

计算机教育丛书



著名计算机教育家
谭浩强教授 主编

全国高等院校计算机
基础教育研究会 联合推出
电子工业出版社

个人电脑系列

学用BASIC语言

杨丽华 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

全国高校计算机基础教育研究会 联合推出
电子工业出版社
计算机教育丛书 个人电脑系列

谭浩强 主编

学用 BASIC 语言

杨丽华 编著

电子工业出版社

内 容 提 要

你想在个人电脑上自己动手编程序,解决问题吗?本书将通过具体的实例,通俗的讲解,让您学会编程序,并上机运行。本书包括的内容有: BASIC 语言的产生与发展, BASIC 语言的初步知识, BASIC 的基本语句, 怎样运行 BASIC 程序, 顺序、选择、循环三种结构的程序设计, 数组, 函数与子程序, 字符串处理, 基本的图形功能, 有关文件的基本知识等。本书力求做到: 由浅入深、叙述清楚、通俗易懂。通过阅读本书, 使读者不仅学会编程, 而且对 BASIC 语言的基本内容有一个完整的了解。本书介绍的 MS - BASIC 可以在 XT、286 及以上的所有 IBM PC 及兼容机上运行。

JS563/10

计算机教育丛书

个人电脑系列

谭浩强 主编

学用 BASIC 语言

杨丽华 编著

责任编辑 张新华 陈碧凤

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京大中印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 10 字数: 238 千字

1996 年 11 月第 1 版 1996 年 11 月第 1 次印刷

印数: 10100 册 定价: 12 元

ISBN 7-5053-3879-X/TP·1668

《计算机教育丛书》序

90年代初,在我国出现了第二次计算机普及高潮。与80年代初出现的第一次计算机普及高潮相比,这次高潮具有全方位、多层次的特点。各行各业的人都迫切地要求学习计算机知识,掌握计算机的应用。计算机知识已成为当代知识分子知识结构中不可缺少的重要组成部分。计算机既是先进科学技术的结晶,又是大众化的工具。这个特点只有计算机才具备。

过去,计算机只能为少数人所掌握,今天我们要向全中国千百万人民群众普及计算机知识。我们的目标是:把计算机从少数专家手中解放出来,使之成为广大群众手中的工具。我们要破除对计算机的神秘感。实践表明:高中以上文化程度的人,能够学会计算机的初步操作和应用。

当然,计算机的应用是分层次的,不同的人在不同的层次上使用着计算机。计算机科学技术内容极为丰富,浩如翰海,它的发展又极为迅速。要在短时期内全部、深入地掌握计算机的知识和应用,几乎是不可能的。我们必须循序渐进、由浅入深,逐步提高。我们说,入门不算难,提高需要下功夫。

对各行各业学习计算机的人员来说,学习计算机的目的是为了应用。应当强调:以应用为目的,以应用为出发点,根据不同工作岗位的特点,需要什么就学什么。实践证明,从学习计算机的应用入手,是学习计算机知识的捷径。

普及计算机教育需要有适用的教材和参考用书。他们应当百花齐放,风格各异,让读者在琳琅满目的书架上能找到自己所需要的书。几年前,我们开始出版《计算机教育丛书》,根据读者的需要,陆续出版了十几本书(主要是供大学生用的教材),受到社会广大读者的欢迎。许多读者热情地鼓励我们扩展题材,区分层次,不拘一格,推动应用。我们愿意为推动计算机教育与普及贡献自己微薄之力。

本丛书的作者大多数是在各高等学校或研究单位工作、具有丰富教学和研究经验的专家教授,其中有的同志在我国计算机教育界享有盛名,颇有建树,并且编写过多种计算机书籍。本丛书的对象主要是计算机的初、中级应用人员和初学者。我们力图用通俗易懂的语言把复杂的计算机概念说清楚。

经过研究,本丛书暂定包括三个系列:①非计算机专业教材系列(由谭浩强负责);②个人电脑系列(由秦笃烈负责);③流行软件系列(由周山美负责)。以后将根据需要增加其他新的系列。

由于我们水平所限,加之计算机技术发展十分迅速,本丛书必然会有不足之处甚至会出现一些错误,诚恳地欢迎广大专家、读者提出意见。

本丛书的出版得到全国高等院校计算机基础教育研究会、奥斯克电脑图书中心、电子工业出版社、中国科学技术出版社的大力支持与帮助,在此表示感谢。

《计算机教育丛书》主编

谭浩强

1996年3月

前 言

90年代出现了全国性第二次普及计算机知识的热潮,无论广度、深度都大大超过了80年代出现的第一次普及高潮。人们已开始把计算机作为人类的第二文化,它是人们进入信息时代不可缺少的工具。为满足广大读者特别是那些具有初中文化程度的人们学习计算机知识的需要,由谭浩强教授担任主编,编写出版一套“计算机教育丛书”,《学用 BASIC 语言》是其中的一本。

BASIC 语言是一种适合于初学者的计算机语言。它不仅简单、易学,而且还具有较广的应用范围。它适用于数值计算、事务管理,尤其适合于面向初学者的计算机教育。过去十几年来,许多读者是从学习 BASIC 语言开始,进而更深入地学习计算机知识,迈入计算机应用的大门的。BASIC 语言是国内外广泛使用的计算机高级语言,对初学者有很大的吸引力,至今仍是初学者学习程序设计的良好语言。因此,“计算机教育丛书”收入了《学用 BASIC 语言》一书。

本书在编写时,针对初学者的特点,力求做到基本概念叙述清楚,通俗易懂,由浅入深,通过举例帮助读者理解和掌握所学内容。

本书由谭浩强教授主审,他对本书的编写给予了具体指导,并提出许多宝贵意见。秦笃烈教授对本书的编写也给予了大力的帮助与支持。在此表示诚挚的感谢。参与本书编写工作的有王实、常毅、李华等。

由于作者水平有限,时间仓促,错误和不妥之处在所难免,敬请读者批评指正。

作者

1996年1月

目 录

第一章 BASIC 语言的发展	(1)
1.1 BASIC 语言的诞生	(2)
1.2 BASIC 语言的发展	(2)
1.3 结构化程序设计	(5)
第二章 BASIC 语言的初步知识	(9)
2.1 什么是 BASIC 程序	(10)
2.2 常量与变量	(11)
2.2.1 常量	(11)
2.2.2 变量	(11)
2.3 表达式	(12)
2.3.1 算术运算符	(12)
2.3.2 算术表达式	(12)
2.3.3 运算规则	(13)
2.4 流程图	(13)
2.5 标准函数	(14)
动脑筋思考	(16)
第三章 编写简单程序	(17)
3.1 LET 语句	(18)
3.2 PRINT 语句和 LPRINT 语句	(18)
3.2.1 PRINT 语句的一般格式	(19)
3.2.2 PRINT 语句的输出格式	(20)
3.3 INPUT 语句	(22)
3.4 READ 语句和 DATA 语句	(24)
3.5 RESTORE 语句	(25)
3.6 END 语句和 STOP 语句	(27)
3.6.1 END 语句	(27)
3.6.2 STOP 语句	(28)
3.7 REM 语句	(28)
3.8 应用举例	(29)
3.9 怎样输入、修改、运行一个 BASIC 程序	(30)
3.9.1 在计算机硬盘上建立 BASIC 系统	(30)
3.9.2 进入、退出 BASIC 系统	(30)
3.9.3 输入、修改、运行 BASIC 程序	(31)

动脑筋思考	(34)
第四章 编写有选择功能的程序	(37)
4.1 关系表达式和逻辑表达式	(38)
4.1.1 关系运算符与关系表达式	(38)
4.1.2 逻辑运算符与逻辑表达式	(38)
4.2 IF 语句和 IF 语句的嵌套	(39)
4.2.1 IF 语句	(39)
4.2.2 IF 语句的嵌套	(42)
4.3 GOTO 语句	(42)
4.3.1 GOTO 语句	(42)
4.3.2 带 GOTO 语句的 IF 语句	(43)
4.4 ON-GOTO 语句	(45)
动脑筋思考	(49)
第五章 编写有循环功能的程序	(51)
5.1 FOR-NEXT 语句	(52)
5.1.1 FOR-NEXT 语句的格式	(52)
5.1.2 FOR-NEXT 语句的执行过程	(52)
5.1.3 关于 FOR-NEXT 的几点说明	(53)
5.2 多重循环	(55)
5.2.1 举例介绍多重循环的概念和应用	(55)
5.2.2 多重循环的执行过程	(57)
5.2.3 关于多重循环的几点说明	(57)
5.2.4 多重循环应用举例	(58)
5.3 WHILE 语句和 WEND 语句	(59)
5.3.1 语句格式	(60)
5.3.2 循环的执行过程	(60)
5.4 应用举例	(61)
动脑筋思考	(65)
第六章 处理大量同类数据的方法——数组	(67)
6.1 什么是数组	(68)
6.1.1 数组与数组元素	(68)
6.1.2 只有一个下标变量的数组	(68)
6.1.3 有两个下标变量的数组	(68)
6.2 数组说明语句(DIM 语句)	(69)
6.2.1 DIM 语句的格式	(69)
6.2.2 DIM 语句的功能	(69)
6.3 数组操作	(70)
6.3.1 数组元素的赋值和输出	(70)
6.3.2 数组的操作	(72)

6.4 应用举例	(75)
动脑筋思考	(79)
第七章 自定义函数和子程序	(81)
7.1 自定义函数	(82)
7.1.1 自定义函数语句格式	(82)
7.1.2 自定义函数的使用规则	(82)
7.1.3 应用举例	(83)
7.2 子程序	(84)
7.2.1 GOSUB 语句和 RETURN 语句	(84)
7.2.2 GOSUB 语句和 RETURN 语句的使用规则	(87)
7.2.3 子程序的嵌套	(88)
7.2.4 ON-GOSUB 语句	(89)
动脑筋思考	(90)
第八章 处理字符型数据	(93)
8.1 字符串变量	(94)
8.2 字符串的输入和输出	(94)
8.2.1 字符串的输入	(94)
8.2.2 字符串的输出	(98)
8.3 字符串的运算	(98)
8.3.1 字符串的连接	(98)
8.3.2 字符串的比较	(98)
8.4 字符串函数	(100)
8.4.1 LEN 函数(求字符串长度的函数)	(100)
8.4.2 VAL 函数(将字符转换成数值的函数)	(101)
8.4.3 STR\$ 函数(将数值转换成字符的函数)	(101)
8.4.4 ASC 函数(求 ASCII 码函数)	(101)
8.4.5 CHR\$ 函数	(102)
8.4.6 LEFT\$ 函数	(102)
8.4.7 RIGHT\$ 函数	(103)
8.4.8 MID\$ 函数	(103)
8.4.9 STRING\$ 函数	(104)
8.4.10 INSTR 函数	(104)
动脑筋思考	(106)
第九章 用 BASIC 作图作曲	(109)
9.1 BASIC 作图	(110)
9.1.1 屏幕显示方式	(110)
9.1.2 屏幕控制语句	(110)
9.1.3 COLOR 语句	(113)
9.1.4 画直线图形	(114)

9.1.5 画圆和椭圆	(119)
9.1.6 图形着色	(122)
9.2 BASIC 作曲	(124)
9.2.1 BEEP 语句	(124)
9.2.2 PLAY 语句	(124)
动脑筋思考	(127)
第十章 保存程序和数据	(129)
10.1 什么是文件	(130)
10.2 源程序文件	(130)
10.3 数据文件简介	(131)
动脑筋思考	(136)
附录	(137)
附录 A 常用字符与 ASCII 代码对照表	(138)
附录 B MS BASIC 语句和函数一览表	(139)
附录 C MS BASIC 出错信息	(144)
附录 D PRINT USING 语句	(146)
参考文献	(147)

第一章

BASIC 语言的发展

- BASIC 语言的诞生
- BASIC 语言的四个发展阶段

1.1 BASIC 语言的诞生

BASIC 语言诞生已经 30 多年了。30 多年来, BASIC 语言不断发展和改进, 从非结构化的语言发展成为结构化的语言, 并朝着面向对象的方向发展。

BASIC 语言是 1964 年 5 月 1 日面世的, 她的创始人是美国 Dartmouth 大学的约翰·凯梅尼 (John G. Kemeny) 和托马斯·卡茨 (Thomas E. Kurtz) 两位教授。说到 BASIC 的产生可追溯到 1956 年。当时计算机只能识别机器语言, 它的指令只能由 0 和 1 组成, 对应于电磁设置的“关”和“开”。为了能使用户更好地使用计算机, 后来生产厂家提供了“汇编语言”, 它使用助记符代替由 0 和 1 组成的代码, 但汇编语言仍是难学难用, 对用户来说还是不方便。凯梅尼和卡茨当时就有一个共同的想法, 要创造一种人们容易学会的语言。

1954 年 FORTRAN 语言的出现, 是计算机语言的一大突破。它允许人们用类似于英语单词和数学公式的语言来编写程序, 该语言是为数值计算而设计的, 特别适合于工程师使用。事实上它是为专家们使用计算机而着想的, 正像凯梅尼和卡茨所说: “在你写第一个 FORTRAN 程序前, 要先接受两个月的培训是很普遍的事。选择它不能较好地达到我们的目的。我们要一个适合非专业性读者的语言。”凯梅尼和卡茨的另一个想法是要让 Dartmouth 大学的学生都会使用计算机。在当时计算机还不具备向所有学生开放条件下, 计算机是成批处理程序的, 即一组程序一个接一个地加以处理。即使是一个简单的程序也要等待一批程序一起处理, 有可能要等好几天才能知道它是否成功。在这种条件下, 让一个学生独占一台计算机来学习如何编写程序是不可想象的。于是一种新的计算机语言的设计思想产生了, 凯梅尼和卡茨认为: 这种语言应具备如下特征: 使初学者容易学习; 应该是通用语言, 可以编写任何程序; 在与计算机进行交互作用中, 给用户提供更多的方便; 用户不必了解更多的硬件知识; 对于小的程序应能很快地响应; 还能给用户提供更清晰的出错信息。

根据以上设计思想, 凯梅尼和卡茨设计了 BASIC 语言。BASIC 是 Beginner's All-purpose Symbolic Instruction Code 的缩写, 译成中文是“初学者通用符号指令代码”。“使初学者容易学习”是设计 BASIC 语言时的重要原则。BASIC 语言吸收了当时流行的 FORTRAN 语言与 COBOL 语言的优点, 并创出了自己的特色。例如, 在交互作用方面, 以前的计算机语言都是为批处理而设计的, BASIC 则是为交互作用计算而设计的, 这使得 BASIC 语言与用户更加友好。1964 年 5 月 1 日与 BASIC 语言同时诞生的还有“分时系统”。凯梅尼和卡茨曾经说过, 在分时系统中, 最重要的交互作用是使用 INPUT 语句。在执行 INPUT 语句时, 要求用户输入的程序暂时中断, 以便提供有关的信息, 然后程序继续运行。这种用户与计算机的交互作用在批处理系统中是不可能的。

在 BASIC 交互系统中, 计算机能对用户的输入进行快速响应, 并在发现程序有错时, 给出有关错误的信息。用户可以随时修改自己的程序, 非常方便。

1.2 BASIC 语言的发展

从 BASIC 语言诞生到现在, BASIC 经历了四个发展阶段, 前后出现了四代 BASIC。

1. 初期的 BASIC 语言(1964~70 年代初)

由于当时计算机硬件条件的限制(主要是内存容量小),BASIC 语言被设计成一种小型的语言,功能相对较弱。1964 年 5 月 1 日诞生的 BASIC 语言只有 14 个语句。

LET、PRINT、END、FOR、NEXT、GOTO、IF THEN、READ、DATA、DIM、DEF、GO-SUB、RETURN 和 REM,真可谓小型、简单、易学。这个版本仅包含数值常量和数值变量。使用 DEF 语句(自定义函数语句)时,只允许定义单行函数,函数的名字从 FNA 到 FNL;GO-SUB 和 RETURN 语句提供了内部子程序的早期形式;变量名的基本部分仅限于单个字母或单个字母后跟一个数字;这时期的 BASIC 虽然功能较弱,但它简单易学的优点,却显示出强大的生命力,很快冲出了课堂,成为一种实用的计算机语言而被迅速地推广开来,应用于社会生活的各个领域。

她作为一种广泛使用的语言,表现了一定的稳定性。并且在多方面有了进一步的发展:例如:

(1)打印语句的功能增强了,增加了紧凑输出格式(输出之间用分号分隔),而且还引入了 PRINT USING 语句。

(2)输入语句也得到发展,引入了 LINPUT 语句,它允许从终端输入整个一行,在这一行中可以包括逗号和头尾空格。LINPUT 语句就是现在标准 BASIC 中的 LINE INPUT 语句。

(3)在矩阵运算中增加了 MAT 语句。

(4)增强了字符串处理的功能。凯梅尼和卡茨为 BASIC 增加了字符串运算和若干个串函数。

(5)增加了文件。当用户的程序比较小,且参加运算的数值或字符串的数目都比较少时,DATA 语句是够用的,但有时虽然程序比较简单,而数值或字符串的数量却很大,这时程序会因为需要很多 DATA 语句变得相当庞大。如果把数据从程序中分离出来,保存在文件里,就会使程序变得很短。这样就引入了两类文件:终端格式文件和随机访问文件。终端格式文件目前称为显示格式文件、字符文件或文本文件。随机访问文件是内部格式文件。

经过多年的努力,早期的不足得到改进,使 BASIC 版本不断发生变化。运行速度加快,能让大量学生在分时系统上同时进行编程、编译和调试。BASIC 第 6 版能联机调试程序,它还能使较大的程序自动调入覆盖的子程序,而不必将需要的每个子程序都包括在一个程序内。所有这一切都表明 BASIC 语言取得了巨大成功。

2. 第二代 BASIC—微机 BASIC 的产生和发展(70 年代中~80 年代中)

1975 年微机的出现,是计算机发展过程中的一次革命性的事件,它使 BASIC 获得重大的发展。

最初的微机存储器很小,例如 MITS 公司的 MITSALTAIR 只有 4K 字节的存储器,只能使用机器语言。在此情况下,现任微软公司(Microsoft)总裁的比尔·盖茨(Bill Gates)看到了为小存储器的微机配上 BASIC 语言的机会,于是就着手进行此项工作。他最早为微机配上了 BASIC 语言,从而使 BASIC 语言得到迅速推广。紧跟比尔·盖茨之后,很多厂商纷纷为其他牌子的微机配上 BASIC 语言。由于其他高级语言规模较大,无法在微机上运行,BASIC 成了与微机天然匹配的语言。在这期间,先后出现了 Cromemco BASIC, Northstar BASIC, TRS-80 BASIC, Apple I BASIC, GW BASIC, MS BASIC(BASICA)等。每一个厂商在

推出其 BASIC 新版本时都极力扩充了 BASIC 的功能。随着计算机硬件的发展, BASIC 不断扩充功能, 由只能处理小程序发展成能处理较大的程序。许多微机上的应用软件是用 BASIC 语言写的。这个时期 BASIC 获得空前的发展, 成为广大程序设计人员手中的有力工具。

但是, 在竞相发展 BASIC 的过程中, 各种不同的 BASIC 版本林立, 互不通用, 这使 BASIC 具有严重的“方言性”, 他们扩展的一些特性过多地依赖于硬件, 从而违背了“通用性”原则。另外, BASIC 语言开始的时候是对源程序进行编译, 而移植到微机上时就改为解释方式。解释方式的运行速度比编译方式慢很多, 不过它可以使用户与计算机进行人机对话, 对初学者是很合适的。

人们开始寻求 BASIC 语言的标准化。1974 年就成立了 X3J2 委员会, 两年后产生了一个小标准, 即 Minimal BASIC, 于 1978 年得到正式批准。X3J2 接着开始制订全面的标准 full BASIC。直到 1987 年正式通过, 使 BASIC 成为美国国家标准化语言之一 (American National Standard for the Programming Language)。

在这期间, 程序设计领域出现了一场革命, 即提出了“结构化程序设计方法”。著名计算机科学家迪克斯特拉早在 1965 年他的一篇论文中就指出: 调试与修改程序的困难和程序中的 GOTO 语句数成正比。也就是说带 GOTO 语句的非结构化程序的调试和维修是困难的。

在结构化程序设计方法提出之后, 老的 BASIC 版本显然不能完全适应这种要求了。

3. 第三代 BASIC—结构化的 BASIC (80 年代中~80 年代末)

1985 年凯梅尼和卡茨推出了 True BASIC。它与这之前的任何一个版本的 BASIC 相比, 可以说有很大的飞跃, 因为它是完全结构化的语言。专家们称它为第三代 BASIC。第三代 BASIC 必须是结构化语言。

结构化的 BASIC 语言吸取了 C 语言和 Modula I 的优点。True BASIC 保持了原 BASIC 语言“小型、通用、会话、易学”的特点。与其他高级语言相比, 它具有语言功能强、语法规则自然、程序结构清晰的特点, 而且具有良好的操作环境。与 True BASIC 相继出现的还有 Turbo BASIC, Quick BASIC 和 QBASIC。它们都属于第三代 BASIC, 均为菜单操作。特别是 Quick BASIC 4.0 和 4.5 版比 Turbo BASIC 更完善, 它大大提高了 BASIC 语言的层次与功能。

近年来, Microsoft 公司推出了 Quick BASIC 的高级子集 QBASIC, 它具有 Quick BASIC 的主要功能, 但采取解释执行方式, 采用菜单驱动, 使用方便灵活, 被认为是 BASIC 和 GW-BASIC 的较佳的更新版本。它作为 DOS 5.0 系统的一个组成部分提供给用户, 因此用户不必单独购买 QBASIC 软件。QBASIC 推出的几年来在国内外迅速得到推广, 是很有前途的, 适合于初学者的“大众语言”。它具有几下特点:

- (1) 扩充了数据类型 (除整型、实型、双精度型、字符串型外, 增加了长整型、定长字符串型变量)。
- (2) 提供了新的选择结构 IF-ELSE-END IF 和多分支选择 SELECT CASE 结构。
- (3) 改进了循环结构, 除了 FOR-NEXT 循环外, 还增加了 WHILE 循环 (WHILE WEND 结构) 和 DO 循环 (DO WHILE-LOOP, DO UNTIL-LOOP 结构等)。
- (4) 子程序和函数作为单独的模块。区分全局变量和局部变量。
- (5) 可以不设行号, 也可以使用行号 (为与第二代 BASIC 兼容), 但行号与执行顺序无关。
- (6) 可以实现递归。

- (7)具有良好的编辑环境,界面友好,可以使用键盘和鼠标。
- (8)提供“联机帮助”,在编辑过程中可以随时查询“帮助信息”。
- (9)提供“分步执行”和“跟踪”等调试工具。
- (10)采用先进的解释程序,使程序运行比一般 BASIC 快很多。

4. 第四代 BASIC——Windows 环境下的 BASIC (90 年代初~)

80 年代中期,在 Microsoft 公司推出 Windows 并取得巨大成功后不久,世界上出现了一批在 Windows 环境下运行的 BASIC,可以说这是第四代 BASIC。所谓第四代 BASIC,其标志是:

- (1)以图形(GUI)方式取代了 BASIC 的字符方式;
- (2)能够在 Windows 环境下运行。

例如,Visual BASIC、GFA-BASIC 以及 CA-REALIZER 都属于第四代 BASIC。Visual BASIC 是第一个 Windows 程序开发环境。人们能用它写 Windows 应用程序,对于开发具有实际应用价值的应用程序,提供了很大方便。

1991 年 Microsoft 公司正式推出 Visual BASIC 1.0,1995 Visual BASIC 已推出第四版,其专业水平版本增加了许多项新功能。由于 Visual BASIC 的编程风格与 Quick BASIC 极为相似,许多专业软件公司竞相推出基于 Visual BASIC 的软件包。例如,Sheridan 软件系统公司推出基于 Visual BASIC 的 Windows 应用开发工具 VB Assist 3.0。还有许多基于 Visual BASIC 开发的软件,这里不一一列举。1994 年 Microsoft 公司总裁比尔·盖茨访华时,在回答有关 BASIC 的现状和发展等问题时曾讲到:“Visual BASIC 是能迎接计算机程序设计挑战的最佳例子。”可见第四代 BASIC 的出现,再次展示了 BASIC 语言的生命力。BASIC 语言在过去计算机应用及普及计算机知识方面起到了重要作用,今后还会继续发挥作用。

本书主要介绍 MS BASIC(即 BASICA),为什么有了第四代 BASIC 后,我们还要介绍 MS BASIC 呢?首先,因为这本书是为非专业的、具有中等文化水平的初学者所写,MS BASIC 的英文保留字较少,对英文水平不高的人学习障碍少一些。第二,MS BASIC 可以在 XT 和 286 以上所有 PC 机及其兼容机上运行,这样拥有中华学习机、XT 等计算机的用户,还可以在自己的机器上使用 BASIC 语言。第三,MS BASIC 可以用来完成科学计算、事务管理和计算机辅助教学等任务,具有实用价值。有一些用 MS BASIC 编写的程序和程序库到目前还在继续使用。第四,用 MS BASIC 编写的程序可以在 Quick BASIC 环境下运行。

本书主要讲解 BASIC 的最基本的知识,读者在学习了本书后,具有了程序设计的初步基础,若需要还可以进一步学习其他语言。

1.3 结构化程序设计

前面我们已经讲过第三代、第四代 BASIC 语言都是结构化的程序设计语言。那么,什么是结构化程序设计呢?下面做一简单介绍。

结构化程序设计是荷兰著名学者 E. W. Dijkstra 等人在 60 年代末提出的一种开创性的程序设计方法,是为了对付当时出现的软件危机而提出的一套程序设计原则和方法。按照这种方法设计的程序结构清晰,易于阅读、修改、证明。

结构化程序设计强调程序设计风格和程序结构的规范化。它的基本特点是结构和思路

清晰。

1. 结构清晰

所谓结构清晰就是使程序模块化；基本单元要采用三种基本结构。三种基本结构是：顺序结构、选择结构、循环结构。

(1)顺序结构 顺序结构是一种最简单、最常用的基本结构。在该结构中，各操作块（也就是程序段）按照各自出现的先后顺序，依次逐块执行。

(2)选择结构 选择结构也是一种常用的基本结构，用它能构成两个或两个以上的选择执行路径——分支。按其所构成的选择执行路径的多寡，又可分为双分支选择结构和多分支选择结构。选择结构的特点是，根据给定的条件成立与否，来决定选择执行某一相应的操作。图 1.1 表示一个双分支选择结构，即根据给定的条件 p 是否成立决定执行 A （条件 p 成立）或执行 B （条件 p 不成立）。

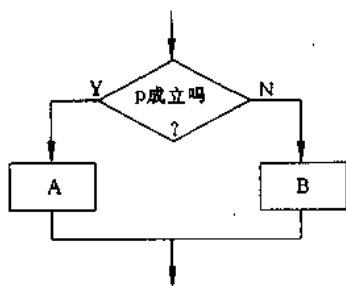


图 1.1

(3)循环结构 循环结构是一种重要的基本结构，按其循环体执行的条件可分为“当型循环”结构和“直到型循环”结构。这两者的主要区别是，前者先判断条件是否满足，再决定是否执行循环体，执行次数大于或等于零；后者是先执行循环体，再判断条件是否满足，执行循环体的次数大于等于 1。图 1.2(a)，(b) 分别表示“当型循环”和“直到型循环”。先看图(a)，当给定的条件 $p1$ 成立时，执行 A 操作，执行完 A 之后再检查 $p1$ 是否成立，若成立，再执行 A ，如此循环直到 $p1$ 不成立不再执行 A 为止。这种结

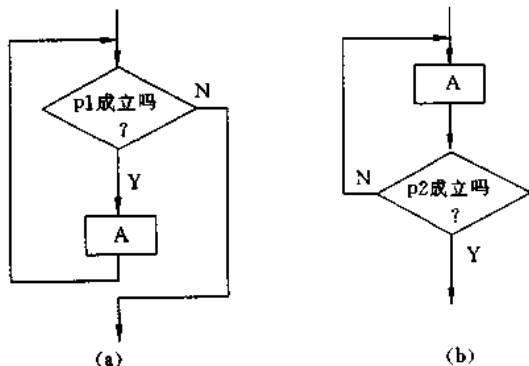


图 1.2

构称为“当型循环”。再看图(b)，它是先执行 A 之后再检查条件 $p2$ 是否成立，若不成立就再执行 A ，之后再检查 $p2$ 是否成立，如此循环，直到 $p2$ 成立为止。这种结构称为“直到型循环”。

2. 思路清晰

所谓思路清晰，就是采用自顶向下、逐步细化的方法。

结构化程序设计方法自从被提出以来，在实践中日益完善，并普遍为人们所接受。

结构化程序设计方法主要有以下内容：

(1)自顶向下、逐步细化

自顶向下、逐步细化的设计方法是结构化程序设计的核心。其步骤是：先分析题目，由总体入手，可先不考虑细节；然后将问题逐步分解为几个部分，若某一部分还是太复杂，还可以再分解，直到分解为上述的基本结构为止。

(2) 模块化

所谓模块化，就是将一个大程序划分为若干个功能相对独立的程序模块，每个模块还可以再分解为若干个子模块，由主程序定好各模块应实现的目标及各模块间的接口。因此各模块可以分别独立地进行设计、编程、调试、维护，而不受其他模块的影响。各个程序模块只有一个入口点和一个出口点。

3. 结构化程序设计

下面我们对计算某工厂职工的平均工资这样一个不太复杂的问题的开发过程做一简单分析，以说明结构化程序设计的实际应用。初学者开始可能不易理解，那么可以先了解开发过程的大致步骤，因此，我们不讲解具体算法及编程，待学了一些基本的程序设计知识后，再看这些内容就很容易理解了。

(1) 对给出的任务进行分析

分析给定的数据、要完成的功能及对输出的要求，并把它们用文字明确地表述出来。本例中给定的数据就是每个职工的工资，要求计算出全厂的平均工资，并打印出来。

(2) 根据要求，用简明的文字写出程序最后应完成的目标

在本例中，“统计全厂职工的平均工资”，这是一个总任务目标，没有具体给出应包含哪些操作，实际上是“顶层设计”，见图 1.3 中的顶层。

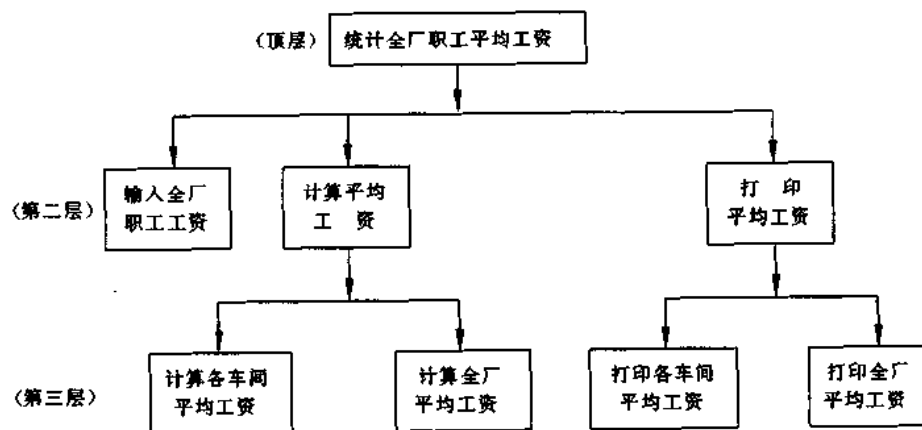


图 1.3

(3) 将总任务分解为若干子任务

也就是划分模块，如图 1.3 中的第二层。即输入全厂职工的工资、根据输入数据计算出平均工资、打印输出，它比顶层具体了一些，如果实现了第二层的子功能，就能实现顶层的总功能。

(4) 再向下分解

如图 1.3 中的第三层，将子功能再分解成更小的子功能。

首先计算出各车间的平均工资，再计算全厂的平均工资；然后先打印各车间的平均工资，再打印全厂的平均工资。这个过程一直继续下去，直到每个子功能不需要再分解为止。这

就是“逐步细化”。

在模块划分中,要使每一个子功能单独构成一个模块,即不要把几个不同的功能放到一个模块中,以增强模块的相对独立性。另外,模块不宜过大,在程序中一般不超过 50~60 行。

(5)根据模块的功能要求,写出每一块的算法。

(6)在算法设计的同时,完成数据结构的设计。

即设计出各模块中使用的数据结构。

(7)按照算法和所设计的数据结构,用计算机语言进行结构化编程。

(8)调试程序,保证程序正确。

从以上过程可以看出:自顶向下,逐步细化;模块化和结构化编程是贯穿于程序开发过程中的。这样编写的程序就是结构化的程序。