



附光盘

举一反三



Delphi

程序设计实战训练

老虎工作室
周松辰 编著

人民邮电出版社
POSTS & TELECOM PRESS

举一反三——

Delphi 程序设计实战训练

老虎工作室 周松辰 编著

orland

Delphi
Enterprise

7

人民邮电出版社

图书在版编目 (CIP) 数据

Delphi 程序设计实战训练/老虎工作室, 周松辰编著.

—北京: 人民邮电出版社, 2004.11

(举一反三)

ISBN 7-115-12644-5

I .D... II. ①老...②周... III. 软件工具—程序设计, IV. TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 103667 号

内 容 提 要

Delphi 是一个功能强大而且应用广泛的程序开发工具。本书系统地介绍 Delphi 在程序开发中需要掌握的常用技能和主要知识点。本书除首尾以外的每一章在详细讲解 1 个实例的基础上, 要求读者尝试完成 3 个与实例相关的练习, 并给出练习中需要注意的关键步骤。书中选择的实例涉及读者在程序设计和实际工作中经常会遇到的问题, 如常用组件的使用方法、流的使用技巧、编写和使用动态链接库、自制图形组件、数据库程序设计等。

本书特色鲜明, 典型实用, 适合于具有一定 Delphi 基础知识的程序设计人员使用, 也可以作为大中专院校师生和 Delphi 初级程序员的参考书。

举一反三——Delphi 程序设计实战训练

◆ 编 著 老虎工作室 周松辰

责任编辑 李永涛

执行编辑 徐宝姝

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67132692

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 22.25

字数: 543 千字

2004 年 11 月第 1 版

印数: 1—8 000 册

2004 年 11 月北京第 1 次印刷

ISBN 7-115-12644-5/TP · 4202

定价: 38.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223



老虎工作室

主 编： 沈精虎

编 委：	许曰滨	黄业清	杜俭业	姜 勇	宋一兵
	谭雪松	向先波	毕丽蕴	高长铎	田博文
	郭万军	詹 翔	宋雪岩	周 锦	冯 辉
	王海英	李 仲	马 震	蔡汉明	张 琴
	赵 晶	张 伟	朱 凯	赵治国	彭 智
	张艳花	孙海侠	姜继红	姚育成	杨平辉

内容和特点

随着信息技术的发展和计算机应用的普及, 计算机软件的应用已经渗透到社会生活的各个领域, 特别是在企业管理、电子商务、数值计算、运动仿真、多媒体应用和网络通信等方面发挥着重要的作用。

计算机应用软件开发工具有很多种, 其中 Delphi 易学易用, 功能强大, 深受广大用户的欢迎。许多软件开发人员都有这样的体会, 学习编程语言需要学习例子, 确实如此, 通过例子可以首先得到感性认识, 然后在此基础上再理解例子中包含的知识内容, 最后通过模仿例子编写出自己的程序。通过循序渐进的实例学习和练习, 就可以逐步掌握这门程序设计语言。从读者反馈和市场调查情况看, 现在许多通过实例介绍 Delphi 的图书存在这样的问题: 实例的实用性不强, 实例背后的知识点不系统, 读者跟着书做例子, 感觉没有问题, 但遇到具体问题, 还是会觉得无从下手。

本书根据实际程序开发时最常用的知识要点, 精选和编写了数十个实例。这些实例绝大多数是作者在担任 Delphi 程序员讲师和在实际程序开发时编写的实例, 每个实例针对一个知识点。如果是初学者, 学习这些实例的同时再配合着学习 Delphi 教程类的书, 效果会更好。

本书除首尾以外的每一章在详细讲解 1 个实例的基础上, 要求读者尝试完成 3 个与实例相关或知识点类似的练习, 并给出了练习中需要注意的步骤。通过这种讲解和练习, 将 Delphi 的主要语言精华、组件使用方法和数据库程序开发要点等内容系统地展现给读者。

本书结合了作者多年的 Delphi 教学经验和程序开发的体会, 特色鲜明, 讲解与练习相结合; 典型实用, 每一章讲述的都是常用的知识和技巧; 简明清晰、重点突出, 在叙述上力求深入浅出、通俗易懂。相信会为读者的学习和工作带来一定的帮助。

全书共 11 章, 各章的主要内容如下:

- 第 1 章: Delphi 预备知识。
- 第 2 章: 组件的综合应用。
- 第 3 章: 二进制文件的应用。
- 第 4 章: 流的应用。
- 第 5 章: 创建图形组件。
- 第 6 章: 多媒体播放器。
- 第 7 章: 动态链接库。
- 第 8 章: 数据库应用。
- 第 9 章: C/S 数据库编程。
- 第 10 章: 【DBGrid】报表输出。
- 第 11 章: 开发实例——宾馆客房管理系统。

读者对象

本书特别适合作为具有一定程序设计语言基础的大中专院校学生 Delphi 实例的辅助教材,还可以作为 Delphi 初级程序员的参考书,以及具有一定 Delphi 基础的程序开发爱好者的提高教材。

配套光盘的使用注意事项

为了方便读者的学习,我们将书中实例程序源文件收录在本书的配套光盘中。下面是本书配套光盘内容的详细说明。

本书中的所有例子保存在光盘的“\Delphi_book\”目录下。各章中的例子和用到的文件分别存放在相应的子目录中。例如第 5 章内容存放在“\Delphi_book\Ch5”目录下。

- 第 2 章至第 10 章的各章都包含典型实例、起步、进阶和提高 4 个小节,这 4 个小节中介绍的例子存放在相应的子目录中。
- 第 8 章的工程“MyFootball.dpr”和“DemoDBGrid.dpr”使用的是 paradox 库,运行该程序前需要先配置数据库别名,别名是“Football”,数据库文件请存放在目录“d:\Delphi_book\Ch8\典型实例\MyDatabase”中。
- 第 8 章的工程“FootballTeam.dpr”使用的是 Access 数据库,运行该程序前需要先配置 ODBC 数据源,数据源名称为“Football_Access”,Access 数据库文件为“\Delphi_book\足彩.mdb”,根据实际位置重新配置数据源,然后编译程序后才能正常运行。登录用户名为“admin”,密码为“3209505”。
- 第 4 章的工程“BlobStream.dpr”,使用的是 InterBase 数据库系统,因此在运行该程序前需要先安装好 Interbase 数据库。相应的数据库文件要存放在“d:\Delphi_book\Ch4\MyStream.gdb”中。
- 运行第 11 章的程序需要先安装好 MS SQLServer2000 系统,然后运行“CheckServer.exe”,通过该程序在“MS SQLServer2000 企业管理器”中创建一个名为“HotelDB”的用户数据库,并在该数据库中自动创建“宾馆客房管理系统”程序用到的所有表、视图和存储过程。如果是第一次运行“CheckServer.exe”,还将创建一个名为“Hotel.ini”的配置文件。将配置文件“Hotel.ini”移到“Client”目录中后可以运行“DemoClient.exe”程序。第一次运行系统如果发现没有任何用户存在,则系统自动创建一个名为“sa”,密码为“123”的用户。再次运行“DemoClient.exe”程序,通过“sa”用户进入系统。注意本例子中登录 MS SQLServer2000 的管理员密码也是“123”。
- 本书用到的一些图标文件保存在“\Delphi_book\图标”目录中。

感谢您选择了本书,希望我们的努力对您的工作和学习有所帮助,也希望您把对本书的意见和建议告诉我们。

老虎工作室网站 <http://www.laohu.net>, 电子函件 postmaster@laohu.net。

老虎工作室

2004 年 9 月

第 1 章 Delphi 预备知识	1
1.1 Delphi 7 开发工具简介	1
1.1.1 Delphi 7 的可视化编程	1
1.1.2 Delphi 7 的面向对象编程	5
1.1.3 Delphi 7 的应用领域	15
1.2 Object Pascal 语法精要	17
1.2.1 简单数据类型	17
1.2.2 枚举、子界和集合	24
1.2.3 数组与记录	28
1.2.4 过程与函数	32
1.3 怎样编写 Delphi 应用程序	35
1.3.1 熟悉 Delphi 的程序结构	35
1.3.2 组织好 Delphi 工程	37
1.3.3 减少全局变量的使用	41
1.3.4 编写程序“十做好”	42
1.4 小结	42
第 2 章 组件的综合应用	43
2.1 相关知识点介绍	43
2.2 典型实例——“文件管理器程序”	44
2.2.1 界面设计	45
2.2.2 编写自定义过程	48
2.2.3 编写组件的响应事件代码	54
2.3 起步——树视图组件的使用	60
2.4 进阶——磁盘文件操作	63
2.5 提高——递归的使用	64
2.6 小结	66
第 3 章 二进制文件的应用	67
3.1 相关知识点介绍	67

3.2 典型实例——“学校人员信息库程序”	69
3.2.1 界面设计	70
3.2.2 定义类型和声明变量	72
3.2.3 编写自定义过程	73
3.2.4 编写组件的响应事件代码	76
3.3 起步——文本文件的应用	80
3.4 进阶——无类型文件操作	83
3.5 提高——配置文件的应用	85
3.6 小结	87
第 4 章 流的应用	89
4.1 相关知识点介绍	89
4.2 典型实例——“基于 InterBase 的文件管理系统”	91
4.2.1 配置 InterBase 服务器	92
4.2.2 创建数据库和表	93
4.2.3 InterBase 数据库组件属性设置	94
4.2.4 编写源程序代码	96
4.3 起步——“积分榜程序”	100
4.4 进阶——文件分割	106
4.5 提高——Delphi 应用程序加密器	110
4.6 小结	116
第 5 章 创建图形组件	117
5.1 相关知识点介绍	117
5.2 典型实例——编写实时显示曲线的组件	119
5.3 起步——基于现有组件编写新组件	135
5.4 进阶——建立新的事件类型	139
5.5 提高——编写仪表组件	143
5.6 小结	151
第 6 章 多媒体播放器	153
6.1 相关知识点介绍	153
6.2 典型实例——“我的媒体播放器”	155
6.2.1 界面设计	156
6.2.2 声明变量和编写自定义过程	157
6.2.3 编写组件的响应事件代码	160
6.3 起步——用【Animate】组件实现动画效果	163

6.4 进阶——播放 Flash 文件	166
6.5 提高——播放文本文件	168
6.6 小结	173
第 7 章 动态链接库	175
7.1 相关知识点介绍	175
7.2 典型实例——窗体动态链接库	177
7.2.1 开发“窗体动态链接库”程序	178
7.2.2 编写“窗体动态链接库”测试程序	183
7.2.3 调试“窗体动态链接库”程序	185
7.3 起步——“函数动态链接库”程序	186
7.4 进阶——“类动态链接库”程序	189
7.5 提高——“动态调用 DLL 函数”	194
7.6 小结	196
第 8 章 数据库应用	197
8.1 相关知识点介绍	197
8.2 典型实例——“足球彩票投注程序”	199
8.2.1 创建数据库和表	200
8.2.2 设置数据模块	203
8.2.3 编写“投注分析和下单”窗体单元	204
8.2.4 编写“球队资料管理”窗体单元	213
8.2.5 编写“对阵表管理”窗体单元	215
8.2.6 设计“投注单”输出报表	219
8.2.7 测试“足球彩票投注程序”	220
8.3 起步——【DBGrid】组件使用技巧	222
8.4 进阶——创建与访问 Access 数据库	224
8.5 提高——数据库程序的友好性设计	228
8.6 小结	236
第 9 章 C/S 数据库编程	237
9.1 相关知识点介绍	237
9.2 典型实例——“订票系统”	239
9.2.1 创建数据库和数据表	240
9.2.2 创建应用程序	242
9.2.3 设计“订票系统”主窗体	248
9.3 起步——使用 ADO 访问 MS SQL Server	257

9.4 进阶——动态 SQL 和视图的操作	261
9.5 提高——触发器的使用	265
9.6 小结	268
第 10 章 【DBGrid】报表输出	269
10.1 相关知识点介绍	269
10.2 典型实例——基于工程数据集的【Rave】报表	274
10.3 起步——独立访问数据库的【Rave】报表	281
10.4 进阶——基于代码的【Rave】报表	283
10.5 提高——动态【DBGrid】报表	287
10.6 小结	295
第 11 章 开发实例——宾馆客房管理系统	297
11.1 系统分析	297
11.1.1 规划系统功能	297
11.1.2 明确用户需求	298
11.1.3 创建数据库结构	299
11.2 组织应用程序工程	307
11.2.1 封面、登录和用户管理单元	307
11.2.2 数据模块	308
11.2.3 主要单元和辅助单元	310
11.2.4 应用程序的主窗体	312
11.3 编写主要功能模块	313
11.3.1 客房类别设置模块	313
11.3.2 客房状态设置单元	314
11.3.3 住宿登记模块	315
11.3.4 换房退房模块	321
11.3.5 预订房间模块	322
11.3.6 宾馆消费记录模块	322
11.3.7 编辑客户登记信息模块	329
11.3.8 离店结算模块	330
11.3.9 收银统计模块	337
11.4 小结	346

第1章 Delphi 预备知识

Delphi 作为一种快速应用程序开发工具受到许多程序开发人员的青睐。在 Delphi 发展过程中有这样的特点：一方面 Delphi 在 Object Pascal 语言、集成开发环境和编程方式等方面保持着相对的稳定，这使得已经熟悉了前一版本的程序员能够轻松地转入新版本的使用；另一方面，由于国际互联网应用的迅速发展，网络应用开发的需求不断增长，Delphi 的每一个新版本在网络支持方面都有提高。除此之外，组件的数量也不断增加，作为缺省方式显示在组件栏上的组件也略有不同。

1.1 Delphi 7 开发工具简介

利用 C 语言进行过程序设计的程序员都知道，为了显示不同颜色和大小中文标题，需要使用汉字库并编写上百行的代码。使用非可视化开发语言编写 Windows 程序时，为了显示一个窗体或者一个按钮也颇费功夫。现在利用可视化开发工具能够轻松实现各种复杂的功能和设计出非常美观的程序界面。

作为一种深受欢迎的快速开发工具，Borland 公司推出的 Delphi 到今天已经历了 7 代产品，每一代产品都伴随着计算机软硬件系统的提高而升级。

1.1.1 Delphi 7 的可视化编程

编写 Windows 应用程序时，设计用户界面的工作量非常大。作为功能强大的可视化开发工具，Delphi 7 能够大大提高用户界面的设计效率。可视化编程不仅要求集成开发环境（Integrated Development Environment, IDE）提供良好的设计和编辑能力，而且还需要大量组件的支持。熟悉尽可能多的组件的使用方法，对提高开发效率特别有帮助。当然没有必要完全掌握所有 Delphi 7 组件的使用方法，在需要的时候，程序员通过联机帮助信息就能够有效地使用这些组件。

一、认识 Delphi 7 的 IDE

可视化开发工具的 IDE 是设计窗体、编辑代码、编译工程和测试程序的一个环境，为了实现以上功能，IDE 提供了丰富的支持以上功能的工具，并对这些工具进行了很好的连接，目的是减小程序开发的难度和提高程序开发的效率。

Delphi 7 的 IDE 组成主要包括主菜单、快捷按钮栏、组件栏、对象监视器、程序代码编辑器、程序浏览器和窗体等，如图 1-1 所示。

二、理解消息响应

所谓消息可以是对鼠标进行的操作，如单击鼠标、双击鼠标、右单击鼠标、拖曳鼠标。



消息也可以是按钮操作，还可以是系统的时钟计时到了某个时间，或者是某个函数传回了一个值等等。各个对象之间就是通过消息来互相通信、互相调用某些操作的。整个 Windows 下的程序都是基于消息驱动的。

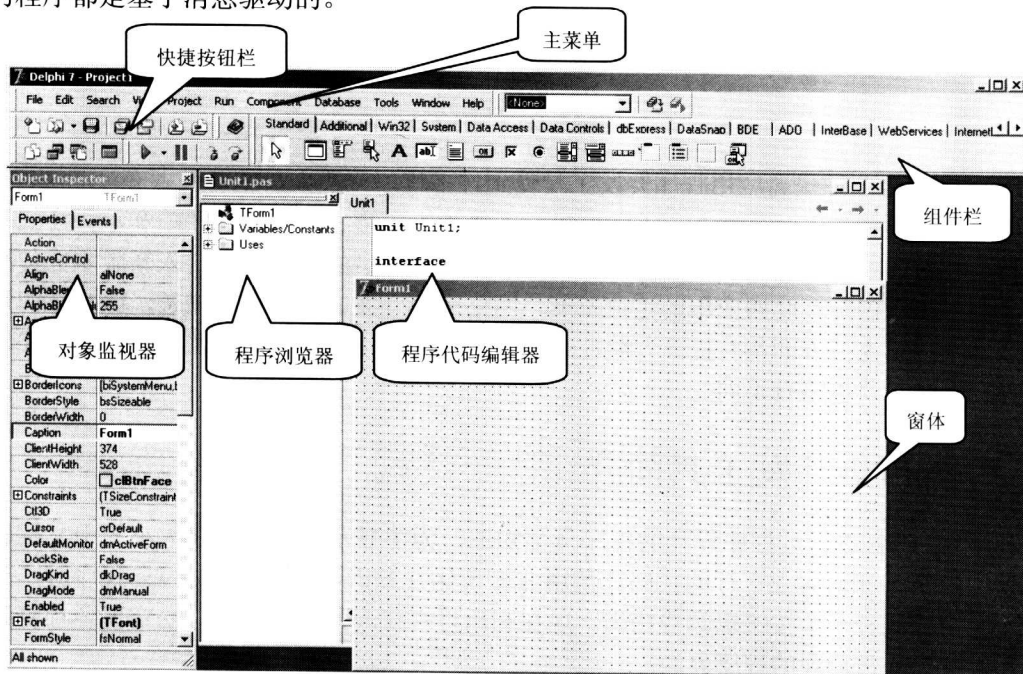


图1-1 Delphi 7 集成开发环境

三、组件、属性和事件

Delphi 结合了可视化和面向对象两大特点。由于它是可视化的，因此在设计时会提供一些可以直接看到、直接使用的组件，比如窗体、文本框、按钮、列表框和菜单等。这些组件具有属性和事件，属性就是这个组件的静态的特性，而事件则是当 Windows 中针对组件发生某些特定的消息时触发的动作。比如对于按钮这个组件来说，它的属性包括按钮的外观大小、颜色、在窗体中的位置、是否可见和快捷键等，它的事件包括单击鼠标左键、双击鼠标、按下键盘键等等。通过修改设置这些组件的属性和事件，就可以让这些组件呈现出需要的外观，完成需要的操作功能。

四、注意窗体设计的技巧

几乎每个 Windows 应用程序的开发都需要设计窗体。设计窗体的工作不仅仅在于设置窗体本身的属性，更多的工作是选择、布置和设置窗体上的组件，这是因为窗体是放置其他组件的容器，是 Windows 应用程序提供给用户进行操作的基本场所。

(1) 窗体环境的网格功能设置。

在缺省情况下，Delphi 窗体上有许多连续的小点将窗体划分成许多网格 (Grid)。当使用鼠标纵向移动组件位置时，组件的上下边缘总是自动捕捉小点构成的横线；当使用鼠标横向移动组件位置时，组件的左右边缘总是自动捕捉小点构成的竖线。这就是所谓的“Snap to grid”功能。关闭这种功能，可以通过改变窗体的“Snap to grid”设置实现，方法是选择【Tools】/【Environment Options】选项，打开【Environment Options】对话框，单击



【Designer】选项卡，取消对【Snap to grid】项的选择，如图 1-2 所示。该选项在缺省状态下是选中的。

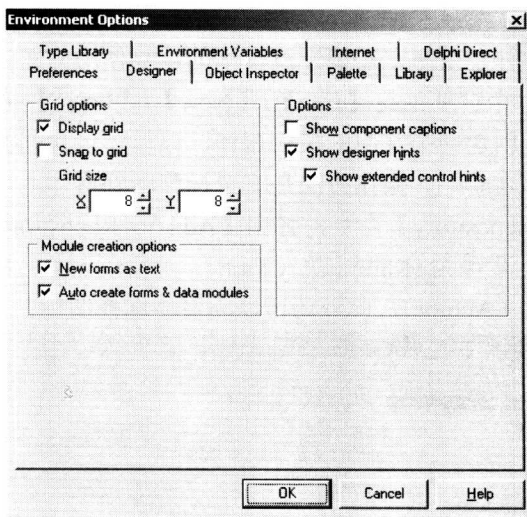


图1-2 取消【Snap to grid】选项

取消【Display grid】选项后，窗体上就不出现小点了，如图 1-3 所示。

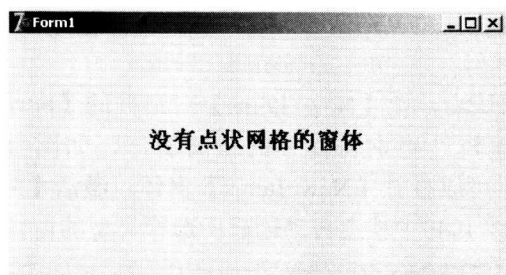


图1-3 没有点状网格的窗体

(2) 使用快捷键控制窗体上的组件。

通过 **Ctrl** 键+方向键可以精确控制组件的位置。

- **Ctrl**+**↑**: 向上移动当前组件。
- **Ctrl**+**←**: 向左移动当前组件。
- **Ctrl**+**↓**: 向下移动当前组件。
- **Ctrl**+**→**: 向右移动当前组件。

通过 **Ctrl** 键+**Shift** 键+方向键可以粗略调整组件的位置。

- **Ctrl**+**Shift**+**↑**: 向上移动当前组件。
- **Ctrl**+**Shift**+**←**: 向左移动当前组件。
- **Ctrl**+**Shift**+**↓**: 向下移动当前组件。
- **Ctrl**+**Shift**+**→**: 向右移动当前组件。

通过 **Shift** 键+方向键可以精确控制组件的尺寸。

- **Shift**+**↑**: 增加当前组件的高度。
- **Shift**+**←**: 减小当前组件的宽度。



- **Shift** + **↓**: 减小当前组件的高度。
- **Shift** + **→**: 增加当前组件的宽度。

(3) 使用窗体仓库。

为了提高应用程序的开发效率，需要注意窗体的重复使用问题。很多应用程序包含多个风格类似的窗体，这时每个窗体通过【File】/【New】/【Form】菜单命令来创建就不是明智之举。合理的作法是先创建一个共用的模板窗体，该窗体包含同类窗体拥有的公共特征，然后把该窗体添加到窗体仓库中。操作方法是鼠标移动到窗体上，单击鼠标右键，从弹出式菜单中选择【Add to Repository】命令，弹出【Add to Repository】对话框，如图 1-4 所示。通过该对话框可以设置窗体的标题（Title）、窗体描述（Description）、存放页面（Page）和窗体的创建者（Author），此外还可以为窗体选择图标。

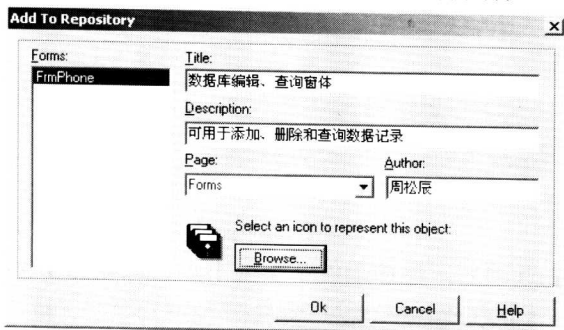


图1-4 添加到窗体仓库的设置

当单击 **Ok** 按钮确认后，在【New Items】对话框的【Forms】选项卡上就会新增一个窗体标题为“数据库编辑、查询窗体”的新的窗体，如图 1-5 所示。执行【File】/【New】/【Other】命令，可以打开【New Items】窗体，激活【Forms】页面，双击【数据库编辑、查询】窗体，则在 IDE 中就会自动创建所选择类型的窗体。

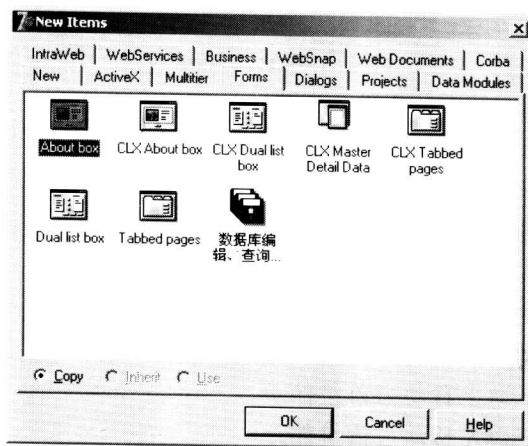


图1-5 【New Items】对话框的【Forms】选项卡

五、使用框架（Frame）

在一个工程中，如果各个窗体之间没有太大的共性，但各个窗体的某个局部都需要具有类似的功能，这时可选择使用框架。框架的使用方法可以参考【例 1-1】。

【例1-1】 框架的使用演示。

1. 建立新工程。

2. 新建框架。

选择 **【File】 / 【New】 / 【Frame】** 命令创建一个框架窗体，命名为“Frame1”，颜色选择“clSilver”。

3. 在窗体上添加一个按钮。按钮的 Onclick 事件代码如下：

```
procedure TForm2.Button1Click(Sender: TObject);  
begin  
    ShowMessage('框架使用演示');  
end;
```

4. 保存框架单元文件为“U_Frame.pas”。

5. 在窗体上添加框架组件，选择刚才创建的框架 **【Frame1】**，如图 1-6 所示。

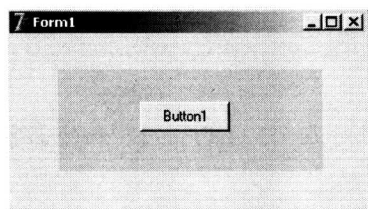


图1-6 框架使用例子

6. 编译运行程序。单击 **【Button1】** 按钮则会弹出信息“框架使用演示”。

六、使用菜单模板

Windows 应用程序一般都需要使用菜单，菜单项可以一项一项添加，但是如果添加的菜单是一种标准行为或者要添加的菜单在其他窗体中已经存在，这时使用菜单模板是更好的选择。使用菜单模板的操作是打开 **【菜单设计器】**，移动鼠标到菜单位置，单击鼠标右键，选择 **【Insert from Template】** 菜单命令，然后选择需要的菜单。列表中的菜单是 IDE 提供的标准菜单，如果以前的程序中用过的菜单最适合现在的工程，可以把原先的菜单保存到菜单模板中供以后使用。

1.1.2 Delphi 7 的面向对象编程

经常有参加程序员培训的学员问：究竟什么是面向对象的编程，面向对象的编程有什么好处？本节就来回答这个问题。

一、什么是面向对象

所谓“面向对象”就是把在程序开发过程中涉及到的事物看成对象或者对象中的一个部分。无论是 Delphi 开发环境直接提供的各种窗体、组件或者程序中自己定义的各种数据结构，都与对象相关。

对象从编程语言角度理解，是一种被称为类（Class）的数据类型，对象中可以包括数据成员、过程和函数，数据成员可以是各种数据类型的变量，当然包括类数据类型。这就意



味着对象中可以包含其他对象。例如窗体是一个类类型的对象，按钮也是一个类类型的对象，向窗体添加一个按钮的操作实际上就是声明按钮对象是窗体对象中的一个数据成员。面向对象具有以下3个基本特性。

(1) 封装。

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某个接口来访问它。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，这也是符合软件工程观点的。

【例1-2】 封装例子。

1. 建立一个新工程，在窗体上添加3个【Button】按钮，如图1-7所示。

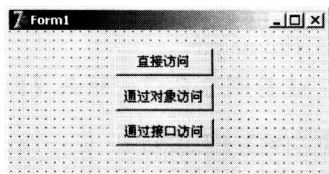


图1-7 封装演示例子

2. 在程序代码中声明新的对象“TMyObject”和外部字符串变量“Ext_Info”。程序代码如下：

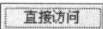
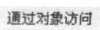
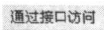
```
unit Unit1;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms,
    Dialogs, StdCtrls;
type
    TMyObject=class
    private
        Pri_Inf:string;
    public
        Pub_Info:string;
        procedure SetPri_Inf(s:string);
        function GetPri_Inf:string;
    end;
    TForm1 = class(TForm)
    .....
```

```
end;
var
    Form1: TForm1;
    Ext_Info:string; //声明一个对象外的字符串变量
implementation
```



```
{ $R *.dfm }  
function TMyObject.GetPri_Inf: string;  
begin  
    GetPri_Inf:=Pri_Inf;  
end;  
  
procedure TMyObject.SetPri_Inf(s:string);  
begin  
    Pri_Inf:=s;  
end;  
  
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Ext_Info:='向外部变量赋值成功!';  
    Caption:=Ext_Info;  
end;  
  
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    with TMyObject.Create do  
    begin  
        Pub_Info:='向对象内部变量赋值成功!';  
        Caption:=Pub_Info;  
    end;  
end;  
  
procedure TForm1.Button3Click(Sender: TObject);  
begin  
    with TMyObject.Create do  
    begin  
        SetPri_Inf('通过接口向对象内部变量赋值成功!');  
        Caption:=GetPri_Inf;  
    end;  
end;  
end.
```

3. 运行程序。

当单击  按钮时，窗体标题为“向外部变量赋值成功! ”；当单击  按钮时，窗体标题为“向对象内部变量赋值成功! ”；当单击  按钮时，窗体标题为“通过接口向对象内部变量赋值成功! ”。