

跟我学系列丛书



Delphi 3.0

Delphi

游宏跃 王 帅 等编著

中国科学技术大学出版社

跟我学 Delphi 3.0

游洪跃 王 帅 等编著

中国科学技术大学出版社

1998. 合肥

TP3

图书在版编目 (CIP) 数据

跟我学 Delphi 3.0/游宏跃 王帅等编著. —合肥: 中国科学技术大学出版社, 1998 年 10 月

ISBN7-312-01042-3/TP · 215

- I 跟我学 Delphi 3.0
- II 游宏跃 王帅 等编著
- III ①Delphi 3.0 ②安装应用
- IV TP

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: 25.25 字数: 624 千
1998 年 10 月第 1 版 1998 年 10 月第 1 次印刷
印数: 1—5000 册

ISBN7-312-01042-3/TP · 215

定价: 35.00 元

内 容 简 介

Delphi 3.0 是 Windows NT、Windows 3.x 及 Windows 95 环境下的快速应用程序开发及数据库开发的工具，Delphi 3.0 是目前唯一的具有可视化开发环境、快速并优化了的 32 位编辑器及紧密集成可调整的数据库引擎所提供的数据库管理能力，它具有更快的速度和更好的执行代码，强大的数据库处理能力、全面支持 Interxet 服务、支持控件对象模型 (COM) 等特征。

本书编写通俗易懂、深浅得当，既可作为广大用户的参考书，也可作为各类 Delphi 培训班的教科书。

前 言

随着科技的发展，电脑的应用越来越广泛。掌握电脑知识是现代年轻人的基本要求，为了普及电脑知识，我们推出了跟我学系列丛书。本套丛书共计6本如下：

跟我学 Visual FoxPro 5.0

跟我学 Visual Basic 5.0

跟我学 Delphi 3.0

跟我学 Windows 98

跟我学 PowerBuilder 6.0

跟我学 Visual C++ 5.0

本套丛书主要从实际角度出发，针对于初、中级计算机爱好者来编写的。本套丛书编写通俗易懂，特别适合作为全国各大专院校在校大学生的教科书。

参加本套丛书编写的老师有：游宏跃、王帅、欧阳、左劫、殷华蓓、宋晓梅、陈兴蜀、黄平等。在本套丛书的编写过程中，得到了中国科学技术大学出版社于文良老师的大力支持，在此表示深切的谢意。

由于时间仓促和作用水平有限，书中错误难免，望读者指正。

编著

1998年9月

目 录

第一章 Delphi 3.0 概述及安装指南	(1)
1.1 Delphi 3.0 的特点	(1)
1.2 Delphi 3.0 安装指南	(1)
第二章 Object Pascal 语言	(5)
2.1 Object Pascal 的基本元素	(5)
2.2 Object Pascal 的数据类型	(11)
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 Wim 32 介绍	(84)
3.1 Wim 32 中级对象	(84)
3.2 Wim 32 系统中的多任务及多线程	(86)
3.3 Wim 32 系统中的内存管理	(86)
3.4 Wim 32 系统中处理异常的方法	(88)
第四章 Delphi 3.0 编程界面	(89)
4.1 Delphi 3.0 的主屏幕	(89)
4.2 Delphi 3.0 的系统菜单	(108)
第五章 开发 Delphi 3.0 程序	(135)
5.1 复习 Object Pascal 程序结构	(135)
5.2 介绍 Delphi 3.0 的工程管理器	(136)
5.3 有关 Delphi 3.0 控件的操作	(139)
5.4 Delphi 3.0 应用程序实例	(140)
第六章 Delphi 3.0 常用工具	(149)
6.1 Delphi 3.0 的对象浏览器	(149)
6.2 Delphi 3.0 的图像编辑器	(151)
6.3 Delphi 3.0 的数据库向导	(154)

第七章 有关窗体的属性、事件及菜单	(159)
7.1 属性和事件	(159)
7.2 Delphi 3.0 的菜单设计	(170)
7.3 Delphi 3.0 的菜单设置器及其实例	(181)
7.4 在窗体中增加工具栏及状态栏	(189)
第八章 Delphi 3.0 的绘图功能	(194)
8.1 图形对象 Shape	(194)
8.2 使用画布进行画面	(197)
8.3 Delphi 3.0 的有关图形对象	(204)
第九章 动态数据交换 (DDE)	(210)
9.1 DDE 会话	(210)
9.2 在 Delphi 中开发 DDE 程序	(211)
9.3 一个 DDE 数据交换的具体应用	(212)
第十章 对象的链接与嵌入 (OLE)	(216)
10.1 OLE 客户	(216)
10.2 OLE 自动化	(217)
10.3 Delphi 3.0 中 OLE 的特点	(217)
第十一章 在 Delphi 3.0 中设计 Windows 风格的联机帮助系统	(221)
11.1 一般原则.....	(221)
11.2 HECF 帮助文件的详细制作	(221)
11.3 在 Delphi 中调用 HLP 文件	(226)
第十二章 多文档界面 (MDI)	(230)
12.1 MDI 的框架 Form	(230)
12.2 MDI 的子 Form	(231)
12.3 合并菜单.....	(232)
12.4 安排打开的子窗口.....	(233)
12.5 与 MDI 有关的几个运行期属性	(235)
12.6 快速生成 MDI 框架	(235)
第十三章 Delphi 3.0 中调试程序	(238)
13.1 出错情况分类.....	(238)
13.2 调试前的准备工作.....	(239)
13.3 定制调试器的显示颜色.....	(240)
13.4 使用调试器.....	(240)

13.5	断点.....	(241)
13.6	观察窗口.....	(243)
13.7	表达式.....	(245)
第十四章	Delphi 3.0 数据库管理系统.....	(246)
14.1	BDE 概述	(246)
14.2	数据库应用程序轻松入门	(248)
14.3	Database Explorer	(277)
14.4	Database Desktop	(281)
14.5	表.....	(286)
14.6	查询.....	(293)
14.7	打印报表.....	(297)
14.8	数据模块.....	(304)
第十五章	Delphi 控件库 (UCL)	(308)
15.1	VCL 概念	(308)
15.2	Standard 页上的控件.....	(309)
15.3	Additional 页上的控件	(318)
15.4	Win32 页上的控件.....	(326)
15.5	System 页上的控件	(336)
15.6	Internet 页上的控件	(343)
15.7	Date Access 页上的控件	(346)
15.8	Data Controls 页上的控件.....	(356)
15.9	Decision Cube 页上的控件	(362)
15.10	QReport 页上的控件	(365)
15.11	Dialogs 页上的控件	(373)
15.12	Win 31 页上的控件	(377)
15.13	Samples 页上的控件	(381)
15.14	Active X 页上的控件	(386)
第十六章	Delphi 在多媒体中的应用	(388)
16.1	MediaPlayer 控件	(388)
16.2	媒体播放器应用实例.....	(393)

第一章 Delphi 3.0 概述及安装指南

1.1 Delphi 3.0 的特点

Delphi 3.0 是 Windows NT、Windows 3.x 及 Windows 9.5 环境下的快速应用程序开发及数据库开发的工具，Delphi 3.0 是目前唯一的具有可视化开发环境、快速并优化了的 32 位编译器及紧密集成可调整的数据库引擎所提供的数据库管理能力，具体地讲，Delphi 3.0 有如下的新特征。

1. 具有更快的速度及更小的执行代码

根据著名的 Pc Week 译测，Delphi 3.0 版所生成的可执行代码比 Visual Basic 5.0 版的编译代码大约快 3~6 倍左右，并且速度可以达到最快的 C++ 编译生成代码的 70%。

2. 强大的数据库处理能力

在 Delphi 3.0 中，对数据库的支持得到了更进一步的强化，并且 Client/Server 版的 Delphi 3.0 更提供了功能强大的服务器端的纠错程序，程序设计人员可以使用此纠错程序方便地跟踪客户程序的请求，从而轻而易举地完成 Client/Server 程序的调试。

3. Active X

Delphi 3.0 能全面地支持 Active X，用户可以创建许多功能强大 Active X 控件和 Active X 文档。

4. 全面支持 Internet 服务

在 Delphi 3.0 中提供了多达 15 个 Internet 相关的控件，能同时支持 Internet Explorer 的 ISAPI 及 Natesape D 的 NSAPI 动态链接接口扩充协议，支持 FTP、HTTP、NNTP、POP、SMTP、TCP 及 UDP 等协议，几乎涵盖了 Internet 的各个方面。

5. 支持控件对象模型 (COM)

Delphi 3.0 提供了自动支持微软提供的控件对象模型 (Component Object Model) 跨平台对象间通讯标准。

6. 新增报表及其他控件

在 Delphi 3.0 中，新增了 Quick Report 报表编写控件，Decision Cube 决策图控件及 Tee Chart 图形显示控件等。

1.2 Delphi 3.0 安装指南

本小节的目的在于帮助初学者成功地安装 Delphi 3.0 的程序开发环境，由于目前 Delphi 3.0 一般都是在 Windows 95 环境使用，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 3.0 的软硬件环境

1. 软件环境

在微机系统上必须安装 Windows 9x、Windows NT 或 Windows 3.x 操作系统，建议选用 Windows 99 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 32Mb 内存空间。

②1.2GB 硬盘，建议安装 3.2GB 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 Microsof 相兼容的鼠标器。

二、Delphi 3.0 的安装步骤（以中文 Windows 95 操作系统为例）

步骤一：启动中文 Windows 95。

步骤二：把 Delphi 3.0 的安装光盘插入光驱中，稍后 Windows 95 将自动启动 Delphi 3.0 的安装画面，如图 1.1 所示。

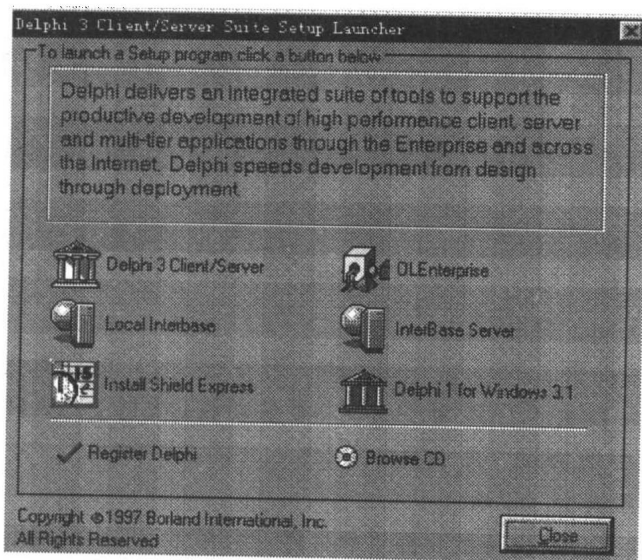


图 1.1 Delphi 3.0 安装画面一

步骤三：单击图 1.1 中的 Delphi 3 Client/server，则 Delphi 3.0 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。

步骤四：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。

步骤五：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。指示用户必须删除已安装过的任何 Delphi 版本，并建议采用该版本 Delphi 所配的反安装程序进行反安装。

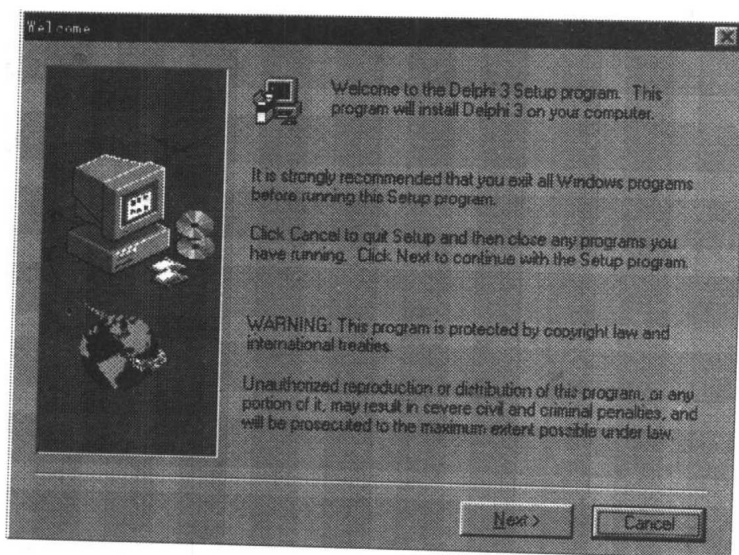


图 1.2 Delphi 版权信息画面

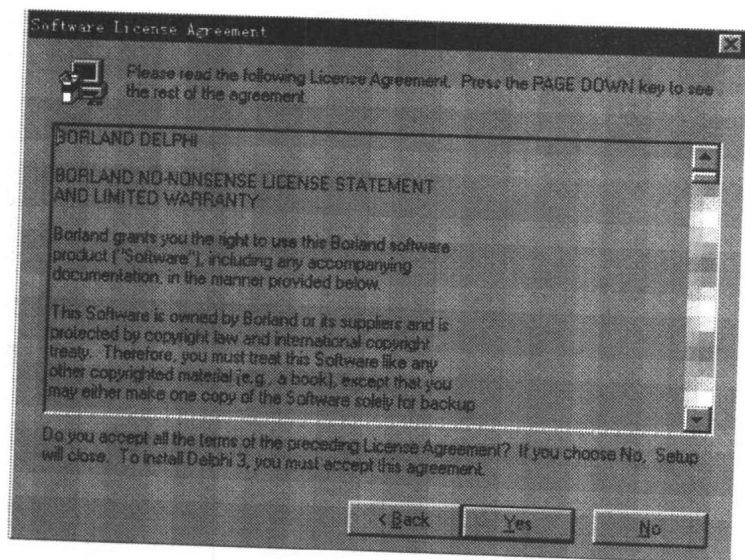


图 1.3 许可协议画面

步骤六：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。

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 3.0 全部功能都安装到用户的硬盘上，目前大硬盘愈来愈流行的今天，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 3.0 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

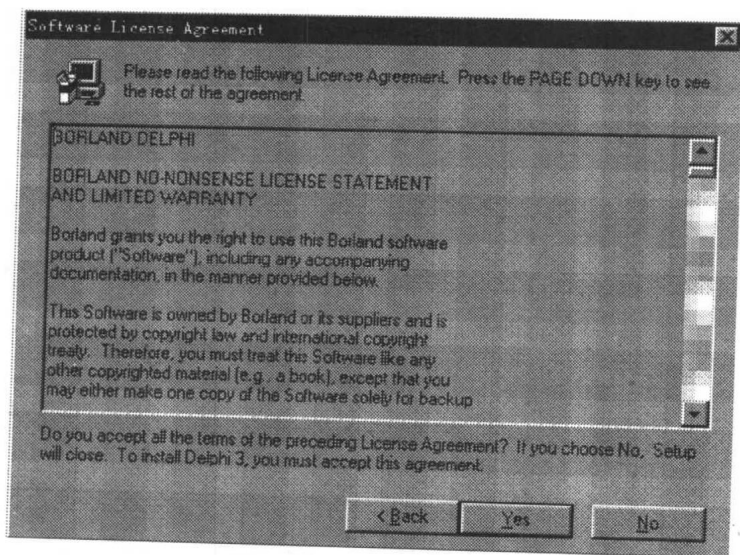


图 1.4 安装信息画面

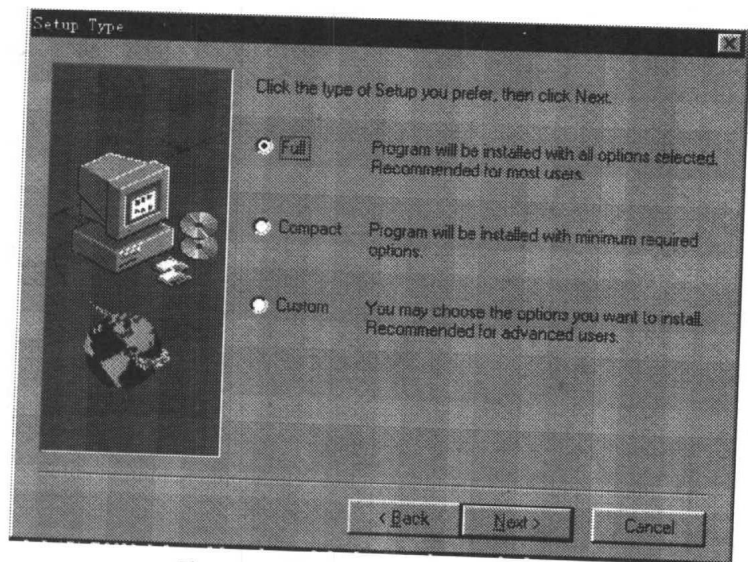


图 1.5 Delphi 3.0 安装选择画面

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

Delphi 3.0 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,用户应当养成做注释的习惯。

在 Object Pascal 中,共有 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: =' 新标题 ';           {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,必须首先定义标识符,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。

1. 标识符不区分英文母的大小写,比如 FirstProc 与 firstpro 是等价的。
2. 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。
3. 标识符的长度无限制,但只有前 63 字符有效。
4. 在标识符中不允许出现空格。
5. 不能以 Object Pascal 的保留字(后面将进行介绍)作为标识符。
6. 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:

```
Panel 2. Caption: ='...';
```

```
Button 2. Caption: ='...';
```

在上面的示例中, 属性都为 Caption, 但一个属于 Panel 2 对象另一个属于 Button 2 对象。

三、变量

Object Pascal 要求用户在过程、函数或程序开始前说明所有的变量, 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 更加容易阅读及可维护。

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 3.0 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: painter=nil;
```

```
    s: string='This is a string';
```

```
    r: real=3.1415926;
```

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等，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, Danger, Danger!';
```

在 Pascal 及 Object Pascal 中都可以，而下面的示例：

```
const
  pi: clouble=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 ()、lindh ()、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 (3.1415926);
  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 For Begin 及 End 等，Object Pascal 不允许把保留字作为标识符。

在 Object Pascal 中，Delphi 3.0 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

(一) 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。

(二) @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。

(三) 算术运算符

算术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。

(四) 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 $E1 \text{ shl } E2$, 表示把 $E1$ 的各位向左移 $E2$ 个位, 也就是乘以 2^{E2} 。

shr (右移): 例 $E1 \text{ shr } E2$, 表示把 $E1$ 的各位向右移动 $E2$ 位, 相当于 $E1$ 整除 2^{E2} 。

(五) 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。

(六) 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。

(七) 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相当, 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,