



# 新大纲

电子工业出版社隆重推出

新编全国计算机等级考试教材

# Pascal

# 语言程序设计

陈 斌 王建新 编著

(二级)



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
URL: <http://www.phei.com.cn>

新编全国计算机等级考试教材

# Pascal 语言程序设计(二级)

陈 斌 王建新 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是根据教育部 1998 年最新修订的全国计算机等级考试大纲编写的。内容覆盖大纲的全部内容,书中全部例题都用 Turbo Pascal 编译器调试通过。本书通俗易懂,简洁清晰,便于自学,而且配有大量针对等级考试的练习题,所有习题均附参考答案,便于检查自学效果。

本书虽为计算机等级考试编写的,亦可作为各类大专院校程序设计语言课程的教材或参考书。

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,翻印必究。

## 图书在版编目(CIP)数据

Pascal 语言程序设计/陈斌,王建新编著. - 北京:电子工业出版社,1999.8

新编全国计算机等级考试教材

ISBN 7-5053-5320-9

I . P… II . ①陈… ②王… III . Pascal 语言-程序设计-教材 IV . TP312

中国版本图书馆 CIP 数据核字(1999)第 13798 号

丛 书 名:新编全国计算机等级考试教材

书 名: Pascal 语言程序设计(二级)

编 著 者:陈 斌 王建新

策 划:胡毓坚

责任编辑:吴明卒

排版制作:电子工业出版社计算机排版室

印 刷 者:北京京安达明印刷厂

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:17 字数:435.2 千字

版 次:1999 年 8 月第 1 版 1999 年 8 月第 1 次印刷

书 号: ISBN 7-5053-5320-9  
TP·2647

印 数:6000 册 定价:22.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,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 语言是由瑞士苏黎世联邦技术学院 N. Wirth 教授于 1969 年研制成功的,是世界上第一个结构化程序设计语言。Pascal 语言常用于编写系统软件、应用软件以及进行科学计算等,是世界上最流行的程序设计语言之一。

Pascal 语言的特点是语法简洁、规范,特别适合于教学和便于自学。近年来,由于基于 Pascal 语言的可视化程序设计语言 Delphi 的出现,使 Pascal 语言旧貌换新颜,身价倍增,越来越多的基于 Windows 的应用程序采用 Delphi 开发。

本书从基本概念出发,循序渐进地介绍 Pascal 语言的程序组成、各种数据类型(包括四种标准类型、枚举类型、子界类型、数组类型、集合类型、记录类型、指针类型和文件类型)以及它们在程序设计中的应用,覆盖了考试大纲的全部内容。整本书贯穿结构化程序设计思想,针对等级考试的题型配有大量例题和练习,所有的练习均附有参考答案。全部程序均使用等级考试要求的,也是目前最流行的 Turbo Pascal 编译器调试通过。本书还对等级考试中的上机考试环境及 Turbo Pascal 的调试环境做了介绍。

本书共分 12 章,第 1~4 章由王建新编写,第 5~12 章由陈斌编写,全书由陈斌统稿。

由于水平有限,加之时间紧迫,书中必有谬误或不妥之处,恳请读者批评指正。

编 者

1999 年 1 月

# 目 录

|                          |        |
|--------------------------|--------|
| 第 1 章 Pascal 语言的基本概念     | ( 1 )  |
| 1.1 Pascal 程序的基本组成       | ( 1 )  |
| 1.1.1 程序首部               | ( 1 )  |
| 1.1.2 程序体                | ( 2 )  |
| 1.2 Pascal 语言的基本元素       | ( 2 )  |
| 1.2.1 Pascal 语言中的字符      | ( 3 )  |
| 1.2.2 Pascal 语言中的符号      | ( 3 )  |
| 习题                       | ( 6 )  |
| 第 2 章 数据与表达式             | ( 8 )  |
| 2.1 四种标准数据类型             | ( 8 )  |
| 2.1.1 整数类型 ( Integer )   | ( 8 )  |
| 2.1.2 实数类型 ( Real )      | ( 9 )  |
| 2.1.3 字符类型 ( Char )      | ( 10 ) |
| 2.1.4 布尔类型 ( Boolean )   | ( 10 ) |
| 2.2 常量和常量的定义 ( Const )   | ( 11 ) |
| 2.3 变量和变量说明 ( Var )      | ( 11 ) |
| 2.4 标准函数                 | ( 12 ) |
| 2.5 表达式                  | ( 14 ) |
| 2.5.1 算术表达式              | ( 14 ) |
| 2.5.2 字符表达式              | ( 15 ) |
| 2.5.3 布尔表达式              | ( 15 ) |
| 2.5.4 表达式的求值规则           | ( 16 ) |
| 习题                       | ( 18 ) |
| 第 3 章 Pascal 语言的基本语句     | ( 21 ) |
| 3.1 赋值语句                 | ( 21 ) |
| 3.2 输入语句 Read 和 Readln   | ( 23 ) |
| 3.2.1 输入语句的进一步说明         | ( 24 ) |
| 3.2.2 输入语句的应用            | ( 25 ) |
| 3.3 输出语句 Write 和 Writeln | ( 26 ) |
| 3.3.1 输出语句的进一步说明         | ( 26 ) |
| 3.3.2 输出格式               | ( 27 ) |
| 习题                       | ( 35 ) |
| 第 4 章 选择语句和循环语句          | ( 39 ) |
| 4.1 IF 语句                | ( 39 ) |

|                                    |              |
|------------------------------------|--------------|
| 4.1.1 IF 语句的第一种格式 .....            | (39)         |
| 4.1.2 IF 语句的第二种格式 .....            | (40)         |
| 4.1.3 IF 语句的嵌套 .....               | (41)         |
| 4.1.4 IF 语句中 BEGIN 和 END 的使用 ..... | (43)         |
| 4.2 Case 语句 .....                  | (44)         |
| 4.3 GOTO 语句和语句标号的使用 .....          | (47)         |
| 4.3.1 语句标号 .....                   | (47)         |
| 4.3.2 GOTO 语句 .....                | (47)         |
| 4.4 WHILE 语句 .....                 | (48)         |
| 4.5 REPEAT 语句 .....                | (51)         |
| 4.6 FOR 语句 .....                   | (52)         |
| 4.6.1 FOR 语句第一种格式 .....            | (52)         |
| 4.6.2 FOR 语句第二种格式 .....            | (55)         |
| 4.7 多重循环 .....                     | (56)         |
| 习题 .....                           | (58)         |
| <b>第 5 章 枚举和子界类型 .....</b>         | <b>(66)</b>  |
| 5.1 枚举类型 .....                     | (66)         |
| 5.1.1 枚举类型的定义 .....                | (66)         |
| 5.1.2 使用枚举类型注意的问题 .....            | (67)         |
| 5.2 子界类型 .....                     | (69)         |
| 5.2.1 子界类型的定义 .....                | (69)         |
| 5.2.2 使用子界类型注意的问题 .....            | (70)         |
| 5.3 子界类型的应用 .....                  | (71)         |
| 习题 .....                           | (75)         |
| <b>第 6 章 数组 .....</b>              | <b>(78)</b>  |
| 6.1 一维数组 .....                     | (78)         |
| 6.1.1 一维数组的定义 .....                | (78)         |
| 6.1.2 数组分量的引用 .....                | (79)         |
| 6.1.3 举例 .....                     | (79)         |
| 6.2 多维数组 .....                     | (88)         |
| 6.2.1 二维数组 .....                   | (88)         |
| 6.2.2 多维数组 .....                   | (90)         |
| 6.2.3 使用数组应注意的问题 .....             | (91)         |
| 6.3 紧缩数组 .....                     | (92)         |
| 6.3.1 紧缩数组简介 .....                 | (92)         |
| 6.3.2 紧缩的布尔数组 .....                | (93)         |
| 6.3.3 紧缩字符数组类型 .....               | (94)         |
| 习题 .....                           | (95)         |
| <b>第 7 章 集合类型和记录类型 .....</b>       | <b>(106)</b> |
| 7.1 集合类型 .....                     | (106)        |
| 7.1.1 集合类型的定义 .....                | (106)        |

|               |                                    |              |
|---------------|------------------------------------|--------------|
| 7.1.2         | 集合的表示 .....                        | (107)        |
| 7.1.3         | 集合的运算 .....                        | (108)        |
| 7.1.4         | 集合的应用 .....                        | (108)        |
| 7.2           | 记录类型 .....                         | (112)        |
| 7.2.1         | 记录类型的定义 .....                      | (112)        |
| 7.2.2         | 记录类型变量的访问 .....                    | (114)        |
| 7.2.3         | WITH 语句 .....                      | (116)        |
| 7.2.4         | 记录的变体 .....                        | (120)        |
| 7.3           | 类型间的关系 .....                       | (121)        |
| 7.3.1         | 类型等同 .....                         | (122)        |
| 7.3.2         | 类型相容 .....                         | (122)        |
| 7.3.3         | 赋值相容 .....                         | (123)        |
| 习题            | .....                              | (124)        |
| <b>第 8 章</b>  | <b>过程和函数 .....</b>                 | <b>(132)</b> |
| 8.1           | 过程(PROCEDURE).....                 | (132)        |
| 8.1.1         | 过程说明 .....                         | (132)        |
| 8.1.2         | 过程调用 .....                         | (133)        |
| 8.2           | 函数(FUNCTION) .....                 | (135)        |
| 8.2.1         | 函数说明 .....                         | (135)        |
| 8.2.2         | 函数调用 .....                         | (136)        |
| 8.3           | 数值参数和变量参数 .....                    | (138)        |
| 8.4           | 全程量和局部量 .....                      | (142)        |
| 8.5           | 递归和向前引用 .....                      | (147)        |
| 8.5.1         | 递归的概念 .....                        | (147)        |
| 8.5.2         | 递归程序 .....                         | (147)        |
| 8.5.3         | 向前引用 .....                         | (150)        |
| 习题            | .....                              | (152)        |
| <b>第 9 章</b>  | <b>指针和动态变量 .....</b>               | <b>(159)</b> |
| 9.1           | 指针类型和指针变量 .....                    | (159)        |
| 9.1.1         | 指针类型的定义 .....                      | (159)        |
| 9.1.2         | 指针变量的操作 .....                      | (160)        |
| 9.2           | 标准过程 New 和 Dispose .....           | (161)        |
| 9.3           | 指针的应用 .....                        | (163)        |
| 9.3.1         | 单链表 .....                          | (163)        |
| 9.3.2         | 循环链表 .....                         | (171)        |
| 习题            | .....                              | (172)        |
| <b>第 10 章</b> | <b>文件 .....</b>                    | <b>(177)</b> |
| 10.1          | FILE 类型文件 .....                    | (177)        |
| 10.1.1        | FILE 类型文件的定义 .....                 | (177)        |
| 10.1.2        | 非 Pascal 标准过程 Assign 和 Close ..... | (178)        |
| 10.1.3        | 若干标准过程和标准函数 .....                  | (179)        |

|               |                                    |              |
|---------------|------------------------------------|--------------|
| 10.1.4        | 建立 FILE 型文件模式 .....                | (179)        |
| 10.1.5        | 读取 FILE 型文件模式 .....                | (181)        |
| 10.1.6        | 复制 FILE 型文件模式 .....                | (182)        |
| 10.1.7        | 修改 FILE 型文件模式 .....                | (182)        |
| 10.1.8        | 举例 .....                           | (183)        |
| 10.2          | TEXT 类型文件 .....                    | (189)        |
| 10.2.1        | 与 TEXT 类型文件有关的标准过程,标准函数及约定 .....   | (190)        |
| 10.2.2        | TEXT 类型文件的读文件模式 .....              | (191)        |
| 10.2.3        | TEXT 类型文件的复制文件模式 .....             | (192)        |
| 10.2.4        | TEXT 类型文件与 FILE 类型文件的比较 .....      | (192)        |
| 10.2.5        | TEXT 类型文件应用举例 .....                | (193)        |
| 习题            | .....                              | (195)        |
| <b>第 11 章</b> | <b>上机考试系统介绍及考试指导 .....</b>         | <b>(198)</b> |
| 11.1          | 上机考试系统简介 .....                     | (198)        |
| 11.1.1        | 系统环境 .....                         | (198)        |
| 11.1.2        | 上机考试登录 .....                       | (199)        |
| 11.1.3        | 考生目录 .....                         | (200)        |
| 11.1.4        | 上机考试时间 .....                       | (200)        |
| 11.1.5        | 上机考试题型及分值 .....                    | (200)        |
| 11.1.6        | 试题内容查阅工具的使用 .....                  | (201)        |
| 11.1.7        | 上机考试过程中注意事项 .....                  | (204)        |
| 11.2          | 上机考试内容 .....                       | (204)        |
| 11.2.1        | 操作系统考试题 .....                      | (204)        |
| 11.2.2        | 程序修改考试题 .....                      | (206)        |
| 11.2.3        | 程序设计考试题 .....                      | (208)        |
| 习题            | .....                              | (209)        |
| <b>第 12 章</b> | <b>Turbo Pascal 5.0 使用简介 .....</b> | <b>(216)</b> |
| 12.1          | Turbo Pascal 的安装 .....             | (216)        |
| 12.1.1        | 将 Turbo Pascal 安装到硬盘 .....         | (216)        |
| 12.1.2        | 将 Turbo Pascal 安装到软盘 .....         | (217)        |
| 12.2          | Turbo Pascal 的集成开发环境 .....         | (217)        |
| 12.2.1        | Turbo Pascal 的启动 .....             | (217)        |
| 12.2.2        | File 菜单 .....                      | (218)        |
| 12.2.3        | Run 菜单 .....                       | (219)        |
| 12.2.4        | Compile 菜单 .....                   | (219)        |
| 12.2.5        | Break/Watch 菜单 .....               | (220)        |
| 12.2.6        | 主菜单快捷键 .....                       | (221)        |
| 12.3          | Turbo Pascal 编程步骤 .....            | (221)        |
| 12.4          | Turbo Pascal 常用编辑命令 .....          | (222)        |
| 12.4.1        | 光标移动编辑命令 .....                     | (223)        |
| 12.4.2        | 插入、删除、块操作类命令 .....                 | (223)        |
| 12.5          | 程序调试 .....                         | (224)        |

|                                    |              |
|------------------------------------|--------------|
| 12.5.1 准备调试 .....                  | (224)        |
| 12.5.2 调试命令 .....                  | (225)        |
| <b>附录 1 标准过程和函数清单 .....</b>        | <b>(227)</b> |
| <b>附录 2 ASCII 码表 .....</b>         | <b>(228)</b> |
| <b>附录 3 习题参考答案 .....</b>           | <b>(229)</b> |
| 第 1 章习题答案 .....                    | (229)        |
| 第 2 章习题答案 .....                    | (229)        |
| 第 3 章习题答案 .....                    | (230)        |
| 第 4 章习题答案 .....                    | (231)        |
| 第 5 章习题答案 .....                    | (234)        |
| 第 6 章习题答案 .....                    | (235)        |
| 第 7 章习题答案 .....                    | (238)        |
| 第 8 章习题答案 .....                    | (239)        |
| 第 9 章习题答案 .....                    | (240)        |
| 第 10 章习题答案 .....                   | (241)        |
| 第 11 章习题答案 .....                   | (241)        |
| <b>附录 4 全国计算机等级考试二级笔试试卷 .....</b>  | <b>(244)</b> |
| <b>附录 5 二级考试大纲(1998 年修订) .....</b> | <b>(259)</b> |
| 基本要求 .....                         | (259)        |
| 考试内容 .....                         | (259)        |
| <b>参考文献 .....</b>                  | <b>(262)</b> |

## 第 1 章 Pascal 语言的基本概念

Pascal 语言是一种结构化的程序设计语言。该语言诞生于 1969 年,是由 Niklaus Wirth 教授在苏黎世联邦工学院研制成功的。后经不断改进与完善以及国际标准组织(ISO)先后两次认证,逐渐成为世界广泛使用的程序设计语言。

1985 年美国 Borland 公司又推出 Turbo Pascal V3.0,将 Pascal 语言发展成一种操作方便,具有友好用户界面,集编辑、编译、运行于一体的程序设计语言。

目前,Pascal 语言成为我国很多大、中专院校计算机专业的第一程序设计语言。而且,越来越多的非计算机专业的学生也开始把 Pascal 语言作为学习程序设计的首选语言。

### 1.1 Pascal 程序的基本组成

我们首先给出一个简单的 Pascal 实例,通过它能够看到 Pascal 程序的基本组成。

【例 1.1】 任意输入三个整数,求出它们的和与平均值。

```
PROGRAM exp1-1 (Input,Output);           程序首部
VAR
    a, b, c, x1: Integer;                } 说明部分
    x2: Real;                             }
BEGIN                                     } 程序体
    Readln(a,b,c);                       }
    x1 := a + b + c;                      } 语句部分
    x2 := x1 / 3;                         }
    Writeln('x1 = ', x1, 'x2 = ', x2)    }
END.
```

一个完整的 Pascal 程序是由程序首部和程序体组成的。

#### 1.1.1 程序首部

程序首部包括三部分内容,它们分别是:

##### 1 PROGRAM

写在最前面的 Pascal 的保留字“PROGRAM”是 Pascal 程序的标记,每一个 Pascal 程序必须以此为开头。

##### 2 程序名

程序名由用户自己给定,但必须符合“标识符”的定义规则。在上例中,程序名就是 exp1-1。

### 3 参数表

参数表是用一对圆括号括起来的文件列表。文件与文件之间用逗号分隔。在上例中, Input 和 Output 是两个标准文件,它说明了程序中的输入与输出操作。在 Turbo Pascal 中,参数表可以省略。

程序首部一定要以分号结束。

#### 1.1.2 程序体

程序体也可称为“分程序”,它是由说明部分和语句部分构成的。

##### 1 说明部分

说明部分必须出现在语句部分之前,用来说明或定义程序中用到的标号、常量、类型、变量、过程和函数。在说明或定义时分别冠以保留字 LABEL、CONST、TYPE、VAR、PROCEDURE 和 FUNCTION。在标准的 Pascal 中规定,这些保留字出现的先后顺序不能改变,但可以省略。上例中,只定义了整型变量 a、b、c、x1 和实型变量 x2,所以只出现了 VAR(变量)说明。

##### 2 语句部分

语句部分是程序中的实体,是程序的执行部分。它由一系列语句组成,每个语句执行一种操作。语句与语句之间用分号分隔。整个语句部分又是一个复合语句。BEGIN 和 END 表示复合语句的开始和结束。在标准的 Pascal 中规定,END 和最后一个语句之间没有分号。最后的 END 必须用圆点结尾。

在上例中,语句:

Readln(a,b,c) 的作用是接收用户从键盘输入的三个整型数据,并分别赋予整型变量 a、b、c。语句:

$x1 := a + b + c$  的作用是将整型变量 a、b、c 的值相加,并将和赋予整型变量 x1。语句:

$x2 := x1 / 3$  的作用是将 x1 的值除以 3,并将结果赋予实型变量 x2(即三个整型数据的平均值)。语句:

Writeln('x1 = ', x1, 'x2 = ', x2) 的作用是输出字符串 'x1 = ' 后输出变量 x1 的值;在字符串 'x2 = ' 后输出变量 x2 的值。

## 1.2 Pascal 语言的基本元素

任何一种高级语言都有自己的一套基本元素。借助这些基本元素,按照一定的语法规则就可构成 Pascal 的语句,并进一步构成 Pascal 语言。

### 1.2.1 Pascal 语言中的字符

#### 1 字母

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| a | b | c | d | e | f | g | h | i | j | k | l | m |
| n | o | p | q | r | s | t | u | v | w | x | y | z |

在使用字母时,除非出现在字符串(见下面字符串说明)之内的字母,否则大小写字母被认为是相同的。

#### 2 数字

0 1 2 3 4 5 6 7 8 9

#### 3 其它

+ - \* / = < > . , ; : ↑ ( ) [ ] { } ' 空格

### 1.2.2 Pascal 语言中的符号

Pascal 语言中的符号包括特定符号、标识符、数、字符串、注释、分隔符和标号。

#### 1 特定符号

在 Pascal 中,那些具有特殊意义的符号被称为特定符号。字特定符号和非字特定符号都属于特定符号。

(1)字特定符号或者叫“保留字”是英文单词或它们的缩写。它们在 Pascal 中具有特殊意义,因此不允许对它们重新定义,也不能再做它用。“保留字”的作用是用来命名 Pascal 语句,表示某些运算,描述某些数据类型,说明程序、过程和函数首部的。

标准 Pascal 中的“保留字”有 35 个,它们分别是:

|        |        |          |           |         |
|--------|--------|----------|-----------|---------|
| AND    | ARRAY  | BEGIN    | CASE      | CONST   |
| DIV    | DO     | DOWNT0   | 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    |

#### (2)非字特定符号

+ - \* / < <= = <> >= > ↑  
 . , ' ; : := .. ( ) [ ] { }

## 2 标识符

标识符是以字母开头的字母数字串。例如:a、b、c、x1、x2、Char、Read 都是标识符。标识符分预定义的标准标识符和用户定义的标识符两种。

### (1) 预定义的标准标识符

预定义的标准标识符是 Pascal 系统预定义的,它们具有特定的含义,可以在程序中直接引用。它们用来标识标准常量、标准变量、标准类型、标准过程、标准函数和标准文件。

标准 Pascal 中有 40 个标准标识符,现分列如下:

|      |         |        |         |       |         |
|------|---------|--------|---------|-------|---------|
| 标准常量 | False   | Maxint | True    |       |         |
| 标准类型 | Boolean | Char   | Integer | Real  | 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 |         |       |         |

在程序的书写过程中,大小写是可以不分的,但为了清楚起见本书中把所有‘保留字’用大写字母表示;预定义的标准标识符第一个字母用大写字母表示,其余用小写字母表示;用户定义的标识符一般用小写字母表示。

### (2) 用户定义的标识符

用户定义的标识符是指编写程序时,用户自己为变量、常量、类型、函数、过程以及程序所起的名字。

用户在定义标识符时一定要注意:不能与 35 个保留字同名;最好也不要与预定义的标准标识符同名。因为在程序中一旦把预定义的标准标识符作为用户定义的标识符,预定义的标准标识符的自身意义将被破坏,这样很容易给程序带来意想不到的混乱。

用户在定义标识符时还必须严格遵守标识符的定义规则,即必须是以字母开头的字母数字串。下面是一些合法的用户自定义标识符:

abc            m32            k            p3            ms586

而以下是一些非法的用户自定义标识符:

386cd            a - b            k \*            mod            55

这是因为 386cd 是以数字开头;a - b 和 k \* 使用了字符“-”和字符“\*”;而 mod 是保留字;55 是常量。

用户定义的标识符的最大长度一般不要超过 31 个字符,如果超出,Pascal 系统只识别前 31 个字符。

### 3 数的表示

在 Pascal 语言中,数的表示采用十进制表示法,并且有整数与实数之分。实数既可以用小数表示法也可以用科学表示法来表示。有关数的进一步说明将在下一章中给出。

### 4 分隔符

分号、逗号、空格以及行结束符称为 Pascal 的分隔符。Pascal 规定任意两个相邻标识符、数、保留字或者两个语句之间必须至少有一个分隔符。比如:

```
IF k > 5 THEN n := n + 1
```

是一个条件语句,其意是如果变量 k 的值大于 5,则将变量 n 的值加 1 再送入变量 n。语句中 IF 与 k > 5 之间必须至少加入一个空格。如果把该语句写成:

```
IFk > 5 THEN n := n + 1
```

那么 Pascal 的编译程序就将把“IFk”看成用户自定义的标识符,整个语句的语义就遭到破坏。除此之外,人们在编写程序时往往要在程序中适当的位置加入一些空格。下面是求解一元二次方程  $ax^2 + bx + c = 0$  两个实根的程序:

```
PROGRAM exp (Input,Output);
VAR
    a,b,c,t,x1,x2:Real;
BEGIN
    Readln(a,b,c);
    t := SQRT(b * b - 4 * a * c);
    IF t < 0.0 THEN
        WRITELN('*****')
    ELSE BEGIN
        x1 := (-b + t) / 2 / a;
        x2 := (-b - t) / 2 / a;
        Writeln('x1 = ',x1,'x2 = ',x2)
    END
END.
```

从这个程序中我们可以看出,由于加入了适当的空格,使得程序层次鲜明,结构清晰,既便于调试,又便于阅读。

在使用分隔符时还应该注意分隔符不能相互替换。该用分号的地方不能用逗号,该用逗号的地方不能用分号。上例中第三行如果写成

```
a;b;c;t;x1;x2:Real;
```

或者写成

```
a b c t x1 x2:Real;
```

都是错误的。

## 5 注释

在 Pascal 语言中,用户可在程序中的任何位置加入注释说明。注释说明的作用在于标出该程序的功能或某些语句的功能,使读程序的人一目了然。注释是用字符“{”和“}”括起来的字符序列。Pascal 的编译程序对“{”和“}”内的所有内容不做任何处理。因此,注释说明不影响程序的执行。当然,“{”和“}”必须成对出现。如果某个计算机键盘上没有花括号,可用“( \* ”和“ \* )”代替。

## 习题

### 一 单选题

1. 一个完整的程序应当包括哪些部分? 下列说法正确的是\_\_\_\_\_。  
(A)程序首部和程序体。  
(B)程序首部和语句部分。  
(C)说明部分和程序体。  
(D)说明部分和语句部分。
2. 对程序首部来说,下列说法正确的是\_\_\_\_\_。  
(A)PROGRAM 可以用 PROG 代替。  
(B)PROGRAM 可以省略。  
(C)PROGRAM 不可省略。  
(D)PROGRAM 必须用英文大写。
3. 对程序首部来说,下列说法正确的是\_\_\_\_\_。  
(A)程序名后必须带有参数表。  
(B)程序名后的参数表可以省略。  
(C)程序名和参数表都可以省略。  
(D)程序名可以省略,参数表不可省略。
4. 对程序体来说,下列说法正确的是\_\_\_\_\_。  
(A)说明部分必须以 VAR 开头。  
(B)说明部分可以出现在语句部分之内。  
(C)语句部分之前一定要出现说明部分。  
(D)说明部分可以省略。
5. 对于字特定符号(保留字)来说,下列说法正确的是\_\_\_\_\_。  
(A)它们可以被重新定义,但必须在说明部分给出明确说明。  
(B)如果太长可以用前四个字母表示。  
(C)它们不能重新定义。  
(D)它们必须用大写字母表示。
6. 下列哪一组是正确的用户自定义标识符\_\_\_\_\_。

- (A) mp3, kl2, program1, windows。
- (B) mn3, set, x1, nn。
- (C) date, 88,  $x - y$ ,  $[x]$ 。
- (D) time, not, div, \*。

7. 下列的说明部分,哪一组是正确的\_\_\_\_\_。

(A) VAR

x; y; z: Integer

(B) CONST

pai = 3.14159,  
min = -1000,  
star = ' \* ';

(C) VAR

x y z: Integer;

(D) VAR

x, y, z: Integer;

## 第 2 章 数据与表达式

Pascal 语言是一种强类型语言。Pascal 程序的每一个数据项都或隐或现地与一种数据类型相联系。数据类型不仅规定了该数据项的取值范围,而且也规定了它们能执行的运算。

Pascal 语言共有 11 种数据类型,可以划分为简单类型、构造类型、指针类型三大类型,具体划分如图 2-1 所示。

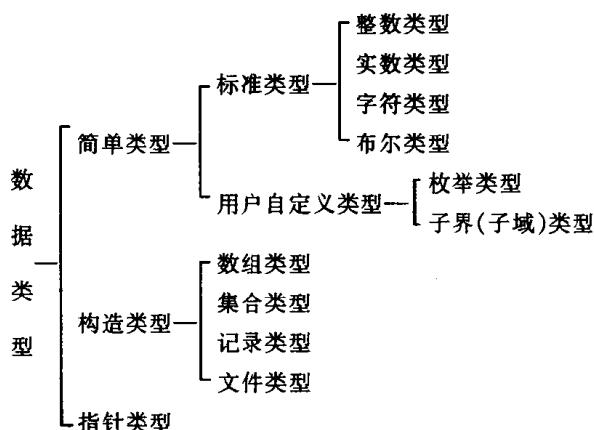


图 2-1 Pascal 语言的数据类型

本章讲授的数据类型是标准类型。其他类型将在后面的章节中逐一介绍。

### 2.1 四种标准数据类型

#### 2.1.1 整数类型 ( Integer )

##### 1 整数类型的值域

因受到计算机硬件和 Pascal 编译器的制约,在 Pascal 中的整数集合只是数学中整数集合的一个子集。标识符 Maxint 是 Pascal 中的最大整型常量。所以,整数类型的值域可以被描述为:

$- \text{Maxint}, \dots, -1, 0, 1, \dots, \text{Maxint}$

Maxint 可以直接在程序中使用,而不必被定义。在一般的计算机上,其值是 32767。下面是几个合法的整型数:

217, 0, -55, Maxint

而 55.0, 0.0, 97635901 都是非法的整型数。因为在描述整型数时不能使用小数点,其值不能超过 32767。