**Turbo Pascal**

Turbo Pascal是Borland公司一个很有代表性的软件开发产品。从语言角度说Turbo Pascal是Pascal语言的一个扩展。它扩展了标准Pascal的功能特性。在语法上，Turbo Pascal吸取了其他语言如C等的特性，简化了标准Pascal的语法；在功能上，Turbo Pascal提供了包括屏幕控制、图形处理、系统调用等有用的函数库（Turbo Pascal中称为单元）。



Turbo Pascal包括CP/M、MS-DOS、Windows等不同操作系统上的版本。

1983年，Borland公司发布了Turbo Pascal的第一个版本。该版本的编译器核心部分由安德斯·海尔斯伯格授权给Borland公司。同时，安德斯·海尔斯伯格也作为雇员加入了Borland公司，并且是后来所有Turbo Pascal版本与Delphi前3个版本的架构师。在第一个版本的Turbo Pascal中，Borland公司的创始人Philippe Kahn为该版本的Turbo Pascal添加了用户界面与编辑器。第一版本Turbo Pascal是个人电脑上编译器发展的一个里程碑。

1989年发布的Turbo Pascal 5.5版是一个重要的版本，从这一版本起，Borland公司把OOP引入Pascal语言，这就是大家所知道的Object Pascal的开端。

Turbo Pascal 6.0推出一个用OOP思想封装的Dos下的应用程序框架（Framework），叫做TurboVision，这个版本的IDE据信即是使用该Framework开发的。

Turbo Pascal的最后一个版本叫做Borland Pascal 7，它包含一个增强的Dos下的IDE和编译器，可以创建Dos和Windows3.x程序，后来为Delphi1.0代替。

1983年11月20日 Turbo Pascal version 1.0发布   
1986年9月17日 Turbo Pascal version 3.02 发布   
1989年5月2日 Turbo Pascal version 5.5 发布

目录

• [关于Turbo Pascal](http://www.techcn.com.cn/index.php?doc-view-140972.html#1)

• [Pascal程序语言入门](http://www.techcn.com.cn/index.php?doc-view-140972.html#3)

• [PASCAL语言的发展](http://www.techcn.com.cn/index.php?doc-view-140972.html#5)

• [数据格式](http://www.techcn.com.cn/index.php?doc-view-140972.html#7)

* • [主要语句：](http://www.techcn.com.cn/index.php?doc-view-140972.html#9)
* • [从Turbo Pascal到Delphi](http://www.techcn.com.cn/index.php?doc-view-140972.html#11)
* • [参考文献](http://www.techcn.com.cn/index.php?doc-view-140972.html#13)

关于Turbo Pascal[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)



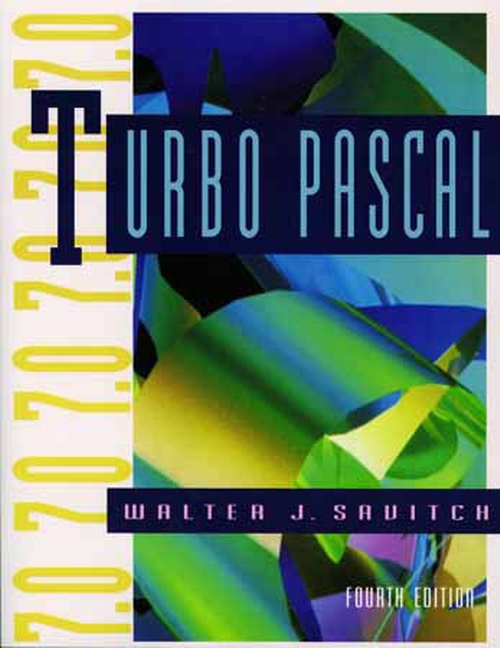
　　Pascal是一种计算机通用的高级程序设计语言。它由瑞士Niklaus Wirth教授于六十年代末设计并创立。   
　　以法国数学家命名的Pascal语言现已成为使用最广泛的基于DOS的语言之一，其主要特点有：严格的结构化形式；丰富完备的数据类型；运行效率高；查错能力强。   
　　正因为上述特点，Pascal语言可以被方便地用于描述各种算法与数据结构。尤其是对于程序设计的初学者，Pascal语言有益于培养良好的程序设计风格和习惯。IOI(国际奥林匹克信息学竞赛)把Pascal语言作为三种程序设计语言之一， NOI(全国奥林匹克信息学竞赛)把Pascal语言定为唯一提倡的程序设计语言，在大学中Pascal语言也常常被用作学习数据结构与算法的教学语言。 现在由于版权等问题，在NOIP中使用Free Pascal，而在NOI和IOI中使用Lazarus 0.9.10和Linux系统。  
　　在Pascal问世以来的三十余年间，先后产生了适合于不同机型的各种各样版本。其中影响最大的莫过于Turbo Pascal系列软件。它是由美国Borland公司设计、研制的一种适用于微机的Pascal编译系统。该编译系统由1983年推出1.0版本发展到1992年推出的7.0版本，其版本不断更新，而功能更趋完善。

Pascal程序语言入门[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)

　　Pascal是由瑞士Niklaus Writh教授于60年代末设计并创立。它是一种计算机通用的高级程序设计语言，是一种优良的程序设计教学语言，整个语言紧凑整齐，概念简洁，数据结构和控制结构比较丰富，程序可读性好，查错能力强，有利于培养学生严谨、清析的程序设计风格和良好习惯，并使他们从中领会和理解精细的程序设计技巧。尤其引入了结构化程序设计的思路，充分考虑了算法设计中的自顶向下、由粗到精的模块化思想，以及总可以将算法步骤剖解为顺序、分支、重复三种基本结构流程的结构化原则，提供了丰富的数据类型和清晰的描述模块化的语句。  
　　Turbo Pascal系列软件作为开发系统软件与就任软件及实施科学计算和教学的有力工具，下发挥着越来越大的作用。也是国际和全国青少年信息学奥林匹克竞赛指定的语言之一。从历届信息学竞赛的情况看，它是最能出成绩和选手最欢迎的语言。以后的例子就以Turbo Pascal 7.0进行程序设计。  
　　下面我们就以一个实例来看一看Pascal程序的结构，从中认识到Pascal语言程序的书写方式，以及其规范的标准设计方式。

PASCAL语言的发展[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)

　　在Pascal问世以来的三十余年间，先后产生了适合于不同机型的各种各样版本。其中影响最大的莫过于Turbo Pascal系列软件。它是由美国Borland公司设计、研制的一种适用于微机的Pascal编译系统。该编译系统由1983年推出1.0版本发展到1992年推出的7.0版本，其版本不断更新，而功能更趋完善。   
　　Turbo Pascal语言是编译型程序语言，它提供了一个集成环境的工作系统，集编辑、编译、运行、调试等多功能于一体   
　　Pascal有5个主要的版本，分别是Unextended Pascal、Extended Pascal、Object-Oriented Extensions to Pascal、Borland Pascal和Delphi Object Pascal。其中，Unextended Pascal、Extended Pascal和Object-Oriented Extensions to Pascal是由Pascal标准委员会所创立和维护的，Unextended Pascal类似于瑞士Niklaus Wirth教授和K.Jensen于1974年联名发表的Pascal用户手册和报告，而Extended Pascal则是在其基础上进行了扩展，加入了许多新的特性，它们都属于正式的Pascal标准；Object-Oriented Extensions to Pascal是由Pascal标准委员会发表的一份技术报告，在Extended Pascal的基础上增加了一些用以支持面向对象程序设计的特性，但它属于非正式的标准。Borland Pascal和Delphi Object Pascal是由Borland公司专门为其开发的编译工具设计的Pascal语言，前者是用于DOS的Turbo Pascal系列和Windows 3.x的Turbo Pascal for Windows的传统高级语言，后者是用于Windows的Delphi和Linux的Kylix的面向对象程序设计语言，它们都不是正式的Pascal标准，具有专利性。但由于Turbo Pascal系列和Delphi功能强大并且广为流行，Borland Pascal和Delphi Object Pascal已自成为一种标准，为许多人所熟悉



数据格式[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)

　　例1：判断这个数是奇数还是偶数。  
　　它的Pascal程序如下:  
　　program e5\_1( input, output) ; {程序首部}  
　　var {变量说明}  
　　a:longint {在tp7中，规定pi约等于3.1415926536}  
　　begin {程序开始}  
　　readln(a); {打入数字}  
　　if a mod 2 <>0{判断是否为奇数}  
　　then writeln(a"-jishu") {输出a-jishu  
　　else writeln(a"-oushu");或a-oushu}  
　　end. {结束程序}  
　　从以上简单的例子可以看出，Turbo Pascal程序是由程序首部、程序说明部分和程序执行部分组成。具体如下所示：  
　　program 程序名; {程序首部，可省略}  
　　说明部分 {说明部分}  
　　begin {程序开始}  
　　语句1； {执行语句}  
　　语句2； {执行语句}  
　　…… {执行语句}  
　　End. (End;){结束程序}  
　　上面程序由如下两部分组成:  
　　1、程序首部  
　　程序首部是程序的开头部分，由保留字program后，接程序名及程序参数表组成，结束时一定要有分号。程序名yzhch是用户自己定义的标识符，参数表一般是文件变量名，用于该程序与外界的数据交流。最常用的参数为input和output。Turbo Pascal程序首部中参数表可以省略。  
　　2、程序说明部分  
　　Pascal语言要求用户将在程序中所使用的标号、常量、类型、变量、记录、文件、以及过程和函数除了Pascal自己预先定义的标准量之外，都必须在说明部分说明后才能在程序执行部分使用。但各个内容部分是可选的，只有执行程序部分需要的时候才进行说明。  
　　3、程序执行部分  
　　紧接着说明部分的begin和end之间的部分为程序的执行部分。它由一系列语句组成，一条语句执行一定的功能，所有语句完成程序设计的任务。语句之间用“;”隔开，允许一行写多个语句，也允许一个语句写多行。最后一行的end后加“.”号表示结束。所跟其后的语句将无任何作用。Begin与end应配对出现，这是每一个Turbo Pascal程序都必须的。  
　　注意：后面将学习到的语句中，也需要引用begin和end作为程序段的分隔标记，但其必须遵守语句规则。  
　　字符集和符号  
　　一、基本字符  
　　（1）字母：A到Z，a到z和\_（下划线）  
　　（2）数字：0、1、2、3、4、5、6、7、8、9  
　　（3）特殊字符：+、-、\*、/、=、^、<、>、(、)、[、]、{、}、.、:、;、'、#、$、@  
　　赋值运算符：:=  
　　关系运算符：>、<、<>、=、>=、<=  
　　子界分隔符：..  
　　括号：[]  
　　注释号：{}  
　　二、符号  
　　（一）保留字  
　　保留字是由字母拼成的字，Turbo Pascal语言预先规定了他们的意义，不允许被再定义。因此，不能作为用户自定义的标示符，否则Turbo Pascal编译器将提示出错信息。Turbo Pascal共有保留字50多个，如下表所示。  
　　保留字 含义 保留字 含义  
　　And 逻辑“与”运算 Mod 求余运算  
　　Asm 汇编语句调用 Nil 空指针  
　　Array 定义数组 Not 逻辑“非”运算  
　　Begin 复合语句结构部分 Object 对象定义符  
　　Case 情况语句结构部分 Or 逻辑“或”运算  
　　Constructor 构造对象定义 Packed 紧缩数组定义  
　　Destrutor 销毁对象定义 Procedure 过程说明符  
　　Div 整除运算 Program 程序首部定义  
　　Do For循环语句结构部分 Record 记录定义  
　　Downto For循环语句结构部分 Repeat Repeat 循环语句结构部分  
　　Else 条件语句结构部分 Set 集合定义  
　　End 复合语句结构部分 Shl 二进制左移运算符  
　　Exports 动态链接库输出项目定义 Shr 二进制右移运算符   
　　File 文件定义 String 字符串定义  
　　For For循环语句结构部分 Then 条件语句结构部分  
　　Function 函数定义 To For循环语句结构部分  
　　Goto 无条件转移语句 Type 类型说明  
　　If 条件语句结构部分 Unit 多个程序模块说明  
　　Implementation 单位实现部分起始标识符 Until Repeat 循环语句结构部分  
　　In 属于 Uses 单元说明  
　　Inherited 对象继承说明符 Var 变量定义  
　　Inline 机器代码插入指令 While While循环语句结构部分  
　　Interface 外挂单元文件接口说明 With 开域语句结构部分  
　　Label 标号定义 Xor 逻辑“异或”运算  
　　Library 动态链接库DLL编译指令  
　　数据类型、常量的变量  
　　一、数据类型的概念  
　　计算机处理数据对象是一个广义的概念。例如，125、12.76是数据，’xiang qj zhong’这一串字符也是数据。前者是数值数据，后者是字符串数据，是非数值数据。显然，为了表示这些数据，它们在内存中必须以不同方式存放。为处理这些数据，计算机对它们施加的运算也不同。为此，Turbo Pascal语言建立了数据类型的概念，对描述的数据进行分类。每一种数据类型定义了一个具有相同性质的数据集合。各种数据类型的数据具有不同的性质。程序中所用到的每一个数据，包括常量和变量都有一个和它相联系的类型。由此决定了数据所具有的值，也决定了对该数据所能进行的操作。  
　　Turbo Pascal语言中数据具有丰富的类型，按它们的特点可以分为简单类型、构造类型、指针类型和过程类型四大类，如图下所示。  
　　其中，标准类型用语言系统预先定义的标准标识符表示，整型用integer表示，实型用real表示，布尔型用boolean表示，字符型用char表示。  
　　二、常量  
　　常量是指在程序中使用的一些具体的整型数、实型数和字符串。  
　　(1)整型数：如9、3、-5、0等。  
　　(2)实型数：如3.1、-6.1E+20等。  
　　(3)字符串：是用单引号括起来的一串字符，如，’book’、’96•5’、’ABC’等。  
　　以上列举的都可以作为常量在程序中使用。为了提高程序的可读性并使程序便于修改，在程序中往往用一些标识符来代表具体的常量。  
　　在Turbo Pascal语言中，可以给一些常量取个名字用一个标识符代表它，这就是常量定义。例如，Cost=60；Blank=’ ’。  
　　经常量定义的标识符又称为常量标识符。  
　　在Turbo Pascal语言中，常量定义要写在常量定义部分中。  
　　常量定义部分的一般形式:  
　　Const  
　　(常量标识符1)= (常量1);  
　　(常量标识符2)=(常量2);  
　　(常量标识符n)=(常量n);  
　　Const是保留宇，表示开始一个常量定义部分，其后可以有若干个常量定义，这些  
　　常量定义之间要用“；”号分隔。例如：  
　　Const  
　　Cost=60;  
　　A=Cost+30;  
　　Pi =3.14159;  
　　Turbo Pascal语言对常量定义有如下要求:  
　　(1)常量定义要放在程序的常量定义部分，即程序首部之后，执行部分之前。  
　　(2)必须遵循先定义后使用的原则，即只有已经定义的常量标识符，才能在程序中  
　　使用。  
　　三、变量  
　　在程序执行过程中其值可以改变的数据，称为变量。每个变量都要有一个名称，这就是变量名。变量名由用户自己定义，但必须符合标识符的规定。  
　　在一个程序中，一个变量只能属于一种确定的数据类型。因此，程序中出现的每个变量都必须说明其数据类型，这样就规定了该变量的取值范围，也决定了对该变量所能执行的运算操作。  
　　变量的类型，可以是标准数据类型integer、real、boolean和char，也可以是用户自定义的各种类型。  
　　变量说明形式是：一个变量标识符或由逗号隔开的多个变量标识符在它的冒号"："后面说明其数据类型。  
　　在Turbo Pascal程序中，变量说明要写在变量说明部分中。  
　　变量说明部分的一般形式:  
　　var  
　　(变量说明1)；  
　　(变量说明2)；  
　　……(变量说明n)；  
　　其中var是保留字，表示一个变量说明部分开始。一个var可以含有多个不同的变量说明，每个变量说明之间用分号隔开，有时称被分号隔开的变量说明为变量说明项。例如：  
　　var  
　　x，y：real；  
　　chl：char；  
　　t，f：boolean；  
　　注意:不同类型的变量一般不能互相串用。  
　　这里还应指出，变量一经说明系统就在计算机内存中为其分配一个存贮空间。在程序中使用到变量时，就在相应的内存中存入数据或取出数据，这种操作称为变量的访问。  
　　标准数据类型  
　　Pascal向程序设计者提供了丰富的数据类型，它们用于专门的目的，但却都是由简单的、非构造型的数据类型所构成的。本节介绍Turbo Pascal中最为基本的几种数据类型：整型、实型、布尔型和字符型。它们都是系统定义的简单数据类型，称为标准数据类型，其对应的名字称为标准标识符。  
　　1、整型  
　　一个整型数据用来存放整数，整型数据可以是正整数、负整数和整数零。  
　　Turbo Pascal中的整型常数必须按规定严格书写。  
　　Turbo Pascal支持五种预定义整型，它们是短整型(Shortint)、整型(Integer)、长整型 (Longint)、字节型(Byte)和字类型(Word)，每一种类型规定了相应的整数取值范围以及所占内存字节数(一个字节为8个二进制位)。因此，用户在具体编程定义变量类型时，要根据它们的特点选用适当的类型，以达到理想的效果。当两个不同范围类型的操作数进行运算时，得到的结果属于较大范围的类型。如下表所示。  
　　Turbo Pascal语言规定可以对整型数据进行算术运算符+、一、\*、Div、Mod。  
　　它们分别表示加、减、乘、整除和取余。这五种运算，要求参加运算的两个数都是整型数，运算结果也是整型数。前三种运算与一般的算术运算加、减、乘相同。Div整除运算，是两个整型数相除取整数部分(商的整数部分)，得到整型结果。Mod取余运算，是两个整型数相除取余数，余数的符号与被除数符号相同。例如：  
　　3 Div 2 = 1 5 Div 7 = 0  
　　6 Div (-4) = - 1 (-12) Div (-5) = 2   
　　7 Mod 4 = 3 (14) Mod (-4) = 2  
　　(-18) Mod (-6) = 0 6 Mod l7 = 6  
　　由此可见，a Mod b，所得结果的符号与a相同，其值(绝对值)在0～∣b∣-1之间。运算符Mod与Div之间有如下关系：  
　　a Mod b = a – (a Div b) \* b (b<>0)  
　　其中Mod运算的结果的符号与a的符号相同。  
　　利用以上两种运算可以对正整数进行分离。例如：  
　　n为四位数8531，可用下法分离出它的个、十、百、千位。  
　　8531 Mod l0 = 1 (个位数)  
　　(8531 Mod l00) Div l0 = 3 (十位数)  
　　(8531 Mod l000) Div l00 = 5 (百位数)  
　　8531 Mod l000 = 8 (千位数)  
　　利用 a Mod b可以判断a能否被b整除。当a Mod b = 0时，a能被b整除。  
　　2、实型  
　　一个实型数据用来存放实数。实型数据可以是正实数、负实数和实数零。实型数据一般用小数或指数形式(亦称科学表示法)表示。例如：  
　　+1993，33，3.5E+5(=3.5×10５)， -0.5E-3(=-0.5×10-３，)，-20.0，，0.0等都是合法实型数。  
　　Turbo Pascal支持一种预定义实型，它们是基本实型(Real)、单精度实型(Single)、双精度实型(Double)、扩展实型(Extended)和装配实型(Comp)。每一种类型规定了相应的实数取值范围和所占内存字节数，以及它们所能达到的精度，即有效数字位数。因此，用户在具体编程时应根据以上的参数适当选用，以达到最佳效果。如下表所示。  
　　对于此类实型数据，若其绝对值大于上界，则产生上溢；绝对值小于下界，则产生下溢，下溢导致结果为0。Comp类型的取值范围是-263+1～238-1之间的整数，相当于十进制的-9.218～9.218。由于Comp类型的数据表示成二进制形式的数，这种类型的变量有时处理起来比较方便，特别对于数值很大的整数间的计算，这种数据类型很有用。  
　　Turbo Pascal语言允许实型数使用下列运算符进行运算。  
　　运算符：+、-、\*、/  
　　分别表示加、减、乘和除。其中"/"叫实数除，即使两个整型数相除，其结果也总是实型，如： 7/2=3.5 6/3=2.0  
　　3、字符型  
　　用标准标识符Char标明字符型。字符型数据可以是字母、符号、数字(0-9)等ASCII码的所有字符。Turbo Pascal支持扩展ASCII码，共包括256个字符。但非印刷字符是不能在标准显示上显示或打印输出。在计算机内部，字符集的元素是以该元素在字符集内的顺序位置来标记的，位置取值范围为0~255，我们称这些整数为字符在字符集内的序数值或序号。每个字符型数据在内存中占一个字节。将字符用单引号括起来，即成字符常数，如，’X’，’7’，’?’。字符常数可按字符的序数值确定大小关系，也就是说它们的大小由它们所对应的ASCII码值决定，如：’Y’，’Z’，’A’<’a’。   
　　由于采用ASCII码，字符依ASCII码序号排列。这样，字符与ASCII码序号有一一对应的映射关系。  
　　4、布尔型  
　　一个布尔型数据用来存放逻辑值，或称布尔值。Turbo Pascal支持预定义布尔型，以标准标识符Boolean表示。Boolean一词，系根据19世纪英国数学家George boole (1815-1864)的名字而得，George boole为现代布尔代数之父。布尔型数据的值只有两个：True(逻辑真)和False(逻辑假)。布尔型是顺序类型，规定False<TRUE，FALSE的序号为0，TRUE的序号为1。  
　　逻辑运算的结果只有两个：True（真）和False（假）。Turbo Pascal提供了六种关系运算符和三种逻辑运算符：  
　　=(等于)、<（小于）、<=（小于等于）、>（大于）、>=（大于等于）、<>（不等于）  
　　NOT（非）、AND（与）、OR（或）  
　　运算关系  
　　函 数  
　　标准函数。Turbo Pascal语言提供了自变量为整型量的标准函数有顺序函数算术函数和转换函数等。  
　　标准函数是Turbo Pascal语言预先定义的，它们实际上是能完成特定功能的称步子程序的程序段。每个标准函数都用一个标识符来标识，每个标准函数都能完成一个特定的功能，在程序中可以直接调用它们。Turbo Pascal语言中某些标准函数与数学中的函数有相似之处。  
　　一、整数类型函数  
　　整型是顺序类型，即所有的整型数都是按一定的顺序排列的。如3的后序数是4，350的后序数是351。以后介绍的布尔型、字符型、枚举类型和子界类型等都是顺序类型。顺序函数可以对顺序类型数据进行操作，但要注意它们自变量的取值范围。  
　　①前趋函数：Pred(x)函数值为x-l，例如：  
　　Pred (6)=5 Pred (-21)=-22  
　　②后继函数：Succ (x)函数值为x+l，例如：  
　　Succ (l5)=16 Succ (-114)= -113   
　　③绝对值函数：Abs (x)函数值为︱X︱，例如：   
　　Abs (-119)=119 Abs (101)=101  
　　④平方函数：Sqr (x)函数值为X2，例如：  
　　Sqr (-5)=，25 Sqr (l0)= 100  
　　以上四个函数的结果仍是整型数。  
　　⑤奇函数：Odd (x)，函数的结果为布尔型。当X为奇数时，函数值为true；当X为偶数时，函数值为false。例如：  
　　Odd (13)= True Odd (16)= False  
　　⑥字符函数：Chr (X)，函数值是序号的ASCII字符，属字符型。例如：  
　　Chr (65)=’A’ Chr (32)=’ ’  
　　二、实数类型函数  
　　在下列算术函数中，X可以是实型或整型数的表达式。对于函数Abs和Sqr，其结果类型和变量X的类型相同，其他算术函数的结果类型都是实型。  
　　绝对值函数Abs(x)：函数值为x的绝对值  
　　平方函数Sqr (x)：函数值为x的平方  
　　小数函数Frac (x)：函数值为x的小数部分  
　　整数函数Int (x)：函数值为x的整数部分  
　　正弦函数Sin (x)：函数值为x的正弦，其中的单位为弧度  
　　余弦函数Cos (x)：函数值为x的余弦，其中的单位为弧度  
　　指数函数Exp (x)：函数值为e（x）  
　　对数函数Ln (X)：函数值为x的自然对数  
　　平方根函数的Sqrt (x)：函数值为x的平方根  
　　反正切函数Arctan(x)：函数值为x的反正切，单位为弧度  
　　随机函数Random：无自变量时，函数值取(0,1)间的随机小数；有自变量且为Word类型时，函数值取(0，自变量)间的随机整数。  
　　三、字符类型函数  
　　Turbo Pascal语言提供如下自变量为字符型的标准函数，其中Chr为字符型。  
　　后继函数Succ (ch)：例如，Succ (’8’)=’9’ Succ (’E’)=’F’  
　　对字符集的最后一个字符，Succ函数无意义。  
　　前趋函数Pred (ch)：例如，Pred (’7’)=’6’ Pred (’B’)=’ A’  
　　序数函数Ord (ch)：:给出字符ch在ASCII字符集中的序号，结果为整型。  
　　注意：Ord (’7’)<>7，正确的是：Ord (’7’)=Ord(’0’)+7=48+7=55  
　　若ch是数字字符，则Ord (ch)-Ord ('0')是该数字字符的数值。例如：Ord (’7’)-Ord(’0’)=7  
　　前面介绍的字符函数Chr (i)是Ord (ch)的逆函数。例如:  
　　Chr (55)= ’7’ Chr (Ord(’A’))=’A’  
　　三、布尔类型函数  
　　Turbo Pascal语言提供布尔型函数主要是几个字符型函数。  
　　Ord (B) 例如：Ord (false)=0 Ord (true)=1  
　　表达式  
　　运算是对数据进行加工处理的过程，得到运算结果的数学公式或其它式子统称为表达式。表达式可以是常量也可以是变量或算式，在表达式中又可分为：算术表达式、逻辑表达式和字符串表达式。  
　　1、算术表达式：  
　　算术表达式是最常用的表达式，又称为数值表达式。它是通过算术运算符来进行运算的数学公式。我们先来看Visual Basic中的算术运算符：  
　　算术运算符  
　　运算符 表达式 说 明 举 例  
　　\* X\*Y 求X乘Y的值 6\*7=42  
　　/ X/Y 求X除Y的值（浮点数运算） 2.76/1.2=2.3  
　　div X div Y 求X除Y的整数商（对整型数计算） 25=5  
　　Mod X mod Y 求X除Y的余数（对整型数运算） 25 mod 4=1  
　　+ X+Y 加法运算 32+2=34  
　　- X-Y 减法运算 48-21=27  
　　由于Visual Basic只能识别按其格式书写的数学表达式，所以必须将我们常用的数学表达式转换成Visual Basic表达式。例如：  
　　数学式 Visual Basic表达式  
　　注:Turbo Pascal中没有乘方函数,但乘方运算可通过指数函数Exp (x)进行,如计算x^y,有  
　　x^y=e^(y\*lnx)=Exp(y\*Ln(x))  
　　2、逻辑运算  
　　逻辑运算的结果只有两个：True（真）和False（假）。Visual Basic提供了六种关系运算符和三种逻辑运算符：  
　　=(等于)、<（小于）、<=（小于等于）、>（大于）、>=（大于等于）、<>（不等于）  
　　NOT（非）、AND（与）、OR（或）  
　　运算关系  
　　p  q  NOT p  p AND q p OR q  
　　True  True  False  True True  
　　True  False  False  False True  
　　False  True True False True  
　　False  False  True False False  
　　例如：5>3 结果为 True, “a”>”b” 结果为False。  
　　3、表达式的运算优先顺序  
　　在进行表达式的转换过程中，必须了解各种运算的优先顺序，使转换后的表达式能满足数学公式的运算要求。运算优先顺序为：  
　　括号→函数→乘、除→加、减→字符连接运算符→关系运算符→逻辑运算符  
　　如果同级的运算是按从左到右次序进行；多层括号由里向外。  
　　例：  
　　（10+6）\*3^2\*COS(1)/2\*8+7  
　　① ④ ③ ⑤ ② ⑥ ⑦ ⑧  
　　Sqrt(Abs(p/n-1))+1   
　　④ ③ ① ② ⑤

主要语句：[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)

　　赋值语句：<变量>:=<表达式>；  
　　输入语句；Read([<文件>,]<变量1>,[<变量2>,[…]]);或ReadLn([<文件>,][<变量1>,[<变量2>,[…]]]);  
　　输出语句：格式同上，Write Writeln  
　　判断语句；if <Boolean表达式> then <语句块1> else <语句块2>  
　　循环语句：  
　　FOR循环1：FOR <顺序型变量>:=<初值> TO <终止量> DO <语句块>;  
　　FOR循环2：FOR <顺序型变量>:=<初值> DOWNTO <终止量> DO <语句块>;  
　　注：FOR循环1 是初值的ascii码值小于等于终止量的（即从小到大），  
　　FOR循环1 是初值的ascii码值大于等于终止量的（即从大到小），  
　　WHILE循环：WHILE <Boolean表达式> DO <语句块>;  
　　REPEAT-UNTIL循环：REPEAT <语句块>; UNTIL <Boolean表达式>;  
　　复合语句（语句至少2句时使用）：begin  
　　（语句1）；  
　　（语句2）；  
　　.....  
　　（语句n）；  
　　end.  
　　pascal编程时常见错误：  
　　1 无效DoS功能号   
　　2 文件末找到   
　　3 路径未找到   
　　4 打开文件过多   
　　5 禁止文件存取   
　　6 无效文件句柄   
　　12 无效文件存取代码   
　　15 无效驱动器号   
　　16 不能删除当前目录   
　　17 不能跨驱动器改文件名   
　　100 Disk read error 磁盘读错误   
　　101 Disk write error 磁盘写错误   
　　102 文件变量末赋值   
　　103 File not open 文件未打开   
　　104 File not open for input 文件未用输入方式打开   
　　105 File not open for output 文件末用输出方式打开   
　　106 Invalid numeric format 无效数字格式   
　　150 磁盘写保护   
　　151 未知单元   
　　152 驱动器未准备好   
　　153 未知命令   
　　154 数据CRC校验错   
　　155 驱动器请求的结构长度错   
　　156 磁盘定位错   
　　157 未知媒介类型   
　　158 扇区末找到   
　　159 打印机缺纸   
　　160 设备写失败   
　　161 设备读失败   
　　162 硬件故障   
　　200 Division by zero 被零除   
　　201 Range check error 数据范围检查错   
　　202 Stack overflow error 堆栈溢出错   
　　203 Heap overflow error 堆溢出错   
　　204 Invalid pointer operation 无效指针操作   
　　205 Floating point overflow 浮点上溢出   
　　206 Floating point underflow 浮点下溢出   
　　207 Invalid floating point operation 无效浮点运算   
　　208 未安装覆盖管理程序   
　　209 覆盖文件读错   
　　210 Object not initialized 对象未初始化   
　　211 Call to abstract method 调用抽象方法   
　　212 Stream registration error 流登计错   
　　213 Collection index out of range 集合下标越界   
　　214 Collection overflow error集合溢出   
　　215 Arithmetic overflow error 算术上溢错误   
　　216 General protection fault 存取非法   
　　217 控制-C   
　　218 授权指令   
　　219 无效的TYPECAST   
　　220 无效的变体TYPECAST   
　　221 无效的变体操作   
　　222 没有变体方法调用DISPATCHER   
　　223 不能建立变体数组   
　　224 变体不包含数组   
　　225 变体数组边界错误   
　　226 TLS初始化错误   
　　这些是编译错误：   
　　l内存溢出   
　　2缺标识符   
　　3标识符未定义   
　　4标识符重定义   
　　5语法错误   
　　6实型常量错   
　　7整型常量错   
　　8字符串常量跨行   
　　9文件嵌套过多   
　　10非正常文件结束   
　　11行过长   
　　12缺类型标识符   
　　13打开文件过多   
　　14无效文件名   
　　15文件未找到   
　　16磁盘满   
　　17无效编译指示   
　　18文件过多   
　　19指针定义中未定义类型   
　　20缺变量标识符   
　　21类型错误   
　　22结构过长   
　　24文件分量不能为文件   
　　25无效字符串长度   
　　26类型不匹配   
　　27无效子界基类型   
　　28下界大于上界   
　　29缺有序类型   
　　30缺整型常数   
　　31缺常数   
　　32缺整型或实型常数   
　　33缺指针类型标识符   
　　34无效的函数结果类型   
　　35缺标号标识符   
　　36缺BEGIN   
　　37缺END   
　　38缺整型表达式   
　　39缺有序表达式   
　　40缺布尔表达式   
　　41操作数类型与操作符不匹配   
　　42表达式错   
　　43非法赋值   
　　44缺字段标识符   
　　45目标文件过长   
　　46未定义外部标识符   
　　47无效\*.OBJ文件记录   
　　48代码段过长   
　　49数据段过长   
　　50缺DO   
　　51无效PUBLIC定义   
　　52无效EXTRN定义   
　　53EXTRN定义过多   
　　54缺0F   
　　55缺INTERFACE   
　　56无效重定位引用   
　　57缺THEN   
　　58缺T0或DOWNTO   
　　59未定义的向前引用   
　　60过程过多   
　　61无效类型转换   
　　62被零除D   
　　63无效文件类型   
　　64不能读写该类型的变量   
　　65缺指针变量   
　　66缺字符串变量   
　　67缺字符串表达式   
　　68单元循环引用   
　　69单元名不匹配   
　　70单元版本不匹配   
　　71单元重名   
　　72单元文件格式错误   
　　73缺IMPLEMENTATl0N   
　　74常数与CASE类型不相匹配   
　　75缺记录变量   
　　76常数越界   
　　77缺文件变量   
　　78缺指针变量   
　　79缺整型或实型表达式   
　　80标号不在当前块中   
　　81标号已定义   
　　82标号未定义   
　　83无效参数   
　　84缺UNIT   
　　85缺“；”   
　　86缺“：”   
　　87缺“，”   
　　88缺“(”   
　　89缺“)”   
　　90缺“＝”   
　　91缺“：＝”   
　　92缺“[”或“(．”   
　　93缺“]”或“．)”   
　　94缺“．”   
　　96变量过多   
　　97无效FOR控制变量   
　　98缺整型变量   
　　99此处不允许用文件和   
　　100字符串长度不匹配   
　　101无效字顺序   
　　102缺字符串常数   
　　103缺整型或实型变量   
　　104缺有序变量   
　　105INLINE错   
　　106缺字符表达式   
　　107重定位项过多   
　　112CASE常量越界   
　　113语句错   
　　114不能调用中断过程   
　　116必须在8087方式下编译   
　　117末找到目标地址   
　　118此处不允许包含文件   
　　120缺NIL   
　　121无效限定符   
　　122无效变量引用   
　　123符号过多   
　　124语句部分过长   
　　126文件必须为变量参数   
　　127条件符号过多   
　　128条件指令错位   
　　130初始条件定义错   
　　13l过程和函数头与前面定义的不匹酉   
　　132严重磁盘错误   
　　133不能计算该表达式   
　　134表达式错误结束   
　　l35无效格式说明符   
　　136无效间接引用   
　　137此处不允许结构变量   
　　138无SYSTEM单元不能计算   
　　l39不能存取该符号   
　　140无效浮点运算   
　　141不能将覆盖编译至内存   
　　142缺过程和函数变量   
　　143无效过程或函数引用   
　　144不能覆盖该单元   
　　147缺对象类型   
　　148不允许局部对象类型   
　　149缺VIRTUAL   
　　150缺方法标识符   
　　151不允许虚拟构造方法   
　　152缺构造方法标识符   
　　153缺释放方法标识符   
　　154FAIL只允许在构造方法内使用   
　　155无效的操作符和操作数组合   
　　156缺内存引用   
　　l57不能加减可重定位符号   
　　158无效寄存器组合   
　　159未激活286／287指令   
　　160无效符号引用   
　　161代码生成错   
　　162缺ASM

扩展阅读：   
1.ASCII码众所周知全称为“美国信息交换标准码，American Standard Code for Information Interchange”。   
2.实例   
3.让我们到幕后去看看，看看ASCII码是如何以数字形式表示文本的。举2个例子：   
4.(1) ASCII码'A' -- 其内存存储字节2进制表示为"01000001" --- 其16进制值为0x41 --- 其10进制值为65(这里的值实际上是'A'在ASCII码表中编号)；   
5.验证过程：   
6.char c = 'A';   
7.printf("%c\n", c);/\* A \*/   
8.printf("%x\n", c);/\* 41 \*/   
9.printf("%d\n", c);/\* 65 \*/   
10.(2) ASCII码'6' -- 其内存存储字节2进制表示为"00110110" --- 其16进制值为0x36 --- 其10进制值为54(这里的值实际上是'6'在ASCII码表中的编号)；   
11.char c = '6';   
12.printf("%c\n", c);/\* 6 \*/   
13.printf("%x\n", c);/\* 36 \*/   
14.printf("%d\n", c);/\* 54 \*/   
15.ASCII码通讯   
16.利用ASCII码作为通讯方式到底是一种什么样的通讯方式呢？（FTP协议中有两种通讯方式，其中一种是ASCII码方式，即文本方式）这里也举例说明：比如我们要传送数值123, 123数值用16进制表示为0x7b，以二进制表示为01111011，那么以二进制方式通讯，01111011就是我们真实传送的数据，但是如果以ASCII码方式通讯，则完全不同了，如果你还传送01111011的话，对方那边的得到的将是'{'('{'对应的ASCII码用16进制表示为7b)。那么我们该如何怎么传呢？正确的方式就是将123每位上的数字转化为其相应的ASCII码，然后传送。这里'1'、'2'和'3'对应的ASCII码用16进制表示分别为0x31、0x32和0x33。这样组合起来后要传送的数据应为"001100010011001000110011"。

从Turbo Pascal到Delphi[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)

我是在高一接触pascal语言，因为参加NOI的需要，顺理成章的要使用Turbo Pascal来写程序了。半年后，我开始想着如何编写Windows程序，又理所当然的找上Delphi。初见Delphi，除了begin，end让我觉得倍感亲切外，Object Pascal里的增加的面向对象的语法却让我很是吃惊，当时的我可根本不懂什么叫面向过程，面向对象；最可恶的是，国内那些教育家们，除了会拿着清华的那本精简的不能再精简的pascal教材照本宣科外，似乎再也没有什么实质性的工作了，传说中的《Turbo Pascal大全》更是无处可寻，所以关于unit，interface这些Delphi里随处可见的关键字我也很不明白。所幸，其后不久，我得到一本名为《计算机反病毒技术》的书，里面统统都是用Turbo Pascal编写的源代码，通过它我迅速明白了早已存在于Turbo Pascal中unit，interface等关键字的含义和用法，又以Delphi中的Help文档为扶手，开始蹒跚学步了。

印象中，国内Delphi作家似乎更偏爱编写应用实例类的技术书籍，至于语法这种东西，没有几个人愿意多去涉及，即使书中必须谈及，也是寥寥数笔，匆匆带过，或者干脆与某本书类似。对Object Pascal语法讲解最好，最权威的恐怕就算《Delphi5开发人员指南》了，这本书至今也是备受推崇的。但与如今泛滥的C++书籍相比，Delphi仍然逊色许多，也难怪很多新手特别是从来没有接触过pascal语言的新手，在学习Object Pascal时会遇到不少困难。自己的感觉是：在从Turbo Pascal向Delphi过渡的过程中，由于没有正确的指引，走了很多弯路；由于没有正确的桥梁，必须要一步实现大跨越。所以，在这里，我提出自己曾经遇见的沟壑，路标性给出我自己的认识和总结，希望给入门的同学们一些帮助。我不打算详细介绍语法知识，并假设你已经有一点pascal语言和面向对象概念的基础。要想学习相关详细知识，我推荐各位一定要阅读《Delphi开发人员指南》和Delphi Help文档中的相关章节。

●     Record，Class，Object

习惯了在一个Program模块内写完所有面向过程代码的我，有几天的时间一直未能彻底明白在非Unit模块中，自定义类的框架，语法是如何的，VCL源代码虽然经典，却过于繁杂，不能让我迅速掌握根本，我需要一个最简单又最能说明问题，完整的可运行的代码，苦于无处寻求答案，只好亲自动手，探索对应关系，终成其下两段代码。

program TP;

{本代码在Turbo Pascal 7.0下编译通过}

type

  MyRecord = record

     {...}

  end;

var

  MR: MyRecord;

  procedure MyProcedure;

  begin

    {MyProcedure Code}

  end;

{=========== main ===========}

begin

{以这个begin为标志，主程序开始，其作用相当于C/C++中的main函数}

  MyProcedure;

end.

上面是一段极其简单的包含记录类型声明和过程声明的代码，二者基本规则如下：用户自定义的数据类型，需要放在以保留字“type”开头的代码段中；过程(procedure)和函数(function)要放在以保留字“var”开头的代码段中；最后一个夹在begin和end间的代码段是主程序的开始，也就是整个程序的入口，作用相当于C/C++里的main函数,请注意，只有在以program保留字开头的代码模块中，这个begin和end才具有程序入口的作用。

再看下面的代码：

program Delphi;

{代码在Delphi7.0下编译通过}

{$APPTYPE CONSOLE}

uses

  SysUtils;

type

TMyClass = class(TObject)

  public

    constructor Create;

    procedure PrintClassName;

  private

    ClassName: string;

  end;

var

  MyClass: TMyClass;

  constructor TMyClass.Create;

  begin

    ClassName := 'TMyClass';

  end;

  procedure TMyClass.PrintClassName;

  begin

    writeln(ClassName);

  end;

{=========== main ===========}

begin

  MyClass := TMyClass.Create;

  MyClass.PrintClassName;

  MyClass.Free;

  readln;

end.

类作为用户自定义的一种数据类型，其声明的规则，成员函数、过程的实现方法都符合经典 Pascal的基本规则，唯一不同的是保留字变了，从记录体变成了类（详细比较代码结构和语法规则），这也说明Object Pascal是在经典Pascal的基础上进行了面向对象内容的语法扩充。当然，内部的运行机制并没有表面语法扩充这么轻松，可那是编译器的事情，在这里，我们完全不用理会。差点忘记告诉读者怎么调试上面的代码了：在IDE环境主菜单里选择 File | New | Other ，在New Item项里选择Console Application，这时出现了代码编辑框，再将上面的代码贴入，F9，完成！

代码内的{$APPTYPE CONSOLE}是一个编译开关，它告诉编译器，这是控制台程序，在格式上它与注释的差别就是那个“$”符号；TMyClass = class(TObject)可以简写为

TMyClass = class，表示TMyClass类从TObject类继承而来，TObject是Delpi中所有类的祖先，这也是为什么我在代码中没有声明Destroy过程却仍然能够使用的原因；Delphi中类的构造很有趣，请注意MyClass := TMyClass.Create这一句，这与C++不同[注1]。readln使程序停顿下来，直到用户按下回车键才结束程序退出。更多详细内容请参考《Delphi开发人员指南》2.17.1节。

   上面两段代码相互对应，虽然很简单，不过我却认为他们在某种程度上很容易让同学发现由经典Pascal向Object Pascal过渡的一些方法，对Object Pascal的类定义语法有个初步了解，这是很重要的一步。当初我要是能够看到这两段代码，或许能少浪费很多时间了。

另外，在TP5.5 –Delphi 7.0中都支持Object类型（对象类型）。语法如下：

object  
Field;  
Field;  
...  
Method;  
Method;  
end;

Method允许以下几种形式：

procedure MethodName(<parameter(s)>:type);  
或者function MethodName(<parameter(s)>:type):type;  
或者constructor MethodName(<parameter(s)>: type [;<parameter(s)>:type]); [virtual];  
或者destructor MethodName[(<parameters>: type)];[virtual];

不错的，构造函数和析构函数都支持virtual，在构造函数中，还有一个有用的东西是Fail函数，当构造函数的初始化失败时，它可以用来释放已经分配的资源。

接下来的代码，是Turbo Pascal的Help文档中，关于Fail函数的演示代码，可以让大家对此有个较深的认识。Turbo Pascal确实是很强大和优秀的。

type

 PBase = ^TBase;

 TBase = object(TObject)  {在这里就已经出现Tobject了，是不是很亲切？}

   constructor Init(FailMe: Boolean);

 end;

 PDerived = ^TDerived;

 TDerived = object(TBase)

   constructor Init(FailMe: Boolean);

 end;

constructor TBase.Init(FailMe: Boolean);

begin

 inherited Init;

 if FailMe then Fail;

end;

constructor TDerived.Init(FailMe: Boolean);

begin

 if not inherited Init(FailMe) then

{判断父类的初始化是否成功}

   { Ancestor failed to construct, we must fail too }

   Fail;

 { Otherwise, proceed with construction }

 {...}

end;

var

 P: PObject;

 X: Boolean;

begin

 for X := False to True do

 begin

   P := New(PDerived, Init(X));

   if P <> nil then

   begin

     writeln('Object constructed sucessfully');

     Dispose(P, Done);

   end

   else

     writeln('Object failed to construct');

 end;

end.

自定义的Object不一定要从Tobject继承下来，因此它也没有内建的构造、析构和其他方法，通常使用New过程和Dispose过程建立和销毁它的实例。Delphi中仍旧支持Object，并有property成员了，但是仍然不允许published。KOL[注2]也使用该类型实现自己的构架。但是对此关键字的保留主要是为了兼容，建议用户改用class关键字。

●  unit模块

在Turbo Pascal的Help文档里，是这样说明unit功能的：Units are the basis of modular programming inBorland Pascal. You use units to createlibraries and to divide large programs into logically related modules。小规模的程序，我们可能将所有代码集中在一个program中，可是面对更加复杂的大型工程时，正确的划分功能封装功能对代码管理和以后的维护有着重要的作用，而使用unit模块正好解决了这些问题。其语法规则如下：

unit identifier;  { Heading }

 interface  { Public symbols }:

  uses      { Uses clause }

  const     { Constants }

  type      { Types }

  var       { Variables }

  procedure { Procedures }

  function  { Functions }

 implementation   { Private symbols }:

  uses            { Uses clause }

  label           { Labels }

  const           { Constants }

  type            { Types }

  var             { Variables }

  procedure       { Procedures}

  function        { Functions }

 begin           { Initialization }

  statement;      { Statements }

    statement

 end.

Interface部分用来声明对外接口，也就是可以被外部引用该文件的程序使用的函数和过程；implementation部分包含接口部分声明的各种函数、过程具体实现的代码；begin一直到最后的end.之间都是初始化部分，可以为本unit内的各种变量，过程，函数初始化。如果没有内容需要初始化，那么begin可以省略，但end.必须存在。

在Delphi下经典Pascal中的unit部分有了变动，请看来源于Delphi Help文档的说明：

unit Unit1;

interface

uses

{ List of units goes here }

  { Interface section goes here }

implementation

uses

{ List of units goes here }

  { Implementation section goes here }

initialization

  { Initialization section goes here }

finalization

  { Finalization section goes here }

end.

可见Initialization部分的开始关键字begin，被Initialization取代了，并且增加了一个finalization部分。Initialization部分的代码可以这样写：

initialization

begin

  {do something…}

end;

也可以这样写：

initialization

  {do something…}

finalization部分的功能，有点类似于析构函数，它主要针对本unit模块中initialization部分初始化的资源进行释放，并且是在程序结束时运行，如果程序以Halt过程结束了，该部分的程序将不能执行。

Unit模块中的interface等关键字和结构初看似乎有些限制程序员的自由度，但也正是这种语法规定体现出Pascal语言的严谨和优美，为减少程序出错的几率做出保证。

program相当文章的提纲挈领，unit则是文章的各个段落。Delphi里，program模块包含在.prj文件中，unit模块包含在传统的.pas文件中。这就是为什么在Delphi中我们经常面对的是为各个窗口服务的unit模块，而非在一个program中写完所有代码。不过我在Delphi的Help中看到这样一句话：In traditional Pascal programming, all source code, including the main program, is stored in .pas files.不知道这算思考角度不同还是算bug，毕竟这种语法并非Delphi中才有，Turbo Pascal程序员也一直在采用这种方法组织程序结构。

给出一个简单的initialization例子。在菜单中选择File | New | Application，再在窗体上放置一个按钮，双击该按钮编写它的Click事件处理代码，完整代码如下：

unit Unit1;

interface

uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

  Dialogs, StdCtrls;

type

  TForm1 = class(TForm)

    Button1: TButton;

    procedure Button1Click(Sender: TObject);

  private

    { Private declarations }

  public

    { Public declarations }

  end;

var

  Form1: TForm1;

  Msg: string;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

  ShowMessage(Msg);

  Msg := 'second';

end;

initialization

  Msg := 'first';

end.

   以上所述，是过渡中两个基本的重要问题，弄懂它们方可初步明白自己为什么要这样编写代码，该在哪里编写代码，如何扩展代码功能。

●  我推荐的参考书籍

     Delphi参考书籍

《Delphi 5 开发人员指南》机械工业出版社

《Inside VCL》李维 电子工业出版社

《Delphi深度历险》陈宽达 科学出版社

《Pascal精要》网络下载电子版

Windows参考书籍

《Programming Windows》Charles Petzold

《Windows2000开发人员指南》中国水利水电出版社

Delphi组件参考书我暂时没有发现特别好的，平时我主要依靠论坛，源代码，Delphi自带的Demo和文档来学习组件的使用。

●     再说点

现在有一些大学取消了Pascal语言课程，去年的ACM大赛也取消了Pascal语言的使用，不禁心寒，启蒙教育没有人做了，这些都使得Delphi在学生中的处境更加艰难。在国内的Delphi论坛上，常常见到许多半吊子程序员在享受Delphi的快速开发的时候，嘴巴里还在责备Delphi功能太弱，不能搞什么底层开发，甚至直接责怪Pascal语言，殊不知，在Dos年代，有多少著名软件使用Pascal开发出来的呢，有多少底层控制程序有着Pascal的身影呢？现在我手头上还有Pascal编写的病毒代码，反病毒代码，IC芯片控制代码。

或许正是Delphi的RAD能力降低了程序开发的门槛，让很多半吊子进入了程序界，RAD开发蒙蔽了许多半吊子编程者（称呼他们为程序员或许稍欠火候）的眼睛，但是国内的教育界同样有着不可推卸的责任，选修课开VB的不少，讲Delphi的很少，我曾经还遇到过一个从心底里就瞧不起Delphi，不啻谈论的老师，而事实上他根本就没有用过Delphi。我现在常去外国网站，发现国内被争论不休的问题，在国外早就有人在做，并且做的非常棒。

似乎我在为Delphi翻案，不停的维护它，没辙，谁要我是它的fans呢，人总要有点信仰，喜好不是吗？所以我认为，与其花时间咒骂一个工具的“不足之处”，不如花时间研究如何解决这个问题。

参考文献[回目录](http://www.techcn.com.cn/index.php?doc-view-140972.html" \l "section)

<http://zh.wikipedia.org/wiki/Turbo_Pascal>  
<http://en.wikipedia.org/wiki/Turbo_Pascal>