

高校非计算机专业计算机等级考试教材丛书

PASCAL语言

程序设计基础

招兆铨
蒋家福 编
张增琪



复旦大学出版社

高校非计算机专业计算机等级考试教材丛书

PASCAL 语言程序设计基础

招兆铨 张增琪 蒋家福 编

复旦大学出版社

(沪)新登字 202 号

PASCAL 语言程序

招兆铿 张增琪 蒋家福 编

复旦大学出版社出版

(上海国权路 579 号)

新华书店上海发行所发行 常熟文化印刷厂印刷

开本 850×1168 1/32 印张 12 插页 1 字数 333,000

1992 年 8 月第 1 版 1992 年 8 月第 1 次印刷

印数 1—8,000

SIN 7-309-06860-X/T·52

定价: 6.10 元

序 言

近年来, 我国的计算机应用得到了迅速的发展。计算机已进入到各行各业, 产生了巨大的社会和经济效益, 已成为实现四个现代化不可缺少的工具。在如今, 掌握一些计算机知识有助于事业的成功已成为人们的共识。大批科技人员和管理人员都把学习计算机技术作为知识更新的关键环节。继续教育是重要的, 但对广大在校的非计算机专业的大学生开展计算机基础教育更是十分必要: 把大学生培养成为既有自己的专业知识又有计算机应用知识的全面人才, 对进一步提高整个社会的计算机应用水平有很大的现实意义。

为了进一步促进高校非计算机专业计算机的教学工作, 普及计算机应用知识和提高计算机应用能力, 上海市高等教育局建立了上海高校非计算机专业学生计算机应用知识和应用能力等级考试制度。该制度针对不同专业对计算机应用知识与应用能力的不同要求, 把考试分为一级、二级、三级, 共三个等级。

根据考试委员会颁布的有关第二等级考试的设置目标和考试范围, 复旦大学组织力量编写了这套教材丛书。包括: 《计算机应用初步》、《TRUE BASIC 语言程序设计基础》、《FORTRAN 语言程序设计基础》、《PASCAL 语言程序设计基础》、《C 语言程序设计基础》、《COBOL 语言程序设计基础》等六种。根据“要求具有使用计算机的基本知识和使用一门高级语言在计算机上编制程序和上机调试的能力”的考试目标, 学生必须学完计算机应用初步和一门高级语言程序设计基础(BASIC、FORTRAN、PASCAL、C、COBOL 中任选一门) 才能顺利通过第二等级的考试。使用这套教材的教学安排为: 应用初步教学 20 学时(含上机), 高级语言程序设计基

础教学 72 学时(含上机),共计为 92 学时。

非计算机专业的计算机教学特点是以应用知识和应用能力为主的。本套教材的作者都是从事这方面教学的教师,有较丰富的教学经验,教材中的许多内容直接取自于多年来积累的教学笔记。

本套教材适用于大学理工科非电类专业、计算机应用专业大专、计算机成人教育和继续教育的各类进修班、培训班,以及广大工程技术人员和管理人员。

欢迎广大师生在使用中对本套教材提出批评与改进意见。

施伯乐

1992.4.15

前 言

计算机程序设计语言 Pascal 是一种系统地体现 E. W. Dijkstra 和 C. A. R. Hoare 所定义结构程序设计概念的语言。这种语言表达能力强,适合于数值计算和非数值计算的程序设计技术。它使用广泛,已成为科学研究上和程序设计教学上的一种极有效的语言。

本书以介绍程序设计方法为出发点,希望读者通过学习 Pascal 语言的基本概念和内容及其程序编制技术,能熟练地掌握这种结构程序设计语言。本书可作为高等学校非计算机专业学生学习 Pascal 程序设计的教材,同时适合用作同等程度人员的计算机程序设计自学教材。

全书分两个部分,共十一章,另设附录。每章内容都有小结,并配备合适的习题。第一章至第四章主要介绍程序和程序设计的基本概念、Pascal 语言的标准数据类型、基本语句和控制结构等。这部分内容由蒋家福编写。第五章至第十一章进一步介绍 Pascal 语言的丰富数据类型,以及以子程序方法为中心的程序开发技术。后一部分内容由张增琪编写。全书是在招兆铨指导和组织下编写的。

夏宽理同志精心地审阅了全书,提出不少修正意见,在此表示感谢。

因限于作者水平,而且时间仓促,书中存在不妥和谬误之处,请读者原谅。

编 者

1992 年 3 月

目 录

序 言 前 言

第一章	概述	1
§ 1.1	程序设计	2
§ 1.2	Pascal 程序结构和基本词汇	6
§ 1.3	Pascal 程序的编译和运行	12
	小结	13
	习题	14
第二章	数据类型、表达式和赋值	15
§ 2.1	数据和数据类型	15
§ 2.2	标准数据类型	17
§ 2.3	常量和变量	23
§ 2.4	表达式和赋值语句	28
	小结	36
	习题	37
第三章	输入和输出	40
§ 3.1	数据的输入和输出	40
§ 3.2	输入语句	41
§ 3.3	输出语句	45
	小结	53
	习题	53
第四章	控制结构	55
§ 4.1	复合语句	55

§ 4.2	条件语句	56
§ 4.3	分情形语句	62
§ 4.4	循环语句	67
§ 4.5	转向语句	81
§ 4.6	程序设计风格	85
	小结	97
	习题	98
第五章	自定义类型	101
§ 5.1	类型定义	101
§ 5.2	已定义类型的重新命名	102
§ 5.3	子界类型	104
§ 5.4	枚举类型	107
	小结	113
	习题	113
第六章	数组类型	115
§ 6.1	结构数据类型	115
§ 6.2	数组类型定义	115
§ 6.3	多维数组	126
§ 6.4	紧缩数组	140
§ 6.5	字符串	145
	小结	146
	习题	147
第七章	记录类型	150
§ 7.1	记录类型定义	150
§ 7.2	记录的引用和赋值	152
§ 7.3	开域语句	157
§ 7.4	带变体的记录	167
§ 7.5	紧缩记录类型	172
	小结	173
	习题	173

第八章	子程序和分程序结构	176
§ 8.1	子程序概念	176
§ 8.2	过程	177
§ 8.3	参数结合方式	183
§ 8.4	函数	189
§ 8.5	过程和函数作为参数的子程序	195
§ 8.6	分程序结构	205
§ 8.7	递归	215
§ 8.8	子程序应用	220
	小结	234
	习题	234
第九章	集合类型	240
§ 9.1	集合的定义	240
§ 9.2	集合赋值	242
§ 9.3	集合运算	244
§ 9.4	集合的输入和输出	249
	小结	252
	习题	252
第十章	文件类型	254
§ 10.1	文件概念	254
§ 10.2	文件类型和说明	256
§ 10.3	文件操作	258
§ 10.4	正文文件	276
§ 10.5	内部文件	283
§ 10.6	文件应用	285
	小结	292
	习题	293
第十一章	动态数据结构	296
§ 11.1	结构基本概念	296
§ 11.2	指针类型和动态变量	299

§ 11.3	动态变量的建立和引用	303
§ 11.4	动态变量的释放	308
§ 11.5	动态数据结构的应用	311
小结		351
习题		351
附录一	保留字和预定义标识符	354
附录二	Pascal 句法图	355
附录三	实习指南	366
附录四	PASCAI 语言考试大纲	372

第一章 概 述

计算机程序设计语言 Pascal, 是在 1971 年由瑞士沃斯 (N. Wirth) 教授正式提出的。它是从 Algol 60 程序设计语言的基础上发展而成的一种良好结构程序设计语言 (语言结构化和数据结构化), 这是最早以条理清楚的方式来表达出 Edsger Dijkstra 和 O. A. R. Hoare 所定义的结构程序设计语言。Pascal 语言的诞生, 可以说是程序设计语言发展过程中的一个里程碑。

Pascal 语言具有丰富完备的数据类型、简明灵活的通用语句、清晰明了的程序结构。因此, Pascal 语言表达能力强、适用面广, 既能描述数值又能描述非数值问题; 既能用于应用程序设计开发又能用于系统程序设计。Pascal 语言编译紧凑方便, 在现代计算机上有较高效率和较好移植性。

Pascal 语言的良好性能和它的程序简明、易编且易读, 促进它广泛应用和发展, 使之成为目前国际上最受欢迎和流行的程序设计语言之一, 特别在教学上, 早就被国内外学校选为程序设计教学语言, 近二十年的教学实践证实它是一门很好的程序设计教学语言。学习 Pascal 语言, 能使学生一开始就接受严谨、良好的程序设计风格和掌握全面的程序设计技巧的训练, 有力地培养学生使用高级语言开发程序的能力。另外, 学习 Pascal 语言有利于学生在今后再学习 Fortran、Cobol 等带有一定应用偏向的程序设计语言。

本章首先说明程序和结构两个基本概念, 从而说明 Pascal 程序结构、词汇、语法和 Pascal 程序的编译和运行等基础知识。

§ 1.1 程序设计

一、程序

有些初学者误认为计算机就是能直接求解问题的机器。其实使用计算机求解问题的一般步骤是：

- (1) 把问题归结成数据(信息)流和算法流程图。
- (2) 编制使计算机能直接或简接受解决的解决问题的程序。
- (3) 精心调试程序，使之输入已知数据后，能产生预期的结果。

程序是计算机能直接或间接地接受的一系列指令的序列。程序概念并不深奥和难以捉摸，在日常生活中，人们经常可遇到实实在在的程序。如一份食谱的烹调说明、乐谱、毛线编织图案、各种零件装配成品的流程等等，它们也可称为程序。程序被精确地编制后，要有处理器去执行程序指令，程序执行的过程称为程序的运行。如演奏家按一份乐谱演奏一段音乐，演奏家好比处理器。计算机的程序比其它类型的程序要更为复杂些，但具有如下的公共特点：

(1) 按前后顺序执行指令序列。这就是从第一条指令开始，依次执行每条指令，直至结束为止。但是遇到某些特别明确规定时，允许按规定变更执行顺序。

(2) 程序的运行将得到预定的结果。计算机程序的运行结果通常是以打印或显示一组信息的形式来表达的。

(3) 程序是对加工的对象施行操作。如要加工成为小菜的对象是肉、蛋和鱼等。计算机程序操作的对象是各种类型的数据。

(4) 时常在指令的前面对操作对象作必要的说明。如在食谱前面经常有所需配料的列表。程序员在编写程序之前，也必须先说明使用数据的属性。

(5) 有些指令要求处理器作出判定。如若用新鲜西红柿，则要剥皮，且在放洋葱之前放入；但若用罐头西红柿，则在最后才放

入,设计指令的作者要求“处理器”——厨师,根据给出西红柿的实际情况,对该指令表示的准则作出判定。

(6) 对需要不止一次地执行一条或一组指令时,必须指明重复次数或表明终止重复的条件。

(7) 程序本身是静态的,执行程序指令的过程是动态的。我们不要把厨师和食谱、钢琴家和乐谱两者混淆起来。同样,不能把处理器和程序两者混淆起来。

由以上几点,说明了程序的实质是程序的作者与处理器用来通话的手段。通话就得用一种专门的术语。如音乐家、厨师和毛线编织者都分别有自己的、完全独特的语言。程序设计语言就是程序员与计算机通话的语言。

二、程序结构

世上公认的第一台电子计算机诞生于1946年,那时程序的指令仅是机器能直接执行的原始指令。使用计算机解决问题编制的程序经常是由几千条指令组成。如果孤立地去看待每条指令,那么程序员就无法把握整个程序功能,使程序既难编写又难调试。主要原因是程序缺乏结构性,于是程序员需对那些原始指令的序列赋予一定的结构性。

结构性首先表现在表达式的计算中,如表达式:

$$\bullet \quad 5 \times 6 + 1$$

相应程序

输入 5

乘 6

加 1

对表达式:

$$5 \times (6 + 1)$$

相应程序

输入 6

加 1

乘 5

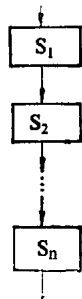
程序员观察到,上面普通代数形式所描述的表达式,能翻译成机器指令的序列,这两者有着固有的对应关系。于是,程序员就用表达式来对应某些指令序列。

数据是程序运行的对象,因此数据结构化也是十分重要的。起初通过程序把数据存放在存储器中,如计算一百个实数的平均值,这些实数被指定放在第 500~599 的单元中,但在编成上千条指令时,有时可能会忘记这些数所存放单元的约定,而放入了其它东西,使计算结果产生了错误。于是,在调试程序时带来了不少麻烦。为此,使用具有数据结构化的描述形式,把它表达为:

averager:ARRAY [1...100] OF integer;

指令本身的结构化也是十分重要的,可以证明所有计算机程序都能用四种基本的结构来表达。这四种基本结构是:顺序,判定,循环(或重复)及过程。

1. 顺序结构



指一条接着一条执行的一组指令。其流程见图 1.1。

BEGIN {开始}

S_1 ;

S_2 ;

$\vdots$

S_n

END {结束}

图 1.1 顺序结构流程图

其中文字说明放入 { } 内, $S_1, \dots, S_n$ 是指令。

2. 判定结构

指根据给定的条件,选定执行那条指令。

(1) 如果条件成立,执行指令 S_1 , 否则执行指令 S_2 。

IF {如果} B {条件} THEN {则}

S_1 {指令}
 ELSE{否则}
 S_2 {指令}

其流程见图 1.2, 其中常用 T 表示 B 为真, F 表示条件 B 为假。

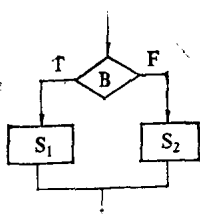


图 1.2 双向条件流程图

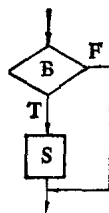


图 1.3 单向条件流程图

(2) 如果条件成立, 执行指令 S, 否则不执行 S。

IF B THEN S

其流程见图 1.3。

3. 循环结构

指满足给定条件时循环地执行一组指令。对给定条件, 主要分为如下两种:

(1) 当条件成立时, 循环地执行一组指令;

当条件不成立时, 循环就结束。

WHILE{当}B DO{做}S

其流程见图 1.4。

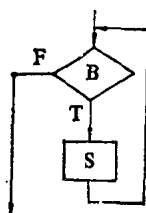


图 1.4 WHILE 循环流程图

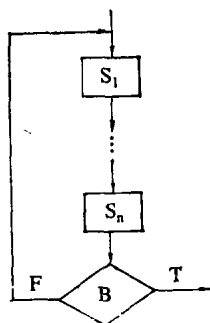


图 1.5 REPEAT 循环流程图

(2) 循环执行一组指令,直至指定的条件成立为止。

REPEAT {重复}

S_1 ;

$\vdots$

S_n

UNTIL {直到} B

其流程见图 1.5。

4. 过程结构

指用单条指令来引用一组指令。程序设计中应用了过程,编制的程序就简洁明了,且赋予层次结构。过程的概念是十分重要的,要编写一个实用的良好的计算机程序是离不开过程的。

程序设计语言的历史,主要是对这些机器指令序列赋予某些结构,Pascal 程序语言是十分注重程序结构的。因此,Pascal 程序语言是表达能力强、易学习、易编写、易使用,且易调试的一门流行广泛的程序设计语言。

§ 1.2 Pascal 程序结构和基本词汇

一、Pascal 程序结构

程序主要是描述对给定的各类数据的处理过程。Pascal 语言是如何描述和表达数据呢?使用 Pascal 语言编写的程序,它的结构和一般格式又是怎样的呢?下面对一组非零整数选择最小值和最大值的简单 Pascal 程序例题,来初步说明完整的程序结构。

例 1

PROGRAM minimax(input, output),
{求一组非零整数的最小值和最大值,且}程序首部
程序在输入整数零时结束}

```

CONST
    min = -32767;
    max = 32767;
VAR
    reading: boolean;
    number, minimum, maximum,
    count: integer;
BEGIN
    reading := true; count := 0;
    minimum := max; maximum := min;
    WHILE reading DO
        BEGIN
            read(number);
            IF number = 0 THEN reading := false
            ELSE
                BEGIN
                    count := count + 1;
                    IF number < minimum THEN
                        minimum := number;
                    IF number > maximum THEN
                        maximum := number
                    END
                END
            END;
            writeln('input numbers:', count);
            writeln('the smallest is', minimum);
            writeln('the largest is', maximum)
        END.

```

程序体

第一部分是程序首部。它由下面内容组成:

(1) 程序标志 PROGRAM。英文字母必须大写。它是 Pascal 语言的保留字, 放在程序的开始部位。