

BASIC 程序设计及
调试技巧



前 言

灵活运用BASIC程序设计技巧和熟练掌握BASIC程序调试方法，是BASIC语言程序设计中的两个极为重要的方面。目前，国内尚无系统介绍BASIC程序设计技巧和调试方法的书籍，为此，我们根据多年讲授BASIC语言程序设计的体会编成此书，旨在帮助学习BASIC语言程序设计的读者更好地掌握常用的编程技巧和调试程序的方法。

本书归纳和讨论了分支程序设计方法与技巧，循环的设计与优化，子程序的设计与调用，数字型程序的设计与简化等4个方面的内容，其中许多编程方法和技巧是作者近几年在多种刊物上发表的文章经加工、深化而成的。本书还介绍了BASIC程序的若干常用的调试方法，以便读者在调试程序过程中更快地发现和修正错误。这些方法必须经过大量的实践才能逐步学会。

书中的例题新颖，具有较强的实用性、趣味性和知识性。所列出的程序分析简明，构思科学，结构清晰，质量较高，均在APPLE机、中华学习机上验证过。

在编写本书过程中，得到张德荣教授、王能超教授的鼓励和指导及华中理工大学出版社的大力支持，王能超教授、陈培军副教授非常认真仔细地审阅了全书，并提出不少宝贵意见，在此一并致谢。

编 者

1988年8月

目 录

第一章 BASIC程序设计概述..... (1)

§ 1 BASIC程序的结构..... (1)

由

1.1 BASIC程序..... (1)

1.2 BASIC程序设计..... (2)

1.3 BASIC程序的结构..... (2)

§ 2 程序设计的步骤..... (6)

2.1 总体设计..... (6)

2.2 确定算法..... (7)

2.3 结构模块化..... (8)

2.4 编写程序..... (9)

第二章 BASIC程序设计技巧..... (10)

§ 1 分支的设计方法与技巧..... (10)

1.1 框图的绘制方法..... (10)

1.2 单分支的设计..... (12)

1.3 多分支的设计..... (13)

1.4 分支的设计技巧..... (22)

§ 2 循环的设计与优化..... (32)

2.1 循环的结构..... (32)

2.2 循环的优化..... (44)

2.3 循环在数组运算中的应用..... (55)

§ 3 子程序的设计与应用..... (57)

3.1 子程序的设计及嵌套调用..... (58)

3.2	子程序的应用	(66)
§ 4	数字型程序的设计与简化	(74)
4.1	求特殊数程序的设计	(74)
4.2	数字算式程序的设计	(78)
4.3	数字图案程序的设计	(84)
第三章	BASIC 程序调试方法	(89)
§ 1	静态调试方法	(89)
1.1	常见错误的检查、判断与修正	(90)
1.2	静态调试方法与步骤	(98)
§ 2	动态调试方法	(99)
2.1	常用的调试命令	(99)
2.2	动态调试方法	(100)
 附录		
附录一	APPLESOFT 错误信息表	(112)
附录二	APPLE 机的 ASCⅡ 码	(114)
附录三	APPLESOFT BASIC 与 IBM PC BASIC 常用语句的主要区别	(116)
附录四	IBM PC BASIC 程序上机操作	(120)

第一章 BASIC程序设计概述

§ 1 BASIC 程序的结构

1.1 BASIC程序

用BASIC语句编写的处理某个问题的一组语句，通常称为BASIC程序。

BASIC程序由一系列程序行组成，若将程序中的语句按其作用进行划分，一个完整的BASIC程序则包含以下的成份：

(1) 变量取值。计算赋值语句LET、成批赋值语句READ/DATA、键盘赋值语句INPUT等都可以给变量赋值。在一个程序中，具体使用哪个语句给变量赋值，要根据数据个数的多少，是已知数据还是临时决定的数据等不同情况灵活选用，有时还可以利用FOR—NEXT语句产生数据。

(2) 计算并保存结果。计算是程序的主要任务，计算工作是由LET语句或PRINT语句实现。值得注意的是，使用PRINT语句所计算出来的表达式的值是不能保存在计算机的内存中的。

(3) 打印（显示）结果。计算或处理的结果，一般需将其打印（或显示）出来，以供人们使用，这项工作是由PRINT语句完成。

(4) 终止运行。一个程序的末尾，一般要设置END

语句（或STOP语句），使程序完成了预定的任务后自行正常停止运行。

1.2 BASIC 程序设计

写一篇文章，要经过一系列的工作过程，即背景或事实→审题→确定题材→提纲→腹稿→拟稿→修改→脱稿。设计一个 BASIC 程序，如同写文章一样，一般须经以下几个阶段：课题的提出→分析计算问题→确定模型→功能设计→程序框图→编写程序→调试程序→上机计算。

学习了中文词法、句法等语法后，不一定就能写文章，学好了语法不一定能写出好文章。BASIC语言中的各种语句，为进行程序设计提供了强有力的手段，但即使对BASIC语言的语句、命令等学得很好，也不一定能设计出高质量的程序。因此，学习用BASIC语言编写计算程序，除严格遵守BASIC语言的有关规定外，还必须注意掌握各类基本程序的编写方法及技巧，不断摸索和总结编程的经验，并通过大量上机实践，才能不断提高程序设计的能力。

1.3 BASIC 程序的结构

高质量的程序，不仅应具有正确性、通用性、可读性和易修改性，而且还要求程序的结构合理、完整和清晰。

程序结构，是指程序中各部分的组合方式。对于BASIC程序而言，则是指BASIC语句的编排方式。

一个BASIC程序，无论它怎样复杂，总是由一些基本形式的结构所组成。程序的基本结构一般分为顺序结构、选取结构和重复结构3种，它们的共同点是，每种结构都具有单入口和单出口，即一个结构只有一个入口和一个出口。

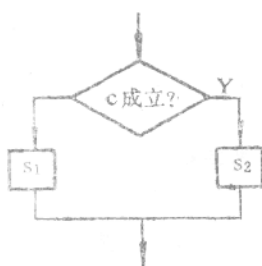
图1-1 所示的程序结构称为**顺序结构**。这种结构中含有



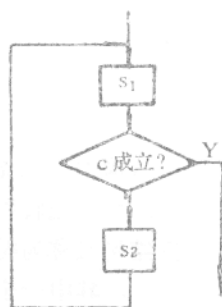
图 1-1

一串语句序列 $s_1, s_2, \dots$, 程序运行时, 是严格按顺序执行语句串。

图 1-2 所示的程序结构称为**选取结构**。这种结构中包



(a)



(b)

图 1-2

含有多重转向语句, 其中, 图 1-2 (a) 表示当条件 c 成立时, 则执行语句序列 s_2 , 否则, 执行语句序列 s_1 。但不论是执行 s_1

还是 s_2 ，执行后都控制到同一出口处。

选取结构中最简单的是“二选一”结构，另外还有一种“多选一”的结构（如图1-3），就是在多种可能中选取其中之一的一种结构。

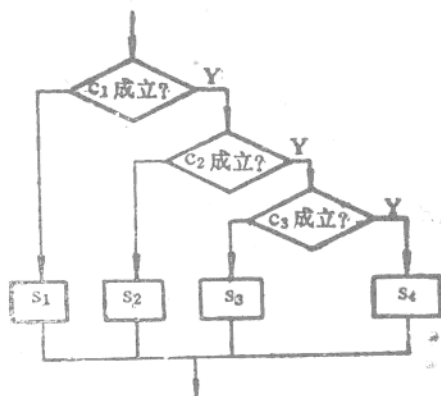


图 1-3

顺序结构和选取结构都属于开型结构，出口处都要转入下一结构，而不是转移到同一结构的入口处。

重复结构又称为循环结构（如图1-4），循环结构中也包含分支，其中一个分支转向出口，另一个分支控制返回结构的入口，从而构成重复，反复执行 s_1 ，直至判断条件发生变化时为止。

循环次数能事先确定的重复结构，可用FOR—NEXT循环结构实现，循环次数由循环变量的初值、终值及步长值确定（如图1-4(c)）。循环次数事先不能确定的重复结构，

是由判断条件决定循环的次数，在适当的时候退出循环，进入下一结构。

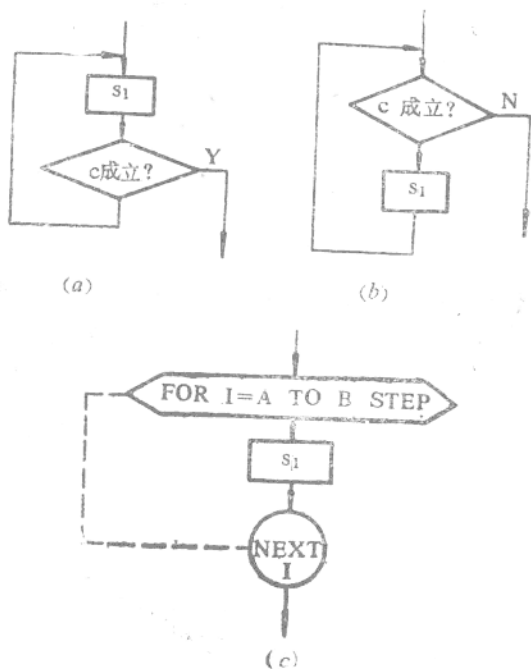


图 1-4

一个复杂的程序，一般不只含一种单一结构，而是由一种结构嵌入另一种结构，从而由几种结构组成复合结构，如图1-5所示。其主结构是重复结构，其中嵌有顺序结构和选取结构，而选取结构又嵌有重复结构，但包含主结构在内的每一个结构都严格地遵循结构的单入口和单出口原则。

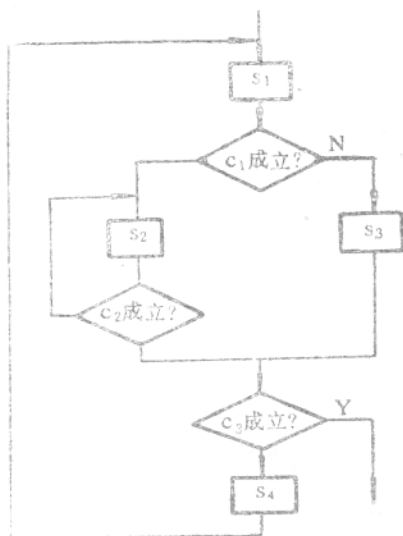


图 1-5

§ 2 程序设计的步骤

学习BASIC语言的目的,是运用BASIC语句编写实际问题的计算程序。设计程序没有统一的模式,设计人员可以尽情发挥,但一般是采用自顶向下设计和自底向上编程的方法,可归纳为总体设计、确定算法、结构模块化和编写程序4步。

2.1 总体设计

无论是大的复杂的问题,还是小的简单的问题,在编写计

算程序前，都要对问题进行分析、规划，即要有一个总体的设想，这就是总体设计。

在具体解决一个实际问题时，特别是复杂问题，一般不可能事先了解到它的全部细节，只能对问题的全局作出一个决策，确定解题的步骤，对每一步做什么进行描述，而不具体地考虑每步如何去做。基于这种考虑，即可进而对问题进行分解，产生对下一步做什么的描述。如此逐步进行下去，一个大的问题就分解成若干相对独立的子问题，每个子问题都解决了，整个问题也就解决了。

总体设计如同写文章前的总体构思一样，对于较复杂的问题，必须先描绘出粗框图，尽可能把有关问题想得周到些、全面些。譬如，整个程序大致分为几个段落，需要编写多少个子程序，它们各实现多少功能，有关参数如何确定，要定义多少个数组，各程序段如何衔接等。

例如，用计算机统计考生的分数，可分解为以下几个子问题：

- (1) 计算每个考生的总分。
- (2) 按总分的高低顺序排队。
- (3) 打印成绩通知单（准考证号、姓名、单科成绩、总分）。
- (4) 将学生的成绩存入档案。

图 1-6 是描述以上问题的总框图，总体工作和各个子问题所完成任务一目了然。

2.2 确定算法

要解决一个实际问题，必然要牵涉到某些算法，例如方程组的解法、插值方法、数值积分方法、数理统计算法等。

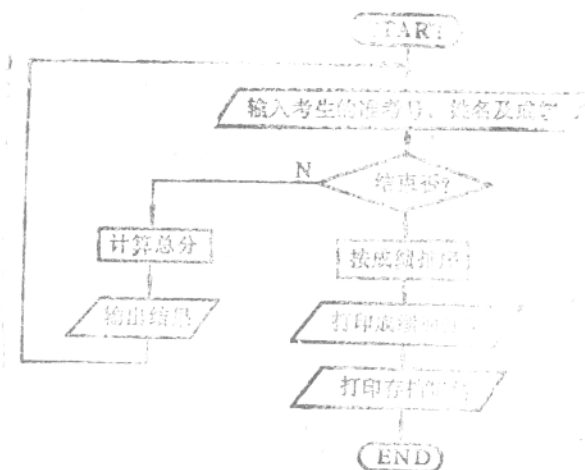


图 1-6

正确的算法是设计BASIC程序的前提,因此,编制程序前,必须了解所采用的算法的特点及应注意的问题;选取算法时,在满足精度要求的前提下,还需考虑计算量要尽可能小及便于程序的编写两个方面。选取算法往往是程序设计中的一个难点,有些甚至要通过比较才能确定一个较好的算法。

2.3 结构模块化

在解决大型问题时,模块化程序设计应用十分广泛。所谓模块化结构,就是对具体问题进行全面分析,按其功能分解为若干部分,每一部分由一个子程序去完成。此时主程序主要由GOSUB语句组成,可看作是控制模块,而子程序则看作功能模块,不同的模块彼此互相独立。主程序好比是总调度,调各个功能模块去分别完成各部分的功能。

采用模块化程序设计，可以使程序具有“积木式”结构，类似于堆积木的办法，最后将各模块拼装成解决实际问题的程序。

模块化程序设计具有以下特点：

(1) 编程方便。各个模块是相互独立的，可分别由不同的人员编制。在统一约定好各模块之间接口的前提下，每个模块具体实现的细节由设计者自行确定。

(2) 便于修改。由于程序模块化，当整个程序实现的任务有所变更，或因发现错误而要对程序进行修改、调整时，修改工作只涉及少数的模块，且局部的更改不致于影响程序的全局。在模块内部进行调整和改动，将不会涉及到模块的外部。

(3) 容易阅读。各个模块功能专一，且各模块间的接口关系非常明确，从而大大降低了阅读和理解整个程序的难度。

(4) 易于验证。由于模块化程序设计的思路十分清晰，只要验证各模块的正确性，并根据各模块功能的实现，即可验证整个程序的正确性。

2.4 编写程序

明确了问题的要求及解决方法之后，可以将总体设计中的粗框图逐步细化，然后根据细框图一步一步地编写程序。

程序设计是极其慎密的工作，不能有丝毫的差错，而且还应讲究程序的结构和风格，尽可能地使程序优化。

程序无论大小，不能编写完毕后就上机计算，应该认真地进行“静态”和“动态”检查，只有经过反复地调试，确认能实现预定的功能后，才能付诸使用。

第二章 BASIC程序设计技巧

分支程序、循环程序和子程序，是BASIC程序中的三个重要内容。本章讨论分支、循环、子程序的设计方法与若干常用技巧。

§1 分支的设计方法与技巧

IF—THEN 语句可以根据某种条件改变程序的执行次序，从而实现程序的分支。含有IF—THEN语句的程序称为**分支程序**。

判断条件是进行分支的依据，每个分支中一定要有特定的计算公式或处理方法。设计分支程序的关键是构造条件，形成分支。

根据程序中所含分支个数，分支程序可分为**单分支程序**（具有两个分支的程序）和**多分支程序**（具有两个以上分支的程序），多分支程序的设计较难掌握，且分支技巧的应用起着决定程序优劣的作用。

1.1 框图的绘制方法

框图是描述解题过程的图形，它是由一些特定的几何图形、文字说明及指向线构成，具有形象、直观的特点，便于程序的编写、检查、修改、阅读。

运用基本框图符号（图2-1）绘制框图，一般是按下列步骤进行：

步1 认真分析计算问题，确定计算公式和计算步骤。

步2 根据计算方案将每个计算步骤用一个框描述出来。一般方法是：先描述计算或处理的核心部分，再逐步展开，反复进行修改、补充，直至所有计算步骤均描述完毕为止。

步3 对于不太复杂的问题，直接绘制细框图，并作为编写程序的依据。对于较复杂的问题，往往是先画出粗框

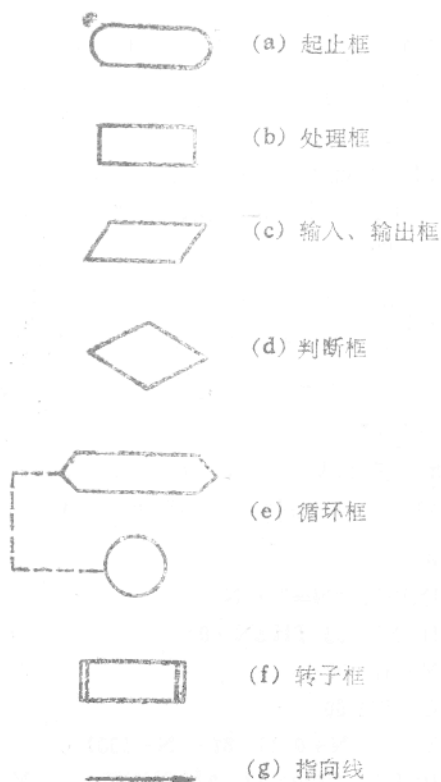


图 2-1

图，再对各部分逐步加工处理成细框图。

步4 将框图与计算公式反复核对，确保框图准确无误。

1.2 单分支的设计

分支程序中的最简单的情形是利用一个IF—THEN语句，形成两个独立的分支。每个分支各自完成不同的运算或处理，然后汇合到一处。

由于单分支程序中仅含一个IF—THEN语句，因而只要注意条件满足和条件不满足时的程序流向，就可以编写出正确的程序。

例2.1 某批发公司规定，购买某种货物100件（含100件）以下，每件单价87元，购买100件以上，超过部分按95折优惠，即

$$Y = \begin{cases} 87 \times N & (N \leq 100), \\ 87 \times 100 + 0.95 \times 87 \times (N - 100) & (N > 100), \end{cases}$$

其中， N 为购物件数， Y 为总价。试编写计算购物总价的程序（ N 由键盘输入。）

分析：在计算公式中，当 N 的值不同时（ $N > 100$ 或 $N \leq 100$ ），计算公式是不同的，由关系式 $N > 100$ 是否成立而确定。

程序

```
10 INPUT "N=" ; N
20 IF N>100 THEN 50
30 Y=87*N
40 GOTO 60
50 Y=87*N+0.95*87*(N-100)
60 PRINT "TOTAL AMOUNT=" ; Y
70 END
```

RUN

N=153

TOTAL AMOUNT=17691.45

程序中的40语句是必不可少的，它的作用是将两个分支分隔开，以免同时进行两个分支中的运算或处理。根据实际问题的需要，分隔的语句还可为END语句或STOP语句。

从例2.1中还可以看出，经过一次判断之后，只能在两个分支中选取其中的一个分支执行。

1.3 多分支的设计

IF—THEN语句可以使程序形成两条支路，如果要使程序分成多路分支时，往往需要用多个(两个以上)IF—THEN语句。

例2.2 某班学生按顺序编号为1、2、3、…，除去编号为1和2的两名学生外，其余学生的编号之和能被100整除。已知学生人数多于37人，编号之和小于1000。编一程序，求出学生的人数。

分析：设学生人数为 N ，除去编号为1和2的学生外，其余学生编号之和应为 $100K$ (K 为正整数)，于是有

$$3 + 4 + 5 + \cdots + N = 100K, \quad (1)$$

其中， $N > 37$ ， $K < 10$ 。

利用等差数列求和公式，(1)式可写为

$$\frac{(N-2)(N+3)}{2} = 100K$$

$$\text{即 } (N-2)^2 + 5(N-2) = 200K \quad (2)$$

(2)式中，右端项是5的倍数，左端第二项也是5的倍数，因此第一项 $(N-2)$ 应是5的倍数。因为 $N > 37$ ，故