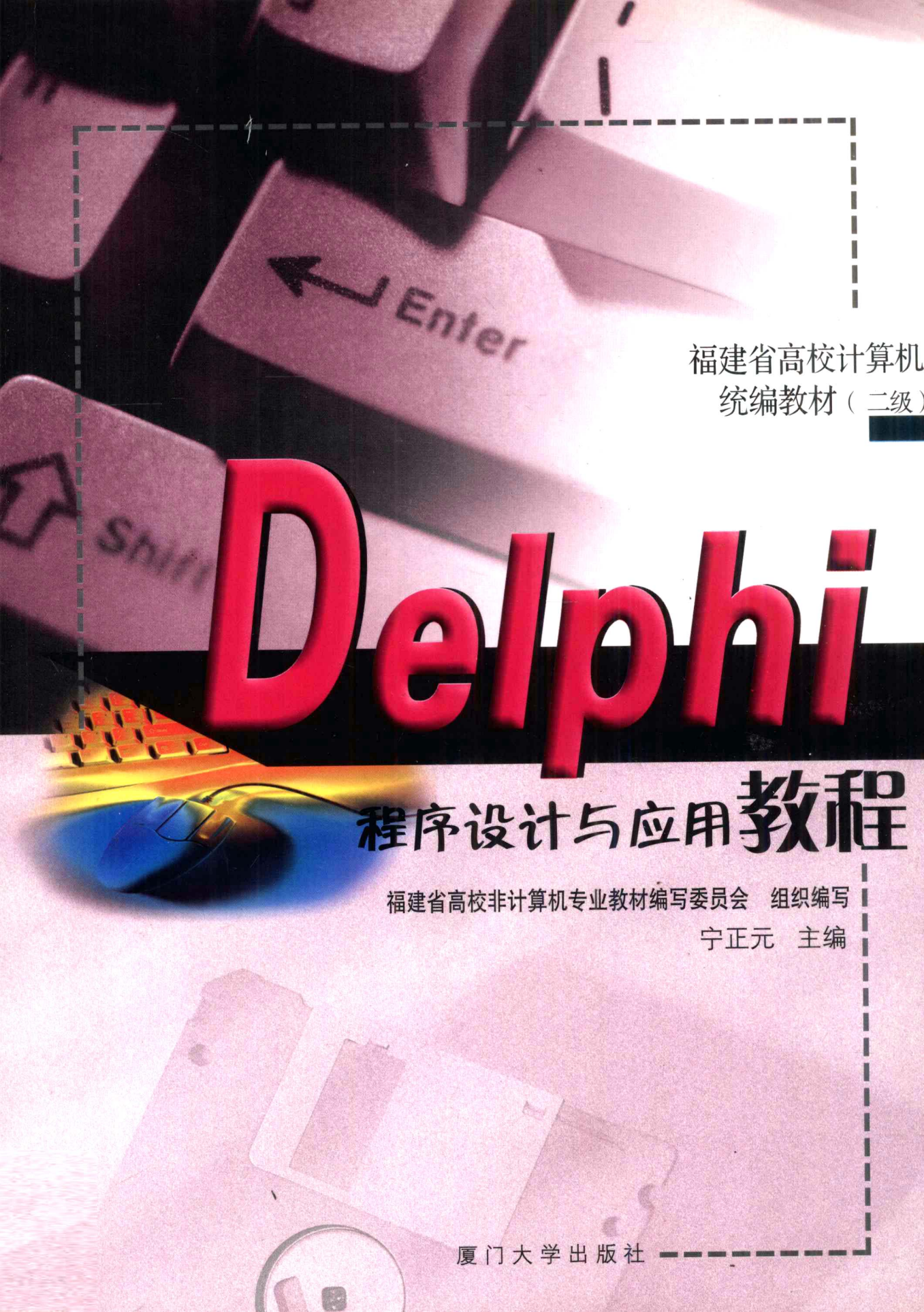




福建省高校计算机
统编教材（二级）

Delphi

程序设计与应用教程

福建省高校非计算机专业教材编写委员会 组织编写
宁正元 主编

厦门大学出版社

福建省高校计算机统编教材

Delphi 程序设计 与应用教程

主 编 宁正元

编 者 (以姓氏笔画为序)

刘雄恩 李更明

陈 琼 陈荔聪

林清波 黄思先

厦门大学出版社

图书在版编目(CIP)数据

Delphi 程序设计与应用教程/宁正元主编. —厦门:厦门大学出版社,2004.7
ISBN 7-5615-2311-4

I. D… II. 宁… III. 软件工具-程序设计-高等学校-教材 IV. TP311.56

中国版本图书馆 CIP 数据核字(2004)第 077253 号

厦门大学出版社出版发行

(地址:厦门大学 邮编:361005)

<http://www.xmupress.com>

xmup@public.xm.fj.cn

福建沙县方圆印刷有限公司印刷

2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

开本:787×1092 1/16 印张:18.75

字数:480 千字 印数:1-2000 册

定价:27.00 元

本书如有印装质量问题请直接寄承印厂调换

前言

随着 Windows 操作系统在微机中的广泛流行,一类将“可视化”手段和“面向对象程序设计”方法完美结合的程序设计软件,如 Delphi、C++Builder、JBuilder、Visual C++、Visual FoxPro 等,已逐渐成为 Windows 应用软件开发的主流平台。其中 Delphi 是最受广大程序员青睐和功能最强大的语言平台之一。Delphi 以具有强大表述力和完全面向对象为特征的 Object Pascal (Pascal 语言的最新版本)作为其编程语言,具备诸多卓越的性能,使其自 1995 年面世以来,在编程界迅速普及。在面向对象的可视化编程工具方面,迄今为止还没有与之相当的开发环境。

如今,Delphi 程序设计之类的书籍充斥书店,而相应的教材却很少。这种状况,一方面表明 Delphi 开发平台的应用已相当普及,另一方面表明在教学环节上却严重滞后。为适应计算机软件技术的发展和提高高校计算机应用教学水平,福建省高校计算机等级考试委员会和厦门大学出版社于一年前策划和组织了“Delphi 程序设计”课程教学和考试的专项研究。本书便是该项研究的成果之一,旨在为高校开设相应课程提供适用的教材,本书即是福建省高等学校计算机应用水平等级考试二级(Delphi 语言)教材。

Delphi 程序设计涉及的内容繁多,然而为满足在有限学时内的教学要求,并使学生掌握在 Delphi 平台上进行程序设计的基本概念、基本方法和应用技能,同时兼顾相应的 IT 认证考试课程的内容,由此形成本教程由浅入深、例程丰富、注重基础、突出方法的特点。

本教程由福建农林大学计算机与信息学院宁正元教授主编,负责全书内容的安排和统稿。第 1 章由福建农林大学林清波编写,第 2 章由福建师范大学陈荔聪和福建农林大学刘雄恩编写,第 3 章由福建农林大学陈琼编写,第 4 章由福建农林大学黄思先和林清波编写,第 5 章由林清波和刘雄恩编写,第 6 章由刘雄恩编写,第 7 章由厦门大学李更明编写。目录中标记“*”号的小节是超出考试大纲的内容,供感兴趣的读者阅读,不作为教学和考试的内容。

在教程编写过程中,福建省高校计算机等级考试委员会的专家们对教材的编写大纲进行了审定,并提出了修改意见;本教程由华侨大学陈维斌教授主审,福建师范大学姚志强老师也曾对部分稿件进行了认真审阅,并提出了宝贵的修改意见或建议;厦门大学出版社对本书的出版给予了大力支持。在此表示诚挚的谢意。

鉴于编者的学识和水平,书中疏漏和不当之处恳请读者指正。

主编 宁正元

2004 年 10 月 8 日

目 录

前言

第 1 章 Delphi 简介	1
1.1 认识 Delphi	1
1.2 第一个应用程序	2
1.2.1 应用程序界面设计	2
1.2.2 设定组件属性和编写事件处理程序	4
1.2.3 编译运行程序	5
1.2.4 保存程序文件	6
1.2.5 退出 Delphi	7
1.3 Delphi 文件的种类	7
1.4 Delphi 集成开发环境简介	8
本章小结	9
习题	9
第 2 章 Object Pascal 语言基础	11
2.1 词法符号	11
2.1.1 基本符号	13
2.1.2 保留字、标识符、标准标识符、指令字	13
2.2 常量与变量	16
2.2.1 常量	16
2.2.2 变量	18
2.2.3 类型常量	19
2.3 简单数据类型	20
2.3.1 有序数据类型	20
2.3.2 实数类型	28
2.3.3 日期时间类型	30
2.4 结构类型	32
2.4.1 集合类型	32
2.4.2 数组类型	33
2.4.3 字符串类型	35
2.4.4 记录类型	39
2.4.5 指针类型	42
*2.4.6 可变类型	43
2.5 表达式与运算符优先级	44
2.6 语句与流程控制	44
2.6.1 基本语句	44
2.6.2 分支语句	46

2.6.3 循环语句	49
2.7 过程和函数	53
2.7.1 过程和函数声明	54
2.7.2 过程和函数调用	56
2.7.3 参数传递	56
2.7.4 过程和函数重载	57
2.7.5 过程和函数的递归调用	58
2.8 程序与单元的结构	60
2.8.1 程序结构	60
2.8.2 单元结构	61
2.8.3 使用单元的一个示例	61
2.9 标识符的作用域	63
本章小结	64
习题	64
第 3 章 Delphi 中的面向对象	67
3.1 面向对象程序设计概述	67
3.2 类与对象	67
3.2.1 类与对象的概念	68
3.2.2 定义类	68
3.2.3 对象的使用	69
3.2.4 构造函数与析构函数	71
3.3 类成员的访问权限	73
3.3.1 封装	73
3.3.2 类成员的访问权限	73
3.4 继承	77
3.4.1 继承的概念及意义	77
3.4.2 派生类的定义及使用	79
3.4.3 派生类的构造函数和析构函数	82
3.5 多态性	85
3.5.1 多态性的概念	85
3.5.2 重载	86
3.5.3 虚方法	88
3.5.4 抽象方法与抽象类	90
3.5.5 类引用及类型信息运算符	91
3.6 属性与方法	93
3.6.1 属性	93
3.6.2 方法	94
本章小结	96
习题	96
第 4 章 运用组件编程	98

4.1 可视化组件库 VCL	98
4.2 窗体设计	98
4.2.1 窗体常见的属性、方法及事件	98
4.2.2 简单窗体设计	100
4.2.3 多窗体设计	101
4.2.4 MDI 多文档窗体	103
4.3 常用组件	105
4.3.1 Standard 页上的常用组件	106
4.3.2 Additional 页上的常用组件	123
4.3.3 Win32 页上的常用组件	125
4.4 其他组件	132
4.4.1 Timer 组件	132
4.4.2 Application 组件	134
4.4.3 Canvas 组件	136
4.4.4 Image 组件	139
4.5 输入与输出	140
4.5.1 输入、输出对话框	141
4.5.2 键盘	144
4.5.3 鼠标	146
*4.5.4 打印	151
本章小结	155
习题	155
第 5 章 程序调试和异常处理	157
5.1 程序调试	157
5.1.1 错误种类	157
5.1.2 使用断点	159
5.1.3 设置程序的执行方式	161
5.1.4 监视变量和相关数据的值	162
5.2 异常保护和处理	164
5.2.1 异常的概念及异常类	164
5.2.2 异常保护和处理	166
*5.2.3 自定义异常类及其应用	169
本章小结	174
习题	174
第 6 章 Windows 深入编程初步	175
6.1 动态链接库 (DLL)	175
6.1.1 什么是 DLL	175
6.1.2 DLL 项目文件	176
6.1.3 创建 DLL	177
6.1.4 DLL 中的初始化代码	179

6.1.5 调用 DLL	180
6.2 Windows API 及其函数	183
6.2.1 Windows API 函数分类	183
6.2.2 API 函数调用	184
6.3 消息处理	187
6.3.1 Windows 的消息机制	187
6.3.2 Delphi 的消息处理	189
6.3.3 发送消息	190
6.3.4 响应消息	191
6.4 多线程编程	196
6.4.1 进程与线程	196
6.4.2 TThread 类	197
6.4.3 创建线程类	199
6.4.4 线程的优先级与线程调度	200
6.4.5 线程的挂起、恢复和终止	201
6.4.6 多线程一例	201
*6.4.7 线程的协调	205
6.5 文件访问	206
6.5.1 Pascal 方式的文件访问	206
6.5.2 Windows API 方式的文件访问	211
本章小结	213
习题	213
第 7 章 数据库应用编程	216
7.1 Delphi 的数据库体系与支持工具	216
7.1.1 使用数据库	216
7.1.2 数据库应用程序的体系结构	217
7.1.3 数据库引擎 (BDE)	217
7.1.4 数据库桌面 (Database Desktop)	218
7.1.5 数据库浏览器 (Database Explorer)	222
7.2 数据集简介	223
7.2.1 数据集的打开和关闭	223
7.2.2 数据集的状态	224
7.2.3 数据集的浏览	225
7.2.4 搜索特定记录	225
7.2.5 数据集的修改	226
7.3 数据访问组件	228
7.3.1 字段对象	228
7.3.2 访问数据表 (TTable)	230
7.3.3 查询数据库 (TQuery)	232
7.3.4 查询语句 Select 简介	235

7.3.5 TBatchMove 组件	236
7.3.6 TDataSource 组件	237
7.4 数据控制组件	237
7.4.1 TDBGrid、TDBNavigator 组件及其应用	238
7.4.2 TDBText、TDBEdit、TDBMemo、TDBImage 组件及其应用	242
7.4.3 TDBListBox、TDBComboBox 组件及其应用	245
7.4.4 TDBCheckBox、TDBRadioGroup 组件及其应用	246
*7.5 ADO 技术	248
*7.5.1 设置 ODBC 数据源	248
*7.5.2 连接 Access 数据库	249
*7.5.3 ADO 常用组件	251
7.6 数据库应用程序实例	253
7.6.1 计算机考试应用程序的设计(BDE)	253
*7.6.2 图书管理应用程序的设计(ADO)	263
本章小结	274
习题	274
附录	278
附录一 术语缩写的中英文对照表	278
附录二 Delphi 常用过程和函数	279
附录三 Delphi 常见错误信息	284

第1章 Delphi 简介

1.1 认识 Delphi

Delphi 是美国著名软件厂商 Borland/Inprise 公司的主要产品之一，是一种基于面向对象的 Object Pascal 语言的可视化集成软件开发平台。和同类产品相比，Delphi 具有以下几个让每个使用者都能感觉到的卓越性能：

- Delphi 提供了高效率的可视化开发环境，其强大的 IDE 功能，可以替开发者自动产生许多代码，大大节省了开发时间。Delphi 是当今所推崇的快速应用程序开发(RAD)的最有力工具之一。
- Borland 公司的 Pascal 编译器，从 DOS 时代的 Turbo Pascal 至今，一直都以快速稳定著称。Delphi 的编译器继承了 Turbo Pascal 编译器的卓越性能，其编译速度在同类产品中是最快的。尤其难能可贵的是，Delphi 对其源代码快速稳定的编译，并没有影响其生成的目标代码的性能，Delphi 的生成代码在大小和运行性能上与在此方面有最好表现的 Visual C++ 完全相当。而后者在开发效率上，在易学性、易用性等方面都与 Delphi 相差甚远。
- Delphi 以 Object Pascal 为编程语言。Pascal 语言是专门为结构化编程而设计的，与其他语言相比，它更接近于自然语言，语法结构严谨，简单明了，没有多义性，便于写出结构严谨、风格优美的程序。Object Pascal 引入了面向对象的编程理念，完全具备了在 Windows 环境下编写各种复杂程序的能力，却又不改其易学易用的特点。这完全归功于 Delphi 设计者对其复杂功能恰到好处的封装，在语言的功能和复杂度之间找到了良好的平衡，使程序员可以把更多的时间和精力放在程序逻辑和算法设计上，而不至于陷入语法的泥潭之中。
- Delphi 在数据库和网络处理功能方面是同类产品中最强的。它提供了开发 Windows 或 Linux 数据库应用程序最迅速最简洁的方法，在数据库和网络方面，Delphi 也是最理想的软件开发平台。

作为开篇，此处不必一一列举 Delphi 的特点。总之，随着学习的逐步深入，读者将切身地领会到 Delphi 是一个完全集成化的、面向对象的、可视化的、能编制各种 Windows 程序的、功能强大而又易学易用的软件开发平台。

1.2 第一个应用程序

在简单了解 Delphi 卓越性能之后，将通过一个简单的应用程序来开始 Delphi 之旅。请读者打开计算机，先不必在意一些难懂的专业术语，只要按照以下步骤逐一完成各个操作，相信每位读者都会发现，原来 Delphi 的入门竟如此简单。

1.2.1 应用程序界面设计

第 1 步，启动 Delphi。

安装完 Delphi 7.0 后，在 Windows 环境中点击【开始】→【程序】→【Borland Delphi 7】→【Delphi 7】运行 Delphi，运行后将出现如图 1-1 所示窗口。

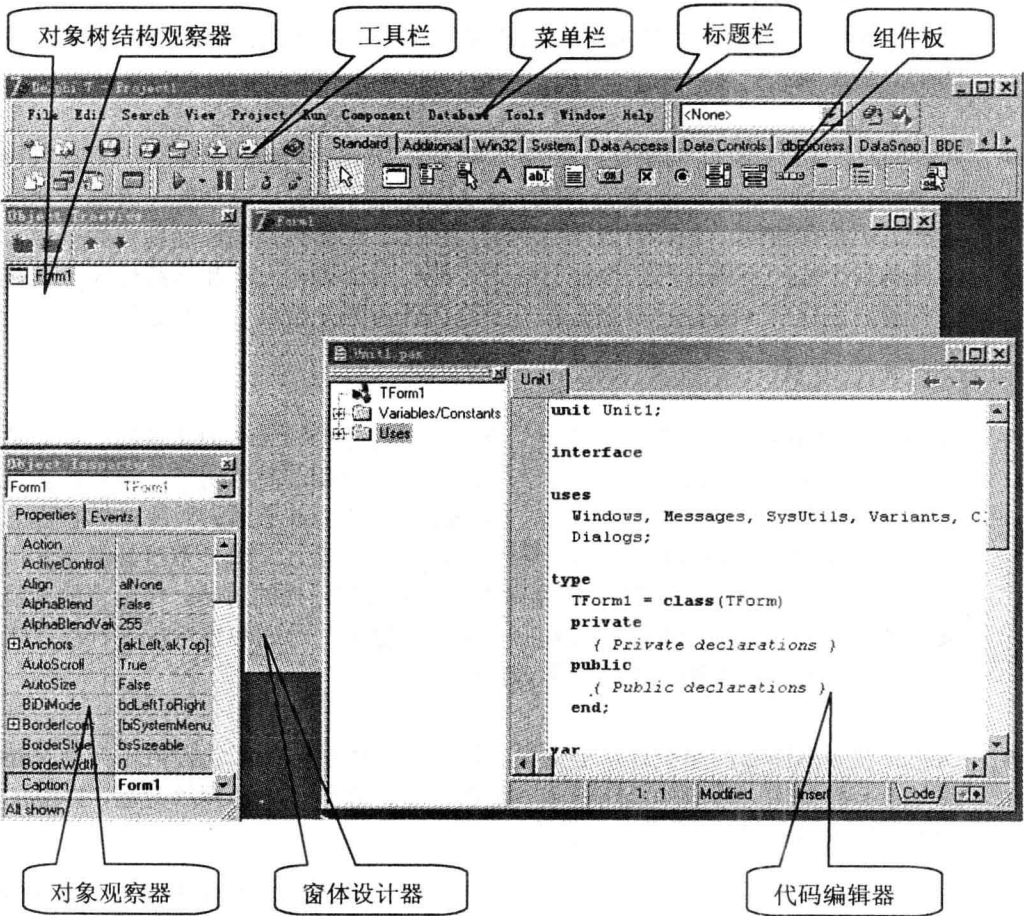


图 1-1 Delphi 的运行界面

第 2 步，选择组件。

在组件板中选择“Standard”页，用鼠标左键单击其中的 Button 组件，使该组件按钮陷

下，如图 1-2 所示。

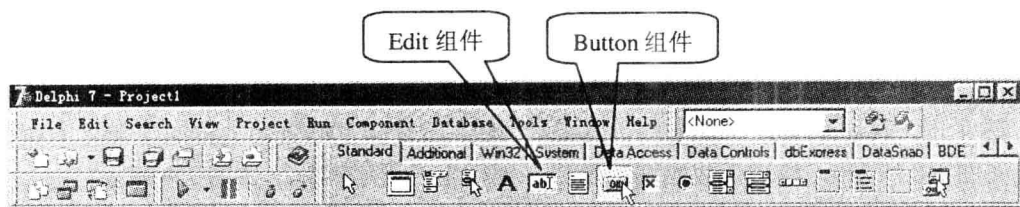


图 1-2 standard 页组件板

第 3 步，放置组件。

将鼠标指针移到 Form1 窗口中，单击左键，这时一个标有“Button1”的按钮便放到了 Form1 窗体上，可以用鼠标改变它的位置和大小，使之满足设计者的需要，如图 1-3 所示。

第 4 步，重复第 2、3 两步的工作。

在 Button1 按钮的右边再放一个 Button2 按钮，如图 1-4 所示。

第 5 步，在组件板找到 Edit 组件（见图 1-2），用第 2、3 两步的方法在 Form1 窗体上放一个编辑框组件，如图 1-5 所示。

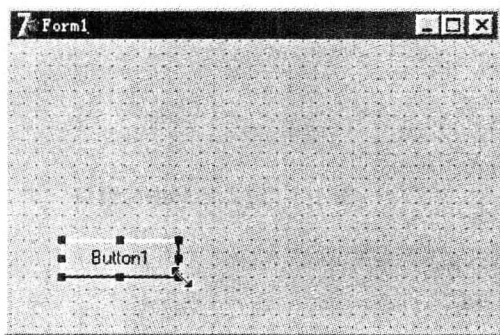


图 1-3 放置按钮 Button1

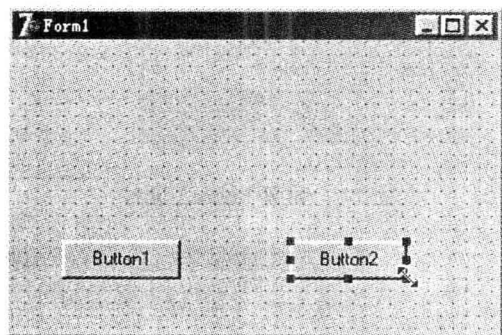


图 1-4 放置按钮 Button2

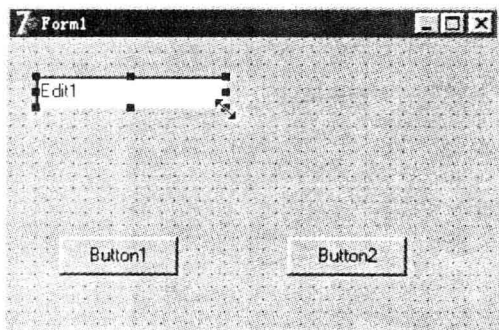


图 1-5 放置编辑框组件

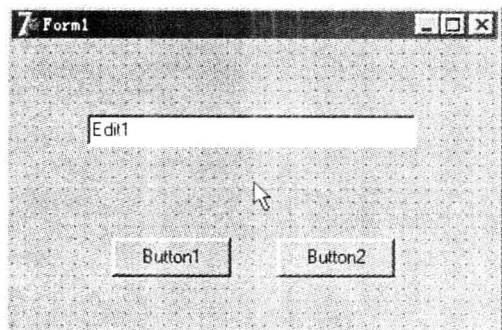


图 1-6 调整组件大小和位置

第 6 步，在 Form1 窗体中调整三个组件的大小和位置，使之排列整齐。这时，一个应用程序的雏形就已经形成了，如图 1-6 所示。接下去的工作就是让每一个组件都发挥作用。

1.2.2 设定组件属性和编写事件处理程序

第 7 步，在对象观察器（Object Inspector）上方的组合框中选择 Button1，在其下方的“Properties”页中找到“Caption”属性，然后把该属性的值“Button1”改为“确定”，如图 1-7 所示。

第 8 步，选择对象观察器窗口“Events”页，用鼠标左键双击 OnClick 事件右侧的输入域（如图 1-8 所示），将进入代码编辑窗口，同时出现程序框架，如图 1-9 所示。

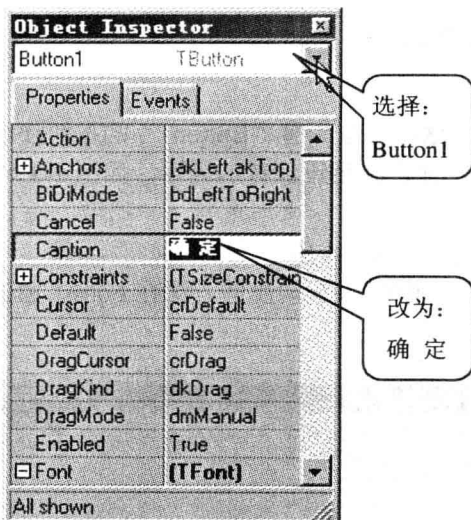


图 1-7 设置 Button1 属性

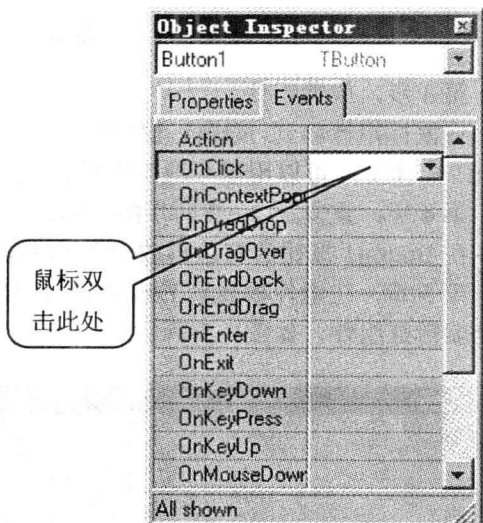


图 1-8 进入代码编辑窗口

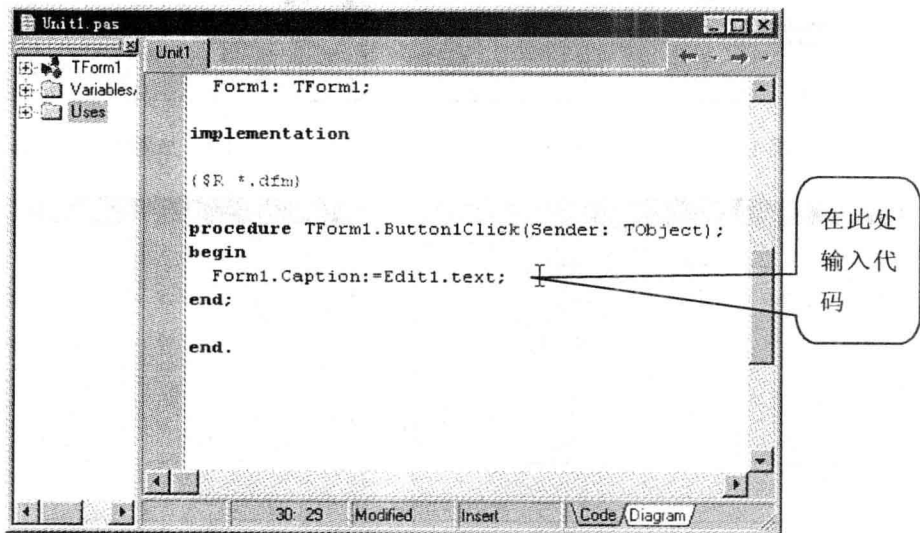


图 1-9 编写事件代码

在图 1-9 所示位置输入如下代码：

```
Form1.Caption:=Edit1.text;
```

第 9 步，在对象观察器窗口上方的组合框中选择 Button2，按第 6、7、8 步的方法，把

Button2 组件的 Caption 属性值改为“放弃”（见图 1-10），在 OnClick 事件代码中输入：Edit1.text:=“”；，如图 1-11 所示。

至此，所有程序设计的工作就都完成了，接下去的工作就是编译并运行该程序，体验一下编制程序的喜悦。

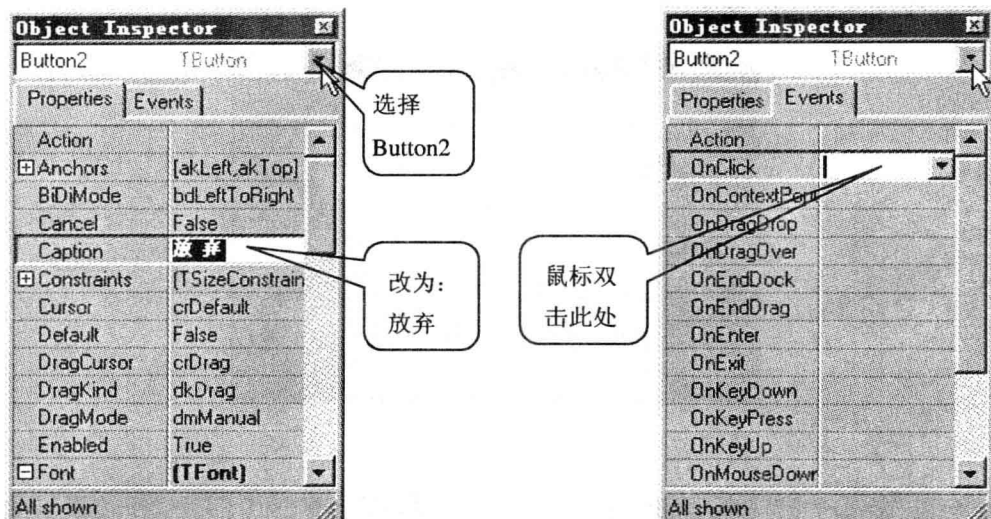


图 1-10 对象观察器

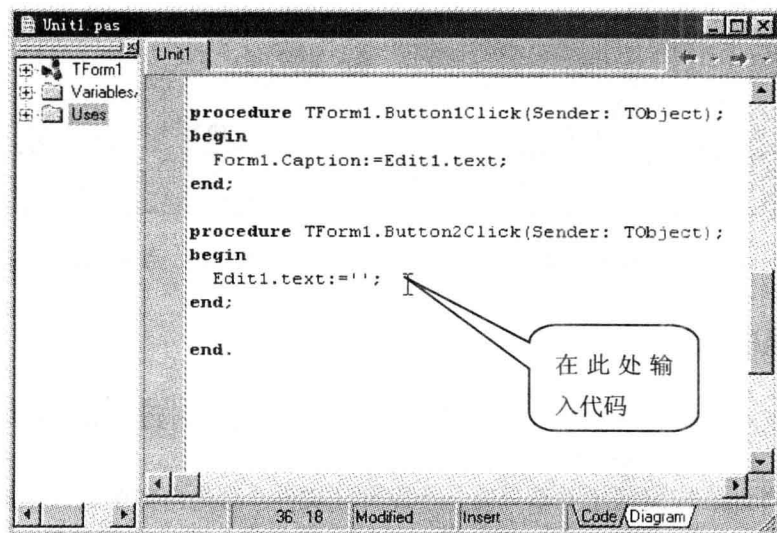


图 1-11 输入代码

1.2.3 编译运行程序

第 10 步，按下 F9 键，程序编译、运行，如果以上操作都没什么错误的话，将会看到如图 1-12 所示的运行结果。

在编辑框里输入：“这是我的第一个 Delphi 程序”，然后点击“确定”按钮，请注意这时

程序窗口标题栏的变化,如图 1-13 所示。

再点击“放弃”按钮,注意编辑框里的变化,如图 1-14 所示。

好了,到此大家会发现,原来用 Delphi 编写一个 Windows 应用程序就这么容易(虽然这个程序并没有什么实用价值)。随着往后学习的深入,相信大家都能用 Delphi 很轻松地编写出各种实用程序。这正是 Borland 公司 Delphi 设计者们希望看到的。

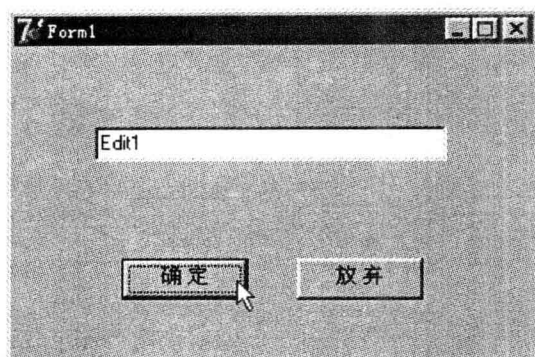


图 1-12 运行结果

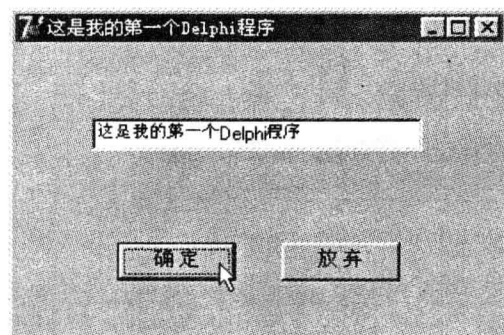


图 1-13 点击“确定”按钮

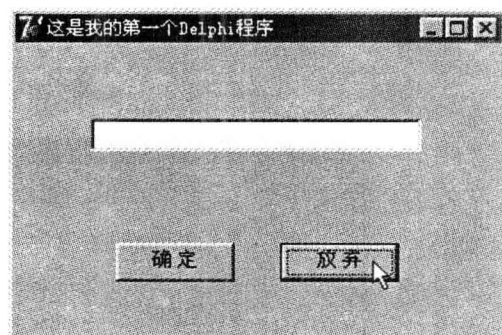


图 1-14 点击“放弃”按钮

1.2.4 保存程序文件

点击应用程序窗口右上角的“×”按钮,关闭应用程序,回到 Delphi 程序设计窗口,接下来的工作就是保存这个编写好的程序。

由于一个 Delphi 应用程序并不只有一个源程序文件,而是由多个文件组成的,所以为了便于将来对源程序进行维护、管理,初学者一开始就要养成一个良好的习惯:对每一个 Delphi 应用程序建立一个独立的文件夹,把与程序相关的所有文件都保存到这个文件夹中。

假设已在 D 盘建立了一个名为 Demo 的文件夹,选择执行菜单【File】→【Save All】菜单项,这时会弹出一个名为“Save Unit1 As”的对话框,在对话框中把盘符改为 D:,文件夹改为 Demo,文件名不变(还是 Unit1.pas),如图 1-15 所示。

然后点击“保存”按钮,存完之后,会另外弹出一个“Save Project1 As”对话框,在对话框中,磁盘和文件夹已默认为 D:\Demo,不必再改了,文件名默认是 Project1.dpr,把它改为 MyFirstPro.dpr,如图 1-16 所示。

点击“保存”后，按 Ctrl+F9，重新编译程序（不运行）。

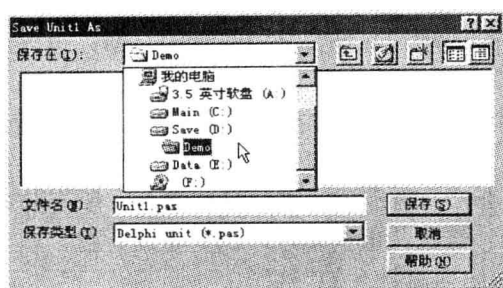


图 1-15 保存 Unit 界面



图 1-16 保存 Project 界面

当然，保存程序文件的操作也可以安排在编译程序之前。或许，养成这样的做法是一个良好的习惯。

1.2.5 退出 Delphi

选择执行【File】→【Exit】菜单项，退出 Delphi。用资源管理器打开 D:\Demo 文件夹，这时大家会看到，刚才设计和编写的程序是由多个文件组成的（见图 1-17）。其中 MyFirstPro.exe 是 Delphi 编译出的可执行程序，它可以复制到任何地方单独运行。用鼠标双击它，看看会有什么结果。还可以在桌面上为它建立快捷方式，通过双击它的快捷方式来运行它。总之，这是一个用 Delphi 编写出的真正的 Windows 应用程序！

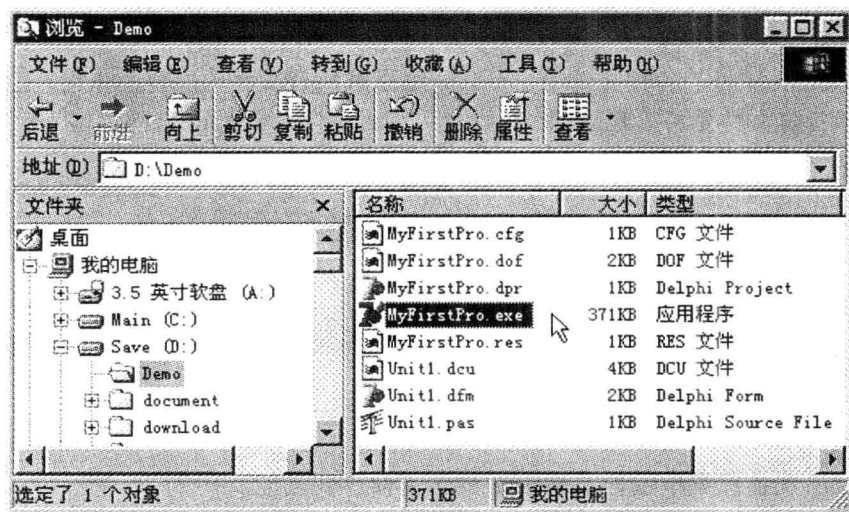


图 1-17 Windows 资源管理器

1.3 Delphi 文件的种类

在上一节介绍过，一个 Delphi 源程序是由多个文件组成的，这些文件组合起来称为一个

项目。一个编制中的 Delphi 应用程序就是一个项目，每个项目由以下几种文件组成。

1. 项目文件 (*.dpr)

每一个 Delphi 应用程序都有一个扩展名为 dpr 的项目文件（按照通常的习惯，也可以把它称为主程序文件），这个文件是由系统自动建立的，一般情况下不必修改它的内容。

在 Delphi 中用【Open Project】操作打开一个项目文件，就意味着装入整个项目。例如：要对 1.2 节所创建的项目进行修改，在 Delphi 的【File】菜单中，选择【Open Project】，在弹出的文件选择窗口中选择“D:\Demo\MyFirstPro.dpr”，确定后，该项目被装入，所有与此项目相关的文件就被自动打开，可以对它进行编辑、修改，重新编译，而后重新存盘。

在【File】菜单中，选择【Close All】可以关闭当前打开的项目。

2. 单元文件 (*.pas)

编程者在 Delphi 中编写的程序代码，绝大多数被保存在这种文件中，单元文件是 Delphi 程序实现的主要组成部分。如 1.2 节中的例子，程序代码就写在 Unit1.pas 文件中。

3. 窗体文件 (*.dfm)

窗体文件中保存了关于设计窗体的各项属性，以及在里面定义的各种组件的属性。

4. 配置文件 (*.dof, *.cfg)

这种类型的文件保存了用户对 Delphi 系统的设置。

5. 中间编译单元 (*.dcu)

Delphi 对一个项目进行编译时，是先把该项目所有单元文件 (*.pas) 编译成 dcu 文件，再把这些 dcu 文件链接成可执行程序。

6. 备份文件 (*.~*)

在 Delphi 中，被修改过的文件重新存盘后，修改前的内容将被自动保存成备份文件。备份文件的名称与原文件不同的地方就是扩展名前多一个“~”，如 Unit1.pas 的备份文件是 Unit1.~pas。

在一个 Delphi 项目的各种文件中，不必详细了解所有文件的内容及作用，但要知道：只有 dcu 文件和备份文件是可以删除的，其他的文件还是交给 Delphi 自动管理，不可随意删除。

1.4 Delphi 集成开发环境简介

集成开发环境简称 IDE，Delphi 的 IDE 由以下几个部分组成，见图 1-1。

1. 标题栏

标题栏显示了当前项目的名称。

2. 菜单栏

菜单栏及所属各级菜单包含了 Delphi 的全部命令和功能。

3. 工具栏

工具栏中放置了菜单栏中某些主要功能的快捷按钮（见图 1-18）。

4. 组件板

组件板包含了 Delphi 的可视化组件，这些组件是编写 Windows 应用程序的基本素材，例如，按钮、列表框、编辑框等等。

5. 对象观察器 (Object Inspector)

对象观察器的功能是设置对象的属性，创建并管理对象的事件处理过程，它是应用程序