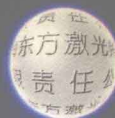


计算机应用软件上机实践教程系列丛书



Delphi 5

即学即用 **上机实践教程**



电子工业出版社

Delphi 5 上机实践教程

策划：何学仪 刘 钢

主编：邹红艳 吴江海

编者：邹红艳 包爱民

东方激光工作室

电 子 工 业 出 版 社

Delphi 5 上机实践教程

(光盘附书)

策 划	何学仪
编 著	东方激光工作室
责任编辑	董 娅 苏宁萍
出版发行	电子工业出版社 电子出版部
经 销	各地新华书店
开 本	1092×787mm 1/16
印 张	17.5
标准盘号	ISBN 7-900074-10-4/TP·11

前 言

Delphi 5 是全新的可视化编程工具, 它为我们提供了一种方便、快捷的 Windows 应用程序开发环境。它使用 Microsoft Windows 图形用户界面的许多先进特性和设计思想, 采用可重复利用的完整的面向对象程序语言(Object-Oriented Language)、是当今世界上最快的编译器以及最为领先的数据库技术。对于广大的程序开发人员来讲, 使用 Delphi 开发应用软件, 无疑会大大地提高编程效率, 而且随着应用的深入, 您将会发现编程不再是枯燥无味的工作——Delphi 的每一个设计细节, 都将带给您一份欣喜。

本书内容丰富, 实用性强且具指导性。采用独特的格式和写作方法, 特别适合于教学和上机实践。抛弃了大量不必要的繁文缛节式的理论说教, 通过简明、精确的操作及步骤, 在提高操作能力的同时, 了解其相关概念和应用法则, 达到即学即用的效果。

本书分为两个部分, 主要介绍 Delphi 5 的基本使用。

第一部分由十三个章节组成, 包括 Delphi 5 入门、窗体设计、对象共同属性与事件、基本组件的使用、简单程序设计、对话框、菜单、工具栏以及多文档的设计、组件的设计、文件管理、数据库程序设计、报表制作与打印、Internet 编程及安装程序的制作。这些内容覆盖了学习 Delphi5 所涉及的最主要以及最基本的内容。每章由以下四个单元组成:

【要点概述】提纲挈领地讲述了上机实践所涉及的相关操作、系统概念和基本原理。

【实践流程】以流程图的形式简洁明了的列出上机实践部分的提纲。

【上机实践】由若干个上机实验组成, 每个实验都列出了比较具体的操作步骤, 引导读者通过实践快速走近 Delphi5。

【思考与练习】由若干个练习题组成, 要求读者自己完成。一般来说, 只要掌握了上机实践中的有关操作, 这些题目也就不难独立完成。

本书第二部分是思考与练习中所有题目的解答, 供读者在完成习题有困难时参考。

总览全书, 作者从实用的角度出发, 由浅入深地介绍了 Delphi 5 的有关编程技术。从每一个程序的界面设计、组件的插入到组件属性的设置、代码的输入都做了详细的介绍和分析, 无论您是 Delphi5 的初学者还是 Delphi 程序员, 相信这本书都能给您带来很大的方便, 并使您的编程水平在实际操作和认真学习中得到可观的提高。本书不但是高等院校相关专业师生教学、自学用书, 而且也是从事 Delphi 5 开发和应用的广大编程人员的开发指导用书和社会相关领域培训班的教材。

本书由何学仪、刘钢策划, 邹红艳、吴江海主编, 《东方激光工作室》的倪君、吴东亮及同济大学继续教育学院的包爱民编写初稿。由于编者水平有限, 不足之处在所难免, 恳请读者对误漏之处批评指正。

编 者

2001 年 6 月

目 录

第一章 Delphi 5 入门	
1.1 Delphi 5 的启动与退出	3
1.2 Delphi 5 的窗口组成及其基本操作	4
1.3 简单实例	9
1.4 调试器的操作	12
第二章 窗体设计	
2.1 建立窗体	15
2.2 添加及编辑组件	17
2.3 运行时改变窗体	20
2.4 窗体模板	21
2.5 框架	23
第三章 Delphi 5 对象共同属性与事件	
3.1 对象的共同属性	27
3.2 Delphi 5 常用对象事件	30
第四章 Delphi 5 基本组件的使用	
4.1 基本控件	35
4.2 图形组件	43
4.3 计时器组件	45
4.4 多媒体组件	47
4.5 分页组件	49
4.6 动画组件	52
4.7 应用程序举例	52
第五章 简单程序设计	
5.1 顺序结构	59
5.2 判断结构	60
5.3 循环结构	63
5.4 函数与过程	65
5.5 应用程序举例	67
第六章 对话框的设计	
6.1 标准对话框	72
6.2 Dialog 组件的使用	75
6.3 用户自定义对话框	79
第七章 菜单、工具栏及多文档界面(MDI) 的设计	
7.1 主菜单设计	86

7.2	快捷菜单设计	92
7.3	工具栏设计	93
7.4	位图菜单设计	98
7.5	多文档界面(MDI)设计	99
第八章 组件的设计		
8.1	创建窗口类组件	111
8.2	建立图形类组件	116
第九章 磁盘与文件管理		
9.1	文件管理	126
9.2	磁盘管理	131
9.3	目录管理	134
9.4	注册表管理	135
第十章 数据库的创建		
10.1	数据库的建立	141
10.2	建立数据库别名	144
10.3	安排窗体组件	146
10.4	数据组件的绑定	147
10.5	窗体编程	148
10.6	数据库 SQL 查询	151
10.7	数据集命令操作	154
10.8	数据库应用程序举例	158
第十一章 报表制作及打印		
11.1	快速建立报表	165
11.2	数据库报表设计	167
11.3	非数据库报表设计	169
第十二章 Internet 编程		
12.1	制作超级链接	176
12.2	开发 Web 浏览器	177
12.3	动态制作 Web 页	181
12.4	电子邮件客户程序	183
12.5	制作 CGI 程序	188
第十三章 制作安装程序		
13.1	制作安装程序	195
13.2	测试安装程序	203
思考与练习参考解答		205

第一章 Delphi 5 入门



要点概述

Delphi 5 是 Inprise 公司推出的全新可视化编程工具。与以前的版本相比, Delphi 5 提供了更加完善的可视化组件(VCL) 和使用更加方便的快速开发工具。

在开始学习 Delphi 5 之前, 先了解一下 Delphi 5 的基本工作环境(见图 1-3) 及相关概念。

1. 主窗口

Delphi 5 的主窗口实际上是整个 Delphi 开发环境的核心, 主要由菜单栏、工具栏和组件面板组成。

2. 组件面板与工具栏

组件面板存放着 Delphi 提供的 VCL(Visual Component Library) 组件, 由可视化组件与非可视化组件组成。可视化组件包括应用程序中能看到的对话框、按钮、标签等; 非可视化组件是在执行时看不到的程序, 如定时器、动态交换与链接(DDE) 的服务器等。

工具栏和一般的应用程序工具栏相同, 用于快速的执行某些操作, 以节省到菜单中选择命令执行的时间。

3. 对象监视器

对象监视器(Object Inspector) 可用来查看和设置对象的功能。其中对象选择器是用于选定查看的对象; “属性”(Properties) 和 “事件”(Events) 选项卡分别用于设置对象的显示方式(如大小、标题、颜色及字体等) 和操作过程(即添加各类事件的响应程序)。

4. 代码编辑器

代码编辑器可供用户书写 Delphi 5 应用程序的源代码, 即编辑单元文件。在其左边有一个 Class Explorer 窗口, 它以树状结构显示程序中的所有方法和对象, 方便用户对源程序代码进行编辑和浏览。

5. 项目管理器

在设计应用程序时, 一个项目中可能包含多个窗体或单元。项目管理器可以同时管理多个项目文件, 通过树状结构及路径, 可以很清楚地了解目前正在处理的是哪一个 Project、Unit 以及相应的 Form。

6. 图像编辑器

图像编辑器(Image Editor) 是 Delphi 5 的一种建立及修改图像的工具, 可将结果保存成文件供窗体使用, 其界面类似于画笔。

7. 工程

工程在 Delphi 5 中扮演着组成程序或是动态链接函数库(DLL, Dynamic Link Library) 集合的角色。工程的内容有些是 Delphi 5 自动产生的, 有些可能是用户自己加入的过程

(Procedure) 或是函数(Function)。下面简要介绍一下 Delphi 5 工程中的各类文件:

- *.PAS: 每个窗体会对应生成一个.PAS 文件, 但每个.PAS 文件不一定会有对应的窗体, 这种文件通常用来记录 Object Pascal 程序代码的内容, 包括窗体事件处理过程和源代码。

- *.DPR: Delphi 5 的项目文件(Project File), 该文件列出了所有组成工程的单元文件。这个文件包含了 Delphi 5 的程序执行顺序, 并表明窗体文件或程序文件的实际路径位置。

- *.DFM: Delphi 5 将每个窗体用一个文件来存储其内容与设置, 这个文件说明了该窗体中使用了哪些组件及组件的属性等信息。

- *.EXE: 项目工程被编译后所生成的可执行文件。

- *.RES: 这个文件是 Delphi 5 用来存储工程所使用的资源信息, 它包含工程所用到的图标(Icon) 和图形(Bitmap) 等资源, 其文件名通常与工程名相同。

- *.CFG: 这个文件用来存储工程的配置信息, 其文件名通常与工程名相同。

- *.DOF: 这个文件用来存储工程所用到的各种编译器的设置, 文件名通常与工程名相同。

- *.~??: 当各类文件有过修改, 系统会自动加以备份, 它即是个备份文件。

- *.DCU: 经过编译.PAS 文件后产生的目标代码, 称为 Delphi Compiler Unit。

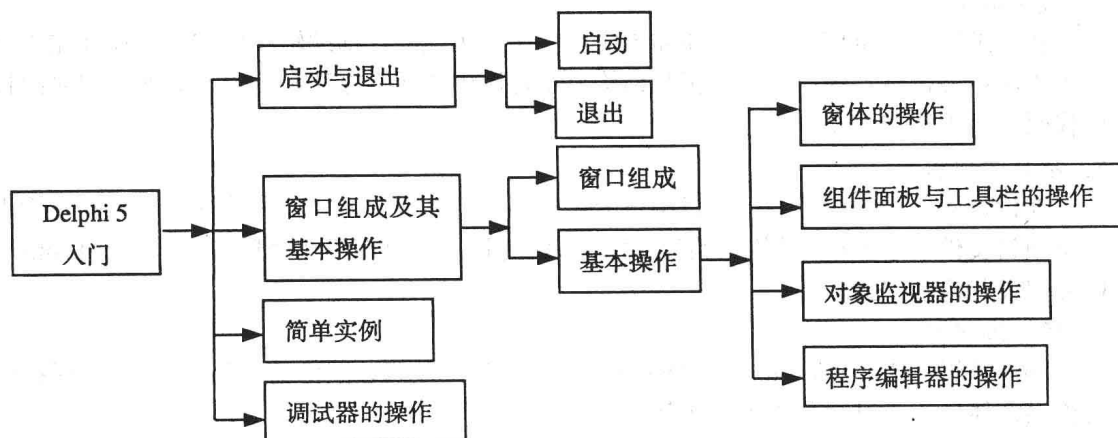
- *.DSK: 存放工程在退出 Delphi 时的各个窗口位置和状态的信息, 以便在重新打开时恢复原状。

8. 调试器

调试器可帮助用户跟踪并调试自己编写的代码, 能方便地查找和修改错误。



实践流程



上机实践

1.1 Delphi 5 的启动与退出

一、启动

【实验】

启动 Delphi 5。

【指导步骤】

方法一：

- (1) 单击 Windows 任务栏中的“开始”按钮。
- (2) 执行“程序→Borland Delphi 5→Delphi 5”菜单命令，如图 1-1 所示。



图 1-1

- (3) 启动 Delphi 5 应用程序，其应用程序启动界面如图 1-2 所示。

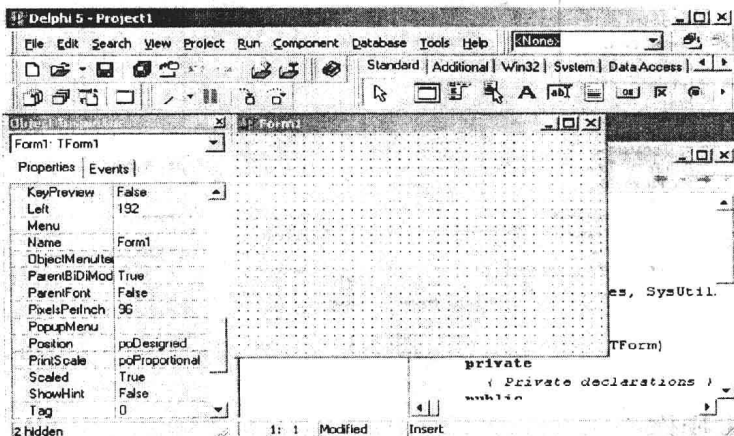


图 1-2

方法二：

- (1) 在桌面上创建一个 Delphi 5 的应用程序快捷图标。
- (2) 双击桌面上的 Delphi 5 应用程序快捷图标，即可启动该应用程序。

二、退出

通常情况下,若关闭一个修改过而未保存过的程序文件时,系统会给出提示,询问是否要保存修改后的程序文件;若是关闭一个用户新建的程序文件且需存盘时,系统会要求用户输入项目(Project) 文件名以及单元(Unit) 文件名。

【实验】

退出 Delphi 5。

【指导步骤】

三种退出方法:

- (1) 选择“File”菜单中的“Exit”命令。
- (2) 单击 Delphi 5 窗口最右上角的关闭按钮。
- (3) 直接按“Alt+F4”键,即可退出 Delphi 5。

1.2 Delphi 5 的窗口组成及其基本操作

一、窗口组成

【实验】

了解 Delphi 5 应用程序窗口,熟悉窗口的组成。

【指导步骤】

- (1) 启动 Delphi 5 应用程序。
- (2) 熟悉其组成及各部分的名称,如图 1-3 所示。

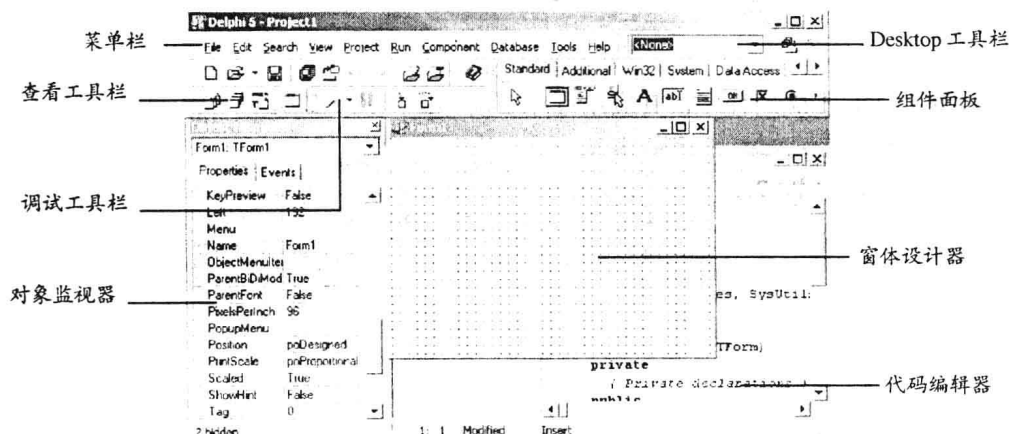


图 1-3

二、窗口的基本操作

1. 窗体操作

窗体(Form) 是在设计 Delphi 5 应用程序时最主要的一部分,它就像一块画板一样,可以把组件一项一项地加上,并调节其位置,以完成应用程序的窗体设计。

【实验 1】

创建两个窗体 Form1 和 Form2,适当调节窗体的大小及位置,并设置 Form2 为主窗体。

【指导步骤】

(1) 启动 Delphi 5, 选择“File”菜单中的“New Application”命令, 开始一个新的应用程序。

(2) 新建应用程序时, 默认窗体名为“Form1”。

(3) 调节窗体位置, 将鼠标置于窗体的标题栏处, 按住鼠标左键, 拖动窗体到合适位置然后松开鼠标即可。

(4) 调整窗体的大小, 将鼠标置于窗体的左下或右下边界处, 此时光标变成双向箭头, 按住鼠标拖动就可改变窗体的宽度或高度。

(5) 选择“File”菜单中的“New Form”命令或单击工具栏中的“New Form”按钮 , 添加新窗体(Form2), 效果如图 1-4 所示。

(6) 设置主窗体。

① 选择“Project”菜单下的“Options”命令或按“Ctrl+Shift+F11”键, 出现 Project Options 对话框。选择“Forms”选项卡, 界面如图 1-5 所示。

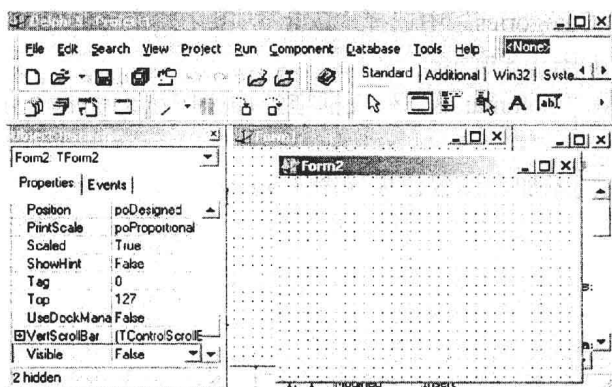


图 1-4

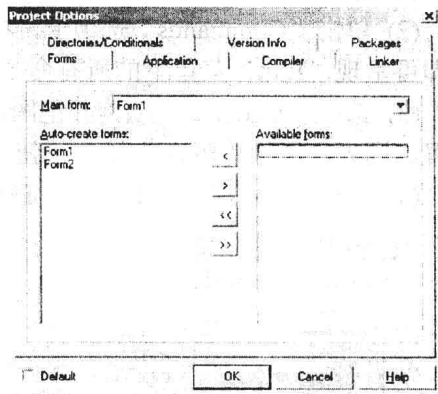


图 1-5

② 单击“Main form”栏右侧的下拉按钮, 选择“Form2”项, 再单击“OK”按钮, 使 Form2 成为主窗体, Form1 为子窗体。

【实验 2】

了解单元与窗体之间的切换。

【指导步骤】

在设计窗体与编写程序代码时, 往往需要将单元与窗体进行反复的切换。切换步骤如下:

单击工具栏中的“Toggle Form/Unit”按钮  或选择“View”菜单中的“Toggle Form/Unit”命令(也可按 F12 键), 可实现单元与窗体之间的相互切换。

2. 组件面板与工具栏操作

组件面板与工具栏包含在 Delphi 5 主窗口中, Delphi 5 主窗口如图 1-6 所示。

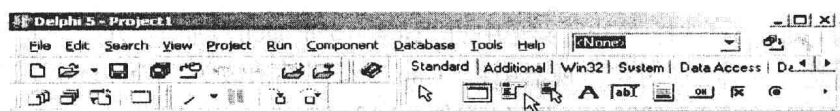


图 1-6

【实验】

向工具栏添加“Undelete”、“Redo”快速按钮。

【指导步骤】

提示：快速按钮是菜单功能的快捷方式，各种快速按钮图标能直观地表现其执行动作。在 Delphi 5 默认情况下，有 16 种快速按钮，如图 1-7 所示。

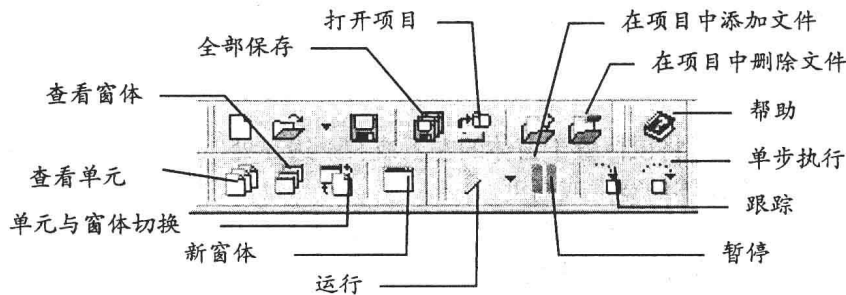


图 1-7

添加快速按钮操作如下：

- (1) 选择“View”菜单下“Toolbars”中的“Customize”命令，出现 Customize 对话框。
- (2) 单击“Commands”选项卡，在“Categories”中选择“Edit”项，在“Commands”中即可呈现“Edit”菜单下的所有命令，如图 1-8 所示。
- (3) 选中“Undelete”项，并把图标从“Commands”列表框拖到加速按钮组中的任意位置。同理，添加“Redo”快速按钮。
- (4) 单击“Close”按钮完成快捷按钮的添加，设置效果如图 1-9 所示。

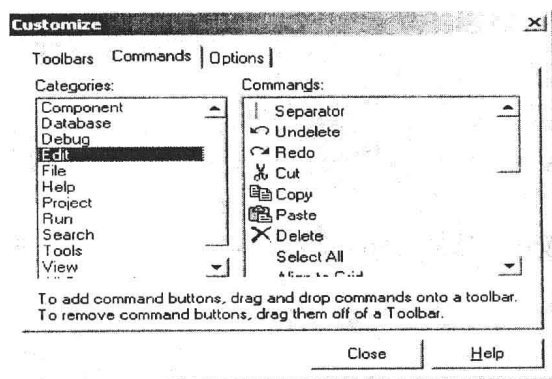


图 1-8



图 1-9

3. 对象监视器操作

【实验】

利用对象监视器设置窗体：标题为“Welcome”，高为“280”，宽为“314”；同时向窗体添加一个 Label 组件和一个 Button 组件，并设置相关属性。

【指导步骤】

- (1) 启动 Delphi 5，选择“File”菜单中的“New Application”命令，开始一个新应用程序。
- (2) 单击选中窗体，在对象监视器中将“Caption”属性值改为“Welcome”(如图 1-10)，同时设置“Height”值为“280”，“Width”值为“314”。
- (3) 在组件面板的【Standard】选项卡中，用鼠标单击 Label 组件 ，然后将鼠标移到窗体中，再单击鼠标左键，该组件(Label1) 即被添加到窗体上了。同理添加 Button 组件 。

说明:

- ① 若要将组件添加到窗体的中心位置上, 则双击组件面板上的组件图标按钮, 即可实现。
 - ② 若要连续添加同一种组件, 则可按下 Shift 键不放, 然后在组件面板上选择该组件, 此时所选择的组件将有蓝色边框标识, 同时对象选择按钮  将弹起; 将鼠标移动到窗体上要添加组件的第一个位置, 单击鼠标即可添加一个组件, 接下来继续移动鼠标, 在要添加该组件的第二个位置上, 再次单击鼠标, 便又添加了一个同种组件。如此反复操作, 即可添加多个同种组件。
 - ③ 如想取消连续添加的操作, 只需用鼠标双击对象选择按钮 , 使之恢复选中状态。
 - (4) 选中窗体中的 Label1 组件, 然后在对象监视器中将其“Caption”属性值改为“欢迎使用 Delphi 5.0”; 双击“Font”属性值右边的省略号按钮, 出现字体对话框, 设置字体为宋体、粗斜、四号、红色, 完成后单击“确定”按钮。
 - (5) 同理将 Button1 组件的“Caption”属性值改为“关闭”。
- 设置完成后的窗体如图 1-11 所示。

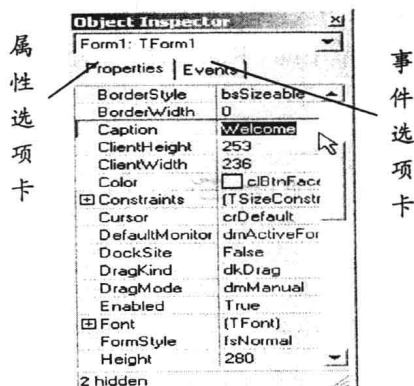


图 1-10

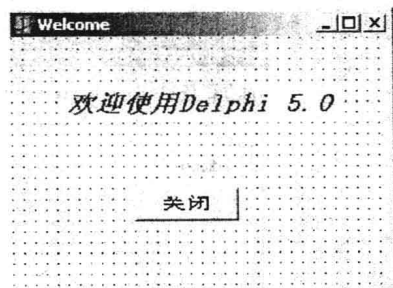


图 1-11

说明: 如果对象监视器不显示, 可以通过单击“View”菜单的“Object Inspector”命令来打开。

4. 代码编辑器操作

【实验 1】

为上一实验的“关闭”按钮添加事件响应代码, 实现在运行程序时, 单击“关闭”按钮即可关闭运行窗口。添加完代码后, 保存该程序。

【指导步骤】

- (1) 选择“Welcome”窗体中的“关闭”按钮。
- (2) 在 Object Inspector 中选择“Events”选项卡。
- (3) 双击 OnClick 事件右侧的属性栏, 即可出现 Button1Click 的事件响应代码段, 在 begin 和 end 之间输入程序代码“Close;”, 如图 1-12 所示。
- (4) 保存文件, 选择“File”菜单中的“Save”命令, 出现 Save Unit1 As 对话框(如图 1-13), 将单元文件保存为“d01_01_01.pas”。
- (5) 同理选择“File”菜单中的“Save Project As”命令, 出现 Save Project1 As 对话框(如图 1-14), 将项目文件保存为“d01_01.dpr”。

注: 项目文件必须和单元文件保存在同一个目录中。

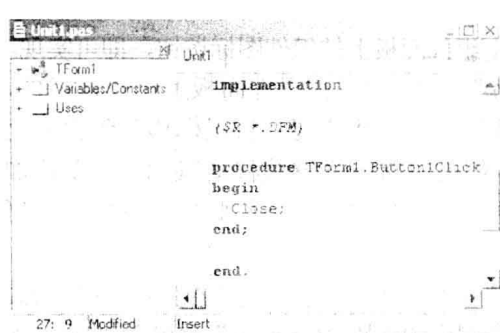


图 1-12

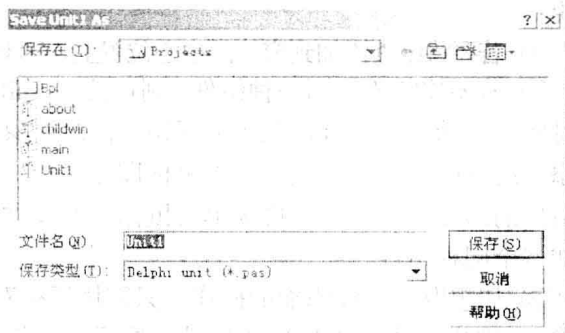


图 1-13

- Save: 保存正在操作的文档或窗体。
- Save As: 将正在操作的文档或窗体以新的文件名保存。
- Save Project As: 以新的名字保存正在操作的工程。

说明: 也可直接单击工具栏中的“Save All”按钮 , 保存应用程序。

(6) 单击工具栏中的  按钮或选择“Run”菜单中的“Run”命令, 即可运行程序。在运行窗体中单击“关闭”按钮, 观察效果。

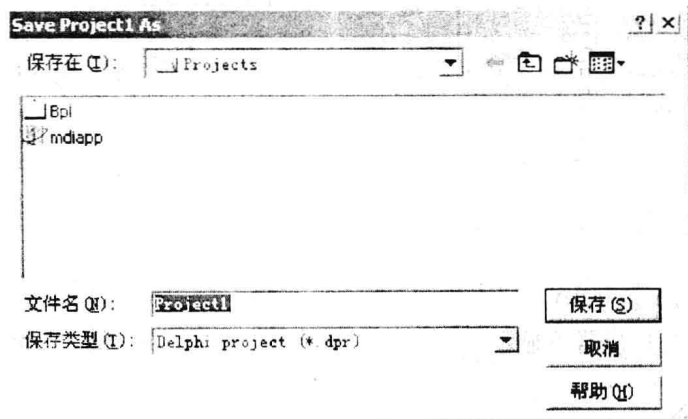


图 1-14

【实验 2】

设置代码编辑器窗口内源代码的显示风格。要求: 设置关键字(Reserved word) 为蓝色, 加粗倾斜; 注释(Comment) 为红色。

【指导步骤】

(1) 在代码编辑窗口中单击鼠标右键, 在快捷菜单中执行“Properties”命令(如图 1-15), 打开 Editor Properties 对话框。

(2) 选择“Color”选项卡, 在“Element”下拉列表框中选择“Reserved word”, 在“Color”颜色区中用鼠标单击蓝色颜色块(如图 1-16), 再在“Text attributes”中选“Bold”、“Italic”复选框, 完成关键字的设置。

(3) 同理, 在“Element”中选择“Comment”, 用鼠标单击“Color”区中的红色颜色块, 再单击“OK”按钮。

说明: 在对话框下边的预览区会显示设置效果。

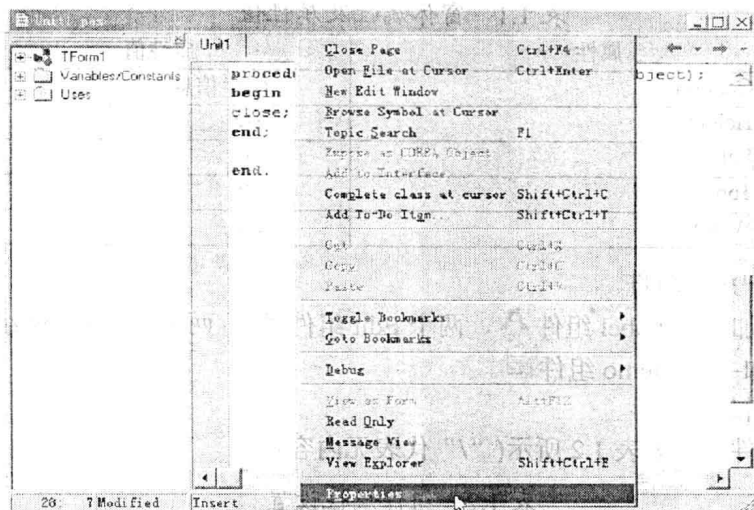


图 1-15

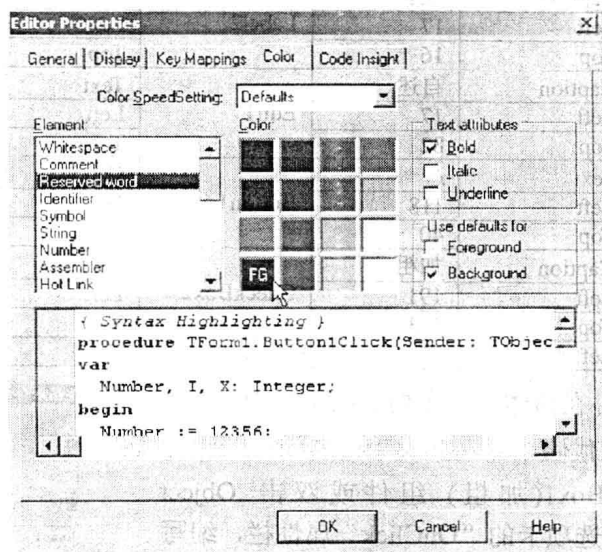


图 1-16

1.3 简单实例

【实验】

通过实验了解窗体及组件的基本设置，代码的编写以及程序的保存与运行。

【指导步骤】

(1) 启动 Delphi 5，系统新建一个工程，并且内含一个窗体 Form1；也可选取“File”菜单中的“New Application”命令，创建一个新应用程序。

(2) 设置窗体属性

按表 1.1 设置窗体属性。

表 1.1 窗体属性及属性值

属性	属性值
Caption	学生信息
Height	313
Left	249
Top	136
Width	258

(3) 向窗体中添加组件

往窗体中添加三个 Label 组件 、两个 Edit 组件 、两个 CheckBox 组件 、一个 Button 组件  和一个 Memo 组件 .

(4) 设置组件属性

窗体中的组件属性如表 1.2 所示(“/”代表无内容)。

表 1.2 组件属性设置

组件	属性	属性值	组件	属性	属性值
Label1	Caption	姓名	Label2	Caption	学号
	Left	17		Left	118
	Top	16		Top	16
Label3	Caption	自述	Edit1	Text	/
	Left	17		Left	17
	Top	82		Top	40
Edit2	Text	/	Botton1	Caption	颜色
	Left	118		Left	184
	Top	40		Top	200
CheckBox1	Caption	加粗	CheckBox2	Caption	倾斜
	Left	191		Left	191
	Top	113		Top	153
Memo1	Left	17	Memo1	Top	104

设置完属性后组件在窗体中的布局如图 1-17 所示。

(5) 编写事件响应代码

① 双击 CheckBox1(加粗) 组件或双击 Object Inspector 中“Event”选项卡的“OnClick”属性栏, 编写 CheckBox1 的 OnClick 事件响应代码如下:

```
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
```

```
    if CheckBox1.Checked then //若选中“加粗”复
```

选框

```
        Memo1.Font.Style:=Memo1.Font.Style+[fsBold] //使“自述”中的文字呈加粗状态
    else
```

```
        Memo1.Font.Style:=Memo1.Font.Style-[fsBold]; //否则使“自述”中的文字呈不加粗状态
end;
```

② 双击 CheckBox2(倾斜) 组件或双击 Object Inspector 中“Event”选项卡的“OnClick”



图 1-17

*注: 加阴影部分为程序设计中需用户输入的内容。(下同)

属性栏，编写 CheckBox2 的 OnClick 事件响应代码如下：

```
procedure TForm1.CheckBox2Click(Sender: TObject);
begin
    if CheckBox2.Checked then //若选中“倾斜”复选框
        Memo1.Font.Style:=Memo1.Font.Style+[fsItalic] //使“自述”中的文字呈倾斜状态
    else
        Memo1.Font.Style:=Memo1.Font.Style-[fsItalic]; //否则使“自述”中的文字呈不
        倾斜状态
end;
```

③ 双击 Button1(颜色) 组件或双击 Object Inspector 中“Event”选项卡的“OnClick”属性栏，编写 Button1 的 OnClick 事件响应代码如下：

```
procedure TForm1.Button1Click(Sender: TObject);
var
    colordlg:TColordialog;
begin
    colordlg:=TColordialog.Create(Form1); //创建颜色对话框
    colordlg.Execute; //打开颜色对话框
    Memo1.Font.Color:=colordlg.Color; //把颜色设置值赋值给文本
    colordlg.Free; //关闭颜色对话框
end;
```

(6) 保存程序

单击工具栏上的“Save All”按钮，将单元文件保存为“d01_02_01.pas”，将项目文件保存为“d01_02.dpr”。

(7) 编辑运行程序

单击工具栏上的  按钮(或按 F9 键) 运行程序。输入有关内容，单击“颜色”按钮，会出现颜色对话框(如图 1-18)，选择颜色，观察效果。

说明：退出运行模式，可选择“Run”菜单中的“Program Reset”命令，如图 1-19 所示。

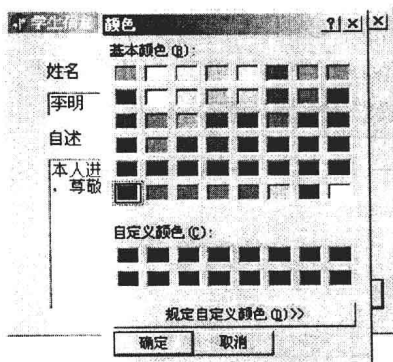


图 1-18

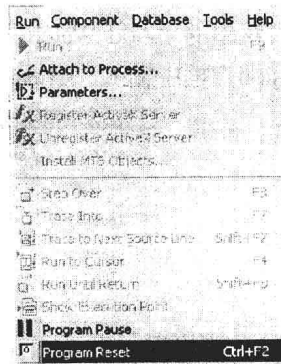


图 1-19