

实例精通系列丛书

Delphi 5

实例精通

徐万力 编著



配光盘



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

实例精通丛书

Delphi 5 实例精通

徐万力 编著

本书附盘可从本馆主页 <http://www.lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制



清华大学出版社

前 言

Microsoft Windows 是基于图形界面的多任务、多窗口的操作系统。自 1983 年问世以来,它历经完善,其 Windows 环境及 Windows 标准已不断被广大用户所接受。Windows 由于其功能齐全、使用方便、用户界面美观友好,使得计算机操作方式和软件开发过程发生了革命性的变化。开发遵循 Windows 标准的应用程序越来越受到各国用户的极大关注和兴趣。

Delphi 5 是 Imprise 公司(原 Borland 公司)推出的 Windows 下快速开发应用程序的工具,它是一个少有的集强大功能与快速开发于一体的 Windows 开发环境。我们在本书里要介绍的是 Delphi 5 的使用。

Delphi 有许多优越的特性:提供了新颖的可视化设计工具、采用面向对象的方法将 Windows 编程的复杂性封装起来、将可视化界面与面向对象的 Pascal 语言完美地结合在一起,实现了可视化设计与 Object Pascal 语言的有机集成,并配有 Database Engine,可通过 SQL Links, ODBC 访问多种数据库,并且提供了强大的开发基于客户/服务器模式的数据库应用的能力。

本书从实用的角度出发,在开发实际项目基础上结合本人的开发体会编写而成。该书内容深入浅出,语言通俗易懂,全书共分 8 章,内容包括:Delphi 概述、Object Pascal 语言、Windows 界面设计、对话框、Delphi 数据库基础、SQL 语言、数据库应用程序的创建以及多线程处理。

阅读本书,读者将掌握 Delphi 的编程方法、Delphi 各种常规部件的综合应用方法和 Delphi 高级应用程序开发技巧,这将大大提高读者的 Delphi 应用开发能力。特别是通过学习 Delphi 数据库内容,读者将掌握 Delphi 数据库开发原理、数据库服务器的联接技术、常用数据库控件的综合应用方法以及创建数据库应用程序。

本书适合于 Delphi 初级用户和有一定经验的程序员。初学者阅读本书可迅速加入 Delphi 高级用户的行列;有一定使用经验的读者可通过本书掌握 Delphi 深层次的开发方法,学会用更巧妙的办法开发出高水平的 Delphi 应用。由于个人水平有限,书中的不当之处在所难免,希望大家不吝赐教。

作 者
1999 年 10 月

第 1 章 Delphi 5 概述

由 Borland 公司开发的 Delphi 是一种全新的可视化编程环境，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它有许多优点，如可定制的开发环境、面向对象、构件库、OCX 和 ActiveX 支持、完全编译等，这些对于广大的程序开发人员来讲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。

对于开发人员来说，选择 Delphi 是非常明智的。它与其他开发工具相比，Delphi 的编程效率和应用程序的性能至少能提高一倍，因而有人曾略显夸张地说：“选择 Delphi 吧，因为选择它就意味着软件开发已经成功了一半！”

如今，Delphi 已经发展到第五版(Borland 公司更名为 Imprise 公司，所以 Delphi 5 是由 Imprise 公司推出的)。它很好地继承了以前版本所具有的优点，同时还新增了许多令人拍案叫绝的功能，从而使自己成为了当今世界上名符其实的最快的编辑器和最为领先的数据库技术。

1.1 Delphi 5 的安装与卸载

1.1.1 安装 Delphi 5 的软硬件要求

Intel Pentium 90MHz 或者更高的 CPU（推荐）。

Microsoft Windows 9x 或者 NT 4.0（Service Pack 3）。

32MB 内存（推荐使用 64MB 或者更高）。

80MB 空余的硬盘空间（最小安装）。

光驱。

VGA 或者更高分辨率的显示器。

1.1.2 Delphi 5 的安装空间

表 1.1 中列出了在安装不同版本的 Delphi 5 时，不同安装要求所需要的硬盘空间的大小（该数目包括了约 10MB 左右的只在安装期间需要的临时交换空间）。

表 1.1

版 本	完 全 安 装	最 小 安 装
Standard（标准版）	129 MB	80 MB
Professional（专业版）	151 MB	118 MB
Client/Server（客户/服务器版）	282 MB	156 MB

1.1.3 安装 Delphi 5

首先启动 Delphi 5 的安装程序。在光驱中插入 Delphi 5 安装盘, 进入 Windows 9x 或者 Windows NT 4.0 操作系统。Setup Launcher 将自动地启动。

通过该 Setup Launcher 我们可以安装 Delphi 5, InterBase 5.5, Remote Debugger Server, Local InterBase, TeamSource 等应用程序, 当然单击 Setup Launcher 中的 Browser CD 图标还可以浏览光盘中的内容, 如图 1.1 所示。



图 1.1 Setup Launcher 窗口

单击 Setup Launcher 中的 Delphi 5 图标, 就可以利用安装向导安装 Delphi 5 了。具体的安装过程并不复杂, 只需按向导中的提示进行即可。不过, 在安装过程中还是有一些需要注意的地方:

1. 选择安装内容

如图 1.2 所示, 你可以在 3 种安装内容中做出选择, 即:

Typical: 选择该项内容, 表明你只选择了 Delphi 5 中一些最常用的内容。

Compact: 选择该项内容, 表明由于你计算机的硬盘空间有限, 所以只选择了 Delphi 5 中最少的内容, 许多有价值的内容已经被你舍弃掉了 (建议你不要选择该项内容)。

Custom: 选择该项内容, 表明你想自定义安装 Delphi 5 中的内容 (建议高级用户选择该项内容)。

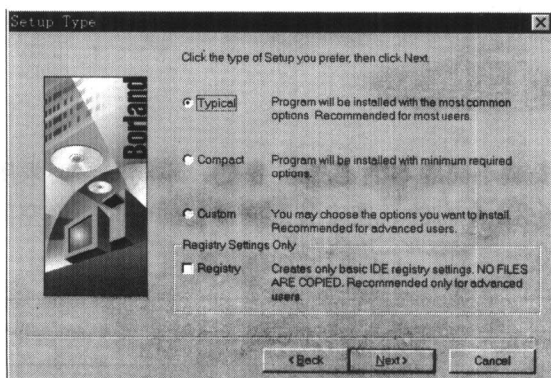


图 1.2 选择安装内容

2. 确定安装目录

如图 1.3 所示, 默认情况下, 安装目录为: \Program Files\Borland\Delphi 5。如果你想在不同位置进行安装, 那么一定要注意不要选在 Delphi 5 以前版本所在的目录上(假定你的机器上已经安装有老版本的 Delphi 产品)。值得注意的是: 尽管你改变了安装目录, 但是安装过后的 README.TXT 文件及其他的文档文件中却依然假定了你使用的是默认安装目录结构。

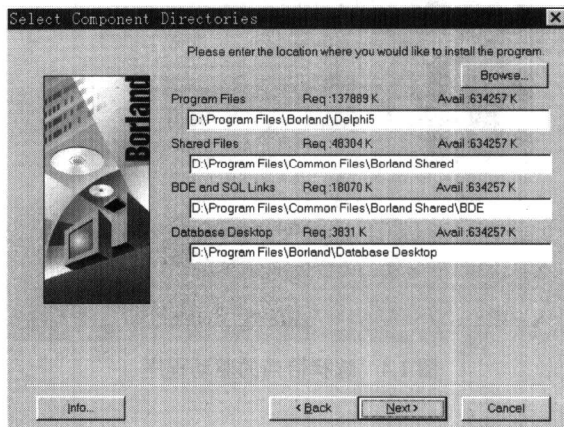


图 1.3 确定安装目录

3. 安装 Borland Database Engine

第五版本的 Delphi, 其 Borland Database Engine (BDE) 将覆盖已有的 32 位 BDE 和 SQL Links。值得注意的是: 只有在 Client/Server 版和 Enterprise 版中才提供了 SQL Links。

4. 安装 InterBase

Delphi 5 包含有最新版本的本地 InterBase 即 InterBase 5.1.1。值得注意的是: 安装 InterBase 的过程与安装 Delphi 5 的过程一定要独立开来。在图 1.1 的 Setup Launcher 窗体中有安装 InterBase 的图标, 当然运行该光盘中 InterBase 目录下的 SETUP.EXE 文件也同样可以安装 InterBase。

5. 驱动程序的选择问题

如图 1.4 所示, 在这里你需要选择四种驱动程序: Sybase 驱动程序、Oracle 驱动程序、MS Access 驱动程序、Informix 驱动程序。如何选择 Oracle, MS Access, Informix 等驱动程序在图 1.4 中一目了然。值得注意的是如何选择 Sybase 驱动程序, 如果你安装的 SQL Links 想使用的数据库管理系统是 Sybase 4.2 或者更低版本, 那么请选用 DB_Lib 驱动程序, 否则最好选用 CT_Lib 驱动程序。

6. 有效地避免安装冲突

在安装过程中, 为了避免安装向导程序在安装和调整系统文件时产生不必要的安装冲突, 建议你最好在安装 Delphi 5 之前关闭正在运行着的所有其他应用程序。如果安装过程中有其他的应用程序正在运行, 要想关闭它们, 那么打开任务栏或者按组合键 Ctrl+Alt+Del 均可以找到这些应用程序, 然后关闭它们。

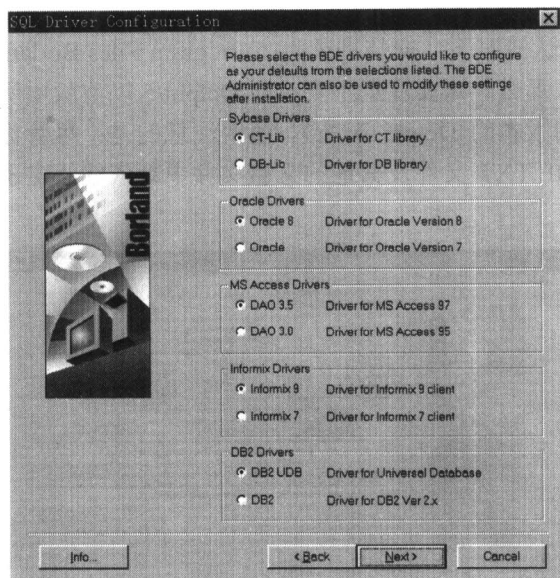


图 1.4 选择恰当的驱动程序

安装完毕后重新启动计算机。

1.1.4 卸载 Delphi 5

如果你的计算机上已经安装了 Delphi 5, 那么你可以随时地卸载它。卸载的具体步骤如下:

步骤 1: 进入 Windows 的控制面板。

步骤 2: 双击“添加/删除程序”图标。

步骤 3: 在列表框中选择 Borland Delphi 5, 如图 1.5 所示。

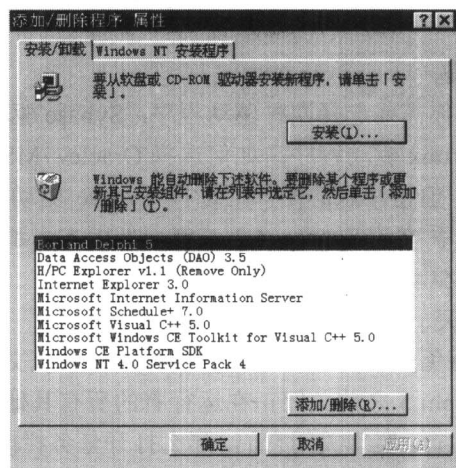


图 1.5 删除 Borland Delphi 5

步骤 4: 单击“添加/删除@...”按钮。

步骤 5: 单击 Yes 按钮。片刻之后卸载即可完毕。

1.2 Delphi 5 的开发环境

1.2.1 进入 Delphi 5

进入 Windows 的“开始”菜单, 选择“程序”中的 Borland Delphi 5 子菜单, 最后双击 Delphi 5 图标, 启动 Delphi 的应用程序, 如图 1.6 所示。

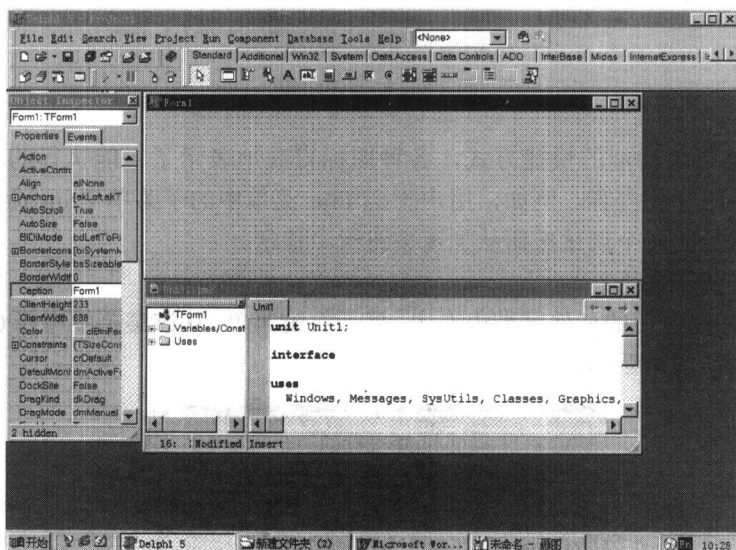


图 1.6 Delphi 的运行界面

首次加载的 Delphi, 界面主要由 4 部分组成:

- (1) 主窗口 Delphi 5-Project1;
- (2) Object Inspector 窗口;
- (3) 窗体 Form1;
- (4) 代码编辑窗口 Unit1.PAS。

以下我们将对这 4 个窗口分别进行介绍。

1.2.2 Delphi 5 开发环境介绍

1.2.2.1 主窗口

Delphi 5 的主窗口是开发过程的控制中心, 它位于屏幕的上端, 包括主菜单、加速按钮和控件选项板, 如图 1.7 所示。

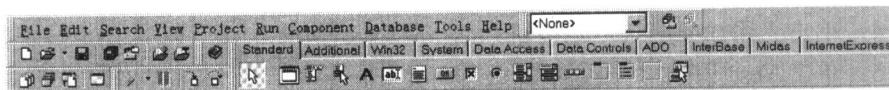


图 1.7 Delphi 的主窗口

1. 主菜单

它位于 Delphi 5 主窗口的最上面，所包含的菜单项提供了 Delphi 5 的全部功能。通过这些菜单项，用户可以控制包含在应用程序中的文件，并进行日常维护、编译和调试等方面的管理。

2. 加速按钮

它位于主窗口的左下端，一般情况下它由两排共 16 个按钮组成，如图 1.8 所示。

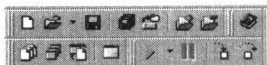


图 1.8 加速按钮组

这些按钮是菜单功能的快捷方式，各种图标直观地表示了它能执行的动作。用户可以修改加速按钮组中的内容。例如，用户想把 File 菜单中的子菜单 Print 命令放到加速按钮组中，而把常见的 4 个调试按钮从加速按钮组中移走。

具体的操作步骤如下：

步骤 1：在加速按钮的任何区域内右击，显示右键菜单，从该菜单中选择 Customize...，显示定制加速按钮的对话框，如图 1.9 所示。

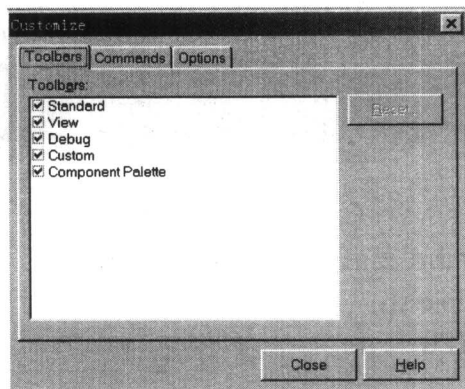


图 1.9 定制加速按钮的对话框

步骤 2：在 Toolbas 页中的 Toolbas 列表框内选择 Debug 项，清除 Debug 字样前面的检查框，并仔细观察加速按钮组中内容的变化。

步骤 3：在 Commands 页中 Categories 列表框内选择 File 项，此时 Commands 列表框内将显示与 File 项有关的命令内容。

步骤 4：滚动 Commands 列表，找到 Print...图标并选中它，如图 1.10 所示。

步骤 5：把 Print...图标从 Commands 列表框拖动到加速按钮组中的任意位置。仔细观察此时加速按钮组中内容的变化，如图 1.11 所示。

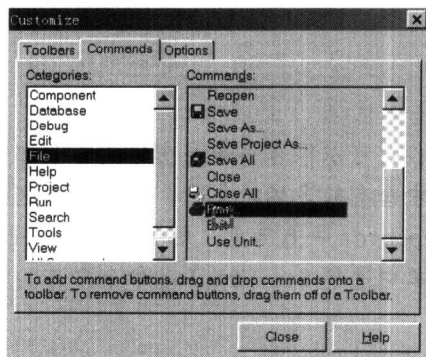


图 1.10 选中 Commands 列表框内的 Print...图标



图 1.11 修改后的加速按钮组

步骤 6: 单击 Close 按钮, 关闭定制加速按钮的对话框。

到此为止, 我们成功地完成了加速按钮的定制任务。

3. 控件选项板

它位于主窗口的右下端, 由一行若干页的按钮组成, 用户可以利用它来选择所需要的部件, 并将它放在窗体上, 如图 1.12 所示。

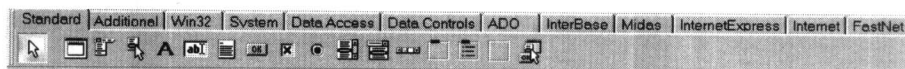


图 1.12 控件选项板

控件选项板每一页上均有若干个控件, 这些控件是建立 Delphi 应用程序的基本要素。Delphi 5 为用户提供了丰富的控件库, 既有可视的控件 (如文本框、标签等), 也有非可视的控件 (如系统菜单、数据表等)。它们按照各自功能分别排列在控件选项板的各个页面上。关于它们的用法将在下一节给出详细的介绍。

1.2.2.2 Object Inspector 窗口

Object Inspector 窗口含有两页: Properties 页和 Events 页。Properties 页中显示的是窗体中当前被选择控件的属性信息, 并允许改变该对象的属性; 而 Events 页中则列出了当前控件可以响应的事件。此外, Object Inspector 窗口的顶部还有 Object Selector, 该下拉式菜单中显示了包括窗体在内的、窗体上所有控件的名称和类型, 如图 1.13 所示。

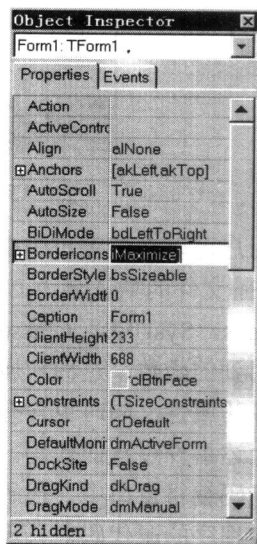


图 1.13 Object Inspector 窗口



指点迷津:

用户使用 Object Selector 可以很容易地在窗体的各个部件之间进行切换,也可以快速地回到窗体本身。当窗体中含有较多的对象时,您会发现这是切换对象是最快捷途径。另外,如果想使 Object Inspector 一直可见,那么可在 Object Inspector 上任意位置右击,显示右键菜单,在该菜单中选择 Stay On Top 命令即可。

1.2.2.3 窗体 Form

Form 窗口位于屏幕的右侧,它是进行大部分程序设计的工作区。首次启动 Delphi 5 时显示的是窗体 Form1。用户可以把程序所需要的控件放在窗体中,通过移动位置、改变尺寸大小等操作随心所欲地安排它们,以此来开发应用程序的用户界面。

1.2.2.4 代码编辑窗口

首次启动 Delphi 5 时代码编辑窗口处于窗体 Form 的下面。当部件被放到窗体上时,Delphi 会自动生成大部分的用户界面代码。移开窗体 Form,显示代码编辑窗口,如图 1.14 显示了空窗体 Form1 的代码窗口。

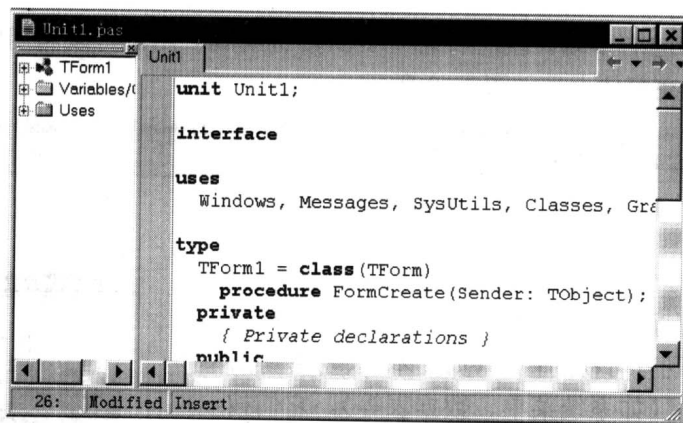


图 1.14 窗体 Form1 的代码编辑窗口

用户可以在该窗口中书写 Delphi 应用程序的源代码,即编辑单元文件。在一个单元文件中,文件的默认名为 Units1, Units2...。Uses 语句后面出现的一系列名字,如 Windows, Messages, SysUtils, Classes..., 它们都是 Delphi 预先定义的单元库,里面含有在编程时经常用到的函数、过程和属性等。Type 语句后面出现的内容是 Form 窗口及其里面所出现的控件的属性,同时也含有这些控件的事件过程的头标识。在 Implementation 中则是用来实现上述事件过程的代码。当程序中含有不止一个窗口时,会有几个单元文件的源程序出现在代码编辑窗口中。代码编辑窗口的标题条中显示了当前所有的单元文件名。要查看某一特定程序的源代码,只需用鼠标选中写有该单元文件名的页就可以了。

1.3 一个简单的例子

上述几节内容简要介绍了 Delphi 5 的安装及开发环境, 本节将演示一个用 Delphi 5 制作的简单例子——日历程序。

该例子的运行结果如图 1.15 所示。

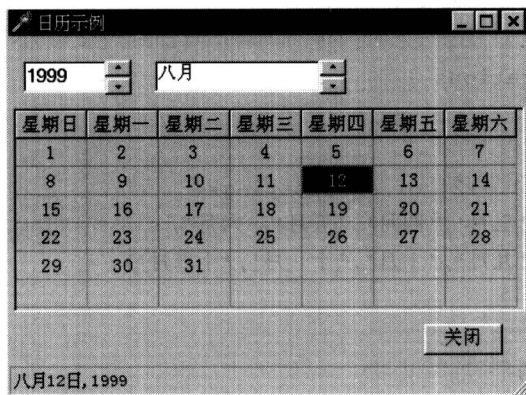


图 1.15 日历示例

使用该示例时, 用户可以在界面上部的年度栏中设定任意的年度、月份, 随着用户选择, 程序将把该年度、月份的日历显示界面中部。用户可以查看某日是星期几, 同时, 随着用户所选日期的不同, 界面下部的状态栏中将显示出该日期字样。显示完毕后, 单击“关闭”按钮, 关掉该程序。

日历示例的程序清单如下:

```
unit u101;  
  
interface  
  
uses  
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
    ComCtrls, StdCtrls, Grids, Calendar, Spin;  
  
type  
    TForm1 = class(TForm)  
        Edit1: TEdit;  
        Edit2: TEdit;  
        Calendar1: TCalendar;  
        Button1: TButton;  
        StatusBar1: TStatusBar;  
        UpDown1: TUpDown;
```

```
UpDown2: TUpDown;
procedure FormCreate(Sender: TObject);
procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);
procedure UpDown2Click(Sender: TObject; Button: TUDBtnType);
procedure Calendar1Change(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
  public
  { Public declarations }
end;
const
  Months: array[1..12] of shortstring=(
    '一月','二月','三月','四月','五月','六月',
    '七月','八月','九月','十月','十一月','十二月');

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  edit1.ReadOnly:=true;
  edit1.Text:=inttostr(calendar1.year);
  edit2.ReadOnly:=true;
  edit2.Text:=months[calendar1.month];
  updown2.Position:=calendar1.Month;
  with updown1 do
  begin
    min:=1000;
    max:=3000;
    position:=calendar1.year;
  end;
end;

procedure TForm1.UpDown1Click(Sender: TObject; Button: TUDBtnType);
begin
  with calendar1 do
  begin
```



```
if (button=btnext) then
  year:=year+1
else
  year:=year-1;
  edit1.Text:=inttostr(year);
  end;
end;

procedure TForm1.UpDown2Click(Sender: TObject; Button: TUDBtnType);
begin
  with calendar1 do
    begin
      if (button=btnext) then
        month:=month+1
      else
        month:=month-1;
      edit2.Text:=months[month];
    end;
end;

procedure TForm1.Calendar1Change(Sender: TObject);
begin
  with calendar1 do
    statusbar1.SimpleText:=months[month]+' '+inttostr(day)+
      '日'+', '+inttostr(year);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  close;
end;

end.
```

第2章 Object Pascal 基础

1972年, DR.Niklaus Wirth 先生设计了一种新型的编程工具, 并以法国哲学家 Blaise Pascal 的名字为其命名, 这就是著名的 Pascal 语言。由于该语言具有可读性强、编写容易等特点, 这使得它作为应用程序开发语言迅速流行于世。1985年, Borland 公司成功地推出了闻名遐迩的编译器——Turbo Pascal。该编译器把语言的简便性和强大的功能有机地结合在一起, 致使 Pascal 语言得以在 PC 平台上格外流行。

后来, Delphi 的语言基础选择了 Pascal, 即形成了 Object Pascal。Object Pascal 是一个真正通俗易懂的面向对象语言。任何能在 C++ 中完成的任务, 都能在 Delphi 中以更为简单直接的方法完成。尤为值得一提的是: 在 Delphi 5 中使用了最新的程序设计语言特性, 使 Object Pascal 能更自如地满足当今编程的需要, 并对 Delphi 4 中的一些特性给予了全面的加强, 例如:

1. 动态数组

动态数组的特点是: 无需给定数组上限。它给予开发者较大的灵活性。当然这都归功于 Delphi 使用了高级内存管理技术来控制动态数组。在 Delphi 4 中虽然出现了该功能, 但使用并不方便, 而在 Delphi 5 中, 开发者操作数据更灵活, 编写的代码也更短。

2. 方法共用

和 Delphi 4 一样, Delphi 5 允许对象拥有同名方法。方法间仅通过不同的参数加以区分。这使得对象更灵活, 与其他对象交换数据更方便。但与 Delphi 4 不同的是, Delphi 5 在使用第三方控件时表现得更灵活。

3. 64 位整型

整型在数字问题的编程中扮演着至关重要的角色, 因为它没有精度问题, 而实型数据做不到这一点。Delphi 5 全面支持 64 位整型, 给开发者更强的数字处理能力。这在发展金融、科学记数法、工程问题的应用程序上有很广的用途。

本章中, 我们将讨论 Delphi 应用程序中最常用的 Object Pascal 语法, 并介绍在事件处理过程和其他应用程序中, 如何用它来编制程序代码。

2.1 Delphi 的编程风格

在编写 Object Pascal 程序时, 一定要注意程序的可读性。由于 Pascal 语言是英式结构语言, 因此在程序中要选择适当的缩排、大小写、语法突显以及需要在需要时将程序代码分行, 同时在必要时应该给程序加上适当的注释, 这样会使得你编写的程序代码能够很容易地被他人读懂。这就是所谓的“Delphi 编程风格”。

2.1.1 缩排

一般而言, 缩排应该遵守以下规则:

1. 编写复合语句时, 将当前的语句向右缩进两个空格。
2. 包含在一个复合语句中的另一个复合语句应该缩进四个空格, 并依次类推。

例如:

```
if expression then
begin
    statement1;
    statement2;
end;

if expression1 then
begin
    if expression2 then
        statement1;
    else
        statement2;
end;
```

必须指出的是: 这两条规则仅仅是建议而已, 不同的程序员对规则的理解也不尽相同, 例如, 对于规则 2 也可以有如下写法:

```
if expression1 then
begin
    if expression2 then
        statement1;
    else
        statement2;
end;
```

除此之外, 还有许多其他的写法, 在此不一一列举了。

2.1.2 大小写的使用

并不像 C/C++ 语言那样, Pascal 编译器是不区分字母的大小写的 (C/C++ 语言区分字母的大小写), 所以, 单字 `Myname`, `MyName`, `myname`, `myName` 和 `MYNAME` 对于 Pascal 编译器来说是完全等价的。但是, 我们建议用户最好还是正确地使用字母的大小写。不区分字母的大小写会产生混用标识符等错误, 这样会影响程序代码的可靠性。使用字母大小写的规则如下:

1. 对于单字标识符, 大写单字的第一个字母。
2. 对于由多个单字组成的标识符, 大写每个单字的第一个字母。

例如:

`MyName`

MyNameIsTom

2.1.3 语法突显

为了使用户在读写 Pascal 代码时更加容易, Pascal 编译器提供了一种名为 Color Syntax Highlighting 的功能。用户在编辑窗口中输入字母时, 编译器根据该字母在 Pascal 中意义的不同, 会显示成不同的颜色。默认状态下, 关键字用粗体表示, 字符串和注释用彩色表示等等。当然, 用户可以自己随心所欲地改变关键字、字符串和注释等的彩色表示, 例如, 在默认状态下注释的彩色表示为蓝色, 用户可以把蓝色改为绿色。操作步骤如下:

步骤 1: 在选择主菜单 Tools 中的 Editor Options... 菜单命令, 显示编辑器属性窗口。

步骤 2: 选择编辑器属性窗口中的 Color 页面, 如图 2.1 所示。

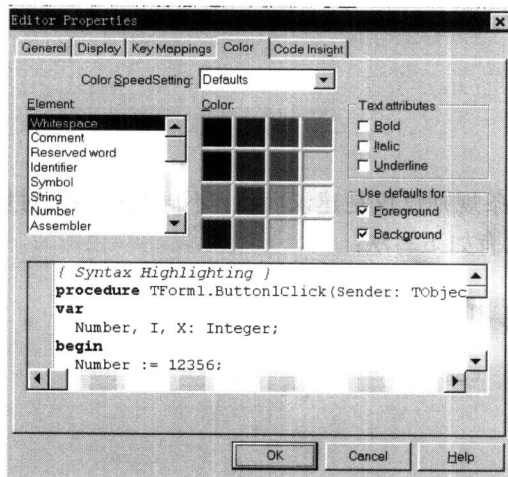


图 2.1 编辑器属性窗口中的 Color 页面

步骤 3: 在 Color 页面中, 选择 Element 下面列表框中的 Comment (该列表框中列出了可更改的所有环境元素, 如关键字、字符串、标识符以及数字等), 然后在 Color 下面的颜色框体中选择适当的色彩 (选择绿色)。

完成后, 用户注意观察图 2.1 窗体下部的显示窗内的变化, 如果满意, 那么单击 OK 按钮, 否则单击 Cancel 按钮, 取消当前的操作。

用户可以使用该方法更改关键字、字符串、标识符等的色彩表示。

2.1.4 注释

在 Pascal 中有 3 种表示注释的方法:

1. 使用大括号表示。

```
{this is a comment line}
```

2. 使用带星号的小括号表示。