

Borland

Delphi 4.0

实用数据库编程



李晓勇 王平等编著
中国科学技术大学出版社

Delphi 4.0 实用数据库编程

李晓勇 王 平 等编著

中国科学技术大学出版社

内 容 提 要

Delphi 是由美国 Borland 公司推出的、在 Windows 95 和 Windows NT 下进行数据库系统开发的强有力的优秀软件。Delphi 所基于的语言是 Object Pascal, Object Pascal 是 Turbo Pascal 系列的升级语言,除了保留了 Turbo Pascal 的基本功能特色以外,主要增加了面向对象特征。Delphi 集成了可视化开发环境、具有优化了的 32 位编译器和由紧密集成可调整的数据引擎。

全书共分为 10 章,主要介绍 Delphi 的最新版本——Delphi 4.0 的新特性和在数据库方面的编程技巧。全书列举了许多实例(都经过作者调试通过),供读者借鉴和参考。

本书编写通俗易懂、深浅得当,对广大读者来说是一本不可多得的 Delphi 4.0 编程参考书,同时也可作为各类 Delphi 培训班的教科书。

图书在版编目(CIP)数据

Delphi 4.0 实用数据库编程/李晓勇等编著. —合肥:中国科学技术大学出版社,1999.7
ISBN 7-312-01117-9

I. D… II. 李… III. Delphi 语言-数据库管理系统-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 26928 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号 邮编:230026)

中国科学技术大学印刷厂印刷

全国新华书店经销

开本:787×1092/16 印张:26.75 字数:670 千

1999 年 7 月第 1 版 1999 年 7 月第 1 次印刷

印数:1—4000 册

ISBN 7-312-01117-9/TP·239 定价:42.00 元

目 录

第一章 Delphi 4.0 安装指南

- 1.1 引 言..... (1)
- 1.2 Delphi 4.0 安装指南..... (1)

第二章 Object Pascal 语言基础

- 2.1 Object Pascal 的基本元素..... (4)
- 2.2 Object Pascal 的数据类型..... (10)
- 2.3 关于 Object Pascal 的语句..... (36)
- 2.4 在 Object Pascal 中调用动态链接库 (DLLs)..... (66)
- 2.5 Object Pascal 的异常处理..... (71)

第三章 Delphi 4.0 编程环境

- 3.1 Delphi 4.0 的主屏幕..... (82)
- 3.2 Delphi 4.0 的系统菜单..... (102)

第四章 编辑器与调试器

- 4.1 编 辑 器..... (128)
- 4.2 代码编辑器的类完成与模块导航..... (132)
- 4.3 使用编辑器编辑一个小程序..... (133)
- 4.4 Delphi4.0 的调试工具..... (136)
- 4.5 小 结..... (148)

第五章 利用 Delphi 创建 Internet 应用程序

- 5.1 创建自己主机上的本地服务器 Localhost..... (150)
- 5.2 HTTP 和 HTML 语言..... (153)
- 5.3 利用 Delphi 生成 HTML 文件..... (154)
- 5.4 在 Delphi 中利用 CGI 技术编程..... (162)
- 5.5 将 CGI 应用程序转换为 ISAPI DLL..... (167)
- 5.6 Web Server 与数据库的连接..... (170)
- 5.7 使用窗体与客户交互..... (173)
- 5.8 在 Delphi 中使用 ActiveForm..... (180)
- 5.9 调试 Web 服务器应用程序..... (197)

5.10 小 结.....	(199)
---------------	-------

第六章 用 Delphi 开发数据库应用程序

6.1 Delphi 数据库系统概述	(200)
6.2 Delphi 数据库应用程序的开发步骤	(201)
6.3 BDE 和 SQL Links 简介	(202)
6.4 学习使用 Form Wizard.....	(206)
6.5 学习使用 Database Desktop 创建自己的数据库表.....	(208)
6.6 基于表 Cominfo.db 的数据库应用	(211)
6.7 处理一个以上的数据库表（主要-明细数据库）	(214)
6.8 利用代码访问数据库	(217)
6.9 数据浏览部件的应用及编程	(241)
6.10 使用 QuickReport 制作报表.....	(246)
6.11 在 QuickReport 报表上创建图表.....	(260)
6.12 在 Delphi 中使用 SQL 编程.....	(265)
6.13 小 结.....	(277)

第七章 动态链接库（DLL）编程

7.1 DLL 在 Windows 中的作用	(278)
7.2 DLL 的访问和编写	(279)
7.3 在 DLL 中实现数据传输	(290)
7.4 利用 DLL 实现窗体重用	(304)
7.5 小 结.....	(318)

第八章 在 Delphi 中使用 OLE 技术

8.1 什么是 OLE	(319)
8.2 创建 OLE 客户的一般步骤	(321)
8.3 TOLEContainer 组件的属性与方法	(323)
8.4 OLE 自动化.....	(331)
8.5 小 结.....	(356)

第九章 动态数据交换（DDE）

9.1 DDE 原理和 Delphi 的 DDE 实现机制	(361)
9.2 开发 DDE 程序的一般步骤	(363)
9.3 简单的 DDE 范例.....	(364)
9.4 复制与粘贴 DDE 链接.....	(368)
9.5 使用计时器的 DDE 连接.....	(373)
9.6 小 结.....	(376)

第十章 创建 VCL 组件与 ActiveX 组件

10.1 关于创建 VCL 的一些基本概念	(377)
10.2 创建并安装组件	(378)
10.3 编写可视组件	(382)
10.4 修改现有组件——TComboBar	(392)
10.5 Clock 组件	(399)
10.6 ActiveX 组件概述	(404)
10.7 将可视组件转化为 ActiveX 组件	(406)
10.8 直接给 ActiveX 组件添加方法	(419)
10.9 在可改写的 Web 页上使用 ActiveX 组件	(420)
10.10 小 结	(422)

第一章 Delphi 4.0 安装指南

1.1 引言

Delphi 4.0 是 Windows NT、Windows 3.x 及 Windows 9.x 环境下的快速应用程序开发及数据库开发工具，是目前唯一的具有可视化开发环境和快速且优化了的 32 位编译器的数据库工具软件，同时它还具有紧密集成可调整的数据库引擎所提供的数据库管理能力。

1.2 Delphi 4.0 安装指南

本小节的目的在于帮助初学者成功地安装 Delphi 4.0 的程序开发环境，由于目前 Delphi 4.0 一般都是在 Windows 95 环境下使用，所以只讲在 Windows 95 环境下的安装过程，在其他环境下的安装类似，对于有经验的用户可略过此小节。

一、安装 Delphi 4.0 的软硬件环境

1. 软件环境

在微机系统上必须安装 Windows 9.x、Windows NT 或 Windows 3.x 操作系统，建议选用 Windows 95 操作系统。

2. 硬件环境

(1) 至少 80486 的 CPU，8MB 内存空间，建议采用 Pentium 200 以上的 CPU 及 64 MB 内存空间。

(2) 1.2GB 硬盘，建议安装 5.1 GB 的 FireBall 硬盘。

(3) 一个 CD ROM 驱动器，建议安 4 倍速或 8 倍速的 CD ROM 驱动器，对于速度更高的光驱，一般对光盘兼容性较差。

(4) 同 Microsoft 相兼容的鼠标器。

二、Delphi 4.0 的安装步骤（以中文 Windows 95 操作系统为例）

(1) 启动中文 Windows 95。

(2) 把 Delphi 4.0 的安装光盘插入光驱中，稍后 Windows 95 将自动启动 Delphi 4.0 的安装画面，如图 1-1 所示。

(3) 单击图 1-1 中的 Delphi 4，则 Delphi 4.0 将进行安装向导的准备工作，稍后屏幕将显示图 1-2 所示的版权信息画面。

(4) 单击图 1-2 中的 Next 命令按钮，这时屏幕将显示软件许可协议画面，如图 1-3 所示，如果用户同意许可协议中的各项，可选择 Yes 命令按钮否则选 No 命令按钮。

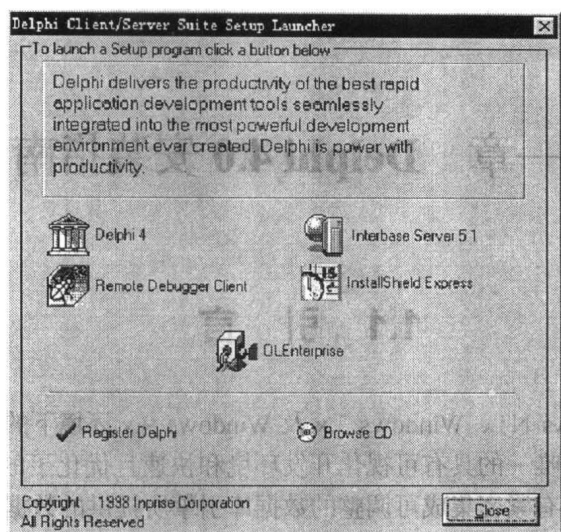


图 1-1 Delphi 4.0 安装画面一

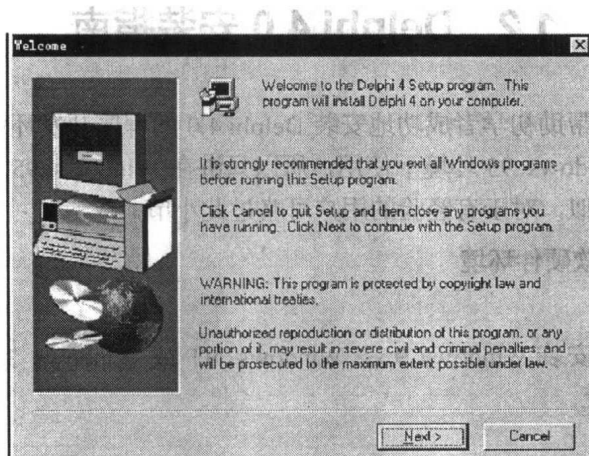


图 1-2 Delphi 4.0 安装画面二

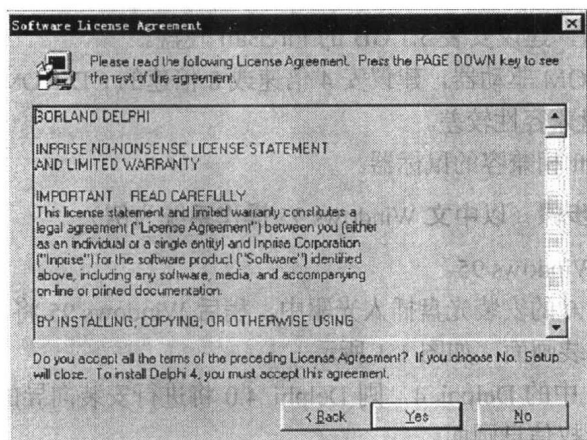


图 1-3 Delphi 4.0 安装画面三

(5) 单击 Yes 命令按钮, 稍后屏幕将显示安装信息画面, 如图 1-4 所示。

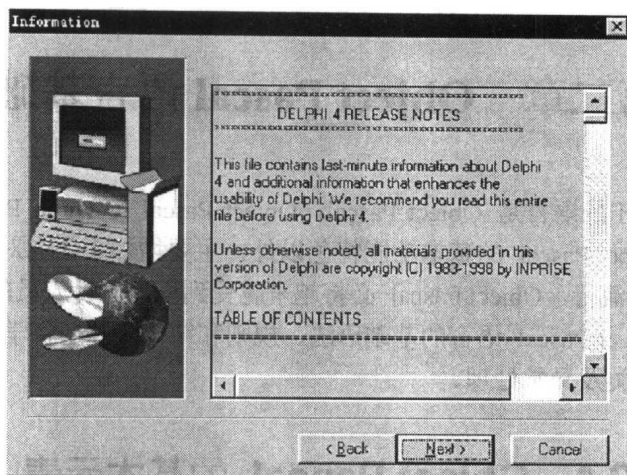


图 1-4 Delphi 4.0 安装画面四

(6) 单击图 1-4 中的 Next 命令按钮, 此时安装程序将进行一些初始化工作, 稍后屏幕将显示如图 1-5 所示的安装选择画面。

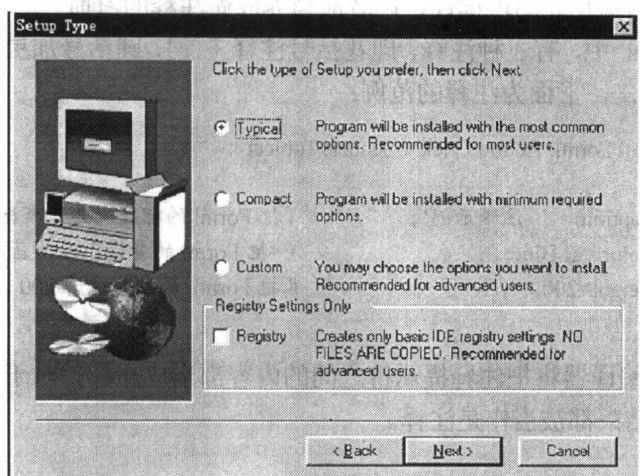


图 1-5 Delphi 4.0 安装画面五

下面介绍图 1-5 中的单选按钮的功能。

1. Full 单选按钮

选择此单选按钮后, 安装程序将把 Delphi 4.0 最常用的功能都安装到用户的硬盘上, 对于初学者, 建议采用此种安装方法。

2. Compact 单选按钮

选择此单选按钮后, 安装程序将只安装 Delphi 4.0 的必须内容, 对于硬盘空间较小或使用笔记本电脑的用户建议使用此种安装方式。

3. Custom 单选按钮

选择此种单选按钮将根据用户的选择作进一步的安装控制, 对于高级用户可使用此种安装方法。

对于后面的步骤将根据用户的不同选择而有不同的安装步骤, 用户只需根据屏幕提示进行操作即可。

第二章 Object Pascal 语言基础

Delphi 4.0 所基于的语言是 Object Pascal, Object Pascal 是 Turbo Pascal 系列的升级语言,除了保留了 Turbo Pascal 的基本功能特色以外,主要增加了面向对象特征,相信随着 Delphi 在全球的迅速流行, Object Pascal 必将越来越受到程序设计人员的喜爱。

本章将介绍 Object Pascal 语言的基础语法、规则和结构,并还将学到 Object Pascal 中的高级部分:单元、类及异常处理。

2.1 Object Pascal 的基本元素

一、注 释

注释主要用于增强程序的可读性,读者应当养成做注释的习惯。

在 Object Pascal 中,有 3 种注释,即花括号注释 { },圆括号加星号注释 (**)及 C++ 类型的双斜线注释 //,下面为注释的范例:

```
procedure TForm1. Button Click ( Sender: Tobject );
begin
    Caption:= ' 注释示例';           {把 Form1 的标题改为' 新标题' }
    Color:=clBlue;                   (*把 Form1 的背景色改为蓝色*)
    Height=200;                       // 把 Form1 的高度改为 200
end;
```

Object Pascal 编译器将把注释提示符之间的内容都当成注释;对于 C++类型注释来讲,从“//”后直到行结束都被当作是注释。

二、标识符

Object Pascal 语言包含有常量、变量、类型、过程、函数、单元等词法单位,用户在使用这些词法单位前,必须首先定义名称,标识符在这里指这些词法单位的名称。

在 Object Pascal 中,定义标识符时请注意以下各点。

- 标识符不区分英文字母的大小写,比如 FirstProc 与 firstproc 是等价的。
- 标识符只能以英文字母或下划线开头,不能以其他字符开头。
- 标识符的长度无限制,但只有前 63 个字符有效。
- 在标识符中不允许出现空格。
- 不能以 Object Pascal 的保留字(后面将进行介绍)作为标识符。
- 在 Object Pascal 中,许多对象都具有标识符的属性及方法等,用户在引用这些特性及方法时,一定要加上限定符号,如:

```
Panel1.Caption:= ' ... ' ;
Button1.Caption:= ' ... ' ;
```

在上面的示例中, 属性都为 Caption, 但一个属于 Panel 对象另一个属于 Button 对象。

三、变 量

Object Pascal 要求用户在过程、函数或程序开始前先说明所有的变量, 比如在 C++ 语言中的如下程序段:

```
void example ( void )
{
    int x=1;
    x++;
    int y=3;
    float f;
    ... ..
}
```

Delphi 中必须改写为:

```
procedure example;
var
    x, y: integer;
    f: real;
begin
    x:=1;
    inc ( x );
    y:=3;
    ...
end;
```

Object Pascal 的语法结构较 C++ 或 Visual Basic 更加容易阅读、可维护并具有更好的结构性。

Object Pascal 允许用户将多个相同类型的变量放在一行中进行定义, 如下所示:

```
x, y: real;
```

注意: 在 Object Pascal 中, 说明的变量都要放在类型标识符之前, 并且变量与类型之间要有一个冒号 “:”, 还有变量的初始化一般与变量的说明是分开的。

在 Delphi 4.0 中, 对于全局变量允许用户在变量说明部分进行初始化, 如下所示:

```
var
    i: integer=10;
    P: pointer=nil;
    s: string='This is a string';
    pi: real=3.1415929;
```

四、常 量

在 Pascal 中, 常量由 const 加以说明, 类似于 C 语言的 const 关键字, Pascal 与 Object Pascal 中常量的主要区别是 Object Pascal 的常量可以加上类型标识符, 而 Pascal 中的常量不能加上类型标识符, 例如:

```

const
  pi=3.1415926;
  m=10;
  errorstring=' Danger!' ;

```

在 Pascal 及 Object Pascal 中都可以，而下面的示例：

```

const
  pi:double=3.1415926;
  m:integer=10;
  errorstring:string=' Danger, Danger, Danger!';

```

只能在 Object Pascal 中出现。

在 Object Pascal 中，允许用户在 const 及 var 说明中使用编译时间函数，比如 ord ()、chr ()、 trunc ()、 round ()、 length ()、 low () 及 sizeof () 等等，下面的说明都是正确的。

```

type
  avr=array [ 1...10 ] of integer;
const
  w:word=sizeof ( byte );
var
  i:integer=9;
  j:smallint=ord ( ' a' );
  l:longint=trunc ( 3.1415929 );
  y:shortint=round ( 2.71828 );
  b:byte=high ( a );
  c:char=chr ( B )

```

五、Object Pascal 中的保留字和特殊符号

在 Object Pascal 中，保留字指 Object Pascal 语言中具有特殊及固定含义的单词，比如 if 、 begin 及 end 等，Object Pascal 不允许把保留字作为标识符。

在 Object Pascal 中，Delphi 4.0 的代码编辑器用黑体字显示保留字以示区别。

除了保留字外，Object Pascal 预定义了一些特殊符号，如下所示：

```

+   -   *   /   =   <   >   [   ]   .   ,   (   )
:   ;   ^   @   {   }   $   #   <=   >=   :=   ..
(*  *) (, .)

```

六、运算符

在 Object Pascal 中提供了如下几种基本运算（下面以优先级排列）：

最高优先级运算	@ Not
次优先级运算	* / div mod as and shl shr
再次优先级运算	+ - or xor
最低优先级运算	= <> < > <= >= in is

下面将分别介绍各种运算

1. 赋值操作符

赋值操作符 “:=” 用于为变量赋值，等价于 Visual Basic 中的 “=” 赋值操作符。比如：

```
count:=count + 1
```

表示把 count + 1 赋值给 count。

2. @运算符

@ 是一元运算符，用于获得操作数的地址指针，操作数可以是变量、过程、函数以及类型中的方法，下面为范例：

```
procedure changevalue (x:integer);  
var p:^integer;  
begin  
    p:= @x;  
    writeln (p^);  
    p^:=18;  
end;
```

比如主程序如下：

```
var  
    par:integer;  
begin  
    par:=19;  
    changevalue (par);  
    writeln (par);  
end;
```

在上面范例中，changevalue 过程先说明一个指向整型数的指针 p，然后使用@运算符把 x 地址赋值给 p 指针，并显示 p 所指向的数，然后再改变此数。

当操作数为变量时，@操作符返回指向此变量的指针。

当操作数是过程或函数的数值参数时，@操作符返回形式参数在堆栈中的地址。

当操作数是过程或函数的变参时，@操作符返回指向实参的地址。

当操作数为过程或函数时，@操作符返回过程或函数的入口。

3. 算术运算符

算述运算符是二元运算符，在 Object Pascal 中，共有六种算术运算符：+（加）、-（减）、*（乘）、/（除）、div（整除）及 mod（求余）。

对于+，-，*三个运算符，当两个操作数都是整数时，运算结果为整数，当两个操作数中只要有一个为实数，则运算结果为实数。

对于/运算符，不论操作数为整数或实数，其运算结果永远都为实数。

对于 div 及 mod 运算符，操作数必为整数，运算结果都为整数。

4. 位运算

位运算使用户能够修改变量的某个位，Object Pascal 共有六个位运算符：not、and、or、xor、shl 及 shr。

not（按位非） 把操作数的每一位取反，也就是 1 变 0，0 变成 1。

- and (按位与) 对于两个操作数按位求与, 当两个位都为 1, 结果为 1, 否则为 0。
- or (按位或) 当两个位中有一个位为 1 时, 结果为 1, 否则为 0。
- xor (按位异或) 当两个位中一个为 0, 另一个为 1 时, 结果为 1, 否则为 0。
- shl (左移) 例如 E1 shl E2, 表示把 E1 的各位向左移 E2 个位, 也就是乘以 2^{E2} 。
- shr (右移) 例 E1 shr E2, 表示把 E1 的各位向右移动 E2 位, 相当于 E1 整除 2^{E2} 。

5. 字符指针运算符

字符指针运算符为 +、-，分别用于使指针增大或减少一个偏移量，或求两个字符指针之间的距离。下面以实例形式说明字符指针运算符的用法，这里假设 p, q 为两个字符指针，i 为一个整型数：

- p + i 生成一个新指针，此指针指向 p 后的第 i 个字符。
- p - i 生成一个新指针，指向 p 前面的第 i 个字符。
- p - q 求两个指针之间的距离。

6. 逻辑运算符

逻辑运算符的操作数为布尔表达式，其运算结果也为布尔值，下面分别介绍 Object Pascal 的各种逻辑运算符：

- not (逻辑非) 把 true 变成 false, false 变成 true。
- And (逻辑与) 相当于汉语的“并且”的意思，当两个操作数的值都为 true，运算结果数为 true，否则结果为 false。
- or (逻辑或) 相当于中文“或者”的含义，当两个操作数中有一个的值为 true 时，运算结果为 true，否则运算结果为 false。
- xor (逻辑异或) 当两个操作数的值相异（即一个为 true，另一个为 false）时，运算结果为 true，否则运算结果为 false。

7. 关系运算符

关系运算符将对两个同类型的操作数进行比较，其运算结果为布尔值，下面分别对于 Object Pascal 中的各种关系运算符进行介绍。

- = : 判断两个操作数是否相等，操作数可以为简单类型、类、类引用、指针、集合、字符串或可变类型。
- <> : 用于判断两个操作数是否不相等，操作数可为简单类型、类、类引用、指针、集合、字符串或可变类型。
- <: 用于判断左侧的操作数是否小于右侧的操作数。操作数可为简单类型、字符串、pchar 或可变类型。
- >: 用于判断左侧的操作数是否大于右侧的操作数，操作数可为简单类型、字符串、pchar 或可变类型。
- <=: 用于判断左侧的操作数是否小于等于右侧的操作数，操作数可以为简单类型、字符串、pchar 或可变类型。
- >=: 用于判断左侧的操作数是否大于等于右侧的操作数，其操作数可以为简单类型、字符串、pchar 或可变类型。

<=: 用于集合类型的比较, 相当于数学中的“包含于”的概念。

>=: 用于集合类型的比较, 相当于数学中的“包含”的概念。

8. 字符串运算符

+ 运算符用于把两个字符串相联, 其结果为字符串, 当两个操作数都为短字符串时, 相联后的结果如果超过了 255 个字符, 则超过部分将被截除, 如有一个操作数为长字符串, 则运算结果也为长字符串。

9. 类型强制转换符

在 Object Pascal 中, 提供了一个专门用于类型强制转换符 As, 使用格式如下:

对象 As 对象类型

例如:

```
procedure TForm2.Click ( Sender:Tobject );
begin
    if Sender Is TButton
        then ( Sender AS TButton ).Height:=50;
end;
```

上述示例中, 通过 As 运算符把 Sender 转移成 TButton 类型。

10. 运行期类型识别操作符

Is 为 Object Pascal 提供的用于在运行期访问对象类型的操作符, 运算结果为一个布尔值。

Is 的使用格式为:

对象 Is 对象类型

当对象为指定的对象类型或此对象类型的继承类型时, Is 的运算结果为 true, 否则为 false。

Is 实际上是一种特殊的关系运算符, Is 通常用于 If 语句的条件部分, 下面以实例形式加以说明:

```
procedure TForm2. Click ( Sender: Tobject );
begin
    if Sender Is TButton
        then TButton ( Sender ) . Height:=50;
end;
```

11. 集合运算符

集合运算符用于进行集合类型的数据进行某些操作, 下面分别对各个集合运算符进行介绍。设 A, B 为两个集合, x 为集合中的元素。

A + B (并) 属于集合 A 或集合 B 的元素所组成的集合。

A * B (交) 同时属于集合 A 与集合 B 中的元素所组成的集合。

A - B (差) 属于集合 A 同时不属于集合 B 的元素所组成的集合。

x In A (属于) 如 x 在集合 A 中, 则运算结果为 true, 否则运算结果为 false。

12. 可变类型的操作

在 Object Pascal 中, 提出了一种新的数据类型 Variant (可变类型), 支持此种类型的操作有 +、-、*、/、div、mod、shl、shr、and、or、xor 等等。

2.2 Object Pascal 的数据类型

在 Object Pascal 中, 变量必须有确定的类型, 数据类型确定了变量值的范围及其变量所能进行的运算, 所以变量类型是非常重要的, 下面为 Object Pascal 的六大类数据类型:

- **简单类型** 包含实型及有序类型。
- **字符串类型** 包括有短字符串类型、长字符串类型。
- **构造类型** 包括数组类型、文件类型、记录类型、集合类型、类类型及类引用类型等。
- **指针类型**
- **过程类型** 把过程与函数看作为一个对象类型。
- **可变类型 (Variant)** 此种类型在运行期间是可变的。

一、简单类型

简单类型包括实型和有序类型, 有序类型包括数型、布尔数型、枚举类型和字符类型, 下面将分别进行讨论。

1. 有序类型

在 Object Pascal 中, 有序类型的特点是其值有一个整数序号, Object Pascal 共有如下 12 个预定义的有序类型:

integer	shortint	Smallint	longint
byte	word	cardinal	boolean
bytebool	wordbool	longbool	char

此外有序类型还包括两个用户自定义类型: enumerate 及 subrange。

对于有序类型, 系统提供有如下一些函数:

- ord () 返回参数的序号。
- pred () 返回参数在有序类型中的前一个序号的数, 如此数是有序类型的第一个数, 一般将产生一个异常。
- succ () 返回参数在有序类型中的后一个序号的数, 如果此数已经是最后一个数, 一般将产生一个异常。
- low () 返回有序类型中的最小值。
- high () 返回有序类型中的最大值。

(1) 整型

在有序类型中, integer 为所有符号整型数的统称, cardinal 为所有无符号的整数统称, 它们的字长由 CPU 及操作系统确定, 下面将对此进行讨论。

在十六位数操作系统中, integer 及 cardinal 的字长及值域如表 2-1 所示。

在三十二位操作系统 (Windows 95 及 Windows NT) 环境下, integer 及 cardinal 的字长和值域如表 2-2 所示。

表 2-1 16 位操作系统的 integer 及 cardinal 字长及值域

类型名	值 域	字 长
Integer	-32768 ~ 32767	有符号 16 位长
Cardinal	0 ~ 65535	无符号 16 位长

表 2-2 32 位操作系统下 integer 及 cardinal 字长及值域

类型名	值 域	字 长
Integer	-2147483648 ~ 2147483647	有符号 32 位长
cardinal	0 ~ 2147483647	无符号 32 位长

用户应尽量多使用 integer 及 cardinal 类型来说明类型。

如用户要精确地控制字长及值域, 可使用如表 2-3 所示的类型进行说明。

表 2-3 其他整数类型的字长及值域

类型名	值 域	字 长
shortint	-128 ~ 127	有符号 8 位长
cardinal	0 ~ 65535	无符号 16 位长
longint	-2147483648 ~ 2147483647	有符号 32 位长
byte	0 ~ 255	无符号 8 位长
word	0 ~ 65535	无符号 16 位长

对于表 2-3 中所示的各种数据类型, 字长及值域和 CPU 及操作系统无关。

对整型数据的算术运算, Object Pascal 规定:

- 对于整型常量, 精确类型为包含此常量值在内的值域的最小的类型。
- 对于两个不同类型的操作数进行算术运算时, 系统将首先将此二整型数转化为包含此同二种类所有值在内的值域最小的类型。
- 在进行赋值时, 赋值号右侧的表达式的计算独立于赋值号左侧的类型。

对于整型数来说, 用户可以使用十六进制数表示, 表示方法为在数的前面加上 \$ 符号, 如 \$20 为十进制数的 32。

(2) 布尔类型

布尔类型通常用于关系运算及条件语句中, 在 Object Pascal 中具有四种布尔类型, 如表 2-4 所示。

对于 boolean 类型来说, 值为 0 (false) 或 1 (true), 而对于 bytebool、wordbool 和 longbool 可为有符号的整数, 当值为 0 时, 表示 false, 而当值为非 0 时, 表示 true。