

电子计算机
会话式语言

BASIC

程序设计



李书 嵎 编

甘肃人民出版社

电子计算机会话式语言
BASIC 程序设计

李 书 媚

**电子计算机会话式语言
BASIC程序设计**

李书娟 编

甘肃人民出版社出版

(兰州庆阳路230号)

甘肃省新华书店发行 兰州新华印刷厂印刷

开本787×1092毫米1/32 印张5.75 字数120,000

1979年10月第1版 1979年10月第1次印刷

印数1—20,240

书号: 13096·44 定价: 0.48元

目 录

第一章 电子计算机与程序设计语言	(1)
§ 1.1 电子计算机概述	(1)
§ 1.2 电子计算机的程序设计	(2)
§ 1.3 电子计算机的程序设计语言	(3)
§ 1.4 BASIC语言发展概况	(6)
第二章 BASIC程序的编制和运行概述	(9)
§ 2.1 解题的基本环节	(10)
§ 2.2 BASIC程序的一个例子	(13)
§ 2.3 程序的编辑	(16)
§ 2.4 程序的运行	(17)
§ 2.5 程序的中断	(17)
第三章 BASIC语言的基本成份	(18)
§ 3.1 字母, 数字, 基本名词及符号	(18)
§ 3.2 数的表示	(20)
§ 3.3 简单变量	(21)
§ 3.4 表达式	(21)
§ 3.5 数组	(22)
§ 3.6 标准函数	(25)
§ 3.7 自定义函数	(30)
第四章 赋值语句, 数据输入和输出语句	(33)
§ 4.1 赋值语句 (let)	(33)
§ 4.2 输出语句 (print)	(33)
§ 4.3 键盘输入语句 (input)	(40)

§ 4.4 读数据语句	(42)
第五章 控制语句, 转子语句, 注释语句	(48)
§ 5.1 转向语句 (goto)	(48)
§ 5.2 条件语句 (if—then)	(49)
§ 5.3 循环语句 (for—step和next)	(51)
§ 5.4 子程序和转子语句 (gosub和return).....	(55)
§ 5.5 结束语句 (end)	(58)
§ 5.6 停语句 (stop)	(59)
§ 5.7 注释语句 (rem)	(59)
第六章 键盘命令, BASIC程序的执行	(64)
§ 6.1 键盘命令	(64)
§ 6.2 操作过程	(68)
§ 6.3 错误信息表	(72)
第七章 BASIC程序实例.....	(75)
§ 7.1 用牛顿法解超越方程	(75)
§ 7.2 用矩形法计算积分	(77)
§ 7.3 计算 $C_j^k = \frac{j!}{k!(j-k)!}$	(79)
§ 7.4 用龙格-库塔 (Runge-kutta) 方法解一阶常 微分方程	(81)
§ 7.5 用高斯消去法解线性方程组	(88)
§ 7.6 确定透镜的焦距	(95)
§ 7.7 射线治癌的剂量计算	(97)
§ 7.8 用腰形管测量流量	(100)
第八章 行变量.....	(104)
§ 8.1 行, 行变量和行表达式	(104)
§ 8.2 行dim语句	(105)
§ 8.3 行读数据语句	(106)

§ 8.4 行赋值语句	(107)
§ 8.5 行条件语句	(108)
§ 8.6 行输入输出语句	(110)
第九章 矩阵	(114)
§ 9.1 矩阵下标	(115)
§ 9.2 矩阵运算语句	(118)
§ 9.3 矩阵的输入和输出	(127)
第十章 文件	(130)
§ 10.1 文件名字	(131)
§ 10.2 文件的开启和关闭	(132)
§ 10.3 关于文件的一些语句	(134)
§ 10.4 和文件有关的键盘命令	(137)
附录一: DJS-130电子计算机简介	(142)
附录二: 电传机代码表 (DJS-130机, DJS-131机用)	(151)
附录三: 八单位字符表	(154)
附录四: BASIC的形式语法	(157)
附录五: BASIC语言简表	(160)
习题答案	(162)
参考文献	(175)

第一章 电子计算机与程序设计语言

§ 1.1 电子计算机概述

电子计算机是综合应用无线电电子学，自动控制技术，信息论，控制论等科学成果的产物。它的出现进一步促进了科学技术的飞跃发展。

自一九四四年第一台电子计算机出现以来，计算技术的发展非常迅速。电子计算机已由第一代电子管计算机、第二代晶体管计算机、第三代集成电路计算机到现在的第四代大规模集成电路计算机，大体上每五至八年计算机的运算速度提高十倍，体积缩小十倍，功耗大幅度下降，而稳定性却显著提高。我国在大跃进的一九五八年研制出了每秒运算一万余次的电子计算机，结束了我国不能生产电子计算机的历史。一九六四年我国研制成功每秒运算十万次的晶体管计算机。一九七一年国产集成电路计算机问世，为计算机的小型化、积木化开辟了良好的前景。一九七九年每秒运算五百万次的集成电路计算机研制成功，标志着我国计算技术又向前迈进了一步。

电子计算机分为两个大类，即模拟计算机和数字计算机。本书所提“计算机”一词，均指电子数字计算机。

以电子计算机为重要组成部份的电子科学技术的高度发展和在各个领域的广泛应用，对于社会生产力的发展将起着变革性的推动作用。

电子计算机能够快速解决工程设计、科学研究及发展新技术中遇到的各种繁杂的计算问题。如测地学、天文学、建筑力学中提出来的代数方程组；工程中的火箭弹道学及自动调整理论等方面广泛应用的常微分方程；气体力学、无线电技术、光学、流体力学、热力学、气象学等方面提出的偏微分方程等。因此随着我国社会主义建设事业的日益蓬勃发展，在建筑结构的应力分析，铁路公路的建设、机械加工、电力建设、化工设计、石油开采和地质勘探、人造卫星轨道运算、天文年历的编制、气象的数值预报以及探空技术、热核反应，农林、邮电等方面都有大量的复杂计算问题，只有使用电子计算机才能迅速准确地得到解决。

§ 1.2 电子计算机的程序设计

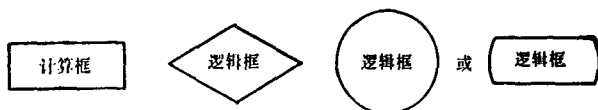
电子计算机的运算速度很快，每秒可达几十万次甚至数百万次以上，但其运算仍是严格按照事先排好的一条条的指令来进行的。一系列的指令总称为程序。编制程序的工作称为程序设计。计算机离开了程序就无法工作。程序安排了计算机所需要完成的任务。计算机的工作完全依赖于程序。所以应用计算机的关键在于程序设计。计算机应用的迅速增长，导致了对程序设计广泛的兴趣。

在电子数字计算机上解题的步骤是：

1. 把生产中或工程中遇到的实际问题转化为数学问题。这是用计算机解决实际问题的前提，这部分工作叫作建立数学模型。

2. 找出该问题适宜于计算机上采用的解法，并按解题的详细步骤画出逻辑上清晰的框图。

框图是解题步骤的图示。有了框图，编写、阅读和检查程序都比较方便，可以减少编制程序中出现的错误。在框图中每一步用一个框表示（见下图）。



框图中的符号

矩形框表示程序中的计算部分。菱形框或圆框为逻辑框，表示按一定条件从两条计算路线中选择一个。

3. 按照框图，编制计算机语言程序。

4. 调试程序，对已编出的程序，进行静态检查，看框图是否完整而正确地体现了解题步骤，程序是否与框图一致，程序有无语法或逻辑错误。然后再安排一些检查数据，在计算机上试算以进一步检验程序的正确性，试算又称为动态调试。

§ 1.3 电子计算机的程序设计语言

程序设计语言是应用领域和计算机之间的桥梁，是人和计算机相互联系的纽带。对用户来说，它是描述解题算法的手段，对计算机加工系统来说，它是信息的源泉和加工的根据。

计算机程序设计语言可分为三类：

1. 机器语言

每一种电子计算机都有自己的指令系统，指令系统就是机器语言，直接用机器语言编的程序叫手编程序，手编程序是编制程序的最原始的形式，用这种方式编程序工作量大，

阅读和修改都比较麻烦，学习和推广也比较困难，但这种程序可直接在计算机上运行，效率较高，运行时间也较短。

2. 汇编语言

汇编语言是为简化程序设计工作而设计的。它的主要思想是用符号来代替操作码，用标识符来代替地址码和变址码，这样就使程序员能摆脱计算机的一些细节问题而专门考虑程序之间的内在联系。

汇编语言又称符号语言，用符号语言编写的程序称为源程序，这种程序不能直接在机器上运行。源程序需经汇编程序加工成目标程序（即用机器语言写的程序）后，才可在机器上运行。

汇编语言比较灵活，用汇编语言写的程序运行效率较高。用它来编写程序比用机器语言较为方便。在阅读、检查和修改时都比机器语言容易。

用汇编语言编的程序是依赖于特定的计算机或计算机族的。所以它又称为面向机器的语言。

3. 高级计算机语言

高级计算机语言是面向过程的语言。虽然汇编语言比机器语言容易掌握，但它仍与通常的数学表示有很大距离，人们希望能有一种与常用的数学表示很相似的程序语言。面向过程的语言就是这种类型的语言。它可使程序员摆脱机器内部的逻辑，集中精力考虑解题算法的逻辑和计算过程的描述。

由于面向过程的语言是不依赖于机器的，所以用这种语言写出的程序，原则上可以在各种不同型号的计算机上运行。

目前世界上面向过程的语言有数百种，真正广泛流行的有以下几种：

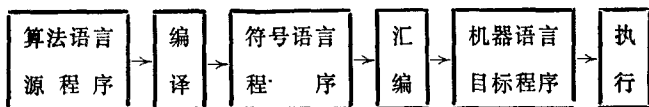
FORTRAN, ALGOL, COBOL, PL/1, BASIC, APL, SNOBOL, LIST。

这种类型的语言学习起来容易掌握，使用也很方便，用它们写出的程序阅读，检查，修改都比较容易，这些语言在世界各地已被广泛采用。

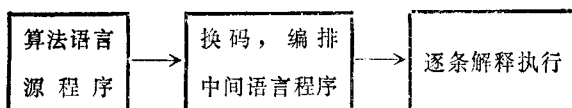
用这种类型的语言写出的程序也叫源程序，它不能直接在计算机上运行。

这种语言从运行的方式上来看又可分为两种类型：

第一种是编辑型：它首先将用算法语言写成的“源程序”经“编译程序”翻译成符号语言程序，再汇编成机器语言的“目标程序”，然后再由机器来执行，其过程可图示如下：



第二种是解释型：它将用算法语言写成的源程序经过换码，编排以后变成由机器代码组成中间语言程序，然后由“解释程序”逐条加以解释后得到结果。DJS-130机，DJS-131机提供的BASIC系统为解释型。这是因为BASIC语言比较简单，便于解释执行，另一方面采用解释方法比较节省内存，这对小型计算机比较有利。但是采用解释方法要耗费较多的机器时间，因而大大降低了解题的速度。解释型的执行过程如下：



§ 1.4 BASIC语言发展概况

BASIC语言是最常用的面向过程的通用程序设计语言之一。

BASIC是英文“Beginner's ALL-purpose Symbolic Instruction Code”的缩写，意思是初学者通用符号指令码。

BASIC语言是美国新罕布什尔州的达特摩斯（Dartmouth）学院的约翰·凯米尼（John Kemeny）和托马斯·库尔特茨（Thomas Kurtz）两教授创造的。于1964年首先在General Electric 225计算机上实现。1965年它的一个版本问世，1967年出第三版，1971年形成标准文本（见参考文献16）。

BASIC语言类似于FORTRAN II，初出现时较为简单，后来内容不断扩充，至1970年已发展成为内容丰富，表达能力强，容易掌握，使用方便，颇受用户欢迎的一种语言。它的设计很成功。国外的很多计算机如：

Burroughs, CDC 300系列, CDC Cyber/6000系列, DEC PDP-8~11, Honeywell 600/6000, IBM370, UNIVAC 1100系列, Xerox Sigma 9等计算机上都配有BASIC语言。

近年来BASIC语言在国内已开始使用，DJS-100系列机均配有BASIC语言，有的DJS-121机也可使用BASIC

语言。

BASIC语言也是一种交互式会话程序设计语言。它提供了人一机对话的功能，即通过电传打字机的键盘和印字装置，人和计算机可以相互应答。

BASIC语言在许多计算机上的实现，都是根据当时的达特摩斯学院的BASIC文本写的，而该学院的BASIC文本本身从1964年的第一个文本到目前的版本已有很大的变化。尽管BASIC语言在各种不同的计算机上的实现不尽相同，但其基本部分是一样的，只是在符号上或扩充部分有些差异。本书是按照BASIC语言在DJS-130机（同样适用于DJS-131机）上的实现来编写的。只要掌握了本书的内容，而在其它类型的计算机上使用BASIC语言也将会是很容易的，只要阅读一下该机的BASIC语言手册，注意两者之间的差别即可。

BASIC系统包括：

1. 单用户BASIC

它对硬件的要求是主机有4 K^(注)内存和一台电传打字机。

它包括BASIC系统的主要语句和键盘操作及查错系统，但不包括矩阵运算语句和行操作等。

2. 分时BASIC

分时BASIC是最通俗最流行的一种分时语言，使多用户（2~16人）以经济而有效的方式来使用计算机。

它除了包含单用户BASIC的全部内容外，还扩充了矩

注：1024个存储单元称为1 K，4 K为4096个存储单元。

阵运算语句和行操作等。

3. 扩充BASIC

它是在磁盘操作系统(DOS)或独立操作系统(SOS)的调度下使用的。

它包含了单用户BASIC和分时BASIC的全部内容,增加了控制输出格式和文件系统,并扩充了命令系统。

分时BASIC和扩充BASIC对硬件的要求较高,内存要求32K,并要求有多台电传机等外部设备,此外扩充BASIC要求硬件配有磁盘。

鉴于目前国产的DJS-130,131计算机仅配有单用户BASIC,介绍单用户BASIC是本书的重点。第一章至第七章的内容都是关于单用户BASIC的,又考虑到行变量,矩阵运算,文件系统这些概念很重要,本书也安排有适当的篇幅加以介绍。第八章,第九章介绍了分时BASIC中的行变量和矩阵运算。第十章文件是属于扩充BASIC的内容。读者如对分时BASIC和扩充BASIC有兴趣,可参阅参考文献2, 3)。

第二章 BASIC程序的编制 和运行概述

在详细介绍 BASIC 语言的各语句之前,本章先对BASIC 程序的编制和运行的整个过程作一简要的叙述,使读者对应用 BASIC 语言编制程序和上机计算运算的全过程有一个概括的了解。

BASIC程序由语句构成,每个语句为一行。每一行后应有回车符号,全部程序结束后打回车和换行符号。在书写程序时,回车,换行符号可以省略,但在穿纸带时,回车和换行符号一定要穿上。

每一语句前都有一个正整数作为它的标号,供编辑和控制转移使用。标号可在 1 ~ 9999 或 1 ~ 99999 之间选取。BASIC程序执行时,一般是按照标号的顺序,由小到大顺次执行的(但是遇到控制转移时例外)。语句的标号习惯上是不连续的,一般以10为间隔,这样的安排是为了在修改程序时,在两语句之间能够插入一些语句而使后面的语句标号无需改动。这给修改程序带来了一定的方便。譬如我们要在标号50和60两语句之间嵌入一些语句,则其标号可以在51~59之间选取。当所有语句全部执行完毕或执行到 end 语句或 stop语句时则停机。

§ 2.1 解题的基本环节

用BASIC语言编制程序有三个基本环节，即提供数据、执行计算和印出计算结果。

1. 提供数据

在BASIC语言中提供数据有两种语句，即赋值语句（let）、读入语句（read）和数据语句（data）。数据个数不多时，一般多使用赋值语句。

例 1：

```
20 let p = 3.14
30 let e = 2.71828
40 let a = 103
```

执行20~40语句后则变量 p 的数值为3.14，变量 e 的数值为2.71828， a 的数值为103，注意这里“=”号的意义与一般数学上的“=”号意义不同。由本例可知计算中需要的数据可由赋值语句直接提供。

若计算中的初始数据较多，则可采用read和data语句来提供数据。

例 2：

```
10 read y
20 let x = 3.14 * y    (其中 * 号表示乘)
30 goto 10
40 data 10.2, 7.3, -56.11, -0.003, 3.4
```

在本例中 y 为直径， x 为圆周长。当第一次执行到10号语句时，将40号语句data中的第一个数10.2读入变量 y ，而第二次执行到10号语句时则将40号语句data中的第二个数

7.3读入变量 y ，依此类推当第5次执行10号语句时则将3.4读入变量 y 。

2. 执行计算

输入数据后，必须计算出结果。在BASIC中简单算术运算也是由赋值语句let实现的。

let语句可用于将计算出来的值赋给某一变量。在let语句中“=”左端为简单变量，“=”右端的求值部分称为表达式。

在BASIC中执行运算的计算符号有：

符号	运算	例子	意义
+	加	$a + b$	a 加 b
	正	$+a$	正 a
-	减	$a - b$	从 a 中减去 b
	负	$-a$	负 a
*	乘	$a * b$	a 乘 b
/	除	a / b	a 除以 b
↑	乘方	$a \uparrow b$	a 自乘 b 次即 a^b

为了便于计算，在BASIC中还提供了一些标准函数（详见第三章 § 3.6）

一些标准函数的例子如下：

$\cos(x)$ 求 x 的余弦，其中 x 为弧度。

$\text{sqr}(x)$ 求 x 的平方根，即 $\sqrt{x}$ 。

$\text{int}(x)$ 舍去 x 的小数部分。

下面是一个表达式求值并赋值给变量 a 的例子

例 3：

```
100 let a = sin(x) + cos(y) + x * y ↑ 3 + sqr(x)
```