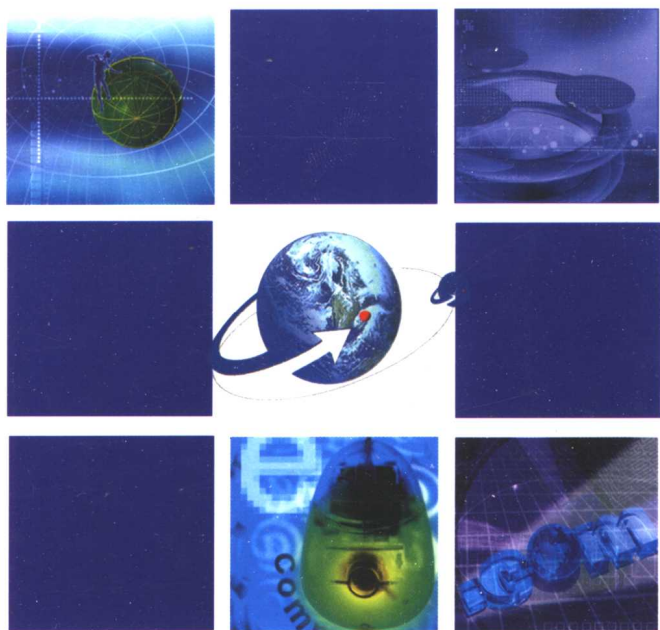


/// 万水创作效果百例丛书 ///



# Delphi 2005

## 精彩编程百例

王振江 等编著



中国水利水电出版社  
www.waterpub.com.cn

万水创作效果百例丛书

# Delphi 2005 精彩编程百例

王振江 等编著

中国水利水电出版社

## 内 容 提 要

Delphi 是 Borland 公司非常著名的应用程序开发工具, 其版本更新的速度表明了它蓬勃发展的生命力和竞争力。新近推出的 Delphi 2005 更增加了对 .NET Framework 的支持, 使其使用范围更加广泛, 具有更大的活力。

本书从实际应用的角度出发, 详细介绍了经典的 Delphi 应用程序开发和新增的基于 .NET Framework 的应用程序开发。第一章介绍在 Delphi 中开发应用程序的一些基础知识和技能; 第二章在第一章的基础上介绍一些比较综合的应用, 以开阔读者的眼界; 第三章介绍图形图像和多媒体的相关开发; 第四章为数据库应用开发, 包括经典的 Delphi 数据库开发和 ADO.NET 的应用开发; 第五章介绍网络应用程序的开发, 包括网络通信和 ASP.NET 应用程序的开发。

本书适合初学者阅读, 也可以作为中级开发人员的参考书。

本书所有实例的完整源代码请从中国水利水电出版社网站 (<http://www.waterpub.com.cn/softdown/>) 下载。

### 图书在版编目 (CIP) 数据

Delphi 2005 精彩编程百例 / 王振江等编著. —北京: 中国水利水电出版社, 2005

(万水创作效果百例丛书)

ISBN 7-5084-3333-5

I.D… II.王… III.软件工具—程序设计 IV.TP311.56

中国版本图书馆 CIP 数据核字 (2005) 第 119326 号

书 名	Delphi 2005 精彩编程百例
作 者	王振江 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: <a href="http://www.waterpub.com.cn">www.waterpub.com.cn</a> E-mail: <a href="mailto:mchannel@263.net">mchannel@263.net</a> (万水) <a href="mailto:sales@waterpub.com.cn">sales@waterpub.com.cn</a> 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 22.25 印张 546 千字
版 次	2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷
印 数	0001—5000 册
定 价	34.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

# 前 言

2001 年，微软推出了全新的 .NET 战略，希望互联网以其产品和服务为中心而发展，这意味着它能让个人电脑、移动电话、编写设备、数码相机甚至家用电器直接彼此交换信息，或者与互联网交换信息，让计算机进入完全的网络时代。此战略的推出开始了一个应用开发的新时代。

.NET 框架是用于建立 Internet 时代应用程序的综合性平台，它允许开发人员建立使用 Windows 平台的完全富于新特色的应用程序。.NET 框架包含一个内容极其广泛的类框架，为程序开发人员提供方便的服务。此外，.NET 框架为安全性、跨平台通信、资源管理和应用扩展性提供完整的服务。

Delphi 2005 是基于 .NET 框架的应用开发平台，它帮助开发者实现基于 .NET 框架的应用和实现从现有的 Win32 Delphi 向 .NET 框架过渡。开发者可以利用现有的 Delphi 语言和框架技术构建基于 .NET 框架的应用，同时，开发者也可以通过现存的 Delphi 应用源代码和技术，轻松实现从 Windows 应用到 .NET 框架的移植，从而保障原先的投资价值。反之，开发者可以轻松实现 .NET 框架应用到 Windows 应用的移植。

Delphi 2005 的设计目的是让 Delphi 开发者能够创建基于 .NET 框架的应用和 Win32 的应用，并且可以移植现存的 Win32 Delphi 程序到 .NET 框架上。由于基于 .NET 框架的开发越来越多，这种从 Win32 到 .NET 的转变正变得越来越重要。

支持微软 .NET 框架的 Delphi 2005 可实现多种可靠应用的快速开发：

- ✓ 支持稳固的 Web 开发的微软 ASP.NET Web Forms 和 XML Web 服务。
- ✓ Windows Forms 和可视控制库 (VCL) Control 及几百种内含的类和控件。
- ✓ 支持高性能数据库应用的微软 ADO.NET 和 Borland Data Provider (BDP)。
- ✓ Borland Enterprise Core Objects (ECO)，以加速设计开发。
- ✓ Delphi 开发者通过现有的 Delphi 技术和设备实现向 .NET 框架的转变非常方便。
- ✓ 熟悉 Delphi 的语法，可免去学习新语言的波折。
- ✓ VCL 控制既能向下兼容已有的 Delphi 源代码，又能与 .NET 框架一起运作。
- ✓ 数据库技术简化了数据库应用向 .NET 框架移植。

本书利用实例由浅入深地介绍了在 Delphi 2005 开发应用程序的方法，由最初的基础开发介绍到后来的专题内容分析，读者在使用本书时，可以循序渐进地阅读，也可以直接查阅自己感兴趣的内容，下面对本书的内容作简单概述。

第一章：使用 VCL.NET 编程方法和全新的 Windows Forms 开发 20 个实例，如果读者对于传统的 VCL 应用开发已经有比较深入的了解，可以跳过此章，直接阅读下一章内容。

第二章：介绍一些比较综合的应用，包括 VCL.NET 和 Windows Forms 的应用，主要是 Windows Forms 应用程序，如 Windows 服务、系统日志、应用程序调试信息等。

第三章:介绍使用 VCL.NET 和 Windows Forms 开发图形图像和多媒体程序的方法及基本技能。

第四章:介绍使用 BDE, dbExpress 和 ADO.NET 开发数据库应用的方法以及几乎全部组件和 .NET 命名空间中数据库部分的内容。

第五章:介绍使用 Socket 组件开发网络应用以及使用 ASP.NET 开发网站的一些方法,还涉及到 XML Web Service 的内容。

除封面署名作者外,参与本书写作、整理、调试、审查的还有董丹丹、王添添、王丽霞、秦国亮、付兴旺、周波、王东亮、王光灿、李志锋、朱鹏、高贵强、刘洪涛、刘景华、杨杰、杨铭、刘颖、阳坚、张华、钟景富、李飞龙、赵凌志等。由于作者水平有限,书中难免有不足和纰漏之处,恳请广大读者批评指正。

作者

2005年8月

# 目 录

前言

## 第一章 Delphi 2005 编程基础

1.1 Delphi 开发语言概述.....	2
1.1.1 VCL Forms Application for .NET.....	3
1.1.2 Windows Forms Application for .NET .....	5
1.1.3 VCL Forms Application for Win32.....	6
1.1.4 ASP.NET Web 应用程序.....	7
1.1.5 IntraWeb Web 应用程序.....	7
1.2 Delphi 2005 开发环境介绍.....	7
实例 1 透明窗体.....	9
实例 2 程序启动窗体.....	12
实例 3 Windows 系统动画.....	15
实例 4 使用 WindowsAPI .....	18
实例 5 进度条和单选按钮组实例.....	21
实例 6 使用 Windows 消息.....	24
实例 7 使用自定义消息（一）.....	27
实例 8 使用自定义消息（二）.....	30
实例 9 模式和无模式对话框.....	33
实例 10 生成奇异窗体.....	36
实例 11 动态创建菜单.....	40
实例 12 操作系统级菜单.....	43
实例 13 制作漂亮的菜单.....	46
实例 14 实现关机功能.....	48
实例 15 多样化显示窗体.....	51
实例 16 隐藏桌面和任务栏.....	55
实例 17 改变桌面墙纸.....	58
实例 18 改变显示器分辨率.....	61
实例 19 创建托盘程序.....	64
实例 20 简单定时器.....	67

## 第二章 组件综合应用

2.1 VCL Forms 应用程序开发.....	71
---------------------------	----

2.1.1	文本相关组件.....	71
2.1.2	按钮和检查框组件.....	71
2.1.3	分组、分界组件.....	72
2.1.4	图形图像组件.....	72
2.1.5	菜单组件.....	72
2.2	Windows Forms 程序开发介绍.....	72
2.2.1	控制台应用程序.....	73
2.2.2	Windows 窗体应用程序.....	74
2.2.3	系统服务程序.....	75
实例 21	制作电子表.....	78
实例 22	制作石英钟.....	81
实例 23	猜数字游戏.....	85
实例 24	查找窗体及其属类.....	89
实例 25	使用线程.....	92
实例 26	多线程同步.....	95
实例 27	模拟控制台程序.....	98
实例 28	光驱关闭程序.....	101
实例 29	简单记事本.....	104
实例 30	文件浏览器.....	108
实例 31	简单计算器.....	110
实例 32	获得系统信息.....	115
实例 33	模拟欠阻尼阶跃曲线.....	118
实例 34	定制开机启动程序.....	122
实例 35	创建指定大小的文件.....	125
实例 36	使用系统事件日志.....	128
实例 37	使用追踪调试程序.....	131
实例 38	经典方式使用 WinForm 事件.....	135
实例 39	创建服务程序.....	138
实例 40	监视系统文件和目录.....	142

### 第三章 图形图像和多媒体

3.1	VCL Forms 系统图形图像开发.....	147
3.1.1	TCanvas 对象.....	147
3.1.2	TPen 对象.....	148
3.1.3	Tbrush 对象.....	149
3.1.4	TColor 类型.....	149
3.1.5	TGraphic 对象.....	150

3.1.6	TPicture 对象 .....	150
3.1.7	TImage 部件 .....	150
3.1.8	TBitmap 对象 .....	150
3.2	Windows Forms 系统图形图像开发 .....	151
3.2.1	Graphics 类 .....	151
3.2.2	Pen 类 .....	152
3.2.3	Brush 类 .....	152
3.2.4	Font 类 .....	153
3.2.5	Bitmap 类 .....	153
3.2.6	图像变换 .....	153
实例 41	图形的多样化显示 .....	155
实例 42	透明的素描板 .....	157
实例 43	简单编辑图片 .....	160
实例 44	像素点 RGB 的转换 .....	163
实例 45	弹出窗口 .....	165
实例 46	动态改变鼠标和图标 .....	167
实例 47	截取图像部分并显示 .....	170
实例 48	文字动态滚动 .....	172
实例 49	飞行的窗口及图片 .....	174
实例 50	媒体播放器 .....	176
实例 51	创建第一个 WinForm 绘图程序 .....	178
实例 52	使用 Windows 画笔 .....	180
实例 53	使用 Windows 画刷 .....	184
实例 54	使用字体 .....	186
实例 55	绘制图像和文本 .....	189
实例 56	处理颜色 .....	192
实例 57	使用坐标转换 .....	196
实例 58	小型画板 .....	198
实例 59	图像格式转换 .....	203
实例 60	将绘图存入文件 .....	206

## 第四章 数据库应用

4.1	VCL Forms 环境下的数据库编程 .....	211
4.1.1	BDE 方案 .....	211
4.1.2	dbExpress 方案 .....	213
4.2	Windows Forms 环境下的数据库编程 .....	214
4.2.1	XML 和 ADO.NET .....	215



4.2.2 ADO.NET 组件 .....	215
4.2.3 使用 DataReader 还是 DataSet .....	216
4.2.4 ADO.NET 的核心对象 .....	216
实例 61 创建第一个数据库应用 .....	219
实例 62 了解 BDE .....	224
实例 63 使用 TTable 动态建表 .....	226
实例 64 使用 TQuery .....	229
实例 65 使用带参数 SQL 查询 .....	231
实例 66 建立主从表 .....	234
实例 67 使用 TBatchMove .....	236
实例 68 用 TTable 的 Filter 实现查询 .....	239
实例 69 了解 dbExpress .....	241
实例 70 使用 TSQLDataSet .....	244
实例 71 了解 ADO.NET .....	247
实例 72 使用 DataSet .....	250
实例 73 使用参数 .....	253
实例 74 写入数据到 XML .....	257
实例 75 使用数据绑定 .....	260
实例 76 使用 DataGrid .....	262
实例 77 使用 DataTable .....	265
实例 78 使用 DataReader 检索多个数据集 .....	268
实例 79 使用 DataSet 操作 XML .....	271
实例 80 修改 DataSet 中数据 .....	273

## 第五章 网络互联

5.1 Windows Forms 网络编程 .....	277
5.1.1 TCPListener 类 .....	277
5.1.2 TCPClient 类 .....	278
5.1.3 UdpClient 类 .....	278
5.1.4 NetworkStream 类 .....	279
5.1.5 Socket 类 .....	279
5.2 ASP.NET Web 编程 .....	280
5.2.1 Web 配置 .....	280
5.2.2 Global.asax 文件 .....	281
5.2.3 页面配置 .....	281
5.2.4 ASP.NET 的 Page 类 .....	282
5.2.5 XML Web 服务 .....	283

实例 81	网络聊天服务器端.....	284
实例 82	网络聊天客户端.....	288
实例 83	邮件发送程序.....	293
实例 84	创建第一个 ASP.NET 程序.....	296
实例 85	使用 Response 对象.....	300
实例 86	使用 Request 对象.....	302
实例 87	使用 Session 对象.....	305
实例 88	使用 Cookie 对象.....	308
实例 89	使用验证组件.....	311
实例 90	在 ASP.NET 中应用 ADO.NET.....	313
实例 91	使用文件.....	315
实例 92	使用 XML (一).....	318
实例 93	使用 XML (二).....	321
实例 94	使用广告组件.....	323
实例 95	网页综合应用 (一).....	326
实例 96	网页综合应用 (二).....	329
实例 97	建立网络服务程序.....	333
实例 98	使用网络服务.....	335
实例 99	简单 Web 浏览器.....	338
实例 100	简单 Ftp 客户端.....	342

## 第一章

# Delphi 2005 编程基础

Delphi 2005 的界面比之以前版本的界面都变得漂亮了，功能也变得更加强大了，使用起来更加方便，其方便快捷的 IDE 环境使得开发应用程序变得非常容易。

本章通过 20 个简单的例子介绍了 Delphi 编程的基础知识，包括传统的 VCL Forms Application 和在 Delphi.Net 中新添加的 Windows Forms Application，其中使用了一些 Windows API 函数。

欢迎大家来到 Delphi 殿堂，与 Oracle 一样，Delphi 带有传奇的神话色彩，Delphi 是著名的希腊古都，因太阳神阿波罗的神殿而著称。Delphi 自诞生之日起，其所见即所得的特点就给了广大的程序员极大的便利，使得程序员可以从繁重的界面代码中解脱出来，将更多的精力投入到高层的业务逻辑中。直到现在，Delphi 仍然是众多的编程工具中的一颗璀璨的明星。

## 1.1 Delphi 开发语言概述

作为本书的开始，首先来看一下 Delphi 所使用的基本语言。Delphi 使用 Object-Pascal，即一种面向对象的 Pascal 作为开发语言。Object-Pascal 是对传统的 Pascal 语言的扩展，在传统意义上的 Pascal 中加入了很多面向对象的特性，如类、对象、异常控制、多线程编程、模块编程、OLE 自动化等，但仍然保留了 Pascal 原有的简单性。

Object-Pascal 是一种模块化的语言，每一个模块叫做一个 Unit，对应一个 .pas 的文件，类似于 Java，文件的名称与 Unit 的名称必须相同。一个 Unit 可以代表一个窗体，它描述了窗体的行为，该窗体可以在 Delphi 的集成编辑环境中进行修改编译，窗体的描述放在一个窗体描述文件中，通常是 .dfm 文件或者 .nfm 文件，.dfm 文件表示 Win32 程序的窗体文件，.nfm 表示 .NET 程序的窗体文件。每一个窗体的描述文件，无论是 .dfm 还是 .nfm 文件，都有一个相关联的 Unit 文件。窗体描述文件是 Delphi IDE 所集成的文件，并不是 Object-Pascal 中的内容。

一个 unit 包含两个部分：Interface 和 Implementation。Interface 部分声明类型、变量、常量以及一些外部可以访问的方法。Implementation 部分实现在 Interface 中声明过的外部方法。

一个 unit 可以引用另外一个 unit，当一个 unit 改变时，需要重新编译其所引用的所有的 unit 文件，不过 Delphi 会自动完成这件事情，而不必使用 makefiles 来手动操作。

在 Delphi 2005 中可以建立三种窗体程序和两种 Web 应用程序，其应用环境也不同。

(1) VCL Forms Application for .NET，这是 Borland 公司将优秀的 VCL 移植到 .NET 平台上的成功作品，在此应用环境下，既可以使用经典的功能强大的 VCL 组件，又可以使用全新的封装于 Microsoft 的 .NET 框架的类库，无论是经典的 Delphi 程序员还是来自微软 .NET 平台的程序员，都可以自如地在此应用环境下进行开发。在用户操作界面上，基本还是沿用原来 Delphi 的 VCL 开发环境的界面，程序构成也非常类似（pas 文件与 nfm 文件联合表示窗体），对于原有 Delphi 的程序员来说几乎是水到渠成的升级。需要注意的是，在此应用开发环境下开发的程序需要微软 .NET Framework 的支持，在没有安装 .NET Framework 的机器上无法运行。

(2) Windows Forms Application for .NET，这是 Borland 公司的 .NET 战略的另一个巨大成就，此应用环境成功地将 Microsoft 的 .NET 平台的开发环境移植到 Delphi 上，可以说是 Delphi 平台的一个全新领域，在此应用环境下最适合与 .NET 无缝集成，也最适合从微软的 .NET 平台转型到 Delphi 的程序员。在此环境下，用户可以完全使用微软 .NET 平台中所有的组件和类库，同时，窗体的所有代码，包括窗体的行为和显示状态，都包含在一个 pas 文件中，在进行代码管理时，会更加方便。另外，由于此应用环境是 .NET 的 Windows Forms，因此同样需要 .NET Framework 的支持。

(3) VCL Forms Application for Win32，这是 Borland Delphi 的经典之作，从最初的 Delphi 到现在，一直经久不衰，其强大的组件库和精巧的程序设计一直使广大的 Delphi 程序员叹为观止，也正是因为其简单易用的特点和强大的功能而吸引了大量的程序员投身于 Delphi 的程

序开发。此开发环境重现了 Delphi 的前期版本，喜爱做 Delphi VCL Forms 的程序员可以在此找到梦中的天堂。此开发环境基于 Win32 的平台，因此开发出的程序在运行时是不需要 .NET Framework 支持的。

(4) ASP.NET Web 应用程序，这是从微软 .NET 平台移植的 Web 应用程序，对于使用 Delphi 开发 Web 应用程序是一个突破性的进展，使得 Delphi 在开发 Web 应用程序方面有了更大的发展空间。

(5) IntraWeb Web 应用程序，这是经典的 Delphi 开发 Web 应用程序的方式，它用现在流行的所见即所得的设计理念来设计网站。用它不仅可以很方便地做出各种论坛，而且还能设计出功能强大的多层数据库应用程序。这又一次让 Delphi 把 RAD 技术发挥到了极致。

### 1.1.1 VCL Forms Application for .NET

为了对 VCL.NET 有一个直观上的了解，先来看一个简单的 Delphi.NET 程序，该程序实现在窗体加载时显示一条消息。

```
1. unit testdemo;
2. interface
3. uses
4.   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs;
5. type
6.   TForm1 = class(TForm)
7.     procedure FormCreate(Sender: TObject);
8.   private
9.     { Private declarations }
10.  public
11.    { Public declarations }
12.    procedure ShowHello(strHello:string);
13.    function HelloBaby(strBaby:string):string;
14.  end;
15. var
16.   Form1: TForm1;
17. implementation
18. {$R *.nfm}
19. procedure ShowHello;
20. begin
21.   ShowMessage('Hello World!');
22. end;
23. procedure TForm1.ShowHello(strHello:string);
24. begin
25.   ShowMessage(strHello);
26. end;
27. function TForm1.HelloBaby(strBaby:string):string;
28. begin
29.   HelloBaby := 'Hello ' + strBaby + ',how are you';
30. end;
31. procedure TForm1.FormCreate(Sender: TObject);
```

```
32. begin
33.   ShowHello(Self.HelloBaby('Jane'));
34. end;
35. end.
```

上面的程序是一个简单的 Delphi.NET 的 VCL 窗体应用程序的 unit 文件,下面来看一看它的结构。

在第 1 行,给出了该 unit 的声明,声明该 unit 的名字为 testdemo。

在第 2~14 行定义了该 unit 的 interface,其中第 3~4 行指出了该 unit 所引用的外部 unit,第 5~14 行定义了一个类型 TForm1,该类型继承自 TForm 类型,是一个窗体类;在 TForm1 类型中声明了三个成员,其中第 7 行声明的过程是与窗体的 OnCreate 事件关联的响应过程,该关联的声明可以在.nfm 窗体描述文件中找到。下面给出与上述 unit 文件关联的窗体描述文件的代码:

```
1. object Form1: TForm1
2.   Caption = 'Form1'
3.   Color = clBtnFace
4.   Font.Charset = DEFAULT_CHARSET
5.   Font.Color = clWindowText
6.   Font.Height = -11
7.   Font.Name = 'Tahoma'
8.   Font.Style = []
9.   Height = 240
10.  Width = 347
11.  Left = 0
12.  Top = 0
13.  OnCreate = FormCreate
14.  PixelsPerInch = 96
15.  TextHeight = 13
16. end
```

可以看出在该窗体内只有一个窗体对象,其中在第 13 行描述了窗体的 OnCreate 事件的关联响应过程,其他都是对窗体显示状态的描述。注意,在 Windows Form 程序中,窗体的事件及其响应过程是定义在同一个文件中的,使用标准的事件关联方法 Include 来实现,对此问题的详细介绍会在后面给出。

再转到 unit 文件,第 8 行和第 10 行的 private 和 public 是方法或者属性限定符,定义在其作用域中的属性或者方法将具有两者所定义的外部可见度,有过编程经验的人,对 private 和 public 的外部可见度的定义应该很了解,在此不再赘述。在 unit 文件中,定义了两个 public 的方法,其中一个是过程 ShowHello,另外一个函数 HelloBaby。

在 unit 文件的 15, 16 行,定义了一个 TForm1 类型的对象 Form1,用于表示本窗体对象。一些变量的声明也可以放在此处。

第 17~35 行实现了上述定义过的 Interface,以关键字 Implementation 开始。第 18 行描述了与 unit 文件关联的资源,存放在.nfm 文件中。

细心的读者可能会发现,在第 19 行和第 23 行,有两个 ShowHello 方法的实现,其中一个有 TForm1 的前缀,这两者是有区别的,有 TForm1 前缀的方法是 TForm1 的类型成员方法,

而没有前缀的方法是全局方法，两者在同一个 unit 中可以并存，但是在 unit 中使用时，类型成员方法有优先权，因此在 unit 的第 33 行的方法调用中将调用第 23 行的 ShowHello 方法，而不是第 19 行的方法。

### 1.1.2 Windows Forms Application for .NET

再来看一下 Delphi for .NET 中 Windows Forms Application 的结构。下面给出一个简单的 Delphi for .NET 中 Windows Form 的 unit 文件，实现的功能同上。

```

1.  unit testWinForm;
2.  interface
3.  uses
4.  System.Drawing, System.Collections, System.ComponentModel, System.Windows.Forms, System.Data;
5.  type
6.  TWinForm = class(System.Windows.Forms.Form)
7.  {$REGION 'Designer Managed Code'}
8.  strict private
9.    Components: System.ComponentModel.Container;
10.   procedure InitializeComponent;
11.   procedure TWinForm_Load(sender: System.Object; e: System.EventArgs);
12.  {$ENDREGION}
13.  strict protected
14.   procedure Dispose(Disposing: Boolean); override;
15.  private
16.   { Private Declarations }
17.  public
18.   constructor Create;
19.   procedure ShowHello(strHello:string);
20.   function HelloBaby(strBaby:string):string;
21.  end;
22.  [assembly: RuntimeRequiredAttribute(typeof(TWinForm))]
23.  implementation
24.  {$AUTOBOX ON}
25.  {$REGION 'Windows Form Designer generated code'}
26.  procedure TWinForm.InitializeComponent;
27.  begin
28.   Self.AutoScaleBaseSize := System.Drawing.Size.Create(6, 14);
29.   Self.ClientSize := System.Drawing.Size.Create(292, 266);
30.   Self.Name := 'TWinForm';
31.   Self.Text := 'WinForm';
32.   Include(Self.Load, Self.TWinForm_Load);
33.  end;
34.  {$ENDREGION}
35.  procedure TWinForm.Dispose(Disposing: Boolean);
36.  begin
37.   if Disposing then
38.   begin

```

```
39.   if Components <> nil then
40.     Components.Dispose();
41. end;
42. inherited Dispose(Disposing);
43. end;
44. constructor TWinForm.Create;
45. begin
46. inherited Create;
47. InitializeComponent;
48. end;
49. procedure ShowHello(strHello:string);
50. begin
51.   MessageBox.Show('HelloWorld!');
52. end;
53. procedure TWinForm.ShowHello(strHello:string);
54. begin
55.   MessageBox.Show(strHello);
56. end;
57. function TWinForm.HelloBaby(strBaby:string):string;
58. begin
59.   HelloBaby := 'Hello ' + strBaby + ',how are you';
60. end;
61. procedure TWinForm.TWinForm_Load(sender: System.Object; e: System.EventArgs);
62. begin
63.   ShowHello(HelloBaby('jane'));
64. end;
65. end.
```

Delphi.NET 的 Windows 窗体程序的结构和 VCL.NET 的结构大致相同，下面只讲述其不同之处。

VCL.NET 所依赖的窗体描述文件 (.nfm 文件) 在 Windows 窗体中将不再存在，窗体的所有信息都保存在 unit 文件中，原来在窗体描述文件中定义的窗体的显示属性在 Window Forms 的 InitializeComponent 方法中实现。注意第 32 行中的 Include 方法，该方法将对象的事件与响应过程联系起来，其使用形式为 Include(event,procedure)，其中，event 可以是系统定义的事件也可以是自定义事件。如果是自定义事件，则需要在 interface 的类型定义中定义件。

第 9 行定义了一个 Components 对象，该对象是 System.ComponentModel.Container 类型，承载了所有的窗体内部组件，是窗体设计器中最重要的对象。

在 Windows 窗体程序中，除了上述的 InitializeComponent 方法外，还另外增加了 Dispose 和 constructor 两个方法，前者的作用是将窗体所占用的资源全部释放，如果在窗体中定义了一些不能在程序结束时自动释放资源的对象，可以在此方法内释放资源；后者是窗体的构造函数，在创建窗体时首先被调用，用来构造窗体的显示状态和行为，如果需要在创建窗体时调用一些方法或者定义一些对象，可以在此方法内进行。

### 1.1.3 VCL Forms Application for Win32

Delphi.Win32 中的 VCL 窗体与 Delphi.NET 中 VCL.NET 的窗体非常类似，在此不再赘述。



但是需要注意的是，在 VCL.NET 中可以使用的微软 .NET 类库在此将不能再使用，不过可以使用经过 Borland 封装的众多 Windows API，从而实现与 .NET 类库同样的功能，而且也不再受 .NET Framework 的限制。

### 1.1.4 ASP.NET Web 应用程序

ASP.NET 应用程序实现了代码隐藏的功能，即将页面的显示和页面的事务逻辑分开在不同的文件中，有点类似于 Delphi 的 VCL Forms 程序中窗体描述文件和 unit 文件分开存储，这样便于不同的开发人员的工作分开进行，网页美工人员只专注于网页文件，而程序开发人员只专注于事务逻辑的开发。

代码隐藏的关键代码在 aspx 文件的第一行，如下所示：

```
<%@ Page Language="c#" Debug="true" Codebehind="WebForm1.pas" AutoEventWireup="false"
Inherits="WebForm1.TWebForm1"%>
```

可以看到在该 Page 指令中，指出了代码隐藏文件为 WebForm1.pas，该页面继承该代码文件中的 WebForm1.TWebForm1 类。

对于具体的程序代码，由于和 Windows Forms Application 都是从微软的 .NET 平台移植而来，因此其结构和风格基本一致，读者可以在编程时非常明显地感受到这一点，这里就不再赘述了，可以参考 1.2 节的描述。

### 1.1.5 IntraWeb Web 应用程序

IntraWeb 是集成的 Web 应用程序开发套件，其丰富的 Web 组件和数据操作工具给网站开发带来了非常大的便利，其多媒体数据的传输和播放组件更是功能强大。一个简单的 IntraWeb 的 Web 应用程序一般由 ServerController, UserSessionUnit 和 Web 窗体等几部分组成，其中 Web 窗体定义网页的显示形式和行为特点，ServerController 定义应用程序级的行为，而 UserSessionUnit 实现用户会话的管理。在 IntraWeb 的 Web 程序中，不会看到任何 Html 代码，其网页中所有的代码都是动态生成的，有效地实现了网址隐藏和代码重用，另外，其网页中生成了大量的 javascript 代码，可以高效地实现客户端的功能。

## 1.2 Delphi 2005 开发环境介绍

下面将简单介绍 Delphi 2005 的集成开发环境，如果您已经对此开发环境比较熟悉，可以跳过此节。

图 1 为 Delphi 2005 的主界面，其中红色的标记表示当前所使用的布局方式。用户可以调整 IDE 开发环境为自己最熟悉和最习惯的样式，然后存储为用户布局方式，在 Delphi 2005 加载时会自动地加载用户的 Layout 方式。图中显示的是笔者所使用的布局方式，以笔者的名字命名。

图 2 给出了 Delphi 2005 的新建工程的菜单。可以看到在 Delphi 2005 中，可以建立的工程类型主要有 ASP.NET Web Application, Windows Forms Application, VCL Forms Application (for .NET 和 for Win32) 等，如果想建立其他的工程，如 IntraWeb 或 ECO 的工程，需要单击 Other (如图 2 中红色标记)，在图 3 所示的对话框中选择。