

Turbo Pascal 5.0

(下) 参考手册

石 放 译
叶 欣 校

- 编辑、编译、调试、运行一体化的集成环境
- 与 Code View 相媲美的源级集成调试器
- 支持独立的 Turbo Debugger 调试器
- 覆盖使用户能解决 CCDOS 下的内存紧张问题
- 丰富的图形函数包
- 支持软件工程中标准的数据，实现隐藏
- 与其它语言接口极其方便

给您带来编程效益的飞跃

BH

北京航空航天大学计算中心软件开发部

目 录

第一部分 编程参考

Turbo Pascal 5.0 介 绍.....1

关于这本手册.....1

第一章 单词和常量.....2

特殊符号和保留字.....2

标识符.....3

标号.....3

数.....4

字符串.....4

常量说明.....4

常量表达式.....4

注释.....6

程序行.....6

第二章 块、位置和作用域.....7

语法.....7

作用域规则.....8

接口和标准标识符的作用域.....8

第三章 类型.....9

简单类型.....9

有序类型.....9

整型.....10

布尔型.....11

字符型.....11

枚举型.....11

子界型.....11

实型.....12

字符串型.....13

结构类型.....13

数组类型.....13

记录类型.....13

集合类型.....15

文件类型.....16

指针类型.....16

过程类型.....16

类型的一致性和兼容性.....17

类型一致性.....17

类型兼容.....17

赋值兼容.....17

类型说明部分.....18

第四章 变量.....20

量说明.....20

数据段.....20

堆栈段.....20

绝对变量.....21

变量引用.....21

限定符.....21

数组、字符串和下标.....22

记录 and 域指示符.....22

指针和动态变量.....22

变量强制类型转换.....23

第五章 类型常量.....25

简单类型常量.....25

字符串类型常量.....25

结构类型常量.....26

指针类型常量.....28

第六章 表达式.....29

表达式语法.....29

操作符.....32

算术操作符.....32

逻辑操作符.....32

布尔操作符.....33

字符串操作符.....33

集合运算符.....34

关系运算符	34
比较	35
集合元素资格测试	35
@操作符	35
函数调用	36
集合构造符	36
值的类型强制转换	37
 第七章 语句	38
简单语句	38
赋值语句	38
过程语句	38
goto 语句	38
结构语句	39
复合语句	39
条件语句	39
1. if 语句	39
2. case 语句	40
循环语句	41
1. repeat 语句	41
2. while 语句	41
3. for 语句	42
with 语句	43
 第八章 过程和函数	45
过程说明	45
提前引用说明	46
External 说明	46
inline 过程说明	46
函数说明	47
参数	48
值参	48
变量参	49
无类类型参数	49
过程类型	50
过程类型说明	50
过程变量	50
过程类型参数	52

表达式中的过程类型	53
 第九章 程序和单元	55
程序语法	55
程序头	55
uses 子句	55
单元语法	55
单元头	56
接口部分	56
实现部分	56
初始化部分	57
单元直接引用	57
单元递归引用	57
与其它说明共享	59
 第十章 输入和输出	60
I/O 简介	60
文件操作的标准过程和函数	60
文本文件处理的标准过程和函数	61
无类型文件处理的标准过程和函数	61
Turbo Pascal 设备	62
Dos 设备	62
CON 设备	62
LP1, LP2 和 LP3	62
COM1 和 COM2 设备	63
漏斗设备 Null Device	63
文本文件设备	63
 第十一章 标准过程和函数	64
流控制过程	64
动态分配过程和函数	64
转换函数	64
算术函数	64
序数过程和函数	65
串过程和函数	65
指针和地址函数	66
杂类过程和函数	66

第十二章 标准单元	67	函数	93
标准单元的依赖关系	67	Turbo3 单元	94
System 单元	67	接口部分	94
Printer 单元	70	kbd	95
Dos 单元	70	CBreak	95
常量、类型和变量	70	过程	95
中断支持过程	73	函数	96
日期和时间过程	73	Graph3 单元	96
磁盘状态函数	73	过程	96
文件处理过程和函数	73		
进程管理过程和函数	73	第十三章 覆盖	98
环境管理函数	73	覆盖单元	98
杂类过程与函数	73	常量和变量	99
Crt 单元	73	结果变量	99
Input、Cutput 文件	75	结果代码	99
窗口	75	过程和函数	99
特殊字符	75	OvrInit	99
行输入	75	OvrInitEMS	99
常量和类型	76	OvrSetBuf	100
Crt 模式常量	76	OvrGetBuf	100
字符颜色常量	76	OvrClearBuf	100
Crt 变量	76	覆盖程序设计	100
过程	78	覆盖代码的产生	100
函数	79	FAR 调用要求	101
Graph 单元	79	初始化覆盖管理模块	101
驱动程序	79	覆盖单元中的初始化部分	103
IBM-8541 支持	80	不能覆盖的单元	104
坐标系统	80	覆盖调试	104
当前指针	80	覆盖块中的外部子程序	104
文本	81		
图像与线型	81	第十四章 使用 8087	106
视口和点阵图象	81	8087 数据类型	106
分页和颜色	82	扩展型精度运算	107
错误处理	82	实数的比较	108
起动	83	8087 运算符	108
用户编写堆管理子程序	85	8087 的实数输出	109
Graph 单元常量、类型和变量	86	使用 8087 的单元	109
过程	92	检测 8087	108
		汇编语言 8087 仿真	110

第十五章 Turbo Pascal 内部结构	111
堆管理程序.....	112
释放方法.....	112
自由块表.....	114
堆错误函数.....	115
内部数据格式.....	116
调用协议.....	119
变参.....	120
值参.....	120
函数结果值.....	120
Near 调用和 Far 调用.....	120
嵌入过程和函数.....	120
入口和出口代码.....	121
寄存器保存协议.....	122
和汇编语言连接.....	122
Turbo 汇编和 Turbo Pascal.....	122
汇编语言子程序的例子.....	123
Turbo Assembler 示例.....	126
Inline 机器码.....	127
Inline 语句.....	127
Inline 指令.....	128
直接存取内存和端口.....	128
Mem, Memw 和 Meml 数组.....	128
Port 和 Portw 数组.....	129
中断处理子程序.....	129
编写 Interrupt 过程.....	129
文本设备驱动程序.....	130
Open 函数.....	131
InOut 函数.....	131
Flush 函数.....	131
Close 函数.....	131
文本设备驱动程序示例.....	131
Exit 过程.....	134
自动优化.....	135
常量编译处理.....	135
常量单一化.....	136
运算顺序.....	136
短路运算.....	136

范围检查.....	136
移位代替乘法.....	136
自动字边界对齐.....	136
不执行代码的删除.....	136
灵巧的连接.....	136

第十六章 Turbo Paseal 过程和函数一览表.....138

Abs 函数.....	138
Addr 函数.....	138
Append 过程.....	138
Arc 过程.....	139
ArcTan 函数.....	139
Assign 过程.....	140
AssignCrt 过程.....	140
Bar 过程.....	141
Bar3D 过程.....	141
BlockRead 过程.....	142
BlockWrite 过程.....	143
ChDir 过程.....	144
Chr 函数.....	144
Circle 过程.....	144
ClearDevice 过程.....	145
ClearViewPort 过程.....	145
Close 过程.....	146
closeGraph 过程.....	146
ChrEol 过程.....	147
ChrScr 过程.....	147
ConCat 函数.....	148
Copy 函数.....	148
Cos 函数.....	148
Cseg 函数.....	148
Dec 过程.....	149
Delay 过程.....	149
Delete 过程.....	149
Decline 过程.....	149
DetectGraph 过程.....	150
DiskFree 函数.....	151
DiskSize 函数.....	151
Dispose 过程.....	151
DosExitCode 函数.....	152

DosVersion 函数	152	GetIntVec 过程	174
DrawPoly 过程	152	GetLineSettings 过程	174
Dseg 函数	153	GetMaxColor 函数	175
Ellipse 过程	153	GetMaxMode 函数	176
EnvCount 函数	154	GetMaxX 函数	176
EnvStrt 函数	154	GetMaxY 函数	177
Eof 函数(文本文件)	154	GetMem 过程	177
Eof 函数(类型、无类型文件)	155	GetModeName 函数	177
Eoln 函数	155	GetmodeRange 过程	178
Erase 过程	155	GetPalette 过程	178
Exec 过程	156	GetRaletteSize 函数	179
Exit 过程	157	GetPixel 函数	179
Exp 函数	157	GetTextSettings 过程	180
FExpand 函数	157	GetTime 过程	181
FilePos 函数	158	GetVerify 过程	181
FileSize 函数	158	GetViewSettings 过程	181
Fillchar 过程	158	GetX 函数	182
FillEllipse 过程	159	GetY 函数	183
FillPoly 过程	159	GotoXY 过程	184
FindFirst 过程	160	GraphDefaults 过程	184
FindNext 过程	161	GraphErrorMsg 函数	184
FloodFill 过程	161	GraphResult 函数	185
Flush 过程	162	Halt 过程	186
Frac 函数	162	Hi 函数	186
FreeMem 过程	163	ImageSize 函数	187
FSearch 函数	163	Inc 过程	197
FSplit 过程	163	InitGraph 过程	188
GetArcCoords 过程	164	Insert 过程	189
GetAspectRatio 过程	165	InsLine 过程	190
GetBkColor 函数	165	InstallUsetDriver 函数	190
GetCBreak 过程	166	InstallUserFont 函数	192
GetColor 函数	167	Int 函数	193
GetDate 过程	167	Intr 过程	193
GetDefaultDalette 函数	167	IoResult 函数	194
GetDir 过程	168	Keep 过程	194
GetDriverName 函数	169	KeyPressed 函数	194
GetEnv 函数	169	Length 函数	195
GetFAttr 过程	170	Line 过程	195
GetFillPattern 过程	171	LineRel 过程	196
GetFillSettings 过程	171	LineTo 过程	196
GetFTime 过程	172	Ln 函数	197
GetGraphMode 函数	172	Lo 函数	197
GetImage 过程	173		

LowVideo 过程	198
Mark 过程	198
MaxAvail 函数	198
MemAvail 函数	199
MkDir 过程	199
Move 过程	200
MoveRel 过程	200
MoveTo 过程	200
MsDos 过程	201
New 过程	202
NormVideo 过程	202
NoSound 函数	202
Odd 函数	202
Ofs 函数	202
Ord 函数	203
OutText 过程	203
OutTextXY 过程	203
OvrClearBuf 过程	205
OvrGetBuf 函数	205
OvrInit 过程	206
OvrInitEms 过程	206
OvrSetBuf 过程	207
PackTime 过程	208
ParamCount 函数	208
ParamStr 函数	208
Pi 函数	209
PieSlice 过程	209
Pos 函数	210
Pred 函数	210
Ptr 函数	210
PutImage 过程	210
PutPixel 过程	212
Ramdom 函数	213
Randomize 过程	213
Read 过程(文本文件)	213
Read 过程(类型文件)	213
ReadKey 函数	214
Readln 过程	215
Rectangle 过程	215
RegisterBGIDriver 函数	216
RegisterBGIFont 函数	216
Release 过程	219

Rename 过程	219
Reset 过程	219
RestoreCrtMode 过程	220
ReWrite 过程	221
Rmdir 过程	221
Round 函数	222
RunError 过程	222
Sector 过程	222
Seek 过程	222
SeekEof 函数	223
SeekEoln 函数	224
Seg 函数	224
SetActivePage 过程	224
SetAllPalette 过程	225
SetAsPectRatio 过程	226
SetBKColor 过程	227
SetCBreak 过程	228
Setolor 过程	228
SetDate 过程	229
SetFAttr 过程	229
SetFillPattern 过程	229
SetFillStyle 过程	231
SetFTime 过程	231
SetGraphBufSize 过程	232
SetGraphMode 过程	232
SetIntVec 过程	233
SetLineStyle 过程	234
SetPalette 过程	235
SetRGBPalette 过程	236
SetFontBuf 过程	238
SetTextJustify 过程	238
SetTextStyle 过程	239
SetTime 过程	241
SetUserCharSize 过程	241
SetVerify 过程	241
SetViewPort 过程	242
SetVisualPage 过程	243
SetWriteMode 过程	244
Sin 函数	245
SizeOf 函数	245
Sound 过程	246
Sptr 函数	246

Sqr 函数	246
Sqrt 函数	246
SScg 函数	246
Str 过程	246
Succ 函数	247
SwapVectors 过程	247
TextBackGround 过程	248
TexColor 过程	248
TextHeight 函数	249
TextMode 过程	250
TextWidth 函数	251
Trunc 函数	252
Truncate 过程	252
UnpackTime 过程	252
Uppcase 函数	252
Val 过程	253
WhereX 函数	253
WhereY 函数	253
Window 过程	254
Write 过程(文本文件)	254
Write 过程(类型文件)	256
Writeln 过程	256

第二部分 附录

附录 A Turbo Pascal 5.0 与 Ansi Pascal 比较	257
与 Ansi Pascal 的差别	257
对 Ansi Pascal 的扩充	258
依赖实现特性	259
错误处理	259

附录 B 编译指令	260
开关指令	260
参数指令	264
条件编译	265

附录 C 略

附录 D 错误信息和代码	268
编译错误信息	268
运行错误	275
DOS 错误	276
I/O 错误	276
严重错误	277
致命错误	277

TURBO PASCAL 简介

欢迎使用《Turbo Pascal 5.0 参考手册》。在《用户手册》中，介绍了 Turbo Pascal 编程的基本概念，以及使用集成环境的方法。本书详细介绍 Turbo Pascal 语言。

关于本手册

本手册分成两个部分：参考部分和附录。第一部分如下安排：

第一章 单词和常量

第二章 块、位置和作用域

第三章 类型

第四章 变量

第五章 类型常量

第六章 表达式

第七章 语句

第八章 过程和函数

第九章 程序和单元

第十章 输入和输出

第十一章 标准过程和函数

第十二章 标准单元

第十三章 覆盖

第十四章 使用 8087

第十五章 Turbo Pascal 的内部结构

第十六章 Turbo Pascal 过程和函数一览表。

第二部分比较了 Turbo Pascal 和 ANSI Pascal，涉及到编译指令，罗列了 Turbo Pascal 编译、运行产生的错误信息。

附录A Turbo Pascal 5.0 与 ANSI PASCAL 比较

附录B 编译指令

附录C 引用材料 (略)

附录D 错误信息和代码

第一章 单词和常量

单词是 Pascal 程序中有意义的最小单位，它分为特殊符号、标识符、标号、数字和字符串常量。

Pascal 程序由单词和分隔符组成。分隔符是空格逗号。如果相邻的两个单词都是保留字、标识符、标号或数字，那么必须用一个或多个分隔符分开。

除了在字符串常量中，分隔符不能成为单词的组成部分。

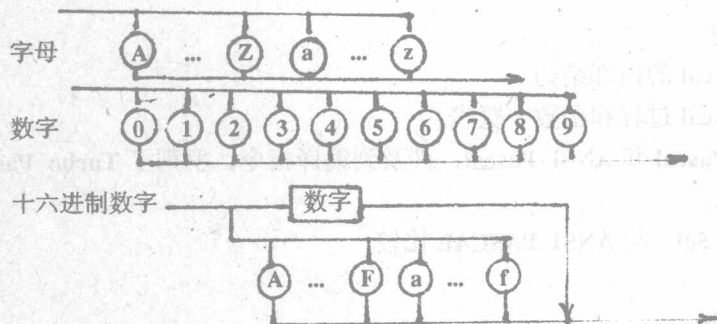
特殊字符和保留字

Turbo Pascal 使用 ASCII 字符集的下述子集

- * 字母—英文字母 A-Z 和 a-z
- * 数字—阿拉伯数字 0-9
- * 十六进制数字—0-9, A-F 和 a-f
- * 空格符—空格键和所有的 ASCII 控制字符，包括行结束符和回车符。

下面是字母、数字和十六进制数字的语法图。随箭头看语法图，经常有多种路径，只有沿着箭头方向从左开始，结束在右边的才算有效。通过含有元素名的框的路径来构造语法。

在长方框中的名字代表非终结符。在圆框中保留字、操作码和标点符号是程序实际使用的单词。



特殊字符或保留字是一个或多个固定意义的单词。最简单是特殊字符：

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

下面字符组是特殊符号：

<= = := .. (* *) (. '.)

一些特殊字符也是操作符。左方括号[和左小括号加句点(.等价。相似地)和.)等价。

下面是 Turbo Pascal 保留字：

absolute	end	inline	procedure	type
and	external	interface	program	unit
array	file	interrupt	record	until
begin	for	label	repeat	uses
case	forward	mod	set	var
const	function	nil	shl	while

div	goto	not	shr	with
do	if	of	string	xor
downto	implemmentation	or	then	
else	in	packed	to	

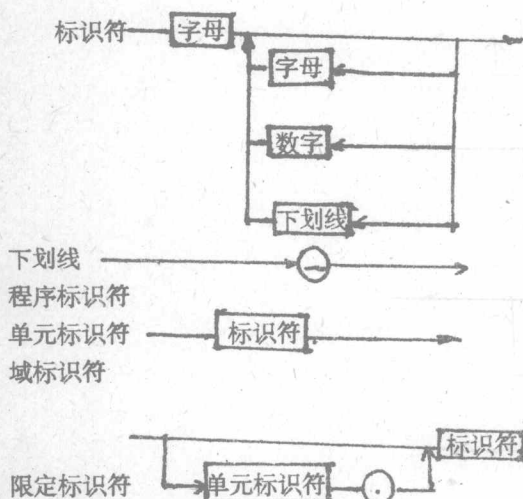
Turbo Pascal 大小写字母没有区别，因此，在编程中能使用大小写。

标识符

标识符表示常量、类型、变量、过程、函数、单元、程序和记录中的域。标识符的长度可以是任意的，但只有前 63 个字符有效。

标识符以字母开始，当中不含有空格。后跟字母、数字和下划线字符(ASCII 码 5F'-')。象保留字一样，标识符没有大小区别。

出现几个同名标识符时，为了选定标识符必须通过单元标识符限定。(单元在《用户手册》第四章和本手册第十二章描述)。例如，用单元标识符 `unitname` 限定标识符 `Ident`，可以写成 `unitname.Ident`。这种标识符称为限定标识符。

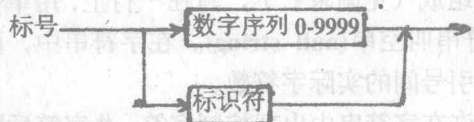


下面是标识符的例子：

`uriteln Exit Realstring System.MemAvail Dos.Exec Crt.Window`

标号

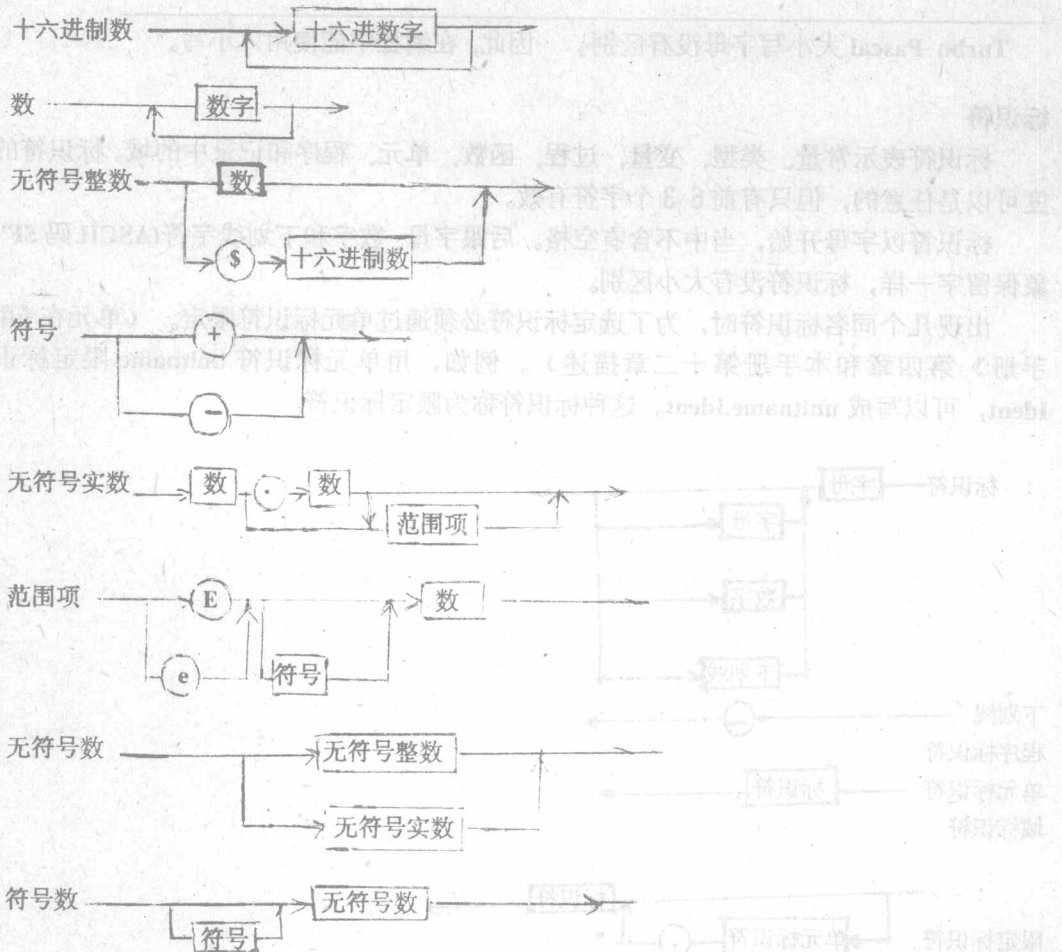
标号是 0-9999 的数字序列，前导零无效。标号和 Goto 语句一起使用。



作为标准 Pascal 扩充，Turbo Pascal 允许使用标识符作为标号。

数

普遍的十进制表示整型和实型的常量。十六进制整数常量用\$作为前缀。科学表示法(E或e后跟指数)表示实数。比如 $7E-2=7 \times 10^{-2}$; $12.25+e6$ 或 $12.25e6$ 都是 12.25×10^6 。数的语法图如下:



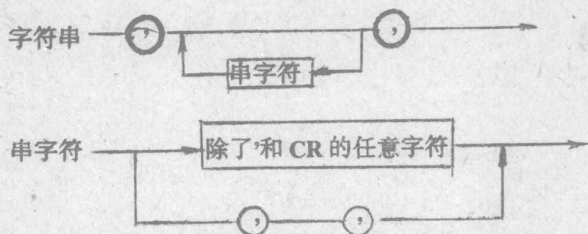
用十进制数和指数表示实型常量;其它十进制表示整型常量,整数为-2147483648..2147483647

十六进制数表示整型常量,范围为\$00000000到\$FFFFFFFF。值的符号隐含在十六进制表示法中。

字符串

字符串是零个或零个以上扩展 ASCII 字母组成(见附录 C),写在一行上,用单引号括起来的序列。在单引号之间没有字符的字符串叫空串(null string)。在字符串中,两个单引号连用表示单引号字符。字符串长度为单引号间的实际字符数。

作为标准 Pascal 的扩展, Turbo Pascal 允许在字符串中出现控制字符。# 字符后跟 0-255 整数表示对应值的 ASCII 码。在 # 字符和整数之间不能有分隔符。如果字符串中有几个连续控制字符,他们之间也没有分隔符。



空串只与字符串类型兼容。单字符字符串和任意字符和字符串兼容。长度为 n 的字符串 ($n \geq 2$) 与字符串类型及紧缩字符数组兼容。

'TuRBo' 'You'll see' "" ';' "

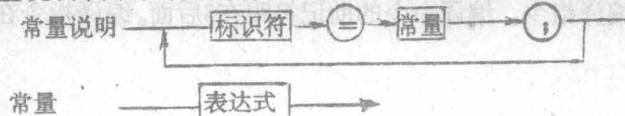
#B#10

'Line 1'#13'Line2'

#7#7'wakeup!'#7#7

常量说明

常量说明表示在含有该说明的块中该标识符代表常量。常量标识符不能包含于自己。



常量表达式

作为标准 Pascal 的扩展, Turbo Pascal 允许使用常量表达式。常量表达式是由编译器运算, 没有实际执行代码的表达式。如:

100 'A' 256-1 (2.5+1)/(2.5-1) 'Turbo'+ ' ' + 'Pascal'

chr(32) Ord('z')-Ord('A')+1

简单常量表达式是单个常量如 100 或 'A'; 标准 Pascal 只允许使用简单常量, Turbo Pascal 允许使用常量表达式。

由于编译时就完成常量运算, 因此下面结构在常量表达式中是不允许的:

* 引用变量和类型变量

* 函数调用

* 地址操作符 除了这些限制, 常量表达式和普通表达式语法差不多(见第六章“表达式”)。

下面的标准函数允许在常量表达式中出现:

Abs	chr	Hi	Length	Lo	Odd	Ord
Pred	Ptr	Round	Sizeof	Succ	Swap	Trunc

常量表达式如:

const

min=0;

max=100;

Center=(Max-Min)div 2;

Beta=chr(225);

NunChars=Ord('Z')-Ord('A')+1;

Message='out of memory';

```

Errstr='Error:'+Message+'.';
Errpos=80-length(Error) div 2;
ErrAttr=Blink+Red*16+unife;
Ln10=2.302585092994045684;
Ln10R=1/Ln10;
Numeric=['0'..'9'];
Alpha=['A'..'Z','a'..'z'];
AlphaNum=Alpha+Numeric;

```

注释

下面结构是注释,注释在编译时忽略

{Any text not Containing right brace}

(* Any text not containing Star/right Parenthesis *)

在{或(*后面紧跟\$是一条编译指令。编译命令的助记符跟在\$后面。编译指令在附录B中介绍。

程序行

Turbo Pascal 程序行最长不能超过 126 个字符。

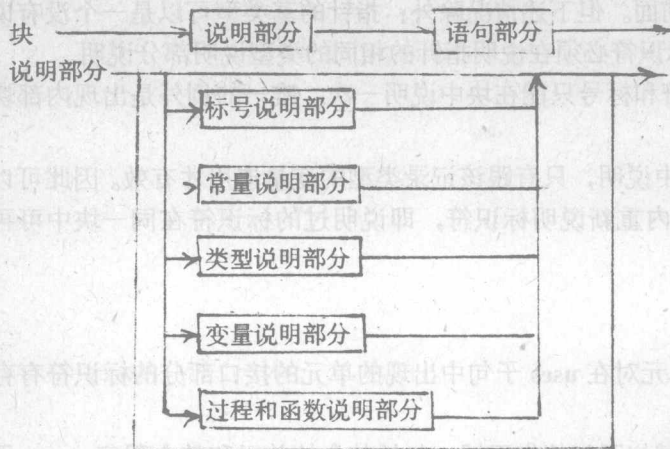
第二章

块、位置和作用域

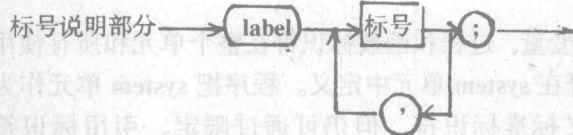
块由任意顺序的说明和语句组成。每个块是过程说明、函数说明或程序及单元的一部分。在块中说明的标识符或标号是该块的局部符号。

语法

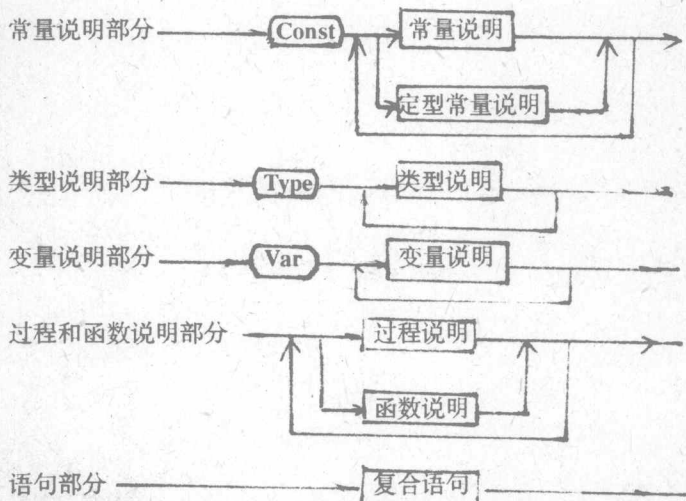
块的语法格式：



标号说明部分是说明标记语句部分中语句的标号的地方。每个标号必须只标记一个语句。



注意用来表示标号的数必须在 0-9999 之间。



作用域规则

标识符或标号在说明中出现叫做它们的定义。以后标识符或标号的出现必须在作用域内。标识符或标号的作用域从所在块中说明开始到该块结束,包括所有内部块。例外情况如下:

- * 在内部块中重新说明:假设 Exterior 块包含 Interior 块,如果 Exterior 和 Interior 有相同的标识符 j ,那么两个块只能存取自己说明的 j 。
- * 块中说明的位置:标识符和标号只有在说明后才能引用。标识符或标号说明必须出现在引用性出现的前面。但下述情况除外:指针的基类型可以是一个没有说明的标识符。然而,这标识符必须在说明指针的相同的类型说明部分说明。
- * 块内重新说明:标识符和标号只能在块中说明一次。唯一的例外是出现内部块中或记录域表中。

记录域标识符在记录类型中说明,只有跟该记录类型的变量连用才有效。因此可以在同一块中但不在同一记录类型内重新说明标识符,即说明过的标识符在同一块中可再说明成域标识符。

接口和标准标识符的作用域

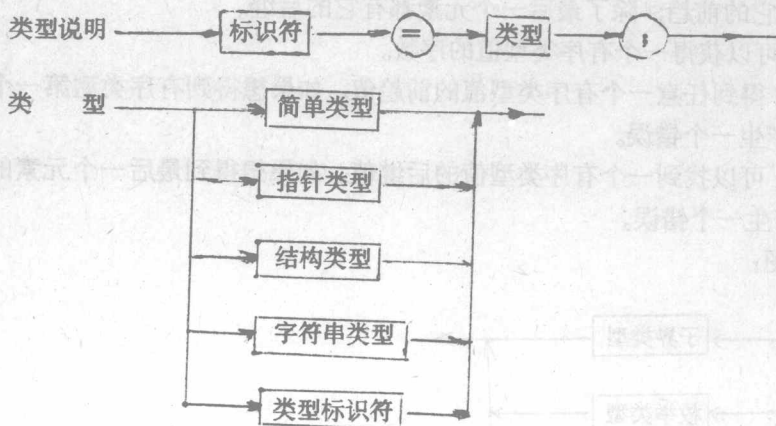
含有 `uses` 子句的程序或单元对在 `uses` 子句中出现的单元的接口部分的标识符有存取权。

`uses` 子句中的每个单元被赋与了新的作用域,包括其余的单元和整个程序。`uses` 子句中的第一个单元代表最外层,最后单元代表最内层。如果两个或更多单元说明同一标识符,没有限定引标识符隐含选择 `uses` 子句最后说明的标识符单元的标识符。然而通过限定标识符,可限定标识符的作用域。

Turbo Pascal 预定义的常量、类型变量、过程和函数标识符在整个单元 and 所有使用的单元都有效。实际上,这些标准标识符在 `system` 单元中定义。程序把 `system` 单元作为真正的最外层。单元或程序可重新定义标准标识符,但仍可通过限定,引用标识符如 `System.Integer` 或 `System.Writeln`。

第三章 类型

说明一个变量时，必须指出类型。变量的类型规定变量的取值范围和可能的操作。类型说明指定一种类型的标识符。



当一个标识符出现在类型定义的左边时，它就作为一个所在块的类型说明标识符。除了指针类型，类型标识符的作用域不包括自身。

有五种主要的类型：

- * 简单类型

- * 字符串类型

- * 结构类型

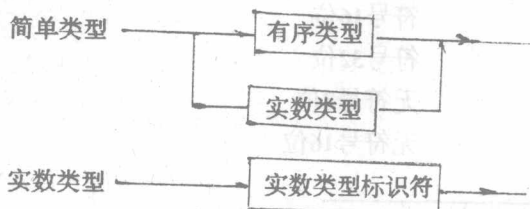
- * 指针类型

- * 过程类型

下面各节分别介绍。

简单类型

简单类型定义值的有序集。



实数类型标识符是标准标识符：`real`, `single`, `double`, `extended` 或者 `Comp`。参考第一章中“数”和“字符串常量”节，可以发现怎样表示常量型整数和实数值。

有序类型