

高等院校适用教材

PASCAL语言 程序设计

孙家启 章欣 吴国凤 主编

机械工业出版社

12

Q/1

PASCAL程序设计语言具有小巧、简洁、连贯、精致、结构性好,数据类型丰富,表达能力强,实书写格式自由,移植容易和程序设计风度优美等特点。它在我国高等院校中正在迅速推广,而且即将成为一种新的标准化工业语言。

本书共分十一章,全面系统地介绍了标准PASCAL 程序设计语言的语句、程序结构、数据类型以及编程方法,并简要地叙述在IBM PC微型机和SIEMENS 7570 -C 中型机上编制、调试、运行 PASCAL 程序的步骤和方法,书末附录中提供PC-PASCAL 对标准 PASCAL 的一些扩充以及 TURBO PASCAL 操作简介。

本书内容丰富,通俗易懂,深入浅出,例题丰富,实用性强,是一本较好的学习和掌握 PASCAL 程序设计的教科书,可供高等院校机、电、仪、经管等有关专业学生用的教材,也可供成人教育各专业学生用的教材或参考书。

PASCAL语言程序设计

主编 孙家启 章 欣 吴国辰

*

责任编辑:李 敬

封面设计:郭景云

*

机械工业出版社出版(北京阜成门外百万庄南街一号)

(北京市书刊出版业营业许可证出字第117号)

三河印刷一分厂印刷

新华书店北京发行所发行·新华书店经售

*

开本787×1092¹/₁₆ • 印张 13³/₄ • 字数 330 千字

1991年9月北京第一版·1991年9月北京第一次印刷

印数 0,001—7,600 • 定价:6.50元

*

ISBN 7-111-02993-3/TP·148

前 言

《程序设计》是计算机科学十分重要的一门基础课。不论计算机专业还是非计算机专业人员，在进入计算机科学大门时，它的知识与技能都是最基本的。因此，《程序设计》可以说是计算机科学的入门课。

PASCAL是一种新型的高级程序设计语言，是根据结构程序设计研究出来的。它具有：基本结构简单，数据类型丰富，编写的程序易读、易维护，实现效率高，查错能力强，移植性好和程序设计风度优美等特点。广泛用于数值计算、非数值计算和编写各种系统软件等。

70年代初，瑞士沃斯（N. Wirth）教授正式发表PASCAL 程序设计语言。这引起了国内外许多计算机公司的重视，几乎所有大、中、小型计算机上都得到实现，而且在微型机上应用起来比较方便，大多数微型机都配有 PASCAL 编译程序。这些优越的客观环境促使越来越多的人学习 PASCAL 程序设计语言。正是在这种形势下，人们迫切地需要一本通俗易懂而又比较详细的，适合我国高等院校作为推广计算机应用的启蒙语言的有关 PASCAL 语言方面的书籍。本书正是为了适应这种形势和要求而编写的。

从我国高等院校的实际情况出发，本书注重实用性，结合国内流行的IBM PC及其兼容机上使用的IBM PASCAL，全面地系统地介绍了标准PASCAL，在书末附录中提供 PC PASCAL 对标准PASCAL的扩充。内容安排力求准确完整，算法分析简明扼要，概念叙述通俗易懂。

本书提供了 IBM PC 微型机和SIEMENS 7570-C小型机上调试程序的一些必要的资料。附录中还扼要介绍TURBO PASCAL特点及其操作步骤。

合肥工业大学孙家启编写本书第一、七、八章和附录；合肥工业大学吴国凤编写第二、三章；华北航天工业学院姜德森编写第四章；哈尔滨电工学院孙宇编写第五章；北京机械工业管理学院张延炬编写第六章；洛阳工学院章欣编写第九、十、十一章。全书由孙家启以及章欣、吴国凤主编，哈尔滨电工学院王杰臣副教授、东北重型机械学院武金木副教授主审并提出有益的建议和修改意见，在此表示感谢。

在本书编写过程中，自始至终得到机械电子工业部教育司、机械电子工业部“高校计算机基础教育研究会”对本书出版给予大力支持；我们得到研究会理事长张奠成教授和研究会理事谢培均、王德民两位副教授热心指导和帮助；另外，我们还得到兄弟院校和其它部门的计算机工作者的热情鼓励和帮助。在此一并致谢。本书不足之处，请专家和读者指正。

编 者

1991年3月

目 录

第一章 计算机系统及程序设计概论	(1)
§ 1-1 计算机系统.....	(1)
§ 1-2 PASCAL 语言的特点.....	(6)
§ 1-3 PASCAL 语言的基本符号.....	(8)
§ 1-4 PASCAL 语言的程序结构.....	(10)
§ 1-5 PASCAL 源程序的编译和运行.....	(12)
习题一.....	(13)
第二章 数据、表达式、赋值语句和输入/输出	(14)
§ 2-1 数据与数据结构.....	(14)
§ 2-2 常量与常量定义.....	(15)
§ 2-3 变量与变量说明.....	(16)
§ 2-4 标准类型.....	(17)
§ 2-5 表达式与赋值语句.....	(21)
§ 2-6 输入/输出语句.....	(23)
§ 2-7 应用程序举例.....	(26)
习题二.....	(28)
第三章 流程控制语句	(30)
§ 3-1 复合语句.....	(30)
§ 3-2 如果语句 (IF 语句)	(31)
§ 3-3 情况语句 (CASE 语句)	(34)
§ 3-4 直到语句 (REPEAT 语句)	(36)
§ 3-5 当语句 (WHILE 语句)	(37)
§ 3-6 循环语句 (FOR 语句)	(38)
§ 3-7 转向语句 (GOTO 语句)	(42)
§ 3-8 应用程序举例.....	(43)
习题三.....	(45)
第四章 过程和函数	(46)
§ 4-1 过程的定义和调用.....	(46)
§ 4-2 函数的定义和调用.....	(48)
§ 4-3 变量的作用域、全程变量和局部变量.....	(50)
§ 4-4 数值参数和变量参数.....	(53)
§ 4-5 过程的嵌套、递归及向前引用.....	(55)
§ 4-6 过程参数和函数参数.....	(61)
§ 4-7 应用程序举例.....	(63)
习题四.....	(67)

第五章 类型定义、枚举类型及子界类型	(69)
§ 5-1 类型定义.....	(69)
§ 5-2 枚举类型.....	(70)
§ 5-3 子界类型.....	(73)
§ 5-4 应用程序举例.....	(75)
习题五.....	(77)
第六章 集合类型	(78)
§ 6-1 集合与集合类型的概念.....	(78)
§ 6-2 集合类型的定义与集合型变量的说明.....	(79)
§ 6-3 集合型常量的表示形式.....	(80)
§ 6-4 集合的运算与输入/输出操作.....	(81)
§ 6-5 应用程序举例.....	(87)
习题六.....	(93)
第七章 数组类型	(94)
§ 7-1 一维数组.....	(94)
§ 7-2 多维数组.....	(97)
§ 7-3 高级数组、紧缩数组和布尔数组.....	(99)
§ 7-4 字符串处理.....	(103)
§ 7-5 应用程序举例.....	(107)
习题七.....	(110)
第八章 记录类型	(112)
§ 8-1 记录类型的数据.....	(112)
§ 8-2 开域语句.....	(115)
§ 8-3 记录的变体部分.....	(117)
§ 8-4 应用程序举例.....	(120)
习题八.....	(124)
第九章 文件类型	(126)
§ 9-1 文件的基本概念.....	(126)
§ 9-2 文件的建立和读写.....	(128)
§ 9-3 文件的更新和合并.....	(132)
§ 9-4 文本文件.....	(136)
习题九.....	(140)
第十章 指针类型和动态数据结构	(142)
§ 10-1 指针.....	(142)
§ 10-2 链表.....	(145)
§ 10-3 栈和队列.....	(154)

VI

§ 10-4 二叉树	(157)
习题十	(162)
第十一章 程序的编制与调试	(164)
§ 11-1 程序设计基础	(164)
§ 11-1 调试和错误处理	(168)
习题十一	(170)
附录	(172)
附录一 ASCII 码	(172)
附录二 标准标识符	(174)
附录三 PASCAL 语法图	(178)
附录四 PC-PASCAL对标准PASCAL的一些扩充	(182)
附录五 IBM PC机 PASCAL 程序上机操作指南	(198)
附录六 Turbo PASCAL 操作简介	(201)
附录七 Siemens 7570-C 机 PASCAL 程序上机操作指南	(205)
参考文献	(211)

第一章 计算机系统及程序设计概论

§ 1-1 计算机系统

计算机不仅是一个复杂的电子设备系统，同时又是一个复杂的信息处理系统。当我们接触到微型计算机时，它的整套硬件设备和应用软件，都会使我们感受到计算机本身是一个复杂的系统。

对于复杂系统的使用，我们可以用系统的观点去掌握它。即把系统的各个组成部分，当作互相联系的整体来理解。

计算机系统的观点应该有如下含义：

- (1) 计算机本身是一个复杂的系统；
- (2) 计算机和它应用的环境合成一个系统；
- (3) 计算机还和它的开发者、使用者一起构成一个系统。

后两点分别强调的是系统配置的适用性和所开发系统的可用性。配置不当，或者开发的软件只有编程的人自己会用，都是微型计算机应用所不希望的。

一、微型计算机系统构成的基本概念

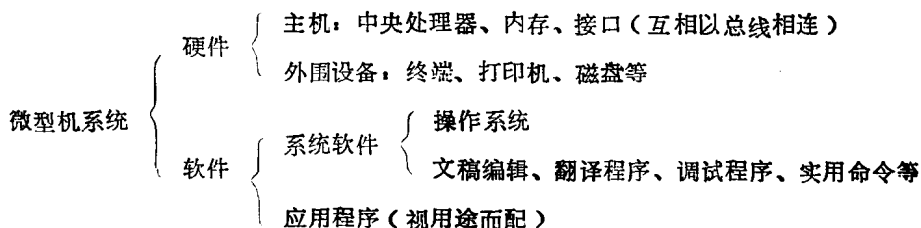
下面将介绍微型计算机应用所必须掌握的概念和术语。

1. 硬件和软件 (Hardware/Software)

计算机系统由硬件和软件两个子系统构成。硬件是指构成计算机所有的物理设备、器件；软件是指计算机所用的信息，信息是程序和数据的集合，软件还应包括相应的使用文档和说明书等。

软件要利用计算机所有的资源(硬件，也包括软件)进行工作。没有软件，计算机无法工作。软件要依赖于硬件的功能加以实现；而硬件要根据软件的规定进行工作。因此，软硬件之间如不配套，计算机系统是无法进行正常工作的。

2. 系统组成



硬件由主机和外部设备（简称外设，又称外部设备或周边设备）组成，外围设备各自经相应的接口和主机相联。

软件由系统软件 and 应用程序组成。

主机由中央处理单元（CPU Central Process Unit），内存储器（简称内存）和接口部件组成。这三者之间通常经总线（BUS）交换信息。

计算机组成的各硬件部件之间的关系见图1-1。迄今为止的计算机都是这一类的模式，称为程序存储式计算机。

3. 程序存储式计算机

从电子管、分立元件晶体管、中小规模集成电路，到LSI、VLSI第四代计算机——微型机，都是程序存储式计算机。

计算机按事先在内存储器中存放好了的程序进行工作。这就是程序存储式计算机。

这里所指的程序，是狭义的定义，即是专指机器码的程序，是用计算机本身能懂得的“机器语言”——

指令系统编制的。而高级语言写的程序应称为源程序，源程序属于信息的数据范畴。

内存储器相当于记忆体，用以存放信息（程序和数据）。取用其中存放的程序并进行工作的是中央处理器（CPU）。

4. 中央处理器（CPU）

中央处理器在主机中处于核心地位。它的全部职能是不断地从内存中取得指令并加以解释执行。除了取得指令和执行指令外，中央处理器不干别的工作。

机器码指令是CPU能懂得的代码。程序是由指令序列构成的。程序是唯一用以规定CPU工作的信息，CPU完全按指令序列——程序，进行规定的工作。

5. 内存（Memory）

存储器分为内存和外存。外存属于外围设备的范畴（如软磁盘、硬磁盘、磁带），可以在断电的情况下长期保存大量的信息。内存中只存放运行时所需的程序和数据。

程序存储式计算机这一工作模式，要求在通电前要准备好CPU开机时的工作程序，为此，内存中需要有ROM（Read only Memory，只读存储器），其中存放用专门设备固化进去的信息（程序和数据），在断电时也不会丢失，也不能被改写的。固化有软件的ROM又称为固件。微型机系统ROM中存放有起动程序。

内存的另一部分是RAM，是可随意读写的存储器，写入的信息在通电时能被保存。运行中需要的程序和数据可以随时由外存调入。

ROM和RAM组成的是记忆的主体，称为存储体，存储体要在存储控制器的控制下进行读写。存储体和它的控制器一起组成内存储器。

内存是一个很大的记忆部件。最小的记忆单元是一位二进制信息：0或1。微型机中通常把这样8位的二进制信息存储空间作为一个最小的工作单元，整个存储体就由许多这样的工作单元组成。访问内存时，根据指定工作单元的地址，可以对它进行访问，即读写该单元的内容。

内存储器的主要技术指标是容量和速度。微型机常用8位（即一个字节）作为内存的基本工作单元，所以内存容量是以多少个字节来表示容量的大小，容量的单位有千字节（KB）和兆字节（MB）。PC机内存空间可有1MB。内存的速度是指存储控制器在接受由CPU

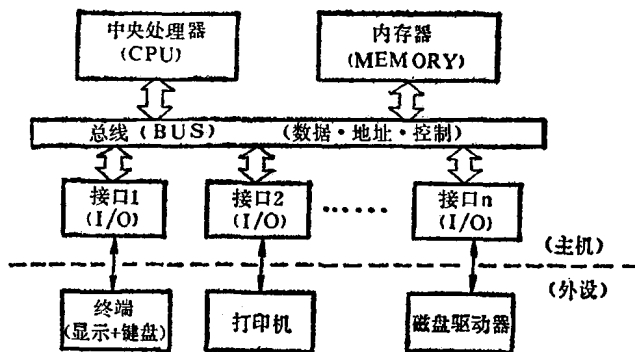


图1-1 计算机的硬件组成原理图

发来的地址和读（写）控制信号后，独立完成一次读写所需的周期时间，其数量级在微秒（ μs ）和若干毫微秒（ns）之间。

6. 接口 (Interface)

接口，在硬件中指的是承担着主机和某种外部设备进行信息交换的控制器件。接口是相当复杂而又种类繁多的一类主机部件。

按信息传输的方向划分，接口有输入、输出两种。有的接口，如对磁盘驱动器，它的接口既要承担读入，又要承担写出的功能。

现代高档一些的微型机，在输入输出管理中，不仅用接口部件，而且还利用类似CPU那样的处理器，称为通道处理机，管理着高速的数据传输，简称通道。主要用于信息块（成组数据）的快速传送，比如读写磁盘成批信息。

7. 指令系统和汇编语言

每一种CPU都有自己可以识别并加以执行的指令系统。指令系统是软件和硬件(CPU)之间的界面。CPU能接受的指令形式是指令的机器代码。

每一条机器码指令都有两个基本的组成部分：操作码和操作数。操作码以代码表示该指令要执行何种功能的操作。操作数是以代码表示被操作的数据（一个或多个）从哪里寻得，故又称为寻址。

为了方便对指令一级的编程，对每一条指令以及寻址方式都规定了助记符。用助记符书写的指令序列（程序）称为汇编语言程序。

$M = A + B$ 这样的表达式可以化为一条汇编程序的指令：

ADD A, B, M (把A, B相加，结果送M)

其中，ADD代表的是这一加法指令的操作码；A, B, M分别是三个操作数，指明指令处理数据的来源与去向。

汇编语言是面向机器来编写程序的，如上述指令中的A, B是CPU内的寄存器名字，M是内存中某一个工作单元的地址。汇编语言和高级语言不同，高级语言是面向问题编程的语言，可以使用许多人类语言中的词汇，接近了人的语言。汇编语言是一种低级语言，它是属于机器这一级的，因而比较难学，要求掌握更多的硬件知识。但是，用汇编语言编程的优点是可以确切地安排程序的执行时间。因为它可以精确求出每条指令的执行时间，故实时性好，汇编语言常用在过程控制的应用场合。

8. 高级语言

汇编语言是和指令对应的，它是面向机器硬件编程的语言。对于非计算机专业人员，计算机系统提供了另一种界面，用高级语言编写程序同计算机打交道。

高级语言编程的特点是采用近似于人类的文字语言。目前国内外比较通用的计算机高级语言有几十种。常见的应用最普遍的有：BASIC, FORTRAN, ALGOL, COBOL, PL/1以及PASCAL等在多数系统中，BASIC语言的源程序是通过BASIC解释程序边解释边执行的。而其他几种语言要经过各自的编译程序编译之后，才能正常运行。各种编程语言（汇编语言、高级语言）各有特点，都在发展，各有各的用途。问题只是你认为哪个好用就可以。

二、计算机系统的层次观点

人们在研究一个复杂系统时，往往将它分解开来进行研讨，同时又要从各部分的联系中

去作整体的认识。计算机是一个复杂系统，在组成计算机的这些模块间，有一定的层次关系。认识计算机系统内在的这些模块间层次关系，有利于总体上去掌握我们要应用的微型机资源。

1. 在哪一级同微型机打交道

层次观点的建立，有助于使用者意识到，应同哪一级的系统资源打交道，从而去研讨怎样进入这一层次，怎样使用这一层次的资源，又怎样透过这个层次和其它层次的资源打交道。粗略地说，使用微型机可以有三个大的档次同微型机打交道。

用户层：使用他人开发的现成的应用程序。

开发层：用某利语言编写程序并作调试。可以是高级语言、低级语言编程，也可以是几种编程语言。他使用时打交道的是开发环境。

系统维护层：维护和扩充系统资源的工作。

不同的档次，目的不同，要求用户具备的计算机知识水平也不同。其中，用户层最高，越是在高层，下面的软、硬件支持越丰富，对使用者越容易。所以，对高层用户的要求反而低。相反，在和计算机低层打交道时，对用户的要求却越高，难度也越大。

我们用图解方法描述微型机系统中各模块之间的层次关系，微型机的层次和模块见图1-2。

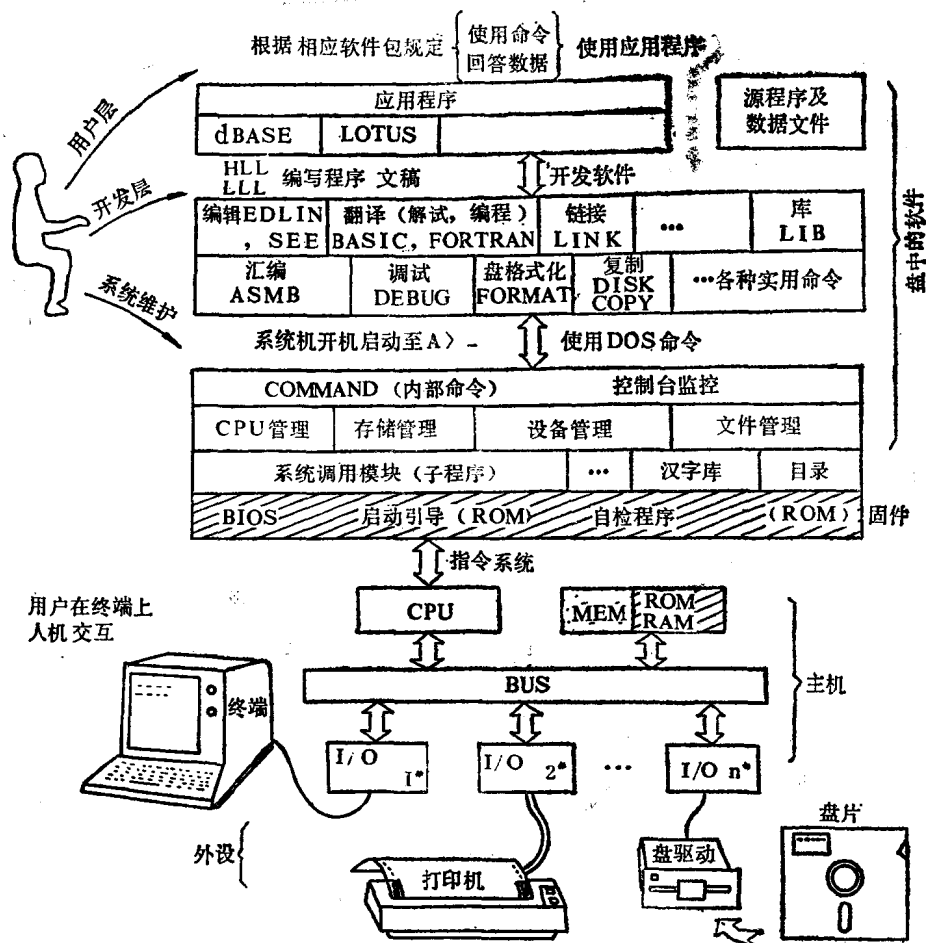


图1-2 微型机的层次和模块

从图 1-2 中, 我们看到, 在终端上用机的用户可以在三个档次 (用户层、开发层、系统维护层) 中和微型机打交道。用户在这三个档次中, 均是透过软件和系统的硬件进行工作的。

2. 硬核和软包

有这样一种形象比喻: 硬件是核心, 软件是一层层包裹在外的。这种比喻把模块间层次关系描绘出来了。

硬件是物质基础, 它支持外层软件实现各种功能; 软件是支配硬件工作的信息, 位于外层的软件要调用内层的资源方能工作。

没有软件的硬件称为裸机。裸机是无法工作的。没有软件的硬件好比人只有躯体而无思想, 什么也干不成。软件对于计算机的重要性, 好比学习知识对于人的重要性, 软件越丰富计算机的能力就越强。

软件本身也分为许多层次, 每个层次内, 还有复杂的信息结构。在许多应用软件中, 流行着结构化的程序设计, 如菜单技术, 用户可以像点菜那样方便地进入应用程序的各个层次, 以实现各种功能的组合。软件分层的另一种意义还在于高层的软件可以获得低层软件的支持。低层软件, 甚至硬件, 都在支持着高层软件的工作。这种支持是“透明”的。所谓透明, 是指它是视而不见, 但确又存在, 不可缺少。

3. 软硬件的配套

在层次关系中, 我们必须注意软件和硬件的协调一致性。有的应用程序, 需要里层的某个软件支持, 调用了里层的功能模块才能工作, 否则, 就无法运行。操作系统往往有不同的版本, 于是, 在某个操作系统支持下开发软件, 换到另一种同类型系列机上, 就有可能因为操作系统的版本号不一样, 无法运行。所以应用程序开发者应向用户说明, 他所开发的程序, 有什么必备的运行环境, 这里指的运行环境, 是指运行该程序时, 计算机系统硬件和软件的最小配置是什么。

4. 软硬件的逻辑等效性

在具备了必需的硬件设备后, 要在这样的微型机上实现某种功能, 可以用开发软件的方法实现, 也可以用增配硬件的方法加以实现。这两者的功能实现, 在逻辑上是等效的。

对于开发软件的人员来说, 懂得这种软硬件的等效性是有一定实际意义的。就是在开发工作中, 要考虑其中某个功能究竟用软件实现, 还是用增加硬件设备来实现。应从技术经济指标来全衡考虑。一般地说, 单纯用软件实现, 必然多花费程序运行的时间, 如用 8 位机去算 16 位或 32 位及以上的数据, 分别处理某个数的一部分, 然后再处理其余部分, 比用 CPU 字长长的算一次要慢得多。但是, 硬件指标低些, 费用也省些。只要时间允许, 可以用软件来实现, 而且软件易复制, 成果容易推广。反之, 在有速度要求的场合, 新增的硬件使用频繁, 单纯由软件实现某种功能的方案就不可取了。

应用微型机和应用高科技设备一样, 不是单纯地依靠资金去不断更新设备, 成败关键还在于掌握它的人, 在于人的知识水平。在有了基础的软硬件设备后, 能不能进一步地利用现有软硬件资源作二次开发, 去满足新的工作需要, 就必须树立计算机系统的层次观点, 懂得软硬件这种等效性, 这样, 开发工作就可以建立在不断地挖掘系统资源的潜力之上。

5. 适用性和可用性

把适用性和可用性放在计算机系统的层次观点中研讨, 也完全是应用的需要,

微型机,按应用场合的不同,应当由不同的模块组成,无论是硬件,还是软件,都是由模块化的层次构成。所以,适用性要求配置的合理性和完整性,完全是同应用的场合、要解决问题的要求相关的。我们绝不应当在流行什么机种时,就买什么机器。比如,现在人们在说现在流行单片机了,单片机比单板机好。殊不知,片级机、板级机各有各的用途。以PC机为例,IBM PC 正宗产品的系列中,单色显示器配的是单色字符卡,无图形功能,和国内广泛流行中的中文操作系统CCDOS 不相容。还是长城0520等兼容机一开始就考虑了中国用户的需要,单色显示器(比彩色便宜)也配了彩色图形卡,中文汉字是图形字符,在这类兼容机上反而好用了。此外,对于中文打印也有过不少的教训,用户往往只注意打印机电气特性的相配。这对于西文字符打印,有ASCII码的通用标准,不存在问题。汉字打印是图形方式,并不是所有打印机都能打印图形。即使有图形方式,还要有相配的软件作信息的控制与转换。不具备相配的打印机驱动模块,原先的DOS 操作系统是打不了汉字的。又如汉字字库,是配硬卡汉字好,还是软卡汉字好,也是不一样的。有的硬卡汉字不占RAM 空间,给用户留了较大的内存空间,开机时硬卡已在内存中,不像软卡汉字要有启动装入过程,的确较为方便。但是,硬卡是固化在ROM 中的字形库,增删和修改不便。软卡不仅有这种灵活性,而且在不使用汉字时,RAM 空间可以用作其它。软卡汉字还可以供用户随意选用,更换时没有设备的浪费。

适用性问题涉及的知识面很广,但是,没有系统机的层次观点是无法理解的。

可用性的问题涉及的是软件工程方面的知识。前面已提到,计算机还和它的开发者、使用者一起构成一个系统。

使用者所处的层次高于开发者。哪怕使用者是只有小学文化水平的乡镇企业工人,开发者即使是高级工程师,他也要考虑到用户的知识水平和使用习惯。不然,他的软件会不受欢迎。与用户不友好的软件,是无法推广使用的。为什么“傻瓜”相机能流行开来,就是开发者把先进技术用于测距和调焦这两项非专业摄影工作者不易掌握的地方,使不会照相的人,也能拍得清晰。这一点正是开发软件的人员要借鉴的。

为了开发可用的应用程序,开发者就要学习这类程序的编程风格,清晰而无二义性的提示,方便的操作,对键入数据的测试和纠错,是许多一系列和可用性有关的知识。请读者在学习使用各种系统软件时,从这些久经使用和修改的软件中,看到颇受欢迎的设计风格,究竟是什么样子。所以,在学习这一门学科中,还要在应用中吸取他人的良好编程风格,应当留心观察,已有的程序是如何同你打交道的。注意掌握一些约定的规定和方法,在许多微型机系统中是一致的,我们在开发应用程序时,也要加以采用。如凡是在要求用户输入操作时,好的软件处处都作必要的提示。有些与用户友好的软件,简直可以不事先看说明书,边操作就可以学会使用,而且功能居然很强,这是值得我们今后留意的。

§ 1-2 PASCAL语言的特点

60年代末瑞士苏黎世联邦工业大学的沃斯(N·Wirth)教授设计了一种新型程序语言,他为纪念一位最早实用计算器的发明者Blaise Pascal 而命名为PASCAL语言。这种语言比FORTRAN、ALGOL和BASIC等语言问世得都要晚。

PASCAL语言是70年代最有影响的一种新型的高级语言,它是从ALGOL60语言演

变发展过来的,但是它作了改进,性能更强。更为重要的是它体现了荷兰计算机科学家E·W·Dijkstra提出的“结构程序设计”的思想。这种思想为使程序具有结构合理、正确性高、易验证而规定了一套程序设计准则。PASCAL语言规则中反应了这些要求和限制。

PASCAL语言具有丰富的数据类型,简明的通用语句,清晰的程序结构,而且书写格式自由,编译紧凑,风度优美,直观易读。使程序设计教学成为逻辑的、系统的训练,适宜培养良好的程序设计风格。因此博得人们的好评。

PASCAL语言是一种系统程序设计语言,可以用来书写各种系统软件,如编译程序等。据了解,世界各国的许多高等院校都用它进行高级语言程序设计的教学,效果良好。它除了广泛应用于高等院校、科学研究、企事业等部门的大型计算机外,在微型计算机上PASCAL语言也已成为最流行的高级程序设计语言,所有主要的微型计算机都配有PASCAL编译程序。因此,使用广泛。

PASCAL语言,已列为我国高等院校计算机高级语言程序设计的基础教学课程。通过学习,掌握程序设计方法与技巧。并加强程序设计能力培养,与后继课程如软件技术基础、数据结构、数值计算与程序、编译方法、数据处理等学习关系很大。在实用的软件系统中,绝大部分的应用软件是用高级语言编制的。学习PASCAL语言的目的,是为了学习已有的应用软件和开发新的应用提供必要的基础。

最后,将沃斯(N·Wirth)教授文章中所提出的PASCAL语言的特征列举如下,供读者学习参考。

- (1) 变量说明是强制性的。
- (2) 某些基字(例如, BEGIN, END, REPEAT等)是“保留”的,不能作标识符。
- (3) 分号(;)看作是语句分隔符,而不是语句终止符(如同PL/1语言一样)。
- (4) 标准数据类型是整数、实数、字符和布尔类型。基本的数据构造工具有数组、记录(对应于COBOL和PL/1语言的‘结构’)、集合和文件。这些结构可以组合和嵌套以形成集合的数组、记录的文件,等等。数据还可以动态地分配并由指针访问。
- (5) 集合SET数据结构提供了类似于PL/1语言的‘位串’的能力。
- (6) 数组可以具有任意维数与任意的界;数组的界是常量,即没有动态数组。
- (7) 像FORTRAN、ALGOL和PL/1语言一样,存在转移语句。标号是无符号整数,必须加以说明。
- (8) 复合语句如同ALGOL语言,相应于PL/1语言里的DO群。
- (9) ALGOL语言的开关和FORTRAN语言的计算转语句的功能在PASCAL中由情况语句表示。
- (10) 循环语句相应于FORTRAN的DO循环,它的步长只能是1(TO)或-1(DOWN-TO),并仅当控制变量的值落在界内时执行。因此,被控制的语句可能完全没有执行。
- (11) 没有条件表达式和多重赋值。
- (12) 过程和函数可以递归调用。
- (13) 对变量不存在(像ALGOL语言里那样)‘OWN’属性。
- (14) 参数有值调用和引用调用,没有名字调用。
- (15) 就不存在匿名的分程序来说,这里的“分程序结构”不同于ALGOL和PL/1语

言的分程序结构，即每个分程序都只有一个名字，并做成一个过程。

(16) 常量、变量等所有对象必须在它们被引用之前加以说明，但允许以下两个例外：

- 1) 指针类型定义里的类型标识符；
- 2) 当存在向前引用时的过程与函数调用。

读者开始接触PASCAL语言，往往感叹它缺少某些“中意的特色”，例如方幂运算符、串的并列、动态数组、布尔值的算术运算、自动的类型转换和省缺说明等，都不是疏忽遗漏，而是有意删除的。因为，在某些情况下，有它们反而招致程序设计解法的效率不高；在另一些情况下，又会感到它们不利于达到程序设计清晰可靠和“风度优美”的目标。最后，还得考虑对繁多的有效的程序设计功能进行严格挑选，以保证编译程序比较紧凑和高效。

§ 1-3 PASCAL语言的基本符号

一、字符集

字符是构成词汇的基本元素。目前常用的是ASCII码。它有128个字符，其中有95个可打印字符（包括大写、小写字母，数字等）和33个控制字符（控制打字机而并不印刷的字符）（见附录一）。

常用的控制字符有：

cr 回车。使打印机的印字头托架回到起始位置的控制字符。

lf 换行。使印刷的位置进至下一行同一位置的控制字符。

ff 换页。使印刷的位置进至下一页开始的控制字符。

为了使打印机的印字头托架进至下一行的第一个字符位置，通常要“回车”、“换行”两个控制字符。这在键盘上通常由“↵”（即RETURN）键实现，而在书写时大多写成“↵”。

字符集往往和设备有关，使用时应注意。例如，字符↑在有的机器上采用∧表示。

字母字符有大写与小写之分，但编写程序时，除在字符串中大小写字母字符有明显的不同含义外，通常采用小写字母。

二、词汇集

PASCAL语言的词汇集由标识符、特殊符号、指示字、无正负号数、字符串及标号组成。

1. 标识符

(1) 作用：标识符是用来标记常量、类型、变量、过程、函数及程序的名，也可作为常量的值。程序人员可以对一些数据对象或一组操作命名，并可以通过名来引用它。一个名即是一个标识符。

(2) 组成：标识符由一串字母字符与数字字符所组成，其第一个字符必须是字母字符。其例见表1-1。

(3) 注意：标识符没有固有的含义。它仅用于标记对象的名（或者有时是标记一个值）。因此，可以随意地选用它们。

表 1-1

正确的标识符	非法的标识符	原 因
a	π	希腊字母不是ASCII字符集字符
monday	begin	begin已作它用，不能作标识符用
third	3rd	第一个字符不是字母字符
x1	x_1	标识符没有下标字符
y2a	y^{2a}	标识符不应含幂指数
myprogram	my program	空格字符使它变成了两个词
xy	x-y	-不是字母或数字字符
iandj	i&j	&不是字母或数字字符
wang	wang's	'不是字母或数字字符

(4) 标准标识符：标准标识符是标识符，它们是语言所预先定义的，因此都有约定的含义（见附录二）。

标准常量：false true maxint

标准类型：boolean char integer real text

标准文件变量：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：dispose get new pack page put read readln reset rewrite
unpack write writeln

这些标准标识符有约定的含义，但并非固有的含义。因此，程序人员可以重新定义它们，使之用于表记其它的对象，这样原来的意义便随之消失。当然，一般并不希望这样做。

2. 特殊符号

(1) 作用：特殊符号在语法上用作界限符号，即语言结构的规则中的分界符号。每个符号联系不同的语法成分。

(2) 组成：特殊符号共56个

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

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 record
repeat set then to type until var while with

(3) 注意：特殊符号中35个字称之“保留字”，它们外形上与标识符一样，但它们不是标识符，这些字是语言保留下来作语法结构规则中的分界符号用的。程序人员不应命名和这些保留字拼写相同的标识符。

在另外号21个符中有5个双字符号：< >、<=、>=、:=、..，它们每个都应视为

单独的一个整体，切忌将它们分开。

特殊符号是语言中具有固定含义的单独符号，程序人员不能将它们挪作它用。有时为了强调其固有含义，在印刷上将其印成黑体或划下横线。例如，**begin**或begin，它们与begin是等价的。

三、词与词的分隔

一个PASCAL程序从外形上看是一串词。习惯上词与词之间应有分隔，因为在一些情况下缺少分隔会改变原来的意义。例如，a div 20有三个词，其本意是a除以20。但若没有分隔即：adiv20，这是一个标识符，和原来就差之千里了。当然a+20，其中有无分隔无所谓。因此，要求在标识符、保留字、数之间，若两个词相连，词与词间应有分隔，通常放置一个空格字符（若放多个也无妨）。

§ 1-4 PASCAL语言的程序结构

一个程序语言的基本功能是对数据进行描述和操作。一个计算机程序，从本质上讲，它是描述对给定数据的处理过程。

PASCAL语言是如何描述数据又是怎样对数据进行操作的呢？用PASCAL语言编写的程序，其一般形式如何？这是读者首先关心的问题。

先通过一个简单的PASCAL程序来初步认识一下完整的程序结构。

```
PROGRAM Example (input, output);    {程序首部}
VAR  {变量说明} {分程序}
    x, square, recipr: real;
BEGIN  {语句部分}
    read(x);
    square := x * x;
    recipr := 1/x;
    writeln ('square of', x, 'is', square, 'reciprocal is', recipr)
END.
```

在这个简单程序中，实际上分三部分。其中：

第一部分是程序的首部，第一行。它指出这是PASCAL语言的程序，其名称为Example（简单的源程序）；程序的参数有两个：input, output。

第二部分是程序的说明部分，包括第二行第三行。它指出程序中使用了三个变量：x, square, recipr。它们都是实数类型的变量。

第三部分是程序的语句部分，包括第四行至第九行。它指出这个程序做四件事：

- (1) 通过读语句read(x)，输入实数x；
- (2) 通过语句square := x * x，计算两个实数之积；
- (3) 通过语句recipr := 1/x，计算实数x之倒数；
- (4) 通过写语句writeln ('square of', x, 'is' square, 'reciprocal is', recipr)，输出计算结果。

PASCAL语言的程序结构见图1-3。它的程序结构是模块结构，层次分明、清晰整齐。一般可以分为三大部分。

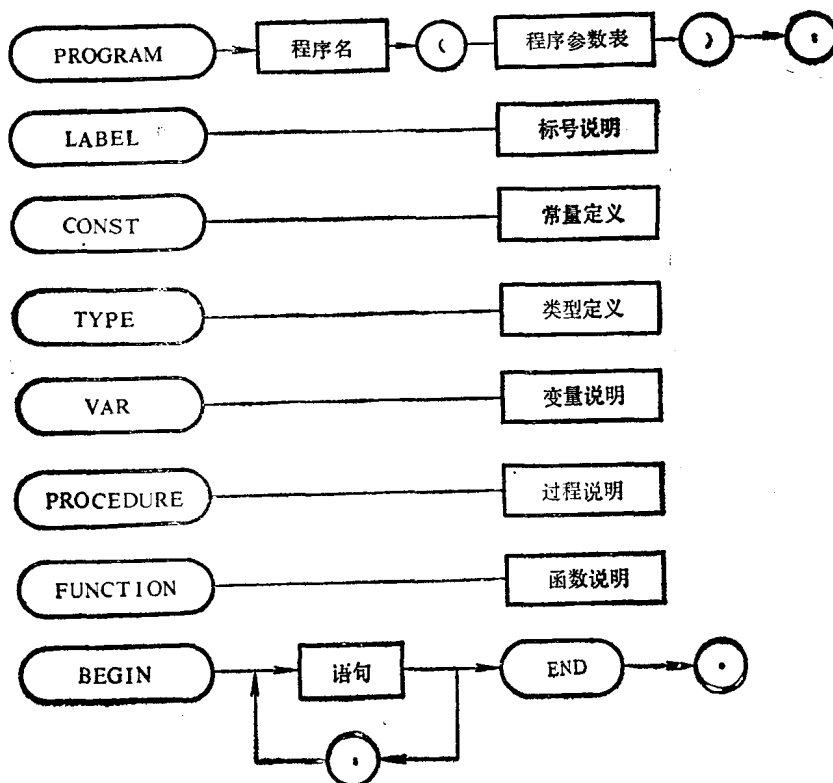


图1-3 程序结构语法图

1. 程序首部

程序首部是保留字PROGRAM开头，它给出程序名 Example，以及程序的参数在圆括号内，(input, output)。程序参数用来表示程序与外界的联系。它从标准文件 input 读出，把结果送到标准文件 output 中。文件的标识符 input 和 output 都是与具体运行环境有关的。例如，在微型计算机中 input 可以是键盘，而 output 是显示屏。通过在程序首部指定附加文件的方法，程序可以存取这些文件。

2. 程序说明部分

PASCAL 语言本身定义了一些标准的常量，标准的数据类型，标准的过程和函数，可以直接使用。而用户自己定义的标号、常量、类型、变量、过程和函数等都必须要在程序的说明部分加以说明，然后才能在程序的语句部分引用。

对于程序的说明部分，在说明时应遵循如下次序，

- ①〈标号说明部分〉；
- ②〈常量定义部分〉；
- ③〈类型定义部分〉；
- ④〈变量说明部分〉；
- ⑤〈过程与函数说明部分〉。

上述简单程序例中未出现标号、常量、类型、过程和函数的说明，仅有变量说明部