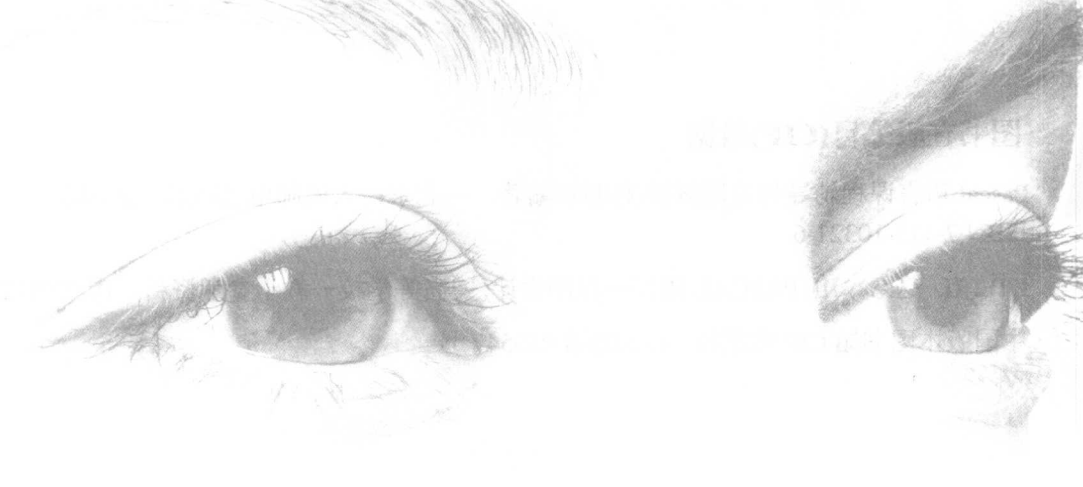




21 世纪计算机辅导系列丛书

Pascal 语言 课程辅导与 习题解析

● 赵明华 编著



21 世纪计算机辅导系列丛书

Pascal 语言 课程辅导与 习题解析

● 赵明华 编著

NBJS11/06

人民邮电出版社

图书在版编目(CIP)数据

Pascal 语言课程辅导与习题解析/赵明华编著. —北京: 人民邮电出版社, 2002.7
ISBN 7-115-10321-6

I. P...II. 赵... III. PASCAL 语言—程序设计 —高等学校—教学参考资料 IV. TP312
中国版本图书馆 CIP 数据核字(2002)第 036547 号

内容提要

本书通过讲解 Pascal 语言的基本概念及众多习题的解答, 来指导读者学习掌握 Pascal 语言。全书介绍了 Pascal 语言的基本语法, 包括数据类型及运算、基本语句、结构控制、数组和函数等, 并介绍了包括编译预处理、指针、结构体、位运算和文件操作等较深层次的内容。

全书习题编排紧扣各相关知识点, 并解答部分历年计算机等级考试的习题, 在各习题解答后附有相关练习。

本书适合参加计算机等级考试的考生作为习题集使用, 亦适合中专、在职培训、高等院校的师生使用。

21 世纪计算机辅导系列丛书

Pascal 语言课程辅导与习题解析

◆ 编 著 赵明华
责任编辑 张立科

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 20
字数: 488 千字 2002 年 7 月第 1 版
印数: 1-5 000 册 2002 年 7 月北京第 1 次印刷

ISBN7-115-10321-6/TP·2886

定价: 26.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

Pascal 语言创立于 20 世纪 70 年代初期, 由瑞士苏黎世联邦技术学院的 N·Wirth 教授推出。为纪念法国数学家帕斯卡, 而命名为 Pascal 语言。

Pascal 语言是典型的结构化程序设计语言。相比于最初的高级语言 BASIC, 它引入了过程、函数、结构类型等概念, 摒弃了数据类型不明显的变量和无条件转向语句的滥用, 使得整个程序的结构更加清晰, 意义更加明显, 便于阅读和更新代码, 可移植和传播性都很强。它的出现, 标志着第二代高级语言的创立, 在程序设计语言发展史上具有里程碑式的意义。

相比其他高级语言, Pascal 语言特别适合于教学, 阐述结构化设计思想的基本概念和方法。它具有丰富的数据类型, 操作简单明了的语句, 强大的编译功能, 高运行效率, 这些都为运用时的灵活方便提供了良好的基础。除此以外, Pascal 语言也用于应用软件设计中。目前程序员常用的 Delphi 工具就是以 Pascal 语言为蓝本编制的。

本书系统地介绍了标准 Pascal 语言的基本概念和语法, 并从中引出结构化程序设计的基本思想和方法。每一章节都分为三部分。在“知识要点”中, 力求深入浅出地讲授每个知识点, 对编程中应着重注意的方面, 进行详细说明, 以提醒读者注意。而对于一般的内容, 则点到即止。在“典型题解”中, 配合知识要点组织了相应例题, 并写有详细的解法和注释, 能够让读者深入地体验编程时的思想, 达到学习和锻炼的目的。在“即学即练”中, 列出了一些习题, 可以让读者更加熟练地运用知识点, 也能对知识要点起到补充的作用。习题的答案在“小节答案”中, 可以作为提示和检测。最好的学习方式不是死记硬背知识点, 而是通过练习来理解, 通过实践来掌握。因此, 本书突出的特点就是以练习题为主进行辅导和训练。每一章的最后一节都是“集中训练营”, 安排了大量习题, 用于对整章知识点应用的综合考察, 增强熟练程度。

全书分 10 章。第 1 章介绍了 Pascal 语言的基本知识, 包括标识符、数据类型、语句体等基本概念。第 2 章讲述几种表达式的组成, 和基本语句的运用。从第 3 章开始到第 7 章, 分别讲述 Pascal 语法的重点内容: 选择、循环结构, 数组类型, 过程与函数, 集合与记录, 指针类型和文件类型。每一章介绍主要数据类型的定义形式和应用方法, 并把它放在具体题目中, 使前后知识连贯起来, 让读者有更明晰的理解。其中, 第 3 章和第 4 章是全书的核心部分, 这是程序设计基本方法的基础, 对于初学编程的读者来说, 是自己通过程序交互让计算机为你工作的初次尝试。后面的第 5 章到第 7 章的内容, 相对独立, 体现了 Pascal 语言丰富的数据类型, 在处理实际问题的能力方面也能上升一个层次。第 8 章是实际操作部分, 讲述了上机操作的注意事项。第 9 章和第 10 章给出了两组综合练习题, 这些习题均选自往届全国计算机水平等级考试二级试题, 以帮助读者学习和掌握 Pascal 语言。

由于时间仓促, 错误与疏漏之处在所难免, 敬请各位读者批评指正。

编 者

目 录

第 1 章	Pascal 语言基础	1
1.1	程序的构成	2
1.2	基本字符集、标识符与保留字	4
1.3	代码与注释的书写规范	7
1.4	数据类型、常量定义与变量说明	11
1.5	运算符与优先级	18
1.6	集中训练营	21
1.7	参考答案	34
第 2 章	表达式与基本语句	36
2.1	预定义标准函数	36
2.2	赋值与算术表达式	39
2.3	关系表达式和条件表达式	42
2.4	输入与输出语句	44
2.5	集中训练营	49
2.6	参考答案	58
第 3 章	选择与循环结构	59
3.1	IF 语句	60
3.2	CASE 语句	63
3.3	FOR 循环结构	66
3.4	WHILE 语句	69
3.5	REPEAT 循环	72
3.6	跳出循环	74
3.7	循环结构嵌套	75
3.8	集中训练营	79
3.9	参考答案	111
第 4 章	数组与函数	113
4.1	一维数组与多维数组	114
4.2	字符串与字符数组	119
4.3	过程与函数的定义	122
4.4	形参与实参	126
4.5	函数调用与返回值	130
4.6	变量的存储类别与作用域	132
4.7	函数的嵌套与递归	134
4.8	集中训练营	144
4.9	参考答案	182

第 5 章 集合与记录	185
5.1 集合 (SET) 的定义和说明	186
5.2 集合的运算	188
5.3 记录 (RECORD) 的定义和说明	193
5.4 记录的访问和应用	197
5.5 记录的嵌套	201
5.6 集中训练营	203
5.7 参考答案	218
第 6 章 指针	220
6.1 指针 (Point) 类型定义和说明	221
6.2 指针的操作	223
6.3 指针与链表	225
6.4 集中训练营	232
6.5 参考答案	245
第 7 章 文件操作	246
7.1 类型文件与正文文件	246
7.2 文件的打开与关闭	248
7.3 文件的读写与定位	250
7.4 集中训练营	255
7.5 参考答案	268
第 8 章 上机操作	269
8.1 编译器的安装	269
8.2 菜单和窗口	270
8.3 程序的调试	272
8.4 常见错误的处理	273
8.5 集中训练营	275
8.6 参考答案	276
第 9 章 综合练习 1	290
9.1 习题	290
9.2 答案	302
第 10 章 综合练习 2	304
10.1 习题	304
10.2 答案	312

第1章

Pascal 语言基础

本章讲述 Pascal 语言的基础部分，主要内容包括 Pascal 程序的构成、主要符号、代码书写规范、数据类型、运算符与优先级。在这些内容中包含了 Pascal 语言的许多基本而重要的概念，是学习以后各章的基础，读者应在理解的基础上牢记。分析例题，做好习题，将会有助于你深入理解本章的知识要点。

✚ 程序的基本构成方式

掌握程序首部、程序体的各部分组成规则，程序标志、程序名、程序参数的书写规范，说明部分和语句部分的语法规则。注意程序代码书写应遵循的顺序，注意将各部分之间当用分号隔开，程序用 END 加圆点结束。

✚ 标识符与保留字

掌握 Pascal 所采用的基本字符集，掌握合法标识符的命名规则，理解标识符和保留字的含义，分析它们的区别，牢记常用的标准标识符。学会自定义标识符的使用方法，另外要注意书中提到的一些细节的问题。

✚ 代码与注释

掌握代码的书写规范以及加入注释的方法。

✚ 数据类型

掌握 4 种简单数据类型：整型、实型、字符型、布尔型。了解它们各自的特点，针对每种类型学会不同的处理方式，另外学习枚举和子界两种自定义数据类型的定义和使用方法。

✚ 常量与变量

掌握常量和变量的说明方法以及在语句中如何使用，并比较它们的异同。

✚ 运算符与优先级

掌握 3 种常用运算符：算术运算符、关系运算符、逻辑运算符，牢记它们使用时的注意事项。理解优先级的概念，记忆常用运算符的优先顺序。

1.1 程序的构成

知识要点

程序是一批数据描述和一组计算机可以执行的语句或规则的汇集。Pascal 程序由程序首部和程序体两大部分组成。

程序首部，按照程序的书写顺序，可以分为程序标志、程序名、程序参数三个部分。程序标志就是保留字 PROGRAM（不区分大小写），这是 Pascal 程序的标志，必须位于程序开头。程序名由程序员自己定义，必须是合法的标识符（参见 1.2 节）。程序参数位于圆括号内，表明程序与外部联系的文件名，文件名之间用逗号隔开。Input 是标准输入文件，Output 是标准输出文件。在 Turbo Pascal 中，程序参数可以缺省。

程序首部以分号结尾，例如：PROGRAM Example (Input, Output);

程序体是程序的主体部分。按照书写顺序，可以分为说明部分和语句部分。

说明部分主要包括常量说明和变量说明，用于定义和说明程序中要用到的数据，描述这些数据的属性。说明部分内容丰富，除了上面讲到的常量定义和变量说明外，还可以有标号说明、类型定义、过程与函数说明。另外，对程序中用到的已知单元，还要加入相应 uses 语句进行说明。

语句部分描述程序所要执行的算法和操作。语句部分以保留字 BEGIN 起始，以 END. 结束，中间是一些执行具体操作的语句，用分号作结尾。BEGIN 和 END 表示复合语句的开始和结束，必须成对出现。在每个 END 之前出现的最后一条语句，其分号可以省略。

例如：简单的 Pascal 程序——Hello, the World!

程序执行后输出“PROGRAM 1: Hello, the World!”字样。

```
PROGRAM Hello(Output);           {程序首部}
VAR                               {说明部分}
    a: Integer;
BEGIN                             {语句部分}
    a:=1;
    Write ('PROGRAM ', a, ': ');
    Writeln('Hello, the World! ')
END.                               {程序结束}
```

在 Turbo Pascal 中，程序首部、说明部分都不是必须要出现的；而语句部分也允许空操作，不包含任何语句。因此，最简单的合法的 Turbo Pascal 程序就是：

```
BEGIN
END;
```


典型题解**【例 1.1.1】**

判断下列程序首部的书写是否有语法错误。

- (1) PROGRAM Ex!(Input)
- (2) PROGRAM (Input,Output);
- (3) PROGRAM abc_d2e(Input;Output)
- (4) Program 2310(Input output);
- (5) PROGRAM _xyz;

解题分析

(1) 的错误显而易见，程序首部没有用分号结尾，而且程序名中含有特殊字符“!”，不符合语法规范。

(2) 中没有程序名，这也不符合语法规范，在编译的时候 Pascal 有可能把括号及其包括的内容认作程序名，显然不是合法的标识符。

(3) 的程序名含有下划线“_”和数字，合乎语法要求，但程序参数之间用分号隔开是个错误。程序首部的结尾也没有加分号。

(4) 的程序标志不全是大写，这没有关系，因为 Pascal 并不区分大小写字母。程序名开头是数字，不是合法的标识符，不能作为合法程序名。程序参数也不该用空格隔开。

(5) 的程序名用下划线起始，是合法的标识符。程序参数缺省，用分号结束程序首部，这些都符合语法规范，因此是正确的程序首部书写方式。

【例 1.1.2】

一个标准 Pascal 程序必须包括。

- | | |
|--------------|-------------------|
| A. 程序首部、说明部分 | B. 程序首部、语句部分 |
| C. 说明部分、语句部分 | D. 程序首部、说明部分、语句部分 |

解题分析

虽然在 Turbo Pascal 中可以省略程序首部而只保留语句部分，但是标准 Pascal 语言明确规定标准 Pascal 程序至少要包含程序首部和语句部分。不选用常量或变量进行操作，也就不必要写出说明部分，因此，本题答案是 B。

即学即练

修改下面程序中的错误之处。

```
PROGRAM s;
CONST
Max=255;
```

```

VAR
    c:Char;
    u:Int;
BEGIN
    Write (Max);
    c:= 'A';
    u:=10;
    Writeln(u-Max);
END

```

小节答案

将“u:Int;” 这一句修改为“u:Integer;” 程序最后的 END 要加上圆点表示程序结束。

1.2 基本字符集、标识符与保留字

知识要点

(1) 基本字符集

Pascal 的基本字符集是 ASCII (American Standard Code FOR InFOR mation Interchange 美国信息交换标准代码) 的一个子集。Pascal 基本字符包括如下。

- ① 大小写英文字母——A 到 Z 以及 a 到 z
- ② 数字——0 到 9
- ③ 空格——ASCII 序号 32
- ④ 其他符号如

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

Pascal 不允许的符号, 有:

α β Ω $\div$ $\pm$ μ π

(2) 标识符

标识符用来标记程序、类型、常数、变量、过程、函数、指令等的名字, 必须由英文字母或下划线开头, 后面是英文字母、数字、下划线的任意组合。

应当注意区分合法与非法的标识符。下面给出一些合法的标识符:

A Ex2 turbo sit_in_the_sun _xyz r2d3_4

Turbo Pascal 不区分英文字母的大小写, 认为是同一个标识符。但是, 从易读性的角度考虑, SetTheName 或 Set_the_name 作为标识符明显要比 SETTHENAME 易读。因此在编程时要养成好的习惯, 增强代码的可读性。

一个标识符的长度由一行的长度（127 个字符）限制，但只有前 63 个字符有效。

标识符可以分为预定义的标准标识符和自定义标识符两类。

在标准 Pascal 中预定义了 40 个标准标识符：

3 个标准常量	False、True、Maxint
5 个标准类型	Integer、Real、Boolean、Char、Text
13 个标准过程	Read、Readln、Write、Writeln、New、Get、Put、 Pack、Unpack、Page、Reset、Rewrite、Dispose
17 个标准函数	Exp、Ln、Sqr、Sqrt、Pred、Succ、Abs、Sin、Cos、 Arctan、Round、Trunc、Odd、Ord、Eof、Eoln、Chr
2 个标准文件	Input、Output

在 Turbo Pascal 中扩充了很多预定义标识符，这里就不一一列举了。

至于自定义标识符，则是根据用户自己的需要来命名。除了应当符合前述的标识符命名规则外，通常还要注意如下问题：

① 自定义标识符不能与保留字同名，否则编译时出错。保留字将在下面介绍。

② 自定义标识符必须在程序的说明部分被合理声明之后，才能在程序的语句部分出现，否则编译时无法通过。

③ 不提倡将标准标识符用作自定义标识符。举个例子，如果在程序说明部分进行了常数定义：

```
CONST Exp=7;
```

则语句中出现的所有 Exp 都将被认为是常数 7，这样指数函数 Exp(x) 就无法应用了。

(3) 保留字

作为高级语言的一种，Pascal 给一些英文单词赋予了特定的含义，这样的特定符号称为“保留字”或“关键字”（Key Word）。标准 Pascal 规定了下面 35 个保留字：

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

Turbo Pascal 7.0 还扩充了 16 个保留字，分别是：

Asm	Constructor	Destructor	Exports	Implementation
Inherited	Inline	Interface	LIBRARY	OBJECT
SHLShr	STRING	UNIT	USES	XOR

在 Turbo Pascal 环境中编写程序代码时，遇到与保留字匹配的字符串，编辑器会自动将整个字符串标为白色，提醒用户这是系统的保留字。

典型题解**【例 1.2.1】**

下列选项中能够用作用户自定义标识符的是：

- A. TYPE B. π C. SET D. True

📖 解题分析

答案是 D。这里 A 和 C 都是保留字，不能用作标识符。B 不属于标准 Pascal 的标准字符集。True 虽然是预定义标准标识符，但用作自定义标识符也符合 Pascal 语法。

【例 1.2.2】

对下面的符号进行分类，那些是保留字，那些是标准标识符？

CONST	Real	AND	CASE	Text	Abs
DO	IN	Char	New	Input	THEN
END	True	Cos	Get	Ord	TO

📖 解题分析

本例考察对保留字和标准标识符的记忆。注意二者之间的区别。

保留字: CONST AND CASE DO IN New THEN END TO

标准标识符: Real Text Abs Char Input True Cos Get Ord

即学即练**【习题 1】**

下面选项中属于非法的用户自定义标识符的是：

- A. dir B. z12_ C. FILE D. list

【习题 2】

填空：标准 Pascal 中一共规定了____个保留字，定义了____个标准标识符。

小节答案

【习题 1】C FILE 是保留字。 【习题 2】35 40

1.3 代码与注释的书写规范

知识要点

作为一个成熟的程序员，在书写 Pascal 程序代码的时候，应当遵循一定的规范。这样写出的程序不仅结构清晰，意义明确，便于自己和别人看懂，而且容易查错，调试起来也非常方便。概括来说，规范的代码就是为了提高程序的可读性和可调试性。下面给出一些建议，具体的很多细节问题请读者在编程实践中慢慢体会。

(1) 代码要分行明确，具有清晰的程序结构

Turbo Pascal 编程环境中的文本编辑器能够自动将程序的起始列与上列对齐。为了保证各层次之间的整齐性，可以用 Tab 键划分代码的层次。每行最多写一条语句，不要好几条语句挤在同一行。必要的时候加入空格和括号，使代码的意义更加明确。

下面是一个编译通过但书写很乱的程序：

```
PROGRAM confuse;
VAR    a,b,c,d:Integer;
BEGIN a:=50; b:=a+1;   b:=1-a*-b DIV a-b;  c:=a+b*a-b;
      IF a<b THEN BEGIN if b<c THEN BEGIN d:=c; c:=a END ELSE BEGIN
d:=b; IF a<c THEN b:=a ELSE b:=c END end ELSE if b>c THEN BEGIN d:=a;
a:=c END ELSE BEGIN if a<c THEN BEGIN d:=c; c:=b END ELSE BEGIN d:=a;
a:=b END end; Writeln(a:3,b:3,c:3,d:3)END.
```

程序中，变量的赋值表达式容易引起歧义，虽然编译器得到的结果是确定的，但这样的表达式不符合习惯。在判断结构中，不通过仔细分析，读者很难知道这段程序执行的是怎样的操作，要达到什么目的。如果改写成下面的结构，则能够使程序清晰很多：

```
PROGRAM clear;
VAR
a,b,c,d:Integer;
BEGIN
a:=50;
b:=a+1;
b:=1-(a*(-b) DIV a)-b;
c:=a+(b*a)-b;
IF a<b THEN
    BEGIN
        IF b<c THEN
            BEGIN
                d:=c;  c:=a
            END
        ELSE
            BEGIN
```

```

                                d:=b;
                                IF a<c THEN b:=a
                                    ELSE b:=c
                                END
                            END
                        ELSE
                            IF b>c THEN
                                BEGIN
                                    d:=a; a:=c
                                END
                            ELSE
                                BEGIN
                                    IF a<c THEN
                                        BEGIN
                                            d:=c; c:=b
                                        END
                                    ELSE
                                        BEGIN
                                            d:=a; a:=b
                                        END
                                    END
                                END;
                            Writeln(a:3,b:3,c:3,d:3)
                        END.

```

相比 confuse 程序, clear 程序在加入了必要的圆括号、空格和换行之后, 表达式的意义一目了然, begin 和 end 的成双成对的对应关系也清晰明确。学完第 3 章之后, 我们可以并不费力地得出这个程序中判断结构的用意: 把算得的三个整数 a、b、c 进行比较, 令 d 等于最大值, 而最大值对应的那个变量则改为最小值。就本例来说, a、b、c 分别为 50、1、99, 这样程序运行之后, a、b、c、d 四个变量值分别变为 50、1、1、99。

(2) 标识符的取名尽可能意义明确, 可读性好

这一点已经讲过, 这里不再赘述。

(3) 表达式的意义应当明显

不同优先级(参见 1.5 节)的运算符出现在同一表达式的时候, 应该注意意义的明确性, 易于理清结构。confuse 程序中 $b:=1-a*-b \text{ DIV } a-b$; 这一句意义就非常不明显, 容易让人看成是 $(1-a)*(-b) \text{ DIV } (a-b)$ 。在 clear 中改为 $b:=1-(a*(-b) \text{ DIV } a)-b$; 的定义方法, 则 b 的值一目了然。

(4) 要有画龙点睛的注释, 格式规范

注释的方法: 以花括号对 {} 或 (*...*) 括住的文字都是注释。Pascal 在编译时不理睬其内容。一个程序注释内容的多少对运行速度没有影响。

可以用两种注释方法形成注释嵌套, 如下例:

```
(*
  IF a<b THEN a:=b;    {Trying}
*)
```

以上的内容都是注释。注意只用一种注释方法无法形成嵌套，如：

```
{ Writeln(a:4,b:4); {prnTo ut} a:=b; }
```

这里黑体的部分不属于注释内容，程序将编译出错。因为第一个}与最左边的{号对应，只注释了这中间的内容。

注释除去说明作用以外，还常用在调试程序中，能够保留住没有把握的代码。当要修改一段编译报错的代码时，把原来的一部分代码加入注释中，这样没有删除代码，编译时这段代码却不起作用。当调试通过后，再将能够删去的地方真正删除。这是程序员常用的调试方法之一。

典型题解

【例 1.3.1】

下面的书写符合标识符语法规范的是：

- A. PROGRAM B. CH₂-COOH C. 2Ca₂Co D. π^2

📖 解题分析

答案是 A。根据标识符的定义，可以知道 PROGRAM 是一个合法的标识符（已用作系统保留字）。B 中有-号，不是合法字符。C 以数字开头，不合规定。D 用到的两个字符都不属于 Pascal 基本字符集。

【例 1.3.2】

下面的程序段，编译时会出现什么结果？

- A.
 {
 注释 1;
 *}
 END.
- B. { 注释 1;
 s:=1;
 { 注释 2; }
- C. { 注释 1; }
 s:=1;
 { 注释 2;
- D. {
 t:=1; {t 初始为 1}
 t:=t*a;
 }

解题分析

A 中 { 与 *) 不能匹配, 因此编译器将继续忽略 *) 及其后面的所有字符。这样, 程序尾的 end. 也被忽略, 编译将产生错误。

B 中的注释 1 忘记加后括号, 但是注释 2 的后括号会跟注释 1 的前括号进行嵌套, 这样, 中间的一句语句 s:=1; 将被忽略。编译时可能没有错, 但是运行结果可能产生意外。

C 的注释 2 忘记加后括号, 这样后面的所有代码将被注释掉, 包括结尾的 end. 错误的性质同 (A)。

D 的注释嵌套用了两组 {}, 在编译时, 外层注释的 { 将与内层注释的 } 匹配, 后面的 t:=t*a; } 将有效, 会产生没有与 “)” 匹配的 “{” 的语法错误。

这道题归纳了加入注释时常见错误的产生原因, 读者在编程时应加以注意。

即学即练

【习题 1】

将下面程序的结构整理清晰:

```
PROGRAM con(Output);
VAR
  s1,s2,temp:Real;
BEGIN
  s1:=-1.0; s2:=1.0;
  s1:=1.0*-s2*s1/-1.0; s2:=1.0/-s1*s2*-1.0;
  IF s1<s2 THEN BEGIN temp:=s1;s1:=s2;s2:=temp END ELSE BEGIN
s1:=1.0*-s2*s1/-1.0; s2:=1.0/-s1*s2*-1.0 END;
  Writeln(s1:6:2,s2:6:2);
END.
```

小节答案

【习题 1】整理后的程序为:

```
PROGRAM con(Output);
VAR
  s1,s2,temp:Real;
BEGIN
  s1:=-1.0; s2:=1.0;
  s1:=1.0*(-s2)*s1/(-1.0); s2:=1.0/(-s1)*s2*(-1.0);
  IF s1<s2 THEN
    BEGIN
      temp:=s1; s1:=s2; s2:=temp
    END
END
```



```

ELSE
    BEGIN
        s1:=1.0*(-s2)*s1/(-1.0);
        s2:=1.0/(-s1)*s2*(-1.0)
    END;
Writeln(s1:6:2,s2:6:2);
END.

```

1.4 数据类型、常量定义与变量说明

知识要点

(1) 掌握 Pascal 的标准数据类型

在程序的说明部分进行变量说明的时候，不但要指明作为变量名的标识符，而且要指出这个变量的数据类型。例如

```
VAR    i:Integer;
```

这里 i 是变量名，Integer（整型）就是变量 i 的数据类型。

Pascal 中有两类数据类型：预定义数据类型和自定义数据类型。其中预定义数据类型内容非常丰富，又可分为若干种，下面分别介绍。

简单数据类型（4 种）

① 整型（Integer）

这是 Pascal 中整数的数据类型，范围-32768~32767。标准标识符 Maxint 表示的就是整型值 32767。

整型数据的运算符包括算术运算符+、-、*、DIV（整除），MOD（取余）以及关系运算符=、<>、<、>等。具体的运算规则将在第2章2.1节介绍。

Turbo Pascal 还支持 4 种预定义整型类型，它们的区别在于表示数的范围不同。列如下：

Shortint	短整型	-128~127
Byte	字节型	0~255
Word	字型	0~65535
Longint	长整型	-2147483648~2147483647

Turbo Pascal 也支持十六进制的整型数，只需在数字之前加符号\$。\$20 相当于十进制数 32，\$AB 相当于十进制数 171。

② 实型（Real）

这是 Pascal 中实数的数据类型，可表示最大实数范围为 1.7×10^{38} ，最小绝对值为 2.9×10^{-39} ，低于这个绝对值按机器零处理，称为数值下溢。绝对值超过上界 1.7×10^{38} 则产生数值上溢(overflow)，编程时要注意数值的下溢与上溢。

实型数据的运算符包括算术运算符+、-、*、/ 和关系运算符。