

● 高等教育自学考试教材 ● 高等教育国家文凭考试教材

PASCAL 程序设计

● 丁文魁 编著
● 陈士龙

12
4/1

北京大学出版社

M

C

A

S

M

C

A

S

M

TP312
DWK/1

高等教育自学考试教材

PASCAL 程序设计

丁文魁 陈士龙 编著

北京大学出版社
北 京

JS371/16

内 容 简 介

本书根据北京市高等教育自学考试大纲编写,作为自学考试和文凭考试的指定教材。全书系统地介绍了标准化PASCAL语言的语法和语义,通过大量的实例讲述构造程序的方法,详细地表述了自顶向下、逐步求精的程序设计方法。每章都有结合本章内容的综合练习和分析,并附有供学生练习的习题和上机题。本书概念清楚,逻辑性强,便于读者理解和掌握,并有利于培养程序设计的能力。

本书作为计算机软件专业和计算机应用专业的教材,也可作为自学者和计算机工作人员的参考书。

图书在版编目(CIP)数据

PASCAL 程序设计/丁文魁,陈士龙编著. -北京:北京大学出版社,1996.1

高等教育自学考试教材

ISBN 7-301-02968-3

I. P… II. ①丁…②陈… III. PASCAL 语言-程序设计 IV. TP312PS

书 名: PASCAL 程序设计

著作责任者: 丁文魁 陈士龙

责任编辑: 杨锡林

标准书号: ISBN 7-301-02968-3/TP·276

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 2502015 发行部 2559712 编辑部 2502032

排 印 者: 北京经纬印刷厂印刷

发 行 者: 北京大学出版社

经 销 者: 新华书店

787×1092 毫米 16 开本 18.875 印张 470 千字

1996 年 1 月第一版 1996 年 1 月第一次印刷

定 价: 22.00 元

前 言

受北京市高等教育自学考试委员会的委托,我们根据考试大纲的要求,编写了《PASCAL 程序设计》这本书,作为自学考试和文凭考试的指定教材,推荐给广大的自学朋友。

计算机软件专业和计算机应用专业的学生学习这门课程后,总体上应达到三方面的要求:
(1) 准确掌握标准 PASCAL 语言的语法和语义;(2) 读懂用 PASCAL 语言编写的程序;
(3) 有一定的编程能力,具有良好的程序设计风格。

本书的开始,我们告诉读者,什么是程序,什么是程序设计。在以后的各章中,通过大量的例题,展现给读者的是构造程序的方法:对于给定的问题,经总体分析,首先给出粗的算法,随着对题解的逐步求精,抽象出数据的类型定义,采用伪码表示算法(这里的伪码,由 PASCAL 的控制结构和自然语言组成),然后获得算法的具体表述——程序。这就是自顶向下、逐步求精的程序设计方法。第六章过程和函数讲解时,编写功能子程序的例子较多,旨在培养学生设计模块程序的能力。

程序设计是一种创造性的脑力劳动。编程能力的培养,除了像学习算法那样读懂书中构造程序的例子外,就是自己动手编写习题中的程序,并且上机调试运行。我们希望读者从第二章起,就应该做足够的编程练习,并在计算机系统上验证它的正确性。除此之外,没有什么“灵丹妙药”。

目前,微机上广泛使用 turbo pascal 系统编译和运行 PASCAL 程序。因此,凡是标准 PASCAL 和 turbo pascal 不同的地方,我们以注释的形式给出它的描述形式,并在附录 D 中给出 turbo pascal 的上机简要说明。

为了帮助理解学过的知识,每章都附有一节综合练习,我们给出了答案,并且做了简单分析。这对读者掌握这一章的知识,可能会起到“抛砖引玉”的作用。

本书根据自学考试大纲编写,抓住了程序设计这个基本问题,适合于除计算机系软件专业外的所有读者使用。但是,缺少递归回溯算法的例题和习题,这方面的知识,请读者查阅有关教材。

书中对于超过大专水平的内容,都注上了星号(*)。

由于条件限制,可能有许多不足之处,恳请广大读者批评指正。

编 者

1995 年元月于北京大学

目 录

第一章 计算机系统和程序设计介绍.....	(1)
§ 1.1 计算机系统	(1)
§ 1.2 算法	(3)
§ 1.3 PASCAL 程序介绍	(5)
习题一	(7)
第二章 数据、表达式和赋值.....	(8)
§ 2.1 字符集和符号	(8)
§ 2.2 标准数据类型	(10)
§ 2.3 常量和变量	(12)
§ 2.4 标准函数	(14)
§ 2.5 表达式	(17)
§ 2.6 赋值语句	(21)
§ 2.7 输入输出语句	(22)
§ 2.8 顺序程序设计	(25)
§ 2.9 综合练习	(28)
习题二	(31)
第三章 选择结构程序设计	(35)
§ 3.1 IF 语句	(35)
§ 3.2 复合语句、空语句	(38)
§ 3.3 IF 语句的嵌套	(40)
§ 3.4 CASE 语句	(45)
§ 3.5 选择结构程序设计	(47)
§ 3.6 综合练习	(49)
习题三	(54)
第四章 循环结构程序设计	(58)
§ 4.1 WHILE 语句	(58)
§ 4.2 REPEAT 语句	(61)
§ 4.3 FOR 语句	(64)
§ 4.4 三种循环语句的区别与关联	(67)
§ 4.5 循环的嵌套	(70)
§ 4.6 循环结构程序设计	(73)
§ 4.7 综合练习	(82)

习题四	(90)
第五章 枚举类型、子界类型和集合类型	(93)
§ 5.1 枚举类型	(93)
§ 5.2 子界类型	(98)
§ 5.3 集合类型	(101)
§ 5.4 类型之间的关系	(109)
§ 5.5 综合练习	(113)
习题五	(115)
第六章 过程与函数	(118)
§ 6.1 过程	(118)
§ 6.2 函数	(124)
§ 6.3 实在参数和形式参数的结合	(127)
§ 6.4 标识符的作用域	(130)
§ 6.5 递归	(132)
§ 6.6 过程和函数应用实例	(136)
§ 6.7 综合练习	(142)
习题六	(147)
第七章 数组类型	(151)
§ 7.1 一维数组	(151)
§ 7.2 二维数组	(155)
§ 7.3 多维数组	(160)
§ 7.4 字符数组和字符串	(161)
§ 7.5 典型实例程序设计	(166)
§ 7.6 综合练习	(179)
习题七	(183)
第八章 记录类型	(187)
§ 8.1 引言	(187)
§ 8.2 记录类型定义和记录分量访问	(187)
§ 8.3 记录的变体	(196)
§ 8.4 记录类型在程序设计中应用	(200)
§ 8.5 综合练习	(206)
习题八	(209)
第九章 动态数据结构	(214)
§ 9.1 引言	(214)
§ 9.2 指针类型定义	(215)
§ 9.3 指针变量和动态存储单元访问	(217)
§ 9.4 链表结构	(224)
§ 9.5 二叉树	(235)
§ 9.6 链表在程序设计中的应用	(241)

§ 9.7 综合练习	(247)
习题九	(250)
第十章 文件类型.....	(254)
§ 10.1 TEXT 类型文件	(254)
§ 10.2 FILE 类型文件	(259)
§ 10.3 FILE 类型文件与 TEXT 类型文件的比较	(267)
§ 10.4 文件在程序设计中的应用	(268)
§ 10.5 综合练习	(275)
习题十	(280)
附录	
附录 A PASCAL 的词汇表	(282)
附录 B PASCAL 语法图	(284)
附录 C PASCAL 的 GOTO 语句	(288)
附录 D turbo pascal 的基本操作	(289)
附录 E MS-DOS 的基本命令	(292)
附录 F ASCII 码字符表	(293)

第一章 计算机系统和程序设计介绍

计算机和程序设计的基本知识,包括计算机系统结构、算法和程序语言的概念,对于学习 PASCAL 程序设计来说是必要的。

本章是为初学者准备的,且以读者能理解和掌握 PASCAL 程序设计作为基本要求。

§ 1.1 计算机系统

1.1.1 计算机的硬件构成

计算机的基本组成部件有运算器、控制器、存储器、输入装置和输出装置五大部分。

1. 运算器

运算器主要包括:一个能对数据进行算术和逻辑运算的部件,通常称为运算部件 ALU (Arithmetic Logic Unit);提供一个操作数和存放操作结果的累加器 A (Accumulator);若干个存放中间结果的寄存器;指令计数器。

2. 控制器

用来命令计算部件做什么操作的一串信息称为指令。控制器的作用是实现对指令的控制,解释指令的操作码和地址码,并根据译码结果将适当的控制信号送到运算器、控制器和机器其他部分,统一指挥整个计算机的工作。

运算器和控制器合称为中央处理部件 CPU (Central Processing Unit)。

3. 存储器

存储器除用来存放数据外,还用来存放表达运算规则及步骤的程序。存储器分为主存储器(内存储器)和辅助存储器(外存储器)。主存储器存放经常使用的信息,主存储器存放不下的数据,或需长期保留的数据存放到辅助存储器上。如同记事一样,大脑起主存储器作用,笔记本起辅助存储器作用。

主存储器存取数据所用的时间短,但容量有限。辅助存储器的容量大,但存取数据所用的时间长,且被执行的指令和被加工的数据一定得存放在主存储器中。常用的辅助存储器有软磁盘、硬磁盘和光盘。它们需要专门的驱动装置和接口卡配合使用。

主存储器中按一定规律划分成许多单元,每个单元称一个字节(Byte),可以存放一个八位二进制的代码信息。每个单元都有一个编号,称为地址码,也用二进制表示。这如同一幢大楼分为许多房间,每个房间都标有房间号一样。一个一个的信息,根据地址码被存入(也称写入)或取出(也称读出)相应单元。注意,存储单元的“地址”和“内容”就像“房间号”和“住在房间的人”一样,是完全不同的两回事。

4. 输入装置

输入装置的作用是将人们需要处理的信息送到计算机的存储器,常用的输入装置有键盘,图形输入板,数字阅读机,视频摄象机等。

5. 输出装置

输出装置的作用是把计算机处理的结果，变为人或其他机器设备能接收和识别的信息形式，如文字、数字、图形、声音、电压等。常用的输出装置有打印机、字符显示器、绘图仪和图形显示器等。

计算机各部件的工作与配合如图 1.1 所示。输入装置将人们所熟悉的信息形式变换为计算机内部所能接收和识别的二进制形式。主存储器用来存储记忆这些信息及运算结果。就微机来说，尽管目前主存储器的容量已达若干 MB（兆字节），但仍需辅助存储器来存储大量信息。运算器的功能是对被处理的信息进行算术逻辑运算，控制器将对每条指令的操作和步骤进行综合，产生实现整个指令系统所需要的全部控制信号，指挥计算机各部件有条不紊地工作。输出装置则将计算机处理的结果变换为人或其他机器所能接收和识别的信息形式。

一台计算机的指令集和它内部的数据表示形式给出了它的功能和使用界面。

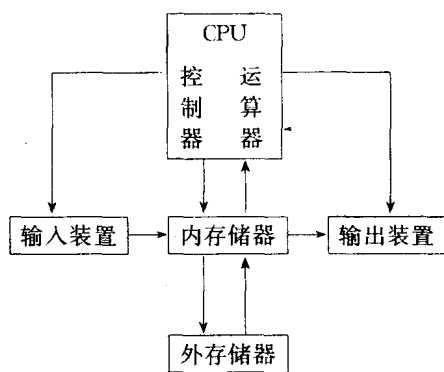


图 1.1 计算机各部件的工作与配合

1.1.2 计算机部件中的数据形式

在数字系统中，数值的大小往往用一串文字或数字表示。人们最熟悉的十进制数就是一种表示方法，例如 185.75。

还可以用二进制的字符“0”和“1”将 185.75 表示为：

10111001.11

以上的数字符号称为数码。从数码序列 185.75 和 10111001.11 中可以看出，数码处于不同的位置所代表的数值是不同的，例如：

$$185.75 = 1 \times 10^2 + 8 \times 10^1 + 5 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

$$10111001.11 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

十进制数 185.75 可以表示成各位的数码与该位 10 的权值乘积的序列。二进制数 10111001.11 可以表示成各位的数码与该位 2 的权值的乘积的序列。前者称为十进制计数法，后者称为二进制计数法。在计算机部件中，数采用二进制形式表示。

机器指令和各种符号也采用二进制形式，例如：

1111 1110 1100 0010

是 PC 机上的指令，它使主存储器中某个特定的存储单元的值加 1。每个字符用 8 位二进

制数表示，占一个字节。例如：大写英文字母 S 用 01010011，而 “=” 用 00111101 表示。

为便于交流，国际上普遍采用一些标准化的代码以便识别和统一使用。例如，ASCII 码 (American Standard Code for Information Interchange, 美国国家信息交换标准字符码) (见附录 F)，EBCDIC 码 (Extended Binary Coded Decimal Interchange Code, 扩展 BCD 交换码) 等。

1.1.3 操作系统

上面我们介绍的计算机部件，像 CPU、存储器、输入输出设备，它们和传统意义上的仪器设备相似，这些称为计算机硬件或硬件设备，总称为裸机。

仅有硬件的计算机使用起来很不方便。操作系统是为了提高计算机的利用效率，方便用户使用以及提高计算机的响应时间而配备的一种软件。操作系统是计算机软硬件资源的管理者。

操作系统统一管理计算机资源，合理组织计算机的工作流程，协调系统各部分之间，系统与使用者之间以及使用者与使用者之间的关系，以利于发挥效率及方便使用。操作系统是直接运行在裸机之上的，我们可以把它看成是对计算机硬件的扩充，也可以看成是用户和计算机硬件的接口。其他的计算机软件都是在操作系统的基础上开发出来的。一个计算机系统是由计算机硬件、操作系统和应用程序组成的 (见图 1.2)。

目前微机上常用的操作系统版本是 DOS (Disk Operating System)，其操作管理的主要文件存放在磁盘上，故称磁盘操作系统。DOS 提供了丰富的内部命令、外部命令以及一系列系统功能，为用户提供了良好的运行环境和高质量的编程工具 (见附录 E)。

操作系统把数据和程序组成文件，每个文件用文件名标识。文件名的一般形式是：

[盘号] [路径] 文件名 [. 扩展名]

其中方括号中的内容可有可无。例如：

C: terel. prg	说明是 C 盘根目录上的 dBASE 程序文件 terel
blam. dbf	当前盘上的数据库文件 blam
text1	当前盘上的文件 text1

文件名是由英文字母开始，后跟随英文字母或数字，最多由 8 个字符组成的字符串。扩展名表示文件的类型，由 3 个字母数字组成。关于盘号，在 DOS 中规定软盘驱动器号为 A 或 B。硬盘为 C 或 D。省略盘号时意指当前盘号，当前盘号表示和提示符 “>” 前的盘号一致。

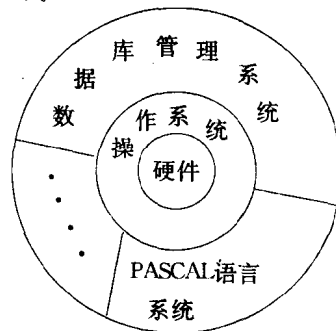


图1.2 一个计算机系统

§ 1.2 算 法

一般地说，为解决一个问题而采取的方法和步骤称为算法，或者说，算法是解题方法的精确描述。解决一个问题的过程就是实现一个算法的过程。请看下面的例子。

例 1.1: 有两个杯子 A 和 B，里面分别盛酒和水，请将杯子 A 和 B 中的液体互换。

根据常识,完成杯子 A 和 B 中液体的互换需要一个杯子 C,算法 1 述描了解决这个问题的方法。

算法 1:

第一步:将 A 杯中的酒倒入 C 杯中;

第二步:将 B 杯中的水倒入 A 杯中;

第三步:将 C 杯中的酒倒入 B 杯中。

例 1.2:求两个正整数 m 和 n 的最大公约数。求解这个问题的方法由算法 2a 给出。

算法 2a:

第一步:以 n 除 m ,并令 r 为所得余数;

第二步:若 $r=0$,则 n 为最大公约数;

当 $r \neq 0$ 时,重复执行下面步骤,直到 $r=0$:

2-1 将 n 的值放在 m 中,将 r 的值放在 n 中。

2-2 以 n 除 m ,并令 r 为当前所得余数。

在我们学过 PASCAL 语言的控制结构后,可以采用它来描述算法。例如,算法 2a 可描述如下:

算法 2b:

第一步:以 n 除 m ,并令 r 为所得余数;

第二步:WHILE $r \neq 0$ DO

BEGIN

2-1 将 n 的值放在 m 中,将 r 的值放在 n 中。

2-2 以 n 除 m ,并令 r 为所得余数

END

从上面的例子可以看出,算法是在有限步骤内求解某一问题所使用的有精确定义的一系列操作规则。每条规则必须是确定的与能行的,不能有歧义性。算法有一个初始步骤,然后每个步骤只能有一个后继步骤,从初始步骤开始到步骤终止组成一个步骤序列。序列终止表示问题得到解答或指出问题没有解答。

另外要指出的是,每个算法应有若干个输入和输出。在算法 2a 中,输入是两个正整数,且存放于 m 和 n 中。输出是 n 中的值,它就是所求的 m 和 n 的最大公约数。

当我们遇到一个算法时,不要期望像读一篇小说那样来读一个算法。如果抱着这样的期望,那就会很难理解和掌握本书的内容。经验告诉我们,学习一个算法直到彻底掌握它的最好方法是反复进行试验。当我们遇到一个算法时,应该立即拿起笔和纸,找出这个算法的一个特例,一步步地实现这个算法。为了能够理解一个给定的算法,这是一个简单且省力的方法,其他途径都是不成功的。

对于算法 2b,不妨取 $m=544$, $n=119$ 。根据算法,执行序列如下:

(1) 用 n 除 m ,把余数放入 r , $m=544$, $n=119$, $r=68$

(2) 在 r 不等于 0 的情况下,重复执行如下步骤:

A $\begin{cases} 2-1 & m=119, n=68 \\ 2-2 & r=51 \end{cases}$

B $\begin{cases} 2-1 & m=68, n=51 \\ 2-2 & r=17 \end{cases}$

$$C \begin{cases} 2-1 \ m=51, \ n=17 \\ 2-2 \ r=0 \end{cases}$$

直到 $r=0$ ，算法的执行终止， n 中的17即为544和119的最大公约数。这个算法比较简单，对于复杂的算法，要经过反复的实验与推敲，才能掌握它的基本要点。

对于给定问题，算法的表示是静止的，执行算法才能给出此问题的解答或指出此问题没有解答，这和“菜谱”与“做菜”一样。算法的书写者书写算法的目的是告诉算法的执行者，执行算法中所给出的计算规则序列，就能给出问题的解，也就是说“如何做”。

现在我们要用计算机作为工具，解决生产中和科学研究中的问题。我们成为算法的设计者，计算机成为算法的执行者。我们若用描述算法1和算法2b使用的语言去描述算法，告诉计算机如何去做显然是行不通的，因为计算机不懂得它。我们用计算机能懂的语言描述算法，算法的这种书面描述形式称为程序。从给定问题到编写出程序的过程称为程序设计。所使用的计算机能懂的语言称为程序设计语言。

PASCAL 语言就是一种高级程序设计语言，它有丰富的数据类型并支持结构化程序设计。一个 PASCAL 语言程序，就是用 PASCAL 语言描述的一个算法，这个程序在支持 PASCAL 语言的计算机系统上运行，能求出问题的解答。

粗略地说，程序设计大致上分三步：第一步分析所求解的问题；第二步用描述算法2b使用的语言写出求解给定问题的算法，算法2b使用的语言有时称为“伪码”；第三步，把第二步得到的算法用 PASCAL 程序设计语言描述出来。

§ 1.3 PASCAL 程序介绍

下面，首先给出用 PASCAL 描述算法1和算法2b的程序，然后介绍程序注释和程序结构，最后介绍如何用计算机系统的 PASCAL 语言系统运行一个 PASCAL 程序。

1.3.1 PASCAL 程序例子

例1.3：交换两个变量 a 和 b 的值。

```
PROGRAM swap(input,output);
VAR
    a,b,c:integer;
BEGIN
    read(a,b);           {输入两个整数分别赋予 a 和 b}
    c:=a;                 {把 a 的值赋予 c}
    a:=b;                 {把 b 的值赋予 a}
    b:=c;                 {把 c 的值赋予 b}
    write(a,b)            {显示 a 和 b 的值}
END.
```

例1.4：描述算法2b 的程序。

```
PROGRAM gcd(input,output);
VAR
```

```

        m,n,r:integer;
BEGIN
    read(m,n);
    r:=m MOD n;
    WHILE r<>0 DO
        BEGIN
            m:=n;
            n:=r;
            r:=m MOD n
        END;
    writeln('The gcd is',n)
END.

```

1.3.2 程序注释

程序中用花括号括起来的部分称为程序注释。程序注释仅帮人们读懂程序，而对程序执行不产生任何影响。程序中的注释，以适量为宜。在例1.3中对每句都加上了注释，为的是让我们读懂它，因我们目前还没有具备 PASCAL 语言的知识。另外，例1.3中用汉语写注释，只有接收汉字的 PASCAL 语言系统才能接收它。若你用的 PASCAL 语言系统不能接收汉字，那只好用英文写注释。

1.3.3 PASCAL 程序结构

程序的第一行称作程序首部，从第二行到最后一行的 END 称作分程序。因此，一个 PASCAL 程序结构如下：

程序首部；分程序。

程序首部由 PROGRAM 开始，接着是程序名，在例1.4的程序中，程序名是 gcd，跟在程序名后面的是用圆括号括起来的程序参数，它是程序使用的文件名表。其中的 input 和 output 是语言系统提供的两个标准文件，分别对应键盘的输入和显示器的输出。因此，程序首部的结构如下：

PROGRAM 程序名（程序参数）；

分程序由两部分组成，前面是说明部分，后面是语句部分。说明部分说明程序中使用的数据，在例1.4中说明部分是

VAR

m, n, r: integer;

语句部分由 BEGIN 开始到 END 结束，其中每个语句以“;”分开。语句序列给出对说明部分说明的数据施加的运算。第一个语句 read(m,n)，读入两个整数分别赋给 m 和 n。最后一个语句 writeln('The gcd is',n)，在显示器上显示：

The gcd is n 的值

中间语句对 m, n, r 的操作如同算法2b 中的描述。

从上面两个程序可以看出，一个 PASCAL 程序由对数据的描述和对数据操作的描述两

部分组成。

1.3.4 从编辑到运行一个程序

目前，一个 PASCAL 语言系统，一般提供文件管理 (file)、编辑 (edit)、编译 (compiler) 和调试 (debug) 等功能。对于 turbo pascal 来说，它们作为选择项显示在窗口上 (操作命令见附录 D)。

程序员应首先选择 file，通过编辑命令输入编制的程序，并以文件形式存放起来，这个文件称为源程序文件。计算机部件并不能直接执行 PASCAL 语言程序，必须经编译程序把源程序翻译成目标程序，这种翻译类似于把一篇英文翻译成中文。目标程序是计算机部件可执行的指令序列，称为机器语言程序。

编译程序首先识别源程序是什么，它是否符合语法，它表达什么意思，然后才能把它翻译成目标程序。编译程序做上述工作的依据是 PASCAL 语言的语法和语义。因此，程序员编制程序必须严格遵守 PASCAL 语言的语法和语义，稍有偏差，编译程序就能查出错误，发出错误信息。这和我们给朋友写信不同，信中有个别语法错误或用词不当，对方根据上下文能理解信中所表达的意思，而编译程序还没有这种能力。

源程序在编译过程中出现错误后，程序员根据错误提示信息，找出源程序中的错误，再次选择编辑 (edit) 项修改源程序，重复这个过程，直到通过编译而生成目标程序。最后，选择运行 (run) 项执行目标程序，给出计算结果 (见图 1.3)。

编译程序仅能查出源程序中存在的语法错误和部分语义错误。语言系统在运行过程中还能查出一些语义错误。至于源程序中存在的逻辑错误，特别是算法本身存在的错误问题，语言系统是不知道的。计算机一丝不苟和有序地执行每条指令，它不能发挥任何独创性。这和程序员操作机器人一样，如果程序员说坐下，机器人就会坐下来，而不管有没有椅子。因此，如果程序员要机器人坐在椅子上，那就必须说“到椅子那儿去”，“把椅子搬过来”，“坐在椅子上”。如果只说“坐在椅子上”而又没有椅子，那将会发生意外事故。因此，程序员在程序设计时必须预测到一切可能发生的事情。计算机只做让它做的事情，不多也不少。如果它不按你的想法去做，那你不应该责备计算机，而应该责备自己。

在结束这一章时，我们强调本章介绍的算法和程序设计的概念，特别是程序设计中应该注意的问题。开始时，读者对这些知识可能理解不深，在以后的学习中，再回过头来阅读这些内容，将会有所裨益。

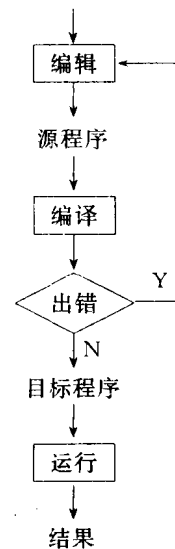


图 1.3 从编辑到运行一个程序

习 题 一

1. 一个算法应该满足什么条件，试举出你所知道的一个算法。
2. 陈述一个 PASCAL 程序的程序结构。
3. 陈述编写程序的步骤。

第二章 数据、表达式和赋值

上一章概述了 PASCAL 的程序结构,并指出:一个 PASCAL 程序是由对数据和对数据操作的描述两部分组成的。研制程序的主要目的是让计算机处理数据,程序中的计算主要是在表达式和赋值语句中进行的。现在就来讨论数据、表达式和赋值,并用这些基础知识进行程序设计。

为此,本章先介绍 PASCAL 的字符集和符号、标准数据类型、常量和变量、标准函数,然后讨论表达式、赋值语句和输入输出语句,最后举例说明如何进行顺序程序设计。

§ 2.1 字符集和符号

先看一个例子:已知任意三角形的两边 a 、 b 和夹角 c ,用公式 $\frac{1}{2}absinc$ 求它的面积。程序是:

```
PROGRAM example(input,output);
CONST
    pi=3.1416;
VAR
    a,b,c,area :real;
BEGIN
    read(a,b,c);
    area :=0.5 * a * b * sin(pi/180 * c);
    write('area=',area)
END.
```

这个程序执行时,将 $a=3$, $b=4$, $c=90^\circ$ 输入之后,输出结果 $area=6.0$ 。

从书写规则上看,这个程序和一篇英文文章相似,它是由基本字符组成的字符序列。字符按一定的构词法组成单词(又称符号),单词按一定的句法构成句子。一个程序可看成若干句子组成的句子序列。

求三角形面积的程序,就是一个符号序列,这些符号序列可分成四类:PROGRAM、CONST、VAR、BEGIN、END,是保留字;()、:=、*、/等是特殊符号;input、output、real、sin、write 等是标准标识符;example、pi、a、b、c 和 area 是普通标识符;最后一类是常数,像 3.1416、0.5、180 和 'area=' ,分别是实型、整型和字符串。

2.1.1 字符集

PASCAL 语言有一套字符集,这些字符按一定词法规则构成语言单词。字符相当于汉语中的字。

字母: A 到 Z, a 到 z。

数字: 0 到 9。

特殊符号: + - * / = ^ < > () [] { } , . : ; \。

注: 数学中的, π , α , β , ..., 不是字符集中的字符, 不能直接使用。turbo pascal 中还使用了: \$, @, #, _ (下划线)。

2.1.2 符号

符号由字符构成, 在语言中有独立意义。符号相当于汉语中的单词。

1. 保留字

保留字在语言中已规定了固定的意义, 程序员不能再重新定义它。这些保留字是:

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	

在本书中, 保留字采用大写, 以便区别于标识符。对于字母来说, 大小写含义一样。

2. 标准标识符

标准标识符是语言中预定义标识符, 程序员不用定义即可使用, 编译系统自动识别它。主要的标准标识符有:

标准常量:	false	true	maxint		
标准类型:	integer	boolean	real	char	text
标准文件:	input	output			
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:	get	new	pack	page	put
	read	readln	reset	rewrite	unpack
	write	writeln			

注: 考虑到程序的可读性和可移植性, 程序员不宜于将标准标识符定义为新的意义。另外, turbo pascal 系统增加了程序中文件变量和磁盘文件的连接与关闭的标准过程:

assign(文件变量, '磁盘文件描述')

close(文件变量)

3. 普通标识符

普通标识符是程序员自己定义的标识符, 用于给常量、变量、函数、过程、类型和文件等命名。普通标识符必须先定义后使用。构成规则是: 以字母打头的字母数字串。命名一个切合题意的标识符, 使程序易读, 易理解。例如:

CONST

```

pi=3.1416;
blank=' ';
TYPE
    color=(red,yellow,blue,green);
VAR
    area:real;
    number,divisor:integer;

```

注: turbo pascal 系统允许程序员定义的标识符中带下划线。如 `v_x`, `v_y`, `end_of_file` 等。

§ 2.2 标准数据类型

程序中的数据可划分为各种类型。比如, 用字符串表示一个人的姓名, 用整数表示他的年龄, 用实数表示他的工资和奖金, 用布尔数表示他是否结婚。整型数据对我们来说是比较熟悉的, 当提到整型数据时, 我们会立刻想到它实际上是数学中的整数, 同时也想到对整数所能施加的运算, 像加、减、乘、除等。因此, 每种数据类型是由两部分组成的, 一个是它的取值, 另一个是在这些值上施加的运算。PASCAL 语言提供了十分丰富的数据类型:

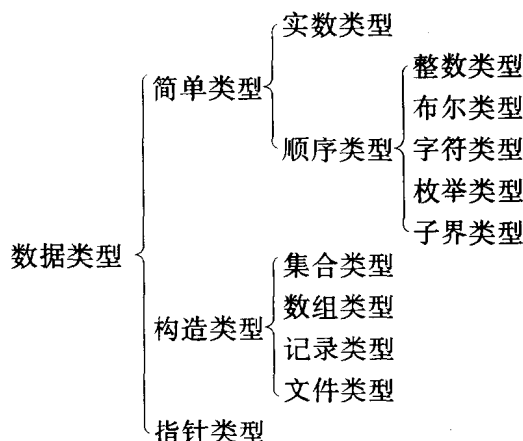


图 2.1 PASCAL 的数据类型

本章先讨论四种标准类型: `real` (实型), `integer` (整型), `boolean` (布尔型), `char` (字符型)。

2.2.1 `integer` (整型)

整数类型标识符是 `integer`, 简称整型。整数有正整数、整数零、负整数。整数常数的书写与数学上的整数十分相似, 值得注意的是不能带小数点。因此, 整数的结构是:

数符 数字串

例如, 618, 0, -618, +618 都是正确的。但是, 0.0, 618.0, 618. 是错误的。

一台计算机只能表示数学中整数的一个有限子集, 取值范围是: