

高等学校教材

BASIC 语言

孙振业

国防工业出版社

出版说明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从 1978 年至 1985 年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材”的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了 1986~1990 年的“七五”（第三轮）教材编审出版规划。列入规划的教材、实验教材、教学参考书等近 400 种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位，广大教师和同学积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

前 言

本教材是按电子工业部制定的工科电子类专业教材 1986~1990 年编审出版规划,由中专电子类教材编审委员会计算机编审小组组织征稿、评选,并推荐出版的。

本教材由北京无线电工业学校孙振业编写,北京无线电工业学校高级讲师曾玉昆担任主审,责任编委周岳山。编审者均依据中专计算机教材编审小组审定的编写大纲进行编写和审阅。

本课程的参考学时数为 60 学时,其主要内容是以 IBM-PC 微型计算机和 APPLE-Ⅱ 微型计算机为背景,介绍 BASIC 语言程序设计的基本概念和方法。具体有以下几部分:电子计算机的基础知识、BASIC 简单程序设计、分支程序设计、循环程序设计、子程序设计,此外还有数组、字符串、文件的使用以及计算机绘图,最后介绍了 IBM-PC BASIC 语言和 APPLE-Ⅱ BASIC 语言的上机操作过程。本教材在注重介绍 BASIC 语言的定义、概念和使用规则的同时,突出介绍了程序设计的技巧和方法。为了使读者尽快掌握编程序的方法,本教材还介绍了程序设计的一般步骤和程序流程图的使用,并配置了大量的程序例题和习题,供读者参考。

本教材在修改过程中得到主审曾玉昆同志的精心指导与帮助,参加审阅的周岳山同志也为本教材提出许多宝贵意见,这里一并表示诚挚的感谢。由于编者水平有限,教材中难免还存在一些缺点和错误,殷切希望广大读者批评指正。

目 录

第一章 概述	1
1.1 计算机系统硬件	1
1.2 计算机语言	4
1.3 计算机系统软件	6
1.4 BASIC 语言的特点	6
习题一	7
第二章 BASIC 语言的词法及程序的构成	8
2.1 BASIC 程序的构成	8
2.2 基本字符	10
2.3 常量	11
2.4 变量	14
2.5 函数	15
2.6 表达式	17
2.7 程序设计的一般步骤	23
习题二	24
第三章 简单的 BASIC 程序	26
3.1 赋值语句	26
3.2 打印语句	30
3.3 键盘输入语句	43
3.4 读数/置数语句	45
3.5 恢复数据区语句	47
3.6 结束语句	49
3.7 简单程序举例	49
习题三	52
第四章 分支程序	54
4.1 无条件转移语句	54
4.2 条件语句	55
4.3 开关语句	58
4.4 交换语句	60
4.5 程序举例	62
4.6 注释语句和暂停语句	67
习题四	69
第五章 循环	71
5.1 循环语句	71
5.2 多重循环	77
5.3 程序举例	83

习题五	88
第六章 数组	90
6.1 数组的概念	90
6.2 数组说明语句	91
6.3 数组操作	93
6.4 程序举例	99
习题六	105
第七章 BASIC 函数的应用	107
7.1 标准函数	107
7.2 自定义函数	118
习题七	121
第八章 子程序	122
8.1 子程序的概念	122
8.2 转子语句和返回语句	124
8.3 程序举例	127
习题八	136
第九章 字符串变量	137
9.1 基本概念	137
9.2 字符串函数	140
9.3 字符串的连接和比较	148
9.4 中文输入方式简介	151
习题九	158
第十章 文件	159
10.1 文件的概念	159
10.2 有关文件的常用命令	161
10.3 顺序文件	165
10.4 随机文件	173
10.5 应用举例	179
10.6 APPLE-Ⅱ BASIC 语言的文件介绍	183
习题十	192
第十一章 图形的绘制	194
11.1 显示方式的选择	194
11.2 画图语句介绍	201
11.3 动画技术	210
11.4 APPLE-Ⅱ BASIC 语言的绘图简介	221
附录	228
附录一 ASCII 字符代码表	228
附录二 BASIC 语言的保留字表	235
附录三 IBM-PC BASIC 语言的上机操作	238
附录四 APPLE-Ⅱ BASIC 语言的上机操作	244

第一章 概 述

电子计算机是一种能够高速、精确、自动地进行数值运算,以及对各种信息进行收集、存储和处理的电子机器。

1946 年在美国宾夕法尼亚大学诞生了世界上第一台电子计算机“ENIAC”。它使用了一万八千多只电子管、一千五百多只继电器和许多其它电器元件。ENIAC 与现代的计算机相比可算是一个庞然大物,它占地一百七十多平方米,总重量达一百三十吨,耗电一百五十千瓦。ENIAC 每秒钟可进行五千次加法运算。

从 ENIAC 问世至今的四十多年中,电子计算机的制造和应用技术得到了迅猛的发展。电子计算机在制造技术方面,每五~八年运算速度就提高十倍,而体积却缩小到十分之一,成本也下降到十分之一。从电子计算机所使用的电器元件方面来说,它已经历了以电子管、晶体管、中小规模集成电路以及大规模集成电路为标志的四次更新换代。电子计算机的这种发展速度远远超过了任何其它工业技术的发展速度。与此同时,电子计算机的应用技术也发展的非常迅速,特别是出现了微型计算机之后,使电子计算机的应用范围扩展到了社会的各行各业、各个领域,加快了整个社会的发展进程。

从构造和运算原理上讲,电子计算机可分为电子数字计算机和电子模拟计算机两大类,从用途上讲,电子计算机又可分为通用电子计算机和专用电子计算机两大类。纵观整个计算机领域,以电子数字通用计算机的发展最为迅速,应用最为普遍。因此,人们所讲的电子计算机(或简称计算机),均指电子数字通用计算机。

电子计算机有以下几个特点:

(1) 运算速度快。普通微型计算机的运算速度已达到每秒几万次、几十万次;巨型计算机的运算速度则可达每秒几亿次、十几亿次。

(2) 计算精度高。一般的计算机可以有十几位有效数字。如果降低运算速度,有效位数还可以增加。

(3) 具有“记忆”和逻辑判断的能力。计算机不仅能进行数值计算,还可以把原始数据、中间结果、运算程序等信息存储起来,以备使用。计算机还可以进行各种逻辑判断,并根据判断的结果做出相应的后续处理。此外,计算机还具有绘图、发声等功能。

(4) 计算机内部的操作运算是自动控制的。用户(计算机的使用者)把程序送入计算机之后,计算机逐行执行程序直到打印出最后结果,不需要人工干预。

1.1 计算机系统硬件

计算机系统由硬件和软件两大部分组成。

硬件是由电子、电磁、光电器件和机械装置构成的计算机设备的总称。它与计算机软件不同,是计算机系统的物质基础,也称为硬设备。

一、硬件的构成

计算机的硬件大体上由以下几个部分构成：控制器、运算器、内存储器、输入、输出设备。其中，前三者合称为主机，后两者又称为外部设备(简称为外设)。电子计算机的硬件结构如图 1-1 所示。

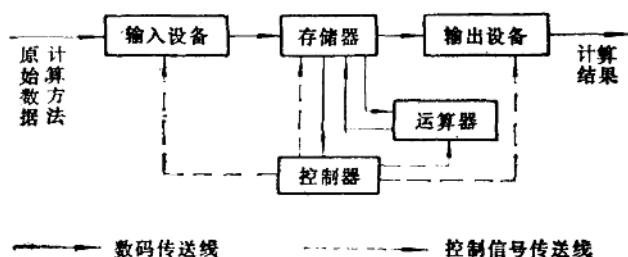


图1-1 电子计算机结构图

二、硬件各个部分的功能

1. 输入设备

输入设备是用户向计算机发布操作命令、输入计算程序和原始数据的设备。常用的输入设备是终端键盘。此外还有盒式磁带、磁盘、图形输入板、声音输入装置等。

2. 输出设备

输出设备是计算机用来输出计算结果或其它信息的设备。常用的输出设备有终端显示、打印机。此外还有盒式磁带、磁盘、绘图仪、声音输出装置等。

3. 存储器

存储器是计算机的“记忆装置”，它能保存和记载原始数据、处理程序、中间结果和最终结果。因此，存储器实际上是计算机的一个信息库。存储器一般分为内存储器和外存储器两大类。

(1) 内存储器。简称内存，是计算机的主存储器。在一台计算机中，内存储器与运算器、控制器直接相连，运算器中参与运算的操作数或运算结果，都要从内存储器中取出或送到内存储器中保存。因此，内存储器是计算机必不可少的部分。

(2) 外存储器。简称外存，是计算机的辅助存储器。外存储器主要有盒式磁带、磁盘等。

4. 运算器

运算器是执行运算的装置，它可以完成各种基本的算术运算和逻辑运算。

5. 控制器

控制器是发布命令的装置，是整个计算机的指挥部。它通过向计算机各个部分发出控制信号，来指挥计算机自动地、协调地进行工作。

控制器与运算器不论在逻辑关系上还是在结构上都有十分紧密的联系，同时它们又是数据加工处理的中心部件，所以把它们合称为“中央处理机”(即所谓的 CPU)。

三、计算机进行数据处理的基本原理

为了了解计算机进行数值计算的基本原理，首先看一下由人、计算器及纸和笔组成的系统进行计算的过程。

例如：计算 $1 + 2 + \frac{2^2}{1 \times 2} + \frac{2^3}{1 \times 2 \times 3}$

(1) 编制数值运算的操作步骤

第一步：计算 $1 + 2$

第二步：计算 2×2

第三步：计算 1×2

第四步：计算 $\frac{(2 \times 2)}{(1 \times 2)}$

第五步：计算 $(2 \times 2) \times 2$

第六步：计算 $(1 \times 2) \times 3$

第七步：计算 $\frac{(2 \times 2 \times 2)}{(1 \times 2 \times 3)}$

第八步：计算 $(1 + 2) + \frac{2^2}{1 \times 2}$

第九步：计算 $\left(1 + 2 + \frac{2^2}{1 \times 2}\right) + \frac{2^3}{1 \times 2 \times 3}$

以上计算步骤记录在纸上。

(2) 依次读出运算的操作步骤，操纵计算器完成具体计算，并将每次运算的结果记录在纸上。表 1-1 列出了具体的操作过程。

表 1-1 操作过程表

	读出操作步骤内容	执行操作	记录结果
(1)	$1 + 2$	加	3
(2)	2×2	乘	4
(3)	1×2	乘	2
(4)	$\frac{(2 \times 2)}{(1 \times 2)}$	除	2
(5)	$(2 \times 2) \times 2$	乘	8
(6)	$(1 \times 2) \times 3$	乘	6
(7)	$\frac{(2 \times 2 \times 2)}{(1 \times 2 \times 3)}$	除	1.333
(8)	$(1 + 2) + \left(\frac{2^2}{1 \times 2}\right)$	加	5
(9)	$\left(1 + 2 + \frac{2^2}{1 \times 2}\right) + \left(\frac{2^3}{1 \times 2 \times 3}\right)$	加	6.333

(3) 操作步骤执行完最后一步所得的结果是最后结果。

如果用计算机解答上述题目，原理是相同的：

(1) 原始数据及操作步骤通过输入设备输入给计算机。

(2) 使用运算器代替计算器，完成各项运算。

(3) 使用存储器代替纸、笔, 记录操作步骤、原始数据及每步计算的中间结果和最终结果。

(4) 使用控制器代替人脑, 控制计算机按操作步骤, 一步一步地完成计算。

(5) 最终结果通过输出设备传送给用户。

1.2 计算机语言

什么是语言呢? 从数学的角度去分析研究, 则语言可以这样定义:

语言是一个字符表(集)上的句子的任何集合。

字符表是字(母)、数字、符号的任何有穷集合。

句子是由字(母)、数字、符号组成的有限长度的集合。

严格的讲, 计算机所使用的各类语言都是在某一个有限的字符集上, 按一定语法规则组成的任何语句和数据的集合。

从计算机语言的发展进程看, 可将计算机的程序设计语言分为两大类: 低级语言和高级语言。低级语言又可分为机器语言和汇编语言。

一、机器语言

在计算机中, 指挥计算机操作的命令称为指令, 指令的全体称为指令系统。

计算机是按二进制原理设计的, 它只识别二进制数, 因此用二进制码 0、1 组成一套指令去控制计算机完成各种操作。这种二进制的指令是面向计算机的, 因此得名机器指令, 它是计算机最原始的指令。这些机器指令的集合称为机器语言。

计算机问世的最初十年(1946~1955), 人们就是用这种与机器紧密相关的机器语言编写程序的。使用机器语言编写程序, 必须了解存储单元的编址方法。由于机器指令是数字化的, 难读难记, 编程枯燥繁琐而且很容易出错。又由于每台计算机的指令系统各不相同, 编写的程序互不通用, 这就给程序移植带来了不易克服的障碍。

二、汇编语言

为了克服机器语言读写困难的缺点, 人们创造了汇编语言。在汇编语言中, 采用了一些助记符号代替原机器指令码。由于汇编语言的符号指令仍然是与计算机的机器指令一一对应的, 因此它仍属于一种面向特定机器的语言。虽然汇编语言在读写指令上比机器语言有了一定改进, 但在编写程序枯燥繁琐、容易出错、不容易修改、没有通用性等方面仍与机器语言一样。因此不便于计算机的推广应用。

用汇编语言编写的源程序上机执行时, 计算机首先将源程序通过已存在计算机中的汇编程序翻译成机器指令, 然后才能运行。

三、高级语言

在使用计算机的实践中, 人们很快认识到用低级语言编制程序和检查错误的繁重劳动, 可以在很大程度上交给计算机去做。这样, 在 50 年代(1957)诞生了第一个高级语言——FORTRAN 语言。

高级语言的共同特点是: 脱离特定的计算机, 是一种类似于自然语言和数学描述语

言的程序设计语言。在用高级语言设计程序时，不再是一条条指令的排序，而是用各种各样的语句，每种语句的功能隐含一串指令。

自高级语言诞生之后，尤如雨后春笋，发展极快。目前世界上使用的高级语言有几百种，常用的有 FORTRAN、ALGOL、COBOL、BASIC、PL/1、LISP、PASCAL、ADA 等。

四、高级语言的执行过程

尽管我们可以用高级语言编写程序，但是计算机只懂得机器语言程序。换言之，计算机不能直接接受和执行用高级语言编写的程序，必须经过“翻译”，将高级语言程序译成机器语言程序。通常把用高级语言编写的程序称为源程序，翻译成能被计算机执行的程序称为目标程序或目的程序。承担翻译工作的程序因高级语言的结构不同而异。象 FORTRAN、COBOL、ALGOL、PASCAL 等为编译结构，承担翻译工作的程序称为编译程序。大多数 BASIC 语言为解释结构，则承担翻译工作的程序称为解释程序。与每一种高级语言相对应的翻译程序是事先已存放在计算机里的机器代码程序，当用某一种高级语言时，调用相应的翻译程序担任翻译。

解释程序的翻译，象自然语言的口译者。口译者每遇到一句话便翻译一句，使听者接受；解释程序每遇到 BASIC 源程序的一个语句，就将它翻译成机器指令并让计算机立即执行。即边解释边执行，直至给出计算结果。解释过程如图 1-2 (a) 所示。

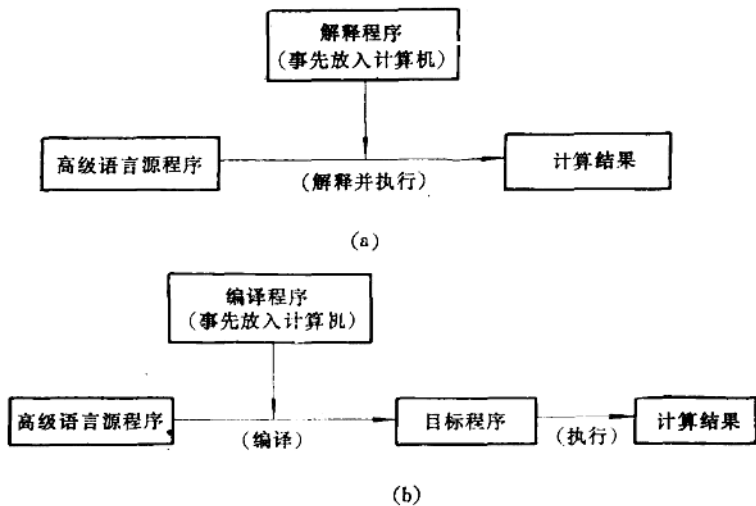


图1-2 高级语言的执行过程

(a) 解释过程；(b) 编译过程。

编译程序象自然语言的笔译者。笔译者将一篇外文资料译成中文让人阅读，一次译成后可被阅读任意次；编译程序将一份源程序翻译成完整的目标程序，然后令计算机执行。并允许重复执行任意次。编译过程如图 1-2 (b) 所示。

解释程序对源程序中重复执行的语句需要重复地解释执行，浪费时间，因而效率低。但解释执行方式便于实现人-机对话，容易修改，这是编译程序所不及的。

1.3 计算机系统软件

计算机的软件，是给计算机配备的一系列程序及资料的总称。它是计算机应用技术的具体体现。软件也称软设备。

计算机系统是硬件与软件结合的有机整体，软件依附于硬件，硬件靠软件发挥作用。

构成软件系统的一系列程序分为两大类：

(1) 控制或管理硬件系统，协调计算机工作的程序称为系统软件或系统程序。

(2) 用户直接用来解题或处理应用项目的程序，称为应用软件或应用程序。

电子计算机的软件构成如图1-3所示。

系统软件主要指操作系统。操作系统是具有多种管理功能的一套复杂程序。它是软件的核心，它指挥整个计算机系统的运转。此外，系统软件还包括各种语言处理

程序，如汇编程序、各种高级语言的翻译程序等。

应用软件主要指一些公用程序及用户编制的专用程序等。

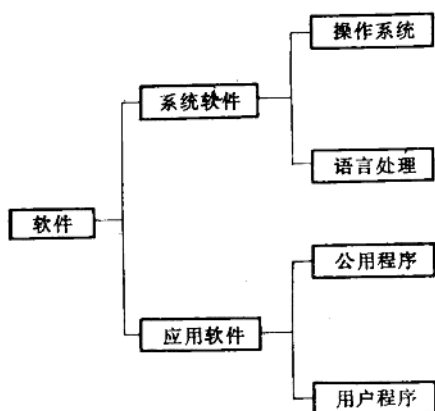


图1-3 电子计算机的软件构成

1.4 BASIC 语言的特点

BASIC 语言是高级语言的一种，BASIC 是 Beginner's All-purpose Symbolic Instruction Code 的缩写。

BASIC 语言是 1964 年由美国 Dartmouth College 的 V.G.Kemeng 和 Thomas F.Kurty 两位教授以 FORTRAN 为蓝本创造的。1964 年首先在 General Electric 225 计算机上实现，1965 年正式发表第一个版本。1978 年，基本 BASIC 建立了美国国家标准文本。1980 年，基本 BASIC 建立了国际标准文本。

BASIC 语言具有如下几个特点：

(1) BASIC 语言相对于其它高级语言（如 FORTRAN、ALGOL、COBOL 等）来说，简单易学。BASIC 语言的命令和语句中使用的词与运算符，同英语中使用的词和数学符号大体一致。例如 BASIC 语言中的“PRINT”是命令计算机进行“打印”，这与其在英语中的意思相同。由此可知，BASIC 语言的语句、命令都很容易理解和记忆。

(2) BASIC 语言是一种“会话式”语言。通过终端显示器和键盘，用户可以和计算机进行“人-机对话”。当源程序逐行进入计算机并运行后，计算机立即检查每行语句有无错误。发现错误立即在屏幕上指出错误所在，并等待用户修改。用户可以通过键盘修改错误，修改后再次运行。这样边运行边修改，直到获得正确结果为止。此外，在程序的运行过程中，计算机还会询问有关信息，用户作出相应回答后，计算机按要求继续执行程序。BASIC 语言上机操作简便、修改程序很方便，很适合初学者使用。

(3) BASIC 语言允许在终端上直接进行键盘运算。所谓键盘运算，是在不加任何

行号的情况下, 输入某个 BASIC 语句, 语句输入完毕, 计算机立即执行这个语句所指示的运算并显示出运算结果, 执行完毕后, 上述语句不予保存。BASIC 语言称这种工作方式为直接工作方式。例如从键盘输入 `PRINT 13*2`, 则计算机立即打印出计算结果 26。调试程序时, 常用直接工作方式检查数据。在 BASIC 语言中用的最多的还是间接工作方式。所谓间接工作方式是从键盘向计算机输送编好的程序, 并由 `RuN` 命令控制执行这个程序。例如:

```
10 PRINT 13*2
20 END
RUN
26
```

其中, 第 10~20 行是输入到计算机内存器的一个程序, 它只有在 `RUN` 命令控制下才被执行并打印计算结果。程序执行之后仍保存在计算机内存器中, 直到删除为止。

BASIC 语言简单易学、调试方便、功能较全, 它不仅可以用于数值计算, 还具有一定的数据处理能力。特别是增加了字符处理、自选打印格式的功能后, 使它的应用范围得以扩大。因此, BASIC 语言成为应用广泛的世界通用语言, 它在计算机语言的教育和普及等方面都起到了极大作用。

本书主要以介绍 IBM-PC BASIC 语言为主, 并对 APPLE-Ⅱ BASIC 语言进行介绍。

在使用本书时, 应注意以下几种符号:

(1) `< >` 括于尖括号内的是语句或命令的基本元素。例如:

`LET <变量名>=<表达式>`

(2) `{ }` 大括号表示必须选择括号内的一个元素。例如:

`IF <判断条件> {`
`THEN<语句组>`
`THEN<行 号>`
`GOTO<行 号>`
`}`

(3) `[ ]` 方括号表示其中的元素是任选项。例如:

`INPUT[ "提示内容"; ]<变量表>`

习 题 一

1. 计算机通常由哪几部分组成? 各部分的基本功能是什么?
2. 什么是计算机的硬件和软件?
3. 简述机器语言、汇编语言、高级语言的优缺点?
4. 计算机怎样执行汇编语言源程序?
5. 计算机怎样执行高级语言源程序?

第二章 BASIC 语言的词法及程序的构成

学习程序设计语言的最好方法是自己动手编写程序，并在计算机上运行和调试这些程序。为此，我们必须了解 BASIC 程序的结构；BASIC 程序中可使用的数据类型、运算符和表达式。此外，还必须了解程序设计的一般步骤。这一章我们分别对这些问题加以说明。

2.1 BASIC 程序的构成

所谓程序，是人们为了解决某个问题，用计算机能够接受的程序设计语言来描述给定数据及对数据的处理过程。BASIC 程序是按照 BASIC 语言的规定编写的程序。那么 BASIC 程序是如何构成的呢？我们看一个简单的例子。

例2-1 已知某班共有学生 40 人，数学考试成绩如下：A 等（5 分）15 人，B 等（4 分）18 人，C 等（3 分）5 人，D 等（2 分）2 人。试计算全班数学考试的平均成绩。

用 BASIC 语言编出的程序如下：

```
10 LET A=15:LET B=18
20 LET C=5:LET D=2
30 LET S=A+B+C+D
40 LET U=(5*A+4*B+3*C+2*D)/S
50 PRINT "U=";U
60 END
RUN
U=4.15
```

其中，第 10、20、30、40 行共含有六条赋值语句；第 50 行是打印语句，它将程序的计算结果打印输出；第 60 行是结束语句；RUN 是用户发出的运行程序的命令；最后是程序的计算结果。

从例 2-1 中可以看出，BASIC 程序是由一些程序行排列而成，程序行是由若干条语句组成，而这些语句又是用一些语言元素按照一定的语法规则构成的。

一般地说，表达式（如例 2-1 中的 $A+B+C+D$ 和 $5*A+4*B+3*C+2*D$ ）/S）、语句关键字（如例 2-1 中的 LET、END、PRINT）等结合在一起，完成一个指定的操作，就构成了一条语句。而若干条语句（它们之间用冒号分隔）组合在一起，再冠以行号（例 2-1 中的 10、20、30 等），完成一个或多个操作，就构成了一个程序行。

一个 BASIC 程序行的格式如下：

〈行号〉〈语句〉[:语句] … 〈程序行终止符〉

一个 BASIC 语句的格式如下：

〈 语句关键字 〉〈 语句体 〉

一、行 号

行号应放在每一个 BASIC 程序行之前，是程序行的标志。

1. 行号的作用

行号的大小指示着程序行将被执行的先后顺序。BASIC 系统通常按由小到大的行号次序执行程序。如例 1 中，程序的执行顺序是 10→20→30→40→50→60。

2. 行号的使用范围

由于各种计算机系统的不同，它们对行号的使用范围的规定也不一样。

IBM-PC BASIC 语言规定，行号的使用范围是 0～65529，并且必须是正整数。

APPLE-Ⅱ BASIC 语言规定，行号的使用范围是 0～63999，并且必须是正整数。

在选择行号时，如果超出规定的范围，则发生错误。

BASIC 程序在计算机存储器中是按行号的大小顺序存放的，与程序行输入的先后次序无关。例如从计算机键盘按下列次序输入程序：

```
60 END
50 NEXT I
10 LET M=0
20 FOR I=1 TO 10
40 PRINT I, I^2, M
30 LET M=M+I
```

在计算机存储器中却是按如下次序保存的：

```
10 LET M=0
20 FOR I=1 TO 10
30 LET M=M+I
40 PRINT I, I^2, M
50 NEXT I
60 END
```

3. 行号可以不连续使用

我们在选用行号时，最好不连续。如例 1 中，使用的行号是 10、20、30、40、50、60，行号不连续并且间隔为 10。这样，当需要调整程序或增加功能时，可以增补一些新的程序行。如在行号为 10 的程序行与行号为 20 的程序行之间，最多可以插入九个新程序行。

二、语 句

前面已经提到，一个 BASIC 程序由许多程序行构成，而构成程序行的则是语句。

(1) 每个程序行可以含有一条或多条语句，语句和语句之间用冒号 (:) 分隔。

例如，程序行

```
100 LET A=5:LET B=6
200 PRINT A+B+C
```


其中，第 100 行含有两个语句，这样的程序行叫多语句行；第 200 行只含有一个语句，这是最常用的书写形式。

一个程序行允许含有多少个语句是没有规定的，但是，一个程序行的全部语句及行号的总计字符个数（包括空格）不能超过 255。

（2）一条语句由语句关键字、语句体组成。

语句关键字（或称语句定义符）通常处于一条语句的开头，它规定计算机执行何种操作。语句中跟在关键字后边的部分叫语句体，它规定计算机执行的具体内容。

例如：程序行

```
10 PRINT 'THIS IS ONE LINE PROGRAM'
```

其中 PRINT 是语句关键字，它规定计算机打印，而后面引号中的内容则是打印的数据。

又如：程序行

```
100 LET S=A+B+C+D
```

其中 LET 是语句关键字，它指挥计算机将变量 A、B、C、D 的值相加，其结果存入变量 S 中。 $S = A + B + C + D$ 是 LET 的语句体。

三、程序行终止符

程序行终止符是一个程序行结束的标志。当通过键盘向计算机输入程序时，输入的内容都将在计算机屏幕上原样显示出来。每输入完一个程序行之后，都要按下 RETURN 键（通常说的回车键）作为终止符，用以通知计算机本程序行输入完毕，此后计算机将等待输入新的程序行。必须注意，程序行终止符并不在屏幕上显示出来，在本书中所列举的程序中，各程序行的末尾也不特别注明，但不能认为那里没有程序行终止符。

2.2 基本字符

任何一种计算机系统所能使用的字符是固定的、有限的，它受计算机设备的制约。

BASIC 语言就是由一个有限的字符集组成的。BASIC 字符集包含下列字符：

（1）英文字母

英文字母表中所有大写与小写的字母。

（2）数字

0 ~ 9。

（3）专用字符

字符

字符名称

=

空格，在键盘上按空格键

等号或赋值号

+

加号或连接符号

-

减号

*

星号或乘号

/

斜线或除号

\

反斜线或整除号

^

插入记号或乘幂符号

(	左括号
)	右括号
%	百分号或整型说明符
#	号码符或双精度实型说明符
\$	美元符号或字符串型说明符
!	感叹号或单精度实型说明符
&	表示AND的字符
,	逗号
.	句号或小数点
'	单引号(撇号)或注释定界符
;	分号
:	冒号或语句分隔符
?	问号或PRINT语句的缩写
<	小于号
>	大于号
"	双引号或字符串定界符
_	下划线

还有一些字符,虽然它们在 BASIC 语言中没有什么特定意义,但它们也属于 BASIC 字符集,并且可以被显示或打印。有关 BASIC 字符集请参见本书附录一。

在 BASIC 语言中,字是由一些字符组成的有意义的字符串,其中对 BASIC 语言有专门意义的字称为保留字(或称关键字)。BASIC 语言的保留字包括全部的命令、语句、函数名字和运算符名字。

由于保留字已经有专门的含义,所以它不能再作为用户使用的变量、数组等的名字。在语句中,应该注意把保留字与数据或语句的其它部分用空格或其它语法所允许的专用字符分隔开,也就是说,必须对保留字加以适当的分界, BASIC 系统才能识别它们。这一点在书写程序时尤为重要。有关 BASIC 语言的全部保留字请参见本书附录二。

2.3 常 量

用户程序是用户按计算机系统提供的能力和规则书写的语句与数据的集合。显然,系统提供的数据类型愈丰富,处理问题的能力也就愈强。

在程序执行过程中,其值始终保持不变的量称作常量。常量是指在程序中直接使用的一些数字或字符。如 13.5、7600、24 等。

常量主要分为数值常量和字符串常量。

一、数 值 常 量

数值常量又分为整型常量和实型常量。

1. 整型常量

整型常量简称为整数,凡是不含有小数点而且值在 $-32768 \sim +32767$ 之间的数,均被视为整数。如 43、6、32576、-96 等。

2. 实型常量

实型常量简称为实数,凡是含有小数点或指数部分以及值超过 $-32768 \sim +32767$ 范

围的数,均被视为实数。如 12.3、326942 等。实数的取值范围是 $\pm 2.9 \times 10^{-39} \sim \pm 1.7 \times 10^{38}$ 。就绝对值而言,比 2.9×10^{-39} 小的数,当作 0 处理;大于 1.7×10^{38} 的数,计算机无法表示,给出“溢出”信息。

在 IBM-PC BASIC 语言中,实数又可分为单精度实数和双精度实数。当一个实数的有效数字的位数[●]小于等于七位时,这个实数称为单精度实数;当一个实数的有效数字的位数多于七位时,这个实数称为双精度实数,双精度实数最多可以有十六位有效数字。如 1.5782、3.1416、0.2 等是单精度实数,而 42384657、3.1415926、284.3276894736 等是双精度实数。

对于非常大或非常小的实数可用指数形式表示,即所谓的科学计数法。如中华人民共和国领土面积是 9600000 平方公里,通常可以写作 9.6×10^6 平方公里。又如蜗牛的爬行速度是 0.0000079 公里/秒,通常可以写作 7.9×10^{-6} 公里/秒。

但是在计算机中,字符只能一个一个地输入/输出,无法表示右上角的指数。因此, BASIC 语言规定:用字母 E 表示底数 10,指数写在同行内 E 的后边(双精度实数采用字母 D 代替 E)。上面两个数可以写成如下形式:

$$9.6E+06, 7.9E-06$$

科学计数法的格式如下:

$$\{\pm\}\text{数}E\{\pm\}n$$

$$\{\pm\}\text{数}D\{\pm\}n$$

其中,数前边的“+”或“-”是数值的符号;E(或D)前边的数称为数字部分或称尾数;n为指数或称阶码,它前边的“+”或“-”是指数的符号。数符和指数符号为正时皆可省略,指数n是一位或两位的十进制数。

在 APPLE-II BASIC 语言中,实数没有单精度与双精度之分,实数的有效数字的位数最多为九位。当计算机输出实数时,若实数的绝对值大于等于 0.01 且小于 999999999.2,按小数格式输出;否则按科学计算法输出。例如:

数字	计算机输出形式
-233.2	-233.2
465497.982	465497.982
0.000000001	1E-09
-12345678900	-1.23456789E+10

3. 数值常量的尾符说明规则

有时为了明确的区分常量的类型,可以给常量加上规定的尾符。BASIC 语言规定:整型常量的尾符是“%”,单精度实型常量的尾符是“!”,双精度实型常量的尾符是“#”。例如:

● 一个常量的有效数字的位数等于从这个常量最左边不是 0 的数字位起到最右边不是 0 的数字位止之间的位数。例如:

常量	有效数字的位数
1000	1
0.0036	2
201.000054	9
8245	4