只有三个文件与程序设计紧密有关：HelloWorld.dpr、Main.pas 和 Main.dfm。它们分别对应 Delphi 中的项目文件(.dpr)、单元文件(.pas)和窗体文件(.dfm)。其它的文件是 Delphi 在编译链接时自动产生的，我们可以先不理睬。

1.2.1 项目文件

项目文件是 Delphi 中的主文件，它是一个 Object Pascal 程序。它的扩展名为“.dpr”，一般由 Delphi 自动生成，很多情况下我们不必对其进行干涉。只有需要在程序运行时进一步控制程序的运行，才对其进行修改（在下一章中读者看到将对它的修改），初学者一般不用过问。单击 View 菜单下的 Units 菜单项（或按快捷键 Ctrl+F12），显示 View Unit 对话框，选取其中的“HelloWorld”，然后单击“OK”即可在代码编辑器中看到 HelloWorld.dpr 的内容，它的内容如下：

程序清单 1.1 HelloWorld 程序的项目文件清单

```
program HelloWorld;
uses
  Forms,
  Main in 'Main.pas' {Form1};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

一个 Pascal 程序以保留字 Program 为标志，表示一个程序的开始，后面紧跟程序名。程序由说明部分和程序体两部分组成。在上述代码中，Uses 语句和 Begin 语句（不包括 Begin 语句）之间的部分为程序的说明部分，Begin 和 End 语句之间的部分为程序体部分，即程序实际执行的部分。读者可以会奇怪项目文件为什么会这么小。这是因为这个程序将它要引用的类型、变量、过程和函数都放在了单元中，使得程序看起来非常简单。Uses 语句说明程序引用的单元，这个程序引用了两个单元：一个是 Delphi 自带的 Forms 单元，一个是用户自定义的 Main 单元。其中自定义单元给出了它所在的单元文件的名称。

在说明部分，有一个用花括号括起来的语句：{\$R *.RES}。它是一个编译开关，表示要连接的资源文件。在这个程序中，对应的资源文件是“HelloWorld.res”，它也由 Delphi 自动管理。

主程序的执行部分只有三行代码，第一行调用 Application 变量的初始化方法 Initialize，进行系统的一些初始化工作，第二行是创建 Form1 窗体，第三行进入程序的运行部分。

1.2.2 单元文件

用 Delphi 编程，我们的大部分编程工作都是在编写程序的单元代码。通常情况下，创建一个窗体就已经创建了一个单元。当然也可以创建一个独立的单元，方法是选择 File 菜单下的 New 菜单项，在打开的 New Item 对话框中双击 Unit 图标。这时就会有新的单元加到程序中。查看单元代码可用与查看项目代码一样的方法，也可以在设计窗体出现时，按 F12 进行切换。以下是 Main 单元的内容：

程序清单 1.2 Main 单元的代码清单

```
unit Main;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TMainForm = class(TForm)
    ButtonShow: TButton;
    procedure ButtonShowClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  MainForm: TMainForm;
implementation
{$R *.DFM}
procedure TMainForm.ButtonShowClick(Sender: TObject);
begin
  showmessage('Hello, World!');
end;
end.
```

下面我们具体分析这个单元文件的结构。单元以 Unit 保留字作为标志，它后面跟以单元的单元名（在这个例子中就是 Main）。保存项目文件时，提供单元文件名的同时就指定了单元的名称，两者是相同的。一个单元分为两个部分：接口部分和实现部分。Interface

和 `implementation` 之间的部分是单元的接口部分, `implementation` 后面的部分是单元的实现部分。

在单元文件中, 接下来的是 `interface` 指令, 它使其后的条目可供使用这个单元的其它单元利用。接口含有单元的公共说明。不在接口中的条目对其它单元是不可见的。

一个单元可以使用其它单元。例如, `Main` 单元使用了很多不同的单元, 所有单元都列在 `uses` 语句的后面。在 `Main.pas` 中, 接下来的是类型说明部分, 由保留字 `type` 进行标识。类型描述可以在内存中创建的一个条目的本质。例如, `integer` 是一种代表整数的数据类型。要使用 `integer` 类型的数, 必须创建一个或多个这种类型的变量, 这些变量可以保存一个整数。其它的数据类型则可以保存字符、字符串、浮点数、记录之类的数据。

数据类型也可以是类, 这是在 Delphi 应用程序中广泛使用的一种面向对象的结构。类描述对象是可以在内存中创建的特征。类封装相关的代码和数据, 它们一起被称为类的成员。类可以继承其它类的成员。例如, `Main.pas` 中的 `TMainForm` 类继承 `TForm` 类的成员, 而 `TForm` 是 Delphi 为创建窗体提供的一个类。由于继承性, `TMainForm` 被称为 `TForm` 类的子类, 而 `TForm` 类称为 `TMainForm` 类的父类。

`Main.pas` 中的 `TMainForm` 类除了继承父类中的成员, 又说明了两个成员。第一个成员是 `TButton` 对象, 叫做 `ButtonShow`。第二个成员是一个过程 (Delphi 中, 类中的过程和函数统称为类的方法, 过程以 `procedure` 标识, 函数以 `function` 标识), 它是完成某一个特定动作的一个例程。在这个程序中, 当用户单击按钮时, `ButtonShowClick` 过程调用 Delphi 中的一个过程 `ShowMessage` 在程序中显示一个消息框。这些就是我们前面在开发程序过程中, 所创建的组件和事件处理过程。Delphi 编写代码来响应插入到窗体中组件对象的属性和事件。

类也可以有附加的说明, 可能是私有说明或者是公有说明。私有说明中的成员只能被同类成员使用, 而公有说明则可以被其它单元中的语句使用。这个例子中没有定义私有和公有说明。由花括号包围的两个注释语句表示这些类型的说明可以存在的位置。

注释在运行时不起作用。它们只是描述程序某一方面的注解。注释可以解释复杂的语句, 或者标识程序的作者、版本号或其它有用信息。Delphi 支持三种注释方法:

- 使用成对的花括号: { }
- 使用括号与星号的组件: (* *)
- 使用双斜杠: //

注意, 前两种方法可以跨行, 括号中间的部分就是注释语句。第三种方法是从 C++ 语句借用过来的。它不能跨行注释, 它后面的语句就是注释语句。

最后, 在单元的接口部分是一个或多个变量的说明, 变量之前是保留字 `var`。可以没有变量, 也可以有一个或多个变量。`Main.pas` 单元的接口部分只有一个变量 `MainForm`。

`var`

`MainForm: TMainForm;`

这条语句说明创建 TMainForm 类的名为 MainForm 的一个变量。换句话说, MainForm 是一个类的对象或者说是类的实例, 在内存中占用空间。这个例子中的对象代表 HelloWorld 程序的主窗体, 其中含有一个按钮和一个事件处理过程。这些代码的生成过程由 Delphi 自动完成, 用户从不需过问。

类的对象类似于像 integer 这样的数据类型的变量。用户可以创建任何一个类的一个或者多个对象, 就像可以创建多个 integer 变量一样。

单元接口部分之后是单元的实现部分, 其中 Delphi 创建语句来完成实时动作, 同时, 用户可以在其中插入自己的说明和其它程序代码, 也可以在单元的实现部分输入新的语句。

在 implementation 语句之后, 读者可以看到一条比较奇怪的语句:

```
{SR *.DFM}
```

它和注释有些相似, 但不是注释语句。凡是花括号后面有 \$ 符号的语句都表示这是一个特殊的指令 (项目文件中也有类似的指令)。这条语句使程序打开并读取一个以 dfm 为文件扩展名的文件。星号告诉 Delphi 寻找与项目文件相关但扩展名为 dfm 的文件。这个文件含有利用对象观察器输入的窗体的属性。

Main.pas 的最后一行代码是保留字 end, 后跟一个句点。从技术上讲, end 保留字表示单元模块的结束, 而句点指示文件的结束。然而, 相对来说, 这个区别不是非常重要。只要记住单元中只有最后那个 end 才能以句点结束即可。

1.2.3 窗体文件

在定制第一个窗体时, 我们给它设置了标题、高度和宽度等属性, 当然还有一些属性我们用的是默认值。没有使用默认值的属性就保存在窗体文件中。在 Delphi 5 以前的版本中, 我们无法看到它的具体内容, 因为 Delphi 将它编译成了二进制文件。Delphi 5 为我们提供了两种选择, 既可以用二进制方式保存, 也可以用文本方式保存。在设计面板上单击鼠标右键, 弹出快捷菜单, 快捷菜单的最后一项是 Text DFM。如果它处于选中状态, 则窗体文件将以文本方式保存, 否则将以二进制方式保存。快捷菜单倒数的第二项是 View As Text, 选择它就可以在代码编辑窗口中查看窗体文件的内容, 具体内容如下:

程序清单 1.3 HelloWorld 程序的窗体文件内容

```
object MainForm: TMainForm
```

```
Left = 183
```

```
Top = 165
```

```
Width = 300
```

```
Height = 150
```

```
Caption = '主窗体'
```

```
Color = clBtnFace
```

```
Font.Charset = DEFAULT_CHARSET
```

```
Font.Color = clWindowText
```

```
Font.Height = -11
```

```

Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
PixelsPerInch = 96
TextHeight = 13
object ButtonShow: TButton
  Left = 104
  Top = 64
  Width = 75
  Height = 25
  Caption = '显示(&S)'
  Font.Charset = GB2312_CHARSET
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = '宋体'
  Font.Style = []
  ParentFont = False
  TabOrder = 0
  OnClick = ButtonShowClick
end
end

```

从上例可以看到各个控件的层次关系，例如 ButtonShow 是放在 MainForm 中的，因为设计窗体时我们的确是将一个按钮放在了窗体中。要取消文本的查看方式回到窗体设计方式，只要选择弹出菜单的 View as Form 项或使用快捷键 Alt+F12。

1.3 编写第二个程序

作为第一个程序的补充，我们来编写第二个程序，让读者进一步体会在 Delphi 下编写程序的容易性和乐趣。这次的程序是一个电子钟，它可以根据系统时间来显示时间，是一个有实用价值的小程序。它用到的组件除了 Standard 页下的 Label 组件外，还有 System 页下的 Timer 组件。Timer 组件是一个不可见组件，在程序运行时它不会出现在主窗体上，但在程序的背后发挥着重要的作用。

提示：Timer 组件是 Delphi 最简单的组件之一。它只有四个属性，其中最重要的是 Interval。它只提供了一个事件，即 OnTimer。该事件将每隔一定的时间自动被触发，这个时间间隔由 interval 属性指定。我们在程序中使用 Timer 组件的目的就是利用它的自动调用 OnTimer 事件的特点。

如果此时读者还没有退出 Delphi 的集成环境，请单击 File|Close All 菜单项关闭当前正在编写的程序。Delphi 可能会提示读者保存文件，这只需按提示一步步操作即可。关闭程序后，按下列步骤操作。

1. 单击 File|New 菜单项，开始一个新的应用程序。在对象观察器中，将窗体的 Caption

属性设置为“电子钟”，并将其 Name 属性设置为 MainForm。选择 File|Save All 菜单项保存所有文件，将单元文件命名为 Main，将项目文件命名为 Clock。

2. 在这个程序中，不允许用户在程序运行时改变窗体的大小。为了消除窗体的最小化和最大化按钮，选中窗体，在对象观察器中双击 BorderIcons 属性。该属性左边有一个小加号，双击后它变成了减号。加号告诉用户这个属性还拥有多个隐藏的子属性。出现减号后，它下面显示的就是子属性。将子属性中的 biSystemMenu 设置为 True，其它的值设为 False。这样做的目的是使用户不能使用最大化和最小化按钮，避免影响窗体的整体布局。接下来设置 BorderStyle 属性，它的右边是一个下拉组合框，单击向下按钮，从下拉列表框中选择 bsSingle。设置这个属性值的目的是锁定窗体的大小，使用户不能用鼠标改变它的尺寸。最后将窗体的 Height 设为 180，Width 设为 250。下面在窗体上放置其它组件。
3. 在设计窗体上先放置一个 Label 组件，考虑该组件的特殊用途，我们必须将其拖大一些，大概 Height 为 40，Width 为 120 左右。再在对象观察器中将其 Name 设置为 LabelTime，而不是使用默认值 Label1，将 Caption 属性的默认值删掉。选中 Font 属性，再单击右边有三个小圆点的小按钮，弹出“字体”对话框，将字体设为粗体，大小设为 24 磅，按“确定”退回到对象观察器。

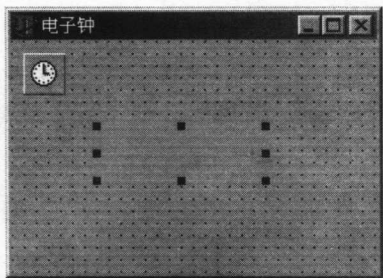


图 1.6 设计中的电子钟窗体

4. 从 System 面板中选取 Timer 组件放入窗体中，因为它是一个不可见组件，所以我们不必关心它的位置，最后的设计窗体如图 1.6 所示。

5. 下面我们编写程序代码。选取 Timer 组件，在对象观察器中切换到 Event 页，双击 OnTimer 右侧的空白区，光标切换到代码编辑窗口中。在 Timer1Timer 过程中输入以下语句：

```
LabelTime.Caption:=TimeToStr(Time);
```

该语句的作用是将系统的当前时间转化为字符串后，再将它赋给 Label 的 Caption 属性，使当前时间以标签标题的形式显示出来。Time 是 Delphi 中的一个函数，它以某种格式返回系统当前时间，TimeToStr 函数将这种特定格式的值转化为字符串形式。Delphi 中的赋值语句用“:=”表示。Timer 组件的 OnTime 事件每隔一定的时间（由 Interval 属性设置，默认值为 1000 毫秒）由程序自动触发，所以加上这个组件并适当编写了代码后，标签的标题就能每一秒每一秒地自动更新。此时我们可以运行程序了，记住在进行之前一定要先事先保存所有文件。这是一个好习惯，在某些情况下它会减少我们很多的麻烦。

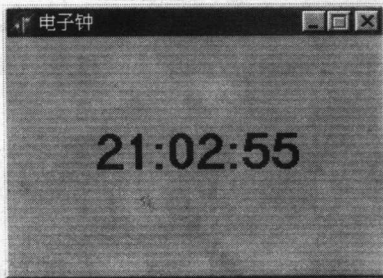


图 1.7 电子钟运行情况

6. 按 F9 或单击加速栏上的三角形按钮运行程序, 我们将看到如图 1.7 所示的一个电子钟。

该电子钟每秒钟自动更新, 栩栩如生。但是还有美中不足的地方, 如果仔细观察就会发现它在开始运行的时候, 窗体上有大约一秒钟的空白, 并不是程序一运行马上就有时间显示, 为什么呢? 这是因为程序在开始运行时, Timer 组件的 OnTimer 事件是在等待了一秒钟后才调用的, 故这前一秒钟的时间里, 窗体上什么都没有。为了解决这个问题, 我们可以在窗体建立时就先调用一次 Timer1Timer 过程。具体做法是, 先选择主窗体对应的 Event 页, 双击 OnCreate 右侧的空白, 在过程 Form1Create 中输入以下语句:

```
Timer1Timer(nil);
```

至此我们的第二个程序就算是编写完了。OnCreate 事件是窗体的一个事件, 它在窗体建立时由程序自动调用, 通常我们将一些变量的初始化工作放在这个事件中进行处理。在这里, 我们在窗体建立时调用一次 Timer1Timer 过程, 这使本来有的那一秒钟的拖延没有了。

说明: 该程序位于配套光盘的 Clock 子目录下。

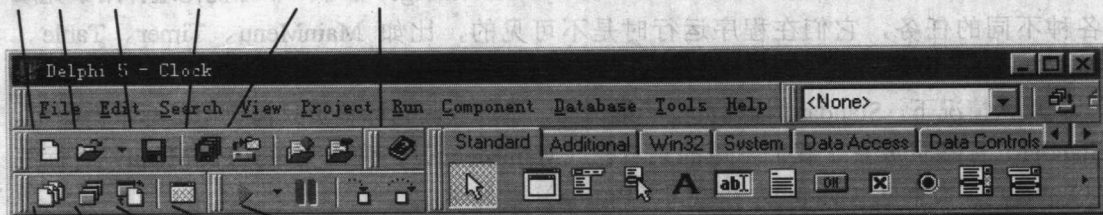
1.4 IDE 简介

Delphi 的 IDE (集成开发环境) 是一个功能非常强大、使用非常方便的开发环境。下面对其进行介绍。

1.4.1 主菜单和加速栏

Delphi 主菜单的标题栏上有当前项目的名称以及最大化、最小化和关闭按钮。它包含有 10 个菜单。加速栏中有很多图标, 它提供对最常用的 Delphi 操作的快捷方式, 这些功能均可在那 10 个菜单的菜单项中找到。加速栏图标的作用见图 1.8。

新建 打开 保存 保存所有文件 打开项目 帮助



单元列表 窗体列表 窗体单元切换 新建窗体 运行程序

图 1.8 主菜单、加速栏和组件面板图示