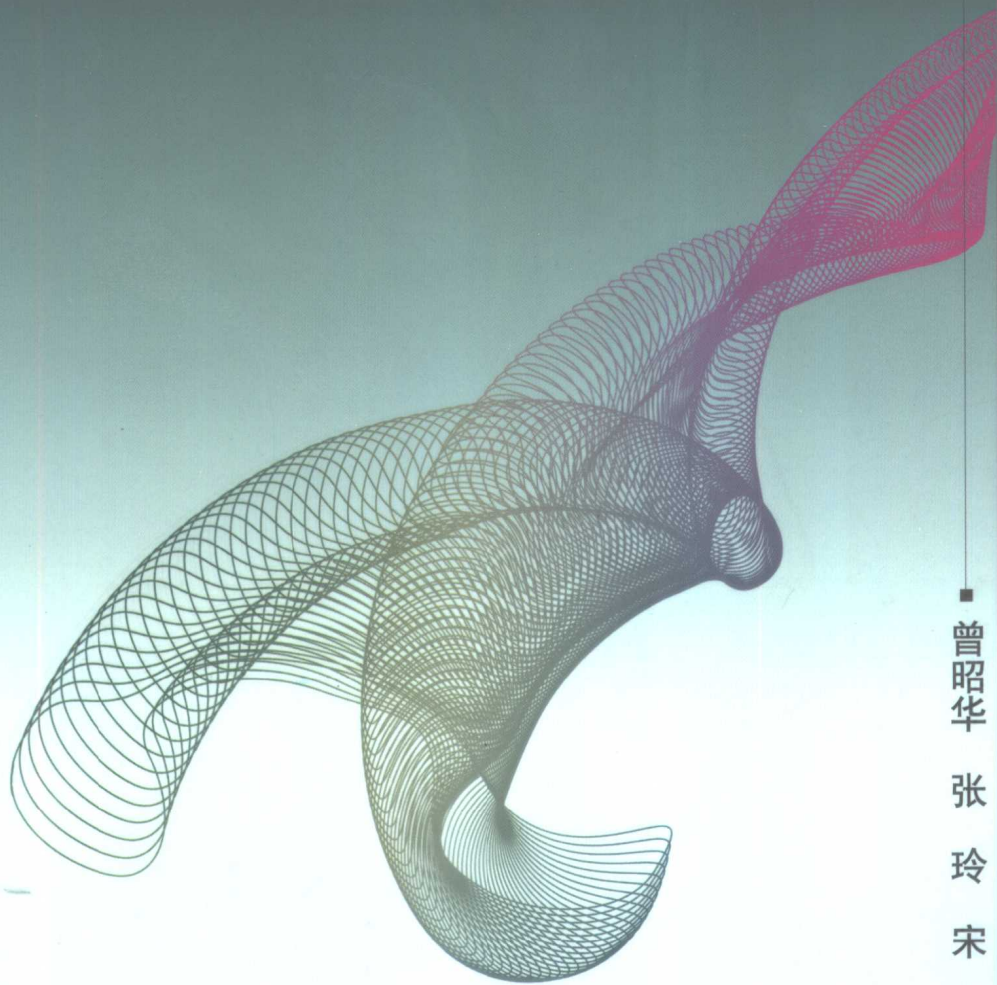


计算机实践指导系列教材



Delphi上机实践 指导教程

曾昭华 张玲 宋菲 等编著



计算机实践指导系列教材

Delphi 上机实践指导教程

曾昭华 张 玲 宋 菲 等编著



机械工业出版社

本书涵盖了 Delphi 最基本的编程实践内容, 以 Delphi 7 版本编写上机操作实例。全书共分 9 章, 介绍了 Delphi 的开发使用界面、语法基础、顺序结构、选择结构、循环结构、数组等基本知识。全书紧密结合可视化编程的思想, 所选取的实例典型而又生动, 每个实例都包括实践结果、实践目的、操作过程、实践源代码、程序分析及注意事项。每章后都配备一些精心设计的练习题。通过对这些例子、练习题和上机操作, 读者可以初步掌握 Delphi 编程的基本思想和方法。本书最后一章有 5 个综合实例, 这是为学有余力的读者准备的。

本书可作为大专院校 Delphi 课程的上机实践指导书, 也可作为广大计算机爱好者的自学指导参考书。

图书在版编目 (CIP) 数据

Delphi 上机实践指导教程/曾昭华等编著. —北京: 机械工业出版社, 2004.6

(计算机实践指导系列教材)

ISBN 7-111-14349-3

I . D . . . II . 曾 . . . III . 软件工具—程序设计—教材 IV . TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 033343 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 陈振虹

责任印制: 施 红

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2004 年 6 月第 1 版·第 1 次印刷

787mm × 1092mm $\frac{1}{16}$ · 12.25 印张 · 298 千字

0001—5000 册

定价: 18.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话: (010) 68993821、88379646

封面无防伪标均为盗版

前 言

Delphi 是第四代编程语言的杰出代表,它结合了可视化技术、面向对象技术、数据库开发技术、网络开发技术等多种先进的软件编程技术和思想,已经成为最佳的应用开发工具之一。Delphi 7 是 Inprise 公司(原 Borland 公司)新一代面向对象和可视化应用程序的开发环境,在 Windows 95/98/2000/NT/XP 操作系统上均能运行。Delphi 语言是标准 Pascal 语言,具有语法严谨,代码结构清晰,可读性好和代码执行效率高等优点。

本书以 Delphi 7 版本编写上机操作实例,选择的每个实例与教材相关内容配合,目的是使学生在上机操作的同时消化和理解教材的内容。每个实例的讲解一般都按如下构架进行:实践结果、实践目的、控件与主要属性介绍、操作过程、实践源代码、程序分析与注意事项。每一章章首介绍本章主要内容,末尾提供了练习题。

本书主要由曾昭华、张玲、宋菲、江洁、左欣、杨景喻编写,另外参加编写工作的还有潘爱先、于晓娜、侯彤璞、马宝柱、孟传、孙棋等。

由于作者水平所限,书中存在的不足与错误之处,敬请读者提出宝贵意见和建议。

编 者

目 录

前言

第 1 章 Delphi 的设计界面	1
实践 1 基本知识准备	2
实践 2 可视化编程实例	9
练习题 1	14
第 2 章 Delphi 的语法基础	15
实践 1 圆锥体体积的计算	16
实践 2 带日期时间的简易科学计算器	21
练习题 2	27
第 3 章 顺序结构	28
实践 1 简单的欢迎程序	29
实践 2 长方体计算公式	32
实践 3 简单的计算器	35
实践 4 简单的选课系统	41
练习题 3	46
第 4 章 选择结构	47
实践 1 带除零纠错的除法运算	49
实践 2 密码输入框	52
实践 3 个人信息登录系统	54
实践 4 个人成绩登记程序	57
实践 5 补交行李费	61
实践 6 日期判断	63
实践 7 延期交税计算	66
实践 8 单选按钮使用	68
练习题 4	71
第 5 章 循环结构	72
实践 1 求阶乘	73
实践 2 求自然对数 e	76
实践 3 韩信点兵问题	79
实践 4 制作数学用表	81
实践 5 求 10000 以内的完备数	84
练习题 5	86

第6章 过程与函数	87
实践1 求任意三角形面积	88
实践2 骑马通信问题	92
实践3 求5个数中的最大数	95
实践4 小数化最简分数	98
实践5 验证歌德巴赫猜想	101
练习题6	105
第7章 枚举与集合	106
实践1 按顺序变化的背景颜色	107
实践2 指定范围内的素数筛选	110
实践3 进制的转换	114
练习题7	117
第8章 数组与记录	118
实践1 起泡法整数排序	120
实践2 方阵的计算	122
实践3 上三角和下三角形矩阵	127
实践4 学生成绩统计系统	130
练习题8	133
第9章 综合实例	134
实践1 方阵的计算	135
实践2 文本编辑器	146
实践3 画板	156
实践4 五子棋游戏	174
实践5 媒体播放器	179
练习题9	188

第 1 章 Delphi 的设计界面

本章介绍 Delphi 可视化的编程环境,并对文件、工程、菜单、控件等进行最基本的操作。最后,通过编写一个小程序,初步了解 Delphi 可视化编程的基本过程,在得到结果的同时,加深感性认识。

本章简单介绍了 Delphi 7 集成开发系统的安装、界面和基本使用。它是一种方便、快捷的 Windows 应用程序开发工具,能大大提高编程效率。

学习完本章之后,希望读者能够做到以下两点:

- 1) 独立安装 Delphi 7 集成开发系统,熟练掌握界面的各种使用方法。
- 2) 初步认识控件及其属性、方法,并使用已掌握的控件开发一个最简单的 Delphi 7 工程。

Delphi 7 集成开发系统的界面主要由以下几个部分组成:

- 1) Delphi 7 主窗口,包括主菜单、工具栏和组件板。
- 2) 对象树状浏览器(Object Tree View)。
- 3) 对象观察器(Object Inspector)。
- 4) 窗体设计器(标题为 Form.1)。
- 5) 代码编辑器(标题为 Unit1.pas)。

对象的属性、事件与方法简介如下:

- 1) 对象的属性:属性描述了所属对象的一个特征,如本章介绍的 Button 控件的 Color 属性。可以通过 Object Inspector 中的属性窗口进行属性的设置或修改,也可以通过程序实现。实现的一般格式如下:

对象名.属性名:=属性值

如本章中的:Shape1.Brush.Color:=clWhite。

- 2) 对象的事件:对象的事件是预先定义好的一种或一系列动作,比如鼠标单击事件(OnClick),对象的事件是固定的,用户不能建立新的事件。设定一个对象的事件,可以在 Object Inspector 中的 Events 选项卡中设置。

- 3) 对象的方法:方法是与对象相关联的过程与函数的统称,被封装在对象之中。





实践 1 基本知识准备



操作过程

启动 Windows,依次单击“开始”、“Borland Delphi 7”、“Delphi 7”,屏幕上会出现如图 1-1 所示的窗口。

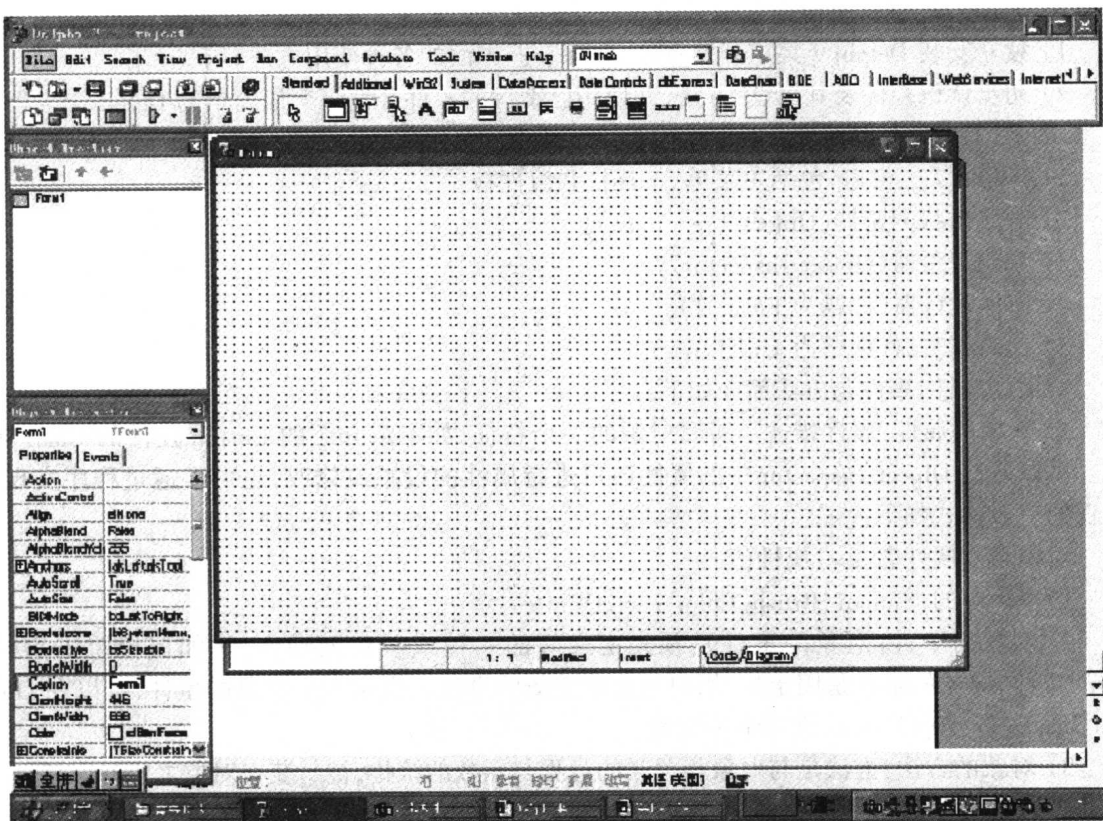


图 1-1

Delphi 7 的工作界面主要包括如下内容。

1) Delphi 的主窗口:标题为 Delphi 7-Project 1,位于屏幕的最上方,主窗口中包括菜单栏、工具栏、组件板。其中,“Project1”是系统自动新建的工程名,使用者可以自己创建任意工程名。

2) 对象树状浏览器(Object TreeView):用来显示工程中建立的所有对象及其所属关系。可用鼠标右击各个对象进行相应的操作。

3) 对象观察器(Object Inspector):可在其中设定所需对象(控件)的属性以及关联事件,每

一项都可以进行修改。

4) 窗体设计器:系统默认标题为“Form.1”,也可自行修改。这是编程中主要对象的显示场所。

5) 代码编辑器:系统默认标题为“Unit1.pas”,这是程序代码的编辑区域。它和窗体设计器之间可以通过 < F12 > 键进行切换,如图 1-2 所示。

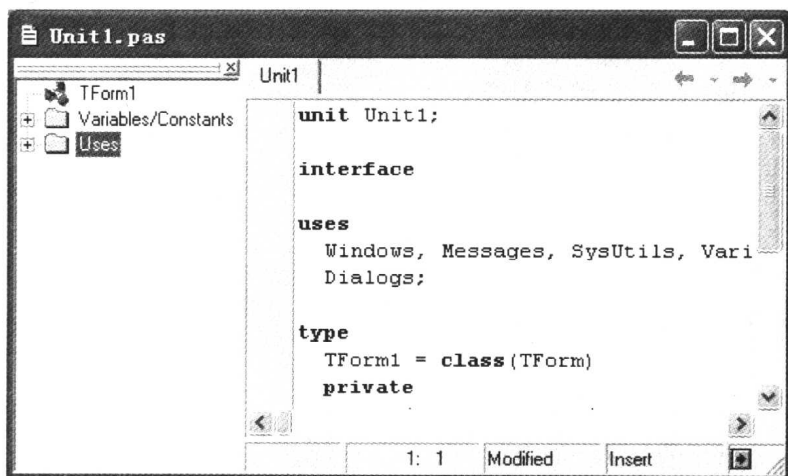


图 1-2

6) 菜单栏:包括“File”、“Edit”等主菜单,各主菜单的功能如表 1-1 所示。

表 1-1 Delphi 7 的主菜单

菜单名	功能
File	提供了各种文件操作命令
Edit	提供在代码编辑和界面设计时所需的复制、粘贴、删除等命令
Search	提供了代码的查找与替换功能
View	可以选择显示或不显示各种工具窗口
Project	和工程(Project)有关的各种操作
Run	提供调试程序所需的设置断点、单步执行等功能
Component	和控件有关的功能
Database	和数据库有关的功能
Tools	开发时会用到的一些工具
Windows	选择已经打开的各个窗口
Help	帮助信息

7) 工具栏:系统默认的工具栏中有 19 个工具按钮,如图 1-3 所示。它们是主菜单中常用功能的快捷方式,熟练地应用这些按钮可以给编程带来很多便利。

将鼠标移到各个按钮之上,稍等片刻,鼠标下方会出现该按钮的功能说明以及某些按钮的

键盘快捷方式。

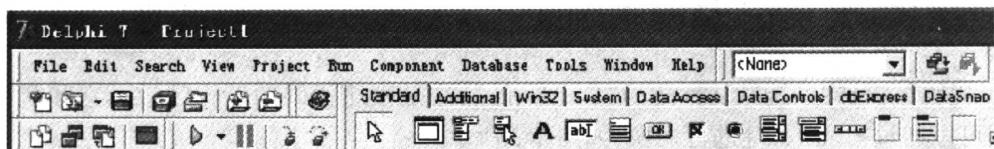


图 1-3

8) 组件板:如图 1-4 所示,组件板包含有 350 多个控件,是 Delphi 编程的核心部分,控件分列在 27 个选项卡中,以图形按钮的形式显示。对于初学者来说,最常用的是 Standard 和 Additional 选项卡中的控件。单击组件板右边的箭头可以左右拉动选项卡标签,以便找到需要的选项卡。

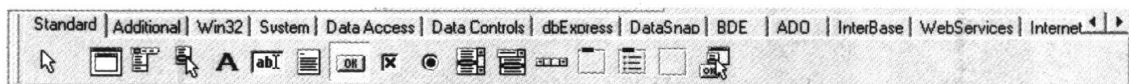


图 1-4

在组件板中最常用的选项卡如表 1-2 所示。

表 1-2 常用组件

选项卡名称	内容及功能
Standard	内置了标准的 Delphi 7 控件,是最常用的
Additional	是 Standard 选项卡功能的扩展,有很多实用的控件
Win32	是在 Windows 95/98/2000/NT/XP 下常用的控件
System	包括了定时器、文件控件、多媒体在内的和系统相联系的控件
Data Access	使用数据库时,该选项卡下的控件用于读写数据库
Data Controls	提供对数据库进行输入、修改等操作的许多控件
Data Snap	用于数据获取(本地或异地)的控件
BDE	提供包括 SQL 语言在内的数据操作控件
Dialog	Windows 常用的对话框控件
InterBase	提供 InterBase 数据库使用的许多控件
Win3.1	与 Delphi 1.0 项目相兼容的控件

下面先做一个小例子。

在选项卡中单击“Standard”(一般来说系统在启动时已经自动选中该项),接下来选中选项卡中的“Button”,可以发现,此控件下凹,表示已经选中。

将鼠标移到 Form1 中,按下左键不放,拖动鼠标,可以按照个人喜好画出一个名为“Button1”的按钮,周围的 8 个黑点可以用来调整按钮的大小。如果要删除此控件,单击左键选中后按 < Delete > 键。

这样,就创造出了第一个对象,如图 1-5 所示。关于属性(Properties)以及事件(Events)将在下一步实践中具体阐述。



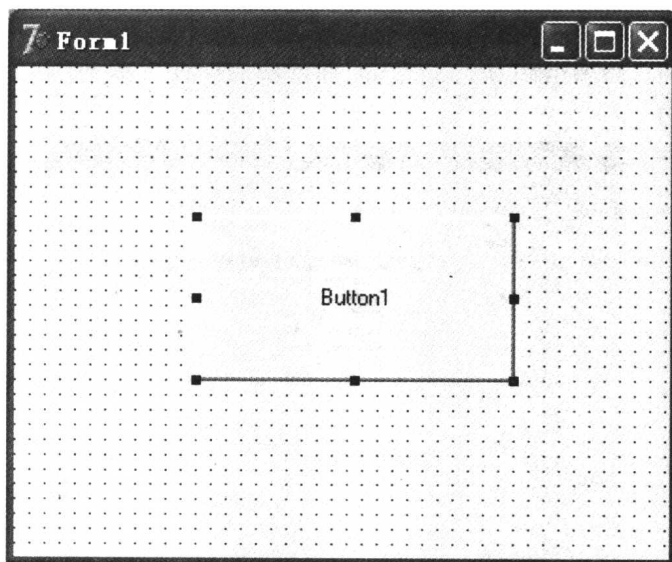


图 1-5



实践源代码

1. 代码编辑器

代码编辑器是程序代码的输入和编辑工具,是 Delphi 提供的一个功能强大、使用方便的代码编写工具。如图 1-6 所示。

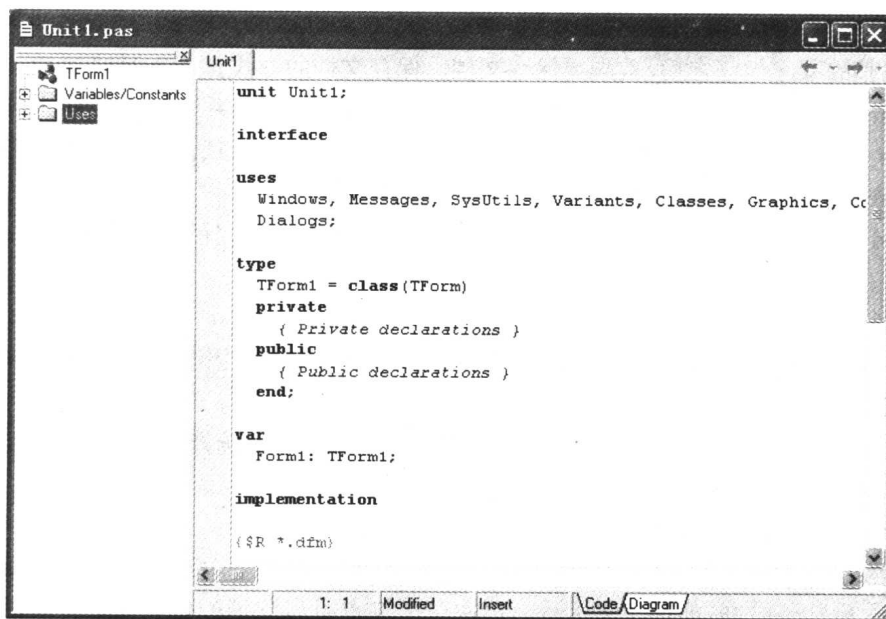


图 1-6

2. 代码洞察技术(Code Insight)

这是 Delphi 7 的代码编辑器为了方便用户而采用的一种技术,使用它可以大大方便代码的编辑,如图 1-7 所示。

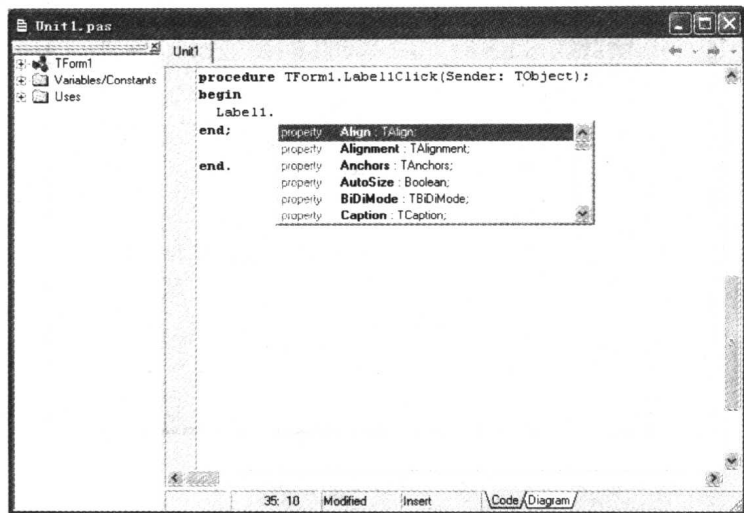


图 1-7

当输入已经创建的对象名称后,系统将自动弹出如图 1-7 所示的提示框,从中选择所需要的属性和方法,代码编辑器就会自动地输入到代码中。

代码洞察技术的适用范围很广,如提示所需输入的参数等,在更为复杂的编程中会起很大的作用。

3. 程序的运行与调试

(1) 保存文档

一个完整的程序编好之后,首先要保存文档。保存文档的方式有以下两种:

1) 单击菜单“File”,选择“Save All”。如果是第一次保存,则系统会先后弹出“Save Unit1 As”和“Save Project1 As”对话框,分别保存代码内容与工程内容,默认地址是 Delphi 7 的安装地址。“Save Project1 As”对话框如图 1-8 所示。

2) 单击菜单“File”,选择“Save As”,将保存当前单元文件;选择【Save Project As】,将保存当前工程文件。

(2) 运行与调试

鼠标单击“Debug”工具条上的“Run”或者直接用 < F9 > 快捷键,就可以开始编译(Compile)、连接(Link)和运行已经建立好的工程。

如果想将工程做成可执行文档(.exe),可以在“Project”项下选择“Build”,则系统生成一个可执行文档。

掌握调试的方法是重要的,尤其是在程序复杂的情况下。

1) 设置断点:在代码编辑器中,用鼠标单击某条语句左侧的空白区,此时语句前会出现一个红点,表示断点已设好,如图 1-9 所示。再次单击红点可以取消断点设置。

运行程序后,执行到断点语句时程序会自动终止,代码编辑器中左侧空白处会出现一个对勾,表示程序已停止在这个位置,如图 1-10 所示。再次按 < F9 > 键程序继续运行。

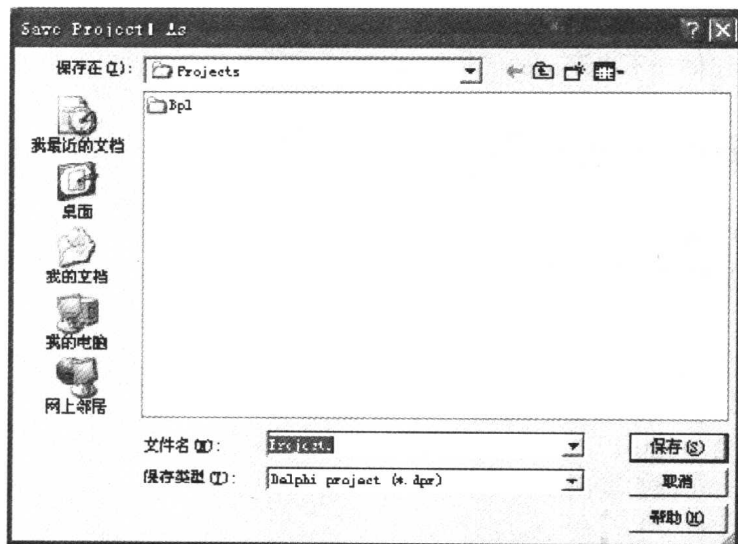


图 1-8

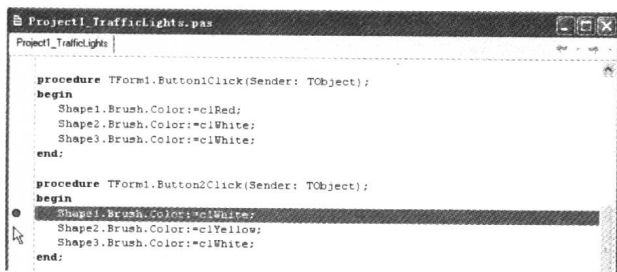


图 1-9

```

procedure TForm1.Button2Click(Sender: TObject)
begin
    Shape1.Brush.Color:=clWhite;
    Shape2.Brush.Color:=clYellow;
    Shape3.Brush.Color:=clWhite;
end;

procedure TForm1.Button3Click(Sender: TObject)
begin

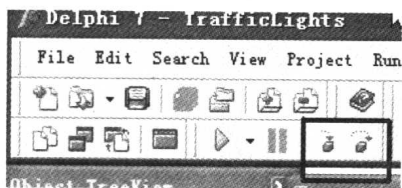
```

图 1-10

在一个程序中可以设置多个断点,而且断点的设置不影响执行断点语句之前的代码。

2) 单步执行:单步执行控制通过加速栏上的单步执行按钮来实现,每按一次按钮,程序执行一句。

方框中为单步执行按钮,如图 1-11 所示,左为【Trace into】按钮,右为【Step over】按钮。



```

procedure TForm1.Button2Click(Sender: TObject)
begin
    Shape1.Brush.Color:=clWhite;
    Shape2.Brush.Color:=clYellow;
    Shape3.Brush.Color:=clWhite;
end;

```

图 1-11

【Trace Into】按钮当执行到调用子程序的语句时,系统会跟踪到子程序的内部,在子程序中也是每按一次按钮就执行一次语句。

【Step Over】按钮当执行到一个调用子程序的语句时,并不进入子程序内部去跟踪,而是接着执行调用语句的下一条语句。

单步执行可以完全地显示程序运行的每一步状况,包括在每一步时各个变量的实际值,显然本例十分简单,还体现不出调试的优越性,但调试的作用必将在以后的学习中逐渐体现。



程序分析与注意事项

一个完整的 Delphi 项目的开发应该包括以下几个步骤:

- 1) 建立一个新的项目(工程)。
- 2) 建立窗体。
- 3) 在窗体上加入对象。
- 4) 检查和设定对象属性。
- 5) 编写事件处理程序。
- 6) 编译、运行(可能需要调试)。
- 7) 保存项目文件。
- 8) 生成可执行文件。





实践 2 可视化编程实例



实践结果

本例实践结果如图 1-12 所示,用鼠标单击某个按钮,对应的圆圈变成相应的颜色,如图 1-13 所示。

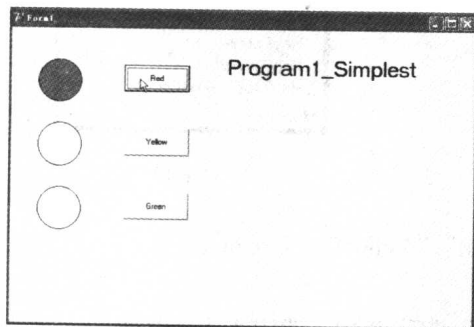


图 1-12

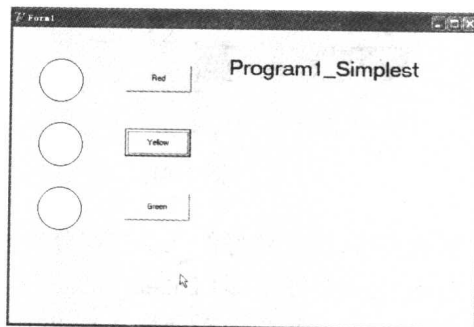


图 1-13

编写过程中,将用到 Shape、Button、Lable 控件。



实践目的

这是一个很简单的 Delphi 编程实例,通过编写该实例,可以使读者熟悉 Delphi 程序设计过程。



操作过程

1. 新工程的创建

在 Delphi 中开发的每个应用程序都被称为工程,虽然每次启动 Delphi 后,系统都会自动生成一个默认的工程 Project1,但应养成新建一个工程的习惯。

单击主菜单中的“File”,在下拉菜单中选择“New”,选择“Application”,如图 1-14 所示,系统将创建一个新的工程,其名字为“Project2”,依此类推。

这里所说的工程,实际上包括了两部分,一部分就是所谓的 Project,另一部分包括源代码等等。在稍后的文件保存操作中,我们将看到二者的不同。

2. 用户界面设计

1) 在生成的 Form1 面板上,建立三个按钮,建立方法如前所述。

改变按钮上的文字,这是在对象观察器里面修改的。选中第一个按钮,然后单击 Object Inspector,选中 Properties,在按钮的属性中找到 Caption,即标题。选中它,可以看到系统默认的 Caption 是“Button1”,如图 1-15 所示,也就是显示在刚建好的按钮上的文字,把它改成“Red”。

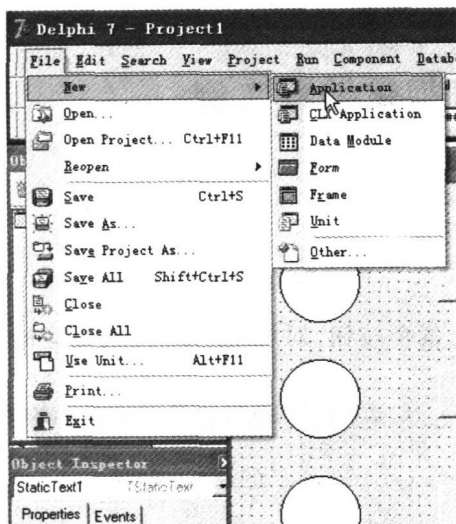


图 1-14

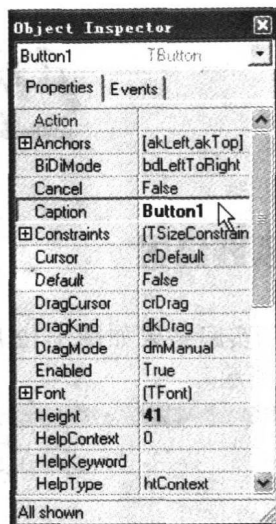


图 1-15

依此类推,把已建好的三个按钮分别改名为“Red”,“Yellow”,“Green”。

2) 建立一个标签,即 Label。

在 Standard 选项卡(就是刚才选 Button 用的那个选项卡)上,选中 Label 控件。

回到 Form1,按住鼠标左键不放,拖动鼠标做出一个 Label。Label 的大小可以自己调整。释放鼠标左键,可以看到面板上出现一个名为“Label1”的标签,底色是灰色。如果这时再单击该灰色区域,可以像上述 Button 一样调整这个标签的大小。

Label 控件的内容修改涉及到的控件属性和 Button 一样,均为 Caption,修改的方法与 Button 类似,此处不赘述。这里将其改为“Program1 _ Simplest”(即“最简单的程序 1”)。

改变 Caption 中文字的字体、颜色等等。这个过程涉及到 Label1 的 Font 属性。

在 Label1 的 Object Inspector 中,单击“Font”属性部分,可以看见最右端出现一个带有“...”的按钮,单击它,弹出“字体”对话框,如图 1-16 所示。

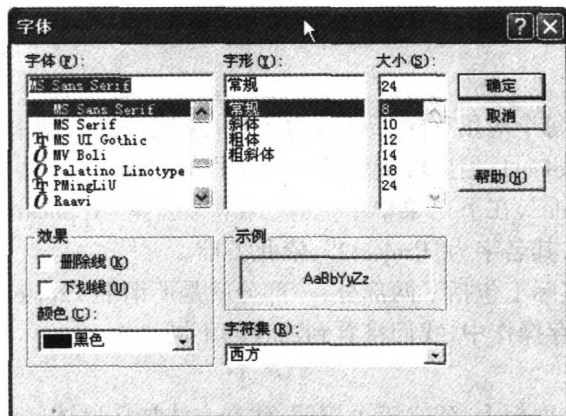


图 1-16

对 Windows 以及 Office 系列软件稍有了解的读者可以很快地掌握字体属性的控制方法,

只需在此窗口中选中需要的字体、字形、大小、颜色、效果等等,此例将字体的大小设定为 24。经过调整对象 Label1 的大小,可以得到较好的效果。

3) 用 Shape 控件画圆。该控件位于组件板中的“Additional”,如图 1-17 所示。

单击该控件后可以在 Form1 中利用鼠标的拖动画出想要的图形,系统默认的是长方形,先画出三个,然后在 Object Inspector 中修改其属性为圆形。选中“Shape”属性,在它的下拉菜单中将默认的 stRectangle 改为 stCircle,如图 1-18 所示,于是长方形就变成了圆形。三个长方形施以同样的步骤即可。当然,我们需要其他形状的时候,可以在下拉式菜单中选择其他的属性。

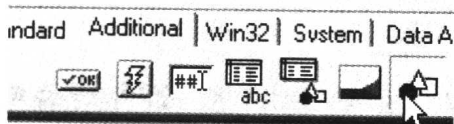


图 1-17

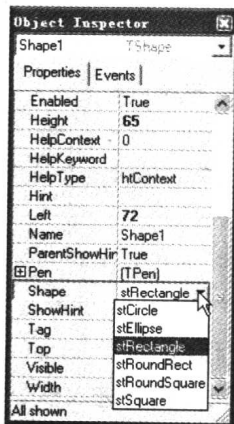


图 1-18

提示:当大量创建某个控件时,一次又一次的单击组件板是一件很烦的事情,这时可以利用剪贴板,用 < Ctrl + C > 组合键实现复制,再用 < Ctrl + V > 组合键实现粘贴。

至此编程的第一步——用户界面设计就完成了。一个好程序离不开友好的用户界面,读者不妨考虑一下如何使界面更为简洁明快,并尝试一下如何实现。

4) 下面要将控件对齐。当窗体上有多个控件时,如果不使之一一对齐,整个界面便会显得杂乱无章,虽然利用鼠标的拖放也能实现,但是操作复杂而麻烦。

下面以三个按钮为例,介绍控件的对齐板(Align)。

➤ 按住 < Shift > 键的同时用鼠标选中三个按钮。

➤ 在菜单中选择“View”、“Alignment Palette”,在窗体中出现如图

1-19 所示的对齐面板(Align),此时窗体如图 1-20 所示。在面板中单击 ,可以看到三个按钮被对齐了,如图 1-21 所示。

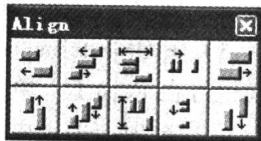


图 1-19

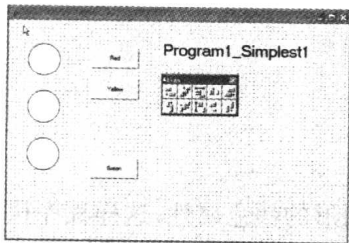


图 1-20

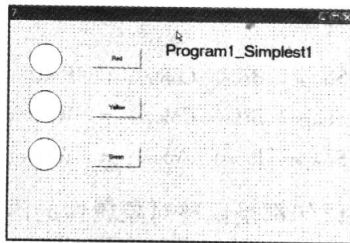


图 1-21