



新大纲

全国计算机等级考试指导

PASCAL 语言 程序设计考试指导 (二级)

李大友 主编
谭国真 李盘林 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

TP312

434995

L210-3

全国计算机等级考试指导

PASCAL 语言程序设计考试指导

(二级)

李大友 主编

谭国真 李盘林 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

JS175/30

内 容 简 介

本书是根据国家教委考试中心颁布的“全国计算机等级二级大纲”中关于 Pascal 程序设计的考试要求和 Pascal 结构程序设计语言自身特点编写的。本书系统地介绍了 Pascal 的基本概念、基本语句、数据类型、程序结构特点以及结构化程序设计方法。

本书除注意典型例题的选讲外,每章末附有较多习题,供练习和理解本章内容之用。为便于自我检查,附录中给出解答。为了更好地掌握 Pascal 内容和考试规律,附录中还给出了九五~九七年试题及答案。

本书可作为考生考前辅导教材,也可作为学习计算机基础知识的自学教材或培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

丛 书 名: 全国计算机等级考试指导
书 名: PASCAL 语言程序设计考试指导(乙级)
主 编: 李大友
编 著 者: 谭国真 李盘林
责任编辑: 吕 迈
排版制作: 电子工业出版社计算机排版室
印 刷 者: 北京天宇星印刷厂
出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>
北京市海淀区万寿路 173 信箱 邮编 100036
经 销: 各地新华书店
开 本: 787×1092 1/16 印张: 15.75 字数: 384 千字
版 次: 1998 年 11 月第 1 版 1998 年 11 月第 1 次印刷
书 号: ISBN 7-5053-5023-4
TP·2492
定 价: 20.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。
若书店售缺,请与本社发行部联系调换。电话 68279077

前 言

Pascal 语言自 70 年代初提出以来,至今将近 30 年。由于该语言程序清晰可靠,易于读懂、便于修改、使用广泛、灵活方便,到目前为止仍没有失去作为各大学计算机教学的第一种高级语言的魅力。

本书是根据国家教委考试中心颁布的《全国计算机等级二级大纲》中关于 Pascal 程序设计》的考试要求和 Pascal 结构程序设计语言自身特点编写的,系统地介绍了 Pascal 的基本概念、基本语句、数据类型、程序结构特点以及结构化程序设计方法。

本书在介绍各章内容时,不仅注重了语法、语义的讲解,而且也十分注意典型例题的搭配,以及注意程序设计方法、风格的训练和培养。每章末附有较多习题,供练习和理解本章内容之用。为便于自我检查,附录中给出解答。对于书中的所有例题和习题,凡结构完整的程序均在 TURBO Pascal 6.0 下调试通过。为了更好掌握 Pascal 内容和考试规律,附录中还给出九五~九七年试题及答案。

本书可作为考生考前辅导教材,也可作为大专院校计算机或非计算机专业的教材和教学参考书,以及自学用书。

本书在编写过程中,自始至终得到了李大友教授的指导和支持,也得到了关浩同志在文字编辑以及齐爱平硕士生 in 程序调试工作中所给予的协助,在此表示深切地谢意!

由于水平有限,书中难免会有不妥或错误之处,请读者批评指正。

作者

1998.4

目 录

第 1 章 Pascal 程序设计快速入门	(1)
1.1 Pascal 程序结构	(1)
1.2 保留字和标识符	(3)
1.3 Pascal 数据类型	(4)
1.3.1 整数类型	(4)
1.3.2 实数类型	(4)
1.3.3 布尔类型	(5)
1.3.4 字符类型	(6)
1.4 常量与变量	(6)
1.5 标准函数	(7)
1.6 表达式和赋值语句	(9)
1.6.1 表达式	(9)
1.6.2 赋值语句	(10)
1.7 输入输出语句及其控制格式	(10)
1.7.1 读语句 Read 和 Readln	(10)
1.7.2 写语句 write 和 writeln	(12)
1.7.3 输出格式	(13)
1.8 简单的 Pascal 程序设计实例	(14)
习题一	(15)
第 2 章 控制结构程序的设计	(18)
2.1 复合语句	(18)
2.2 选择性语句	(18)
2.2.1 IF 语句	(18)
2.2.2 CASE 语句	(22)
2.3 循环语句	(25)
2.3.1 WHILE 循环语句	(25)
2.3.2 REPEAT 循环语句	(29)
2.3.3 FOR 循环语句	(30)
2.4 循环嵌套	(33)
2.5 GOTO 语句	(36)
2.6 应用程序举例	(37)
2.7 结构化程序设计	(41)
习题二	(42)
第 3 章 函数和过程	(48)
3.1 标准函数和标准过程	(48)

3.2	自定义函数的说明与调用	(49)
3.3	自定义过程说明与调用	(52)
3.4	数值参数与变量参数	(54)
3.5	标识符的作用域	(59)
3.5.1	局部量的作用域	(59)
3.5.2	全局量的作用域	(59)
3.5.3	子程序嵌套的作用域问题	(62)
3.5.4	函数名和过程名的作用域	(63)
3.5.5	非局部量与副作用	(65)
	习题三	(66)
第4章	递归程序设计	(69)
4.1	递归的基本概念	(69)
4.2	递归调用	(69)
4.3	函数递归调用	(70)
4.3.1	如何编写正确的递归程序	(73)
4.4	过程的递归调用	(77)
4.5	设计递归过程的技巧	(78)
4.6	栈与递归	(82)
4.7	递归程序应用实例	(85)
	习题四	(87)
第5章	枚举类型和子域类型	(90)
5.1	类型定义	(90)
5.2	枚举类型	(90)
5.3	子域类型	(93)
5.4	枚举和子域类型的程序实例	(96)
	习题五	(98)
第6章	数组类型	(100)
6.1	一维数组	(100)
6.2	一维数组程序举例	(102)
6.3	二维数组	(104)
6.4	二维数组程序举例	(105)
6.5	紧缩数组和字符串变量	(109)
	习题六	(113)
第7章	集合类型	(115)
7.1	集合的定义	(115)
7.2	集合运算	(117)
7.3	集合的输入输出方法	(118)
7.4	集合应用程序举例	(119)
	习题七	(120)
第8章	记录类型	(122)

8.1 记录类型定义、变量说明和引用	(122)
8.2 记录数组	(124)
8.3 WITH 语句	(125)
8.4 记录类型程序应用举例	(129)
习题八	(131)
第9章 动态数据结构	(134)
9.1 指针	(134)
9.1.1 指针类型和指针变量	(134)
9.1.2 开辟和释放动态存储单元	(135)
9.1.3 动态变量的引用	(136)
9.1.4 对指针变量的操作	(137)
9.1.5 指针变量和动态变量的区别与关系	(138)
9.2 链表	(139)
9.2.1 链表的基本概念	(139)
9.2.2 单向链表节点的插入	(141)
9.2.3 单向链表节点的删除	(145)
9.2.4 建立单向链表	(146)
9.2.5 打印单向链表	(147)
9.2.6 带冠节点的单向链表	(147)
9.3 双向链表	(148)
9.4 循环链表	(149)
习题九	(150)
第10章 文件	(154)
10.1 FILE 文件	(155)
10.1.1 FILE 文件的基本概念	(155)
10.1.2 FILE 文件的建立与读写	(156)
10.2 text 文件	(159)
10.2.1 数据类型的自动转换	(159)
10.2.2 文本文件的分行结构	(161)
习题十	(165)
附录 A 习题解答	(167)
附录 B 1995 年 ~ 1997 年全国计算机等级考试二级笔试题 (Pascal 语言部分) 及标准答案	(200)
参考文献	(244)

第 1 章 Pascal 程序设计快速入门

1.1 Pascal 程序结构

程序是一个指令的序列,是对给定数据处理过程的描述。Pascal 语言是如何描述这个处理过程,达到问题的求解?为此,本书将通过一个简单 Pascal 程序来介绍 Pascal 程序的组成。

【例 1】从点 (x_0, y_0) 到直线 $Ax + By + C = 0$ 的距离

$$D = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

写一个程序读入 x_0 和 y_0 , 输出 D 的值。

【程序】

```
PROGRAM PointToLine (input,output);      (* 程序首部 *)
VAR                                       (* 以下两行为说明部分 *)
    xo,yo,A,B,C,D:real;
BEGIN                                   (* 以下六行是执行部分 *)
    writeln('Please input A,B,C,x0 and y0');
    readln(A,B,C,x0,y0);
    D:=abs(A * x0+B * y0+C)/sqrt(A * A+B * B);
    writeln('D=',D)
END.
```

【输出】

```
Please input A,B,C,x0 and y0
D=
```

【说明】该程序分为三部分:

1. 程序的首部

(1)程序的首部总是以 PROGRAM 开头,大小写均可。

(2)程序的名称。程序名称由用户自己定义,它的选取最好能反映出程序的功能。本例中 PointToLine 表示求点到直线的距离。应该注意,在 PROGRAM 和程序名之间必须留有空格。

(3)程序的参数表。它们用来表示该程序与外界的联系。参数表放在一对圆括号中,紧跟在程序名的后面,各参数之间用逗号隔开,每个参数代表一个输入输出文件名。本程序中 input 表示有输入操作,借助于标准文件 input(通常指键盘输入)输入数据;output 表示有输出操作,程序把数据处理结果送到标准文件 output(通常指终端显示器)上。

(4)在参数表后面必须有一个分号表示程序首部结束。

(5)程序的注释。注释用一对花括号 {} 或 (* 和 *) 括起来,它可以写在程序的任何位置。注意,圆括号和 * 之间不能插入空格。注释的作用是增加程序的易读性。Pascal 编译程序并不理睬所有的注释。

2. 程序的说明部分

Pascal 语言严格规定凡是在程序中出现的标识符都必须先说明后使用;如程序中的 VAR 表示对所有变量的说明,VAR 后面不加分号,只表示变量说明的开始。例中

VAR

x0,y0,A,B,C,D:real;

指出程序中使用的六个实型变量 x0,y0,A,B,C,D。

3. 程序的执行部分

执行部分是由 BEGIN 开始、END 结束,中间是可执行的语句序列,这部分的书写规则是各语句间用分号隔开,一行可写多个语句,END 和它前面的一条语句之间可以不加分号,END 之后必须加圆点“.”以表示结束。

本程序由四条语句组成。

writeln('Please input A,B,C,x0 and y0')是输出语句,又叫写语句,该语句的功能是在屏幕上显示提示信息

Please input A,B,C,x0 and y0

readln(A,B,C,x0,y0)是输入语句,又叫读语句。计算机执行此语句时处于等待状态,要求从键盘输入数据,一行输入数据以打“回车”键结束。全部数据输入完毕后,程序继续往下执行。对于上面的读语句,要求从键盘输入五个实数(用空格隔开)分别赋给变量 A,B,C,x0,y0。

D:=abs(A * x0+B * y0+C)/sqrt(A * A+B * B)是赋值语句,它将在左边的运算结果赋给 D。其中 abs(x),sqrt(x)是 Pascal 的标准函数,其意义分别是 $|x|$ 和 $\sqrt{x}$;“*”表示相乘,“/”表示相除。

writeln('D=',D)写语句,执行这条语句将结果按指定的形式输出;引号中的内容照原样显示在显示器上,D 的内容显示在相应位置上。

注意程序对字符的大小写没有要求。在本书中,程序标志符 PROGRAM、变量说明符、起始 BEGIN 和结束 END 等保留字均用大写以示醒目,其余部分用小写或大写。

上述程序虽然简单,但结构较完整。任何一个 Pascal 程序同样是由这三部分组成,其一般组成如下:

PROGRAM	程序名(程序参数表);	一程序的首部
LABEL	标号说明;	一程序说明部分
CONST	常数说明;	
TYPE	类型说明;	
VAR	变量说明;	
PROCEDURE	过程说明;	
FUNCTION	函数说明;	
BEGIN	语句序列	
END		
		一程序执行部分

上面的有关部分将在后面的章节陆续详细介绍。

1.2 保留字和标识符

顾名思义,保留字(保留关键字)是一种程序语言专门留作该语言使用的,具有特殊含义的关键字。标识符是标识某一对象的名字。

1. 保留字

Pascal 中的保留字共有 35 个,这里以字母顺序列出:

AND	ARRAY	BEGIN	CASE	CONST
DIV	DO	DOWNT	ELSE	END
FILE	FOR	FUNCTION	GOTO	IF
IN	LABEL	MOD	NIL	NOT
OF	OR	PACKED	PROCEDURE	PROGRAM
RECORD	REPEAT	SET	THEN	TO
TYPE	UNTIL	VAR	WHILE	WITH

读者可能一下记不住这些保留字,这不要紧,随着以后会逐一介绍,最终不但记住,还应熟练地使用。

Pascal 规定,保留字用大写或小写字母的含义相同,但为了养成良好的书写程序的习惯,建议读者一律用大写来表示保留字,而程序的其它部分建议用英语的书写规范或按照自己的习惯办。

2. 标识符

标识符由字母和数字组成。标识符的第一个字符必须是一个字母。标识符的字符个数由程序语言的编译程序确定。一般来说,所有程序语言的标识符都可以达到 8 个字符长,超过 8 个字符,可能只识别前 8 个字符,其余不起作用。TURBO Pascal、MS Pascal 规定,标识符最长为 31 个字符。建议读者上机编程时参照所使用编译器的版本的标识符规定。

Pascal 语言的标识符分两类:一是 Pascal 预先确定的标识符,如提供给标准常量、标准类型、标准函数、标准过程和标准文件的名字,这类标识符称为标准标识符;另一类是程序员定义的标识符。

Pascal 的标准标识符共 40 个:

标准常量: false, true, maxint

标准类型: boolean, integer, real, char, text

标准函数: abs, arctan, chr, cos, eof, eoln, exp, ln, odd, ord, pred, round, sin, sqr, sqrt, succ, trunc

标准过程: dispose, get, new, pack, page, put, read, readln, reset, rewrite, unpack, write, writeln

标准文件: input, output

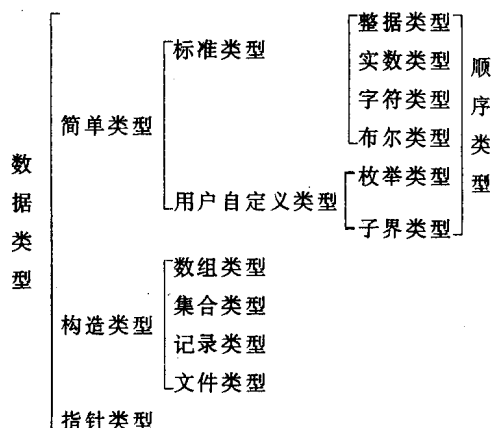
用户定义标识符除遵循标识符的定义规则外,还应该注意以下几点:

- (1) 不要与保留字和标准标识符完全相同;
- (2) 最好选择见名知义的标识符;

(3)如果应用程序非常大,可采用近几年流行的匈牙利标记法。该标记法不但能表明标识符的意义而且还能表示其类型。使用匈牙利标记法,变量以一个或几个小写字母开头以表示变量的类型,这样,便提供了一种反映内在特性的标识。例如,iSum 表示整型数并指示该标识符用来存放累加和。匈牙利标记法在 Windows 程序中被广泛采用,感兴趣的读者可参阅有关 Windows 方面的书。在本书中由于程序都比较小,故仍采用传统的标识符标记方法。

1.3 Pascal 数据类型

数据类型含有三层意义:一是规定了该类型数据的取值范围;二是规定了该类型变量的存储方式;三是定义了一组对该类型数据进行的运算。例如在字长为 2 个字节的计算机中,标准类型 integer 说明了该类型数据的取值范围为-32768 到 32767 及能在这些值上进行+、-、*、DIV、MOD 等运算。Pascal 语言的数据类型如下:



1.3.1 整数类型

【标识符】 integer

【取值】 整数的取值范围与机器有关,在 IBM-PC 机上运行的 TURBO Pascal 和 MS Pascal,规定整数的范围是-32768~32767。

【运算】 在整数集上的运算有算术运算和关系运算。算术运算包括+(加)、-(减)、*(乘)、DIV(整除)、MOD(取余)。这些运算符对整型数进行运算,其结果是整型数。其中运算符 DIV 和 MOD 是保留字。DIV 和 MOD 运算满足下列关系:

$$a \text{ MOD } b = a - (a \text{ DIV } b) * b$$

例如 $14 \text{ DIV } 4 = 3$ $-14 \text{ DIV } 4 = -3$

$$14 \text{ MOD } 4 = 14 - (14 \text{ DIV } 4) * 4 = 2$$

$$-14 \text{ MOD } 4 = -14 - (-14 \text{ DIV } 4) * 4 = -2$$

运算符“-”也可作单目运算符,如-69。

关系运算包括=(等于)、<>(不等于)、<(小于)、<=(小于等于)、>=(大于等于)、>(大于)。两个整数的关系运算,其结果为真(true)或假(false)。

1.3.2 实数类型

【标识符】 real

【取值】 在微机上运行的 MS(或 TURBO) Pascal 可接受的实数范围为 $-10^{38} \sim 10^{38}$ 。绝对值超过 10^{38} 时称为上溢;绝对值小于 10^{-38} 的数称为下溢,以输出零表示。

因计算机只能表示有限位的实数,所以在计算机上的实数不但要注意取值范围,而且要注意它的运算精度。例如某台计算机提供 7 位十进数字的精度,即一个实数有 7 位有效数字,那么当运算结果超过 7 位时,就会产生四舍五入。

【表示法】 实数有两种表示形式:十进制表示法和科学表示法。

十进制表示形式,如 3.14159、0.196,与日常习惯相同。Pascal 规定带小数点的实数在小数点的前后必须有数字,如 .58、10. 是不合法的。

科学表示法用于表示很大或很小的实数,E(e)代表底数 10,E 后面的整数表示 10 的幂次。如 1.5E8 表示 1.5×10^8 。

使用科学表示法应该注意两点:一是若字母 E 前带小数点,则在小数点的前后必须有数字,当然若没有小数点,则用整数表示法;二是字母 E 后必须是整数,十号是可以省略的。

【运算】 适用于实数的运算有算术运算和关系运算。

算术运算有+(加)、-(减)、*(乘)、/(除)。算术运算的两个操作数只要有一个是实数,运算结果为实数。运算符“/”,对于参加运算的数无论是整数还是实数,结果均为实数。

关系运算同样适合于实数类型,但要注意不要去直接测试实数值是否相等(见 1.5 节)。

1.3.3 布尔类型

【标识符】 boolean

【取值】 布尔变量可以取两个值中的一个,这两个值用标准值 true(真)和 false(假)表示,其实际含义是反映一种判断结果或表明事件的两种不同状态。

【运算】 适合于布尔类型的运算有布尔运算(或称逻辑运算)和关系运算。

布尔运算有 AND(与)、OR(或)、NOT(非),这些运算符的对象是布尔型,结果也是布尔型。AND、OR 为双目运算符,参加运算的数是两个。NOT 是单目运算符,参加运算的数据是一个,其作用是对布尔数据取反。表 1.1 列出了布尔运算的真值表。

表 1.1 真值表

P	Q	NOT P	NOT Q	P AND Q	P OR Q
T	T	F	F	T	T
T	F	F	T	F	T
F	T	T	F	F	T
F	F	T	T	F	F

优先级依次为 NOT、AND、OR。

关系运算同样适用于布尔类型。Pascal 语言对布尔型的关系运算约定下面的式子成立

$\text{false} < \text{true}$

尽管布尔运算只有三个运算符,但利用 $\text{false} < \text{true}$ 的约定和关系运算符,就可实现其他的逻辑运算。例如

P 逻辑蕴涵 Q,即 $P \rightarrow Q$;可表示成 $P \leq Q$ (注:P 为 T,Q 为 F 时 $P \rightarrow Q$ 才为 F,其余的真值指派均为真)

P 等价 Q,即 $P \leftrightarrow Q$,可表示成 $P = Q$ (注: $P \leftrightarrow Q$ 为真,当且仅当 P 与 Q 具有相同的真

值)

P 和 Q 的异或运算 $P \oplus Q$, 可表示成 $P \lt \gt Q$ (注: 当 P、Q 取值不同时为真)

1.3.4 字符类型

【标识符】 char

【取值】 全部可打印字符。

不同的计算机系统采用不同的字符集。大多数计算机系统采用 ASC II 字符集, 如 IBM PC 微型机, 因此本书采用 ASC II 字符集(见附录)。

【表示】 用一对单引号将字符括起来, 如 'A'、'+'、'9' 等。

注意: 字符类型只允许单引号内有一个字符, 否则不属于字符类型, 而成为字符串, 如 'ABC' 是字符串而不是字符类型数据。

【运算】 由于字符在 ASC II 码字符集中有唯一的序号, 由小到大排列且一个紧跟一个。例 '0' 的序号为 48, '1' 的序号为 49, 'A' 的序号为 65 等等, 请查阅附录。六个关系运算符均适用于字符类型数据。关系运算是比较字符型数据序号的大小, 运算结果为布尔量, 如 'A' < 'B' 为真, '1' > 'B' 为假。

1.4 常量与变量

1. 常量和常量定义

在程序执行时不允许改变的数据对象称为常量。在 Pascal 语言中有整值常量、实值常量、字符值常量、字符串值常量、布尔值常量等。如 123、9.8、'A'、'ABC'、true 等都是常量。

常量标识符是常量的命名, 即给常量起个名字。为常量命名的语法称为常量定义。

【语法格式】

CONST

<常量标识符> = <常量>;

:

:

<常量标识符> = <常量>;

【实例】

CONST

iNum = 123;

rNum = 9.8;

cNum = 'A';

string = 'ABC';

T = true;

【说明】上面的实例定义了 5 个常量标识符, 其类型由定义它的常量的类型决定。例如 iNum 是整型, iNum 等于整数 123; string 是字符串类型, 它等于 'ABC'。

常量标识符一经定义, 程序运行过程中就不允许改变其值, 只能被引用; 另外也可用已定义的常量去定义新的常量, 甚至允许常量表达式。例如,

CONST

biggest=1E19

smallest=-biggest; {用已定义的常量定义新的常量}

product=biggest * smallest; {用常量表达式定义新的常量}

2. 变量和变量说明

变量是程序执行时其值可改变的一类量。变量说明主要指出该变量属于什么类型数据。Pascal 程序中所用到的每一个变量,都必须在这个程序的变量说明部分中加以说明。

【语法格式】

VAR

<变量标识符表>:<类型>;

:

:

<变量标识符表>:<类型>;

VAR 表示开始一个变量说明部分,<变量标识表>是一个用逗号隔开的标识符序列,冒号后面的<类型>指明其前面各变量的所属类型。

【实例】

VAR

i,j,k:integer; {说明三个整型变量 i,j,k}

rsum:real; {说明一个实型变量 rsum}

flag:boolean; {说明一个布尔型变量 flag}

ch1,ch2:char; {说明两个字符型变量 ch1,ch2}

1.5 标准函数

为了便于程序员编写程序,Pascal 语言将常用的函数关系定义为系统的标准函数,如 x^2 定义为 $\text{sqr}(x)$, $\sqrt{x}$ 定义为 $\text{sqrt}(x)$ 等。Pascal 语言提供了 17 种标准函数,见表 1.2。

函数的使用称为函数的调用,其调用形式如下:

函数名(<自变量>)

其中函数名是系统预定义的系统标识符,它描述了函数值同自变量的对应关系,自变量可以是指定类型的任意表达式。例如 $\text{sqr}(1+x)$ 和 $\text{exp}(x)$ 分别表示数学式子 $(1+x)^2$ 和 e^x 。

函数调用可直接出现在表达式中,系统自动求出该函数的值,并参与表达式的运算。如 $x1:=(-b+\text{sqrt}(b*b-4*a*c))/(2*a)$ 。

在使用标准函数时,要注意自变量的数据类型和函数值的数据类型,以便正确的使用。

表 1.2 标准函数一览表

函数名	中文名	执行运算	自变量类型	函数值类型	实 例
abs(x)	绝对值函数	$ x $	整,实	与 x 相同	$\text{abs}(-3)=3$
sqr(x)	平方函数	x^2	整,实	整或实	$\text{sqr}(1.5)=2.25$
sin(x)	正弦函数	$\sin x$	整,实	实	$\text{sin}(3.14)=0.0$

函数名	中文名	执行运算	自变量类型	函数值类型	实 例
cos(x)	余弦函数	$\cos x$	整,实	实	$\cos(3.14)=-1.0$
arctan(x)	反正切函数	$\text{tg}^{-1}x$	整,实	实	$\arctan(1.0)=0.7419$
ln(x)	对数函数	$\ln x$	整,实	实	$\ln(2.1)=0.7419$
exp(x)	指数函数	e^x	整,实	实	$\exp(0.742)=2.1$
sqrt(x)	开方函数	$\sqrt{x}$	整,实	实	$\text{sqrt}(2.25)=1.5$
trunc(x)	截尾函数	取整	实	整	$\text{trunc}(5.9)=5$
round(x)	四舍五入函数	四舍五入	实	整	$\text{round}(8.5)=9$
ord(x)	序号函数	给出 x 的序号	顺序类型	整	$\text{ord}(\text{true})=1$
chr(x)	字符函数	按序号求字符	整	字符	$\text{chr}(65)='A'$
pred(x)	前趋函数	x 的前趋值	顺序类型	与 x 相同	$\text{pred}('b')='a'$
succ(x)	后继函数	x 的后继值	顺序类型	与 x 相同	$\text{succ}('2')='3'$
odd(x)	奇数函数	判断奇偶	整	布尔	$\text{odd}(5)=\text{true}$ $\text{odd}(6)=\text{false}$
eoln(x)	行结束函数	判断一行是否结束	文件型	布尔	读正文文件时当到行末尾得 true 否则得 false
eof(x)	文件结束函数	判断文件是否结束	文件型	布尔	读文件到末尾得 True 否则得 false

对表 1.2 所示的标准函数的几点说明:

(1) 用于数值计算的算术函数

该类函数共有八种,它们是 $\text{abs}(x)$, $\text{sqr}(x)$, $\text{sqrt}(x)$, $\sin(x)$, $\cos(x)$, $\arctan(x)$, $\ln(x)$, $\exp(x)$ 。除 abs 和 sqr 外,不论自变量是整型或实型都得到实型结果。

Pascal 语言没有提供幂次函数,为了求 a^b 的值,可利用对数的性质 ($a^b = \exp(b \cdot \ln(a))$) 表达成

应该注意的是:

①通过对数运算和指数运算求幂运算的结果,会产生一定误差。

②用这种方法时,必须 $a > 0$,因为只有 $a > 0$ 时 $\ln(a)$ 才有意义,所以

$$a^b = \begin{cases} -\exp(b \cdot \ln(-a)) & a < 0, b \text{ 为奇数} \\ \exp(b \cdot \ln(-a)) & a < 0, b \text{ 为偶数} \end{cases}$$

计算三角函数的值时,自变量的单位是弧度而不是度。度与弧度的换算公式为

$$\text{弧度} = \text{度数} \times 3.14159/180$$

反正切函数 $\arctan(x)$ 是计算 x 反正切的主值,即 $-\pi/2 < \arctan(x) < \pi/2$ 。

(2) 函数 $\text{pred}(x)$ 和 $\text{succ}(x)$

$\text{pred}(x)$ 和 $\text{succ}(x)$ 中 x 必须是顺序类型,前者结果求 x 的前趋值,后者是求 x 的后继值。

例如 $\text{pred}(5)=4$, $\text{pred}('5')='4'$, $\text{succ}(4)=5$, $\text{succ}('4')='5'$

(3) 序号函数 $\text{ord}(x)$

$\text{ord}(x)$ 的定义域是顺序类型。该函数求得值 x 在定义域中的序号。例如 $\text{ord}(\text{false})=0$, $\text{ord}(\text{true})=1$,若计算机采用 ASC II 码字符集,则 $\text{ord}('A')=65$ 。

(4) 字符函数 $\text{chr}(x)$

chr(x)是ord(x)的逆函数,它取字符集中序号为x的字符,x是整型,函数值是字符型。例如 chr(65)='A',chr(ord('A'))='A'。

(5)截尾函数 trunc(x)和四舍五入函数 round(x)

这两个函数用于把实型转换成整型。trunc(x)是通过略去实值的小数部分而得到整值的,如 trunc(9.8)=9, trunc(-5.7)=-5。round(x)求对x进行四舍五入,如 round(8.4)=8, round(9.8)=10, round(-3.8)=-4。

(6)判断函数 odd(x),eoln(x)和 eof(x)

odd(x)称奇数函数,x取值为整型或布尔型。如果x为奇数 odd(x)的值为 true,否则为 false。

eoln(x)称行结束函数,eof(x)称文件结束函数,其中的x是文件类型,在第10章经常用到,这里简单介绍一下。

eoln(x)的功能是当文件x到达当前行的末尾是时,取值为 true,否则为 false。

eof(x)的功能是当文件x到达文件末尾时,取 true,否则为 false。

1.6 表达式和赋值语句

1.6.1 表达式

表达式是由常量、变量、函数、运算符及圆括号组成的有意义运算式,它代表一个值。在Pascal语言中,表达式分为算术表达式和布尔表达式、集合表达式(见第7章)。

1. 算术表达式

算术表达式由算术运算符和操作数组成,其中算术运算符有+、-、*、/、DIV、MOD,操作数是整型或实型的常量、变量、函数。若操作数都是整型数据,则可用+、-、*、/、DIV、MOD组成表达式,结果为整型数据;若操作数都是实型数据,则可以用+、-、*、/组成表达式,结果为实型数据。在+、-、*运算中,若任一个运算符的操作对象中,一个是实型数据,而另一个是整型数据,则运算中整型数据自动转换为实型数据参加运算,结果为实型数据。在/运算中,无论操作对象是整型数据还是实型数据,结果均为实型数据。DIV或MOD运算中,操作数不允许是实型。表达式中不允许两个运算符连续出现。例如 $x * (-y)$ 不能写成 $x * -y$ 。

运算符的优先顺序为:*、/、DIV、MOD属同级,优先级最高;+、-属同级,优先级较前面四个运算符的优先级次之;同一优先级的运算符遵守自左向右的计算的规则。

2. 布尔表达式

布尔表达式是运算结果为真(true)或假(false)的运算式。它是由关系运算符、逻辑运算符及操作数组成。六种关系运算符:<、<=、=、<>、>=、>均要求两个操作数为相同类型的数据,但整型和实型可以在关系运算符两边混合出现,关系比较的结果为布尔值。逻辑运算符有三种:AND、OR、NOT,它们要求操作数为布尔型数据,逻辑运算结果为布尔值。例如,有如下变量说明:

```
VAR  a,b:integer;  
      x,y:real;
```

p,q,r:boolean;

则下列均为布尔表达式:

$a < > b$ $a = b$ NOT q $(a > b) \text{ AND } (x \leq y)$

需要注意的是应该避免直接测试实数数值是否相等,因为计算中,经一系列计算后,理论上相等的数实际上只不过是相近而已。例如 $a/b * b = a$ 可能永远得不到 true 值。比较两个实型量 a 和 b 是否相等,应该用下式代替

$\text{abs}(a-b) < 1\text{E}-6$

关于表达式,还要注意其中运算次序。计算优先顺序规则:

(1)括号内的表达式首先计算;

(2)运算符的优先级有四级,从高级到低级排列为

NOT

* / DIV MOD AND

+ - OR

< <= => <> > >=

同级的运算符从左到右进行

还应注意,当用关系式构成复杂的布尔表达式时,这些关系式必须放在括号内,其原因是关系运算符的优先级最低。例如

$(a < b) \text{ AND } (x \leq y)$ 为正确的布尔表达式,而 $a < b \text{ AND } x \leq y$ 是错误的。

1.6.2 赋值语句

【语法格式】

$\langle \text{变量名} \rangle := \langle \text{表达式} \rangle;$

其中,“:=”是赋值号,其作用是将赋值号右边表达式的值赋给其左边的变量。

通常,赋值号两边的类型必须一致。但是,下面的情况是正确的:

实型变量可以赋以整型表达式或整数类型子界表达式;

表达式类型子界可以赋以变量类型子界表达式。

1.7 输入输出语句及其控制格式

1.7.1 读语句 Read 和 Readln

要想把初始数据从键盘输入到计算机或者把运行结果输出到显示终端,就要使用输入语句(读语句)或输出语句(写语句)。

【语法格式】

read($\langle \text{变量名表} \rangle$);

readln($\langle \text{变量名表} \rangle$);

readln;

其中,变量名表是一个或用逗号隔开的多个变量,它可以是整型、实型、字符型,但不可以是布尔型,即不可以直接读入布尔值。read 和 readln 是读语句的标准过程名,它们可以出现在程序中任何需要输入数据的位置。