

朱国江

中华学习机

编程技巧

高等教育出版社

12

7/1

中华学习机编程技巧

朱 国 江

东南出版社

内 容 简 介

本书系统讲述 BASIC 语言程序设计和技巧, 内容广泛, 取材新颖, 深入浅出。

全书以国家教育委员会指定推广的优秀机型——中华学习机(即将大量生产, 与 APPLE-Ⅱ 完全兼容)为选型机, 兼顾 PC-1500 机、紫金-Ⅱ 机。

本书适用于从事微型计算机应用、管理人员和广大青少年及微机爱好者自学, 也可作为各种类型培训班的教学参考书。

中华学习机编程技巧

朱国江 编著

责任编辑 黄 丽 荣

* * *

高 等 出 版 社 出 版

(北京西郊白石桥路46号)

北京邮电学院印刷厂印刷

新华书店北京发行所发行 全国各地新华书店经售

* * *

开本: 787×1092 1/32 印张: 5.125 字数: 108千字

1988年2月第一版 1988年2月第一次印刷

印数: 1—15,000 定价: 1.25元

ISBN7-5029-0097-7/TP·0003

前 言

近年来,有关计算机的书颇受欢迎,这对我国计算机的普及和应用,起着良好的推动作用。

但是,目前还有不少单位的微型计算机尚未充分发挥其应用的作用,其主要原因是对各种各样的微型计算机开发的还不够。为了帮助广大初、中级计算机应用人员以及学过 BASIC 语言的中、小学生掌握编制程序的方法和技巧,作者根据近年来从事计算机实践和教学经验,拟就上述问题作些补充,力求熔实用性、通用性、技巧性、趣味性于一炉,在编程方法、设计技巧和实际操作诸方面作一些探索性尝试。

本书强调普及和提高结合,实用和技巧兼顾,以满足不同层次读者的需求。

本书的资料主要来自作者多年编程和教学实践,同时还吸收了国内外的先进经验。书中阐述的方法、技巧、思路对其它程序设计语言都有参考价值,而实用程序可以直接付诸使用或稍作修改后引用。

在本书编写过程中,得到南京大学大气科学系邹进上教授的鼓励与支持,并提出不少宝贵意见,在此谨表谢意。

由于作者学识疏浅,成书仓促,错误缺点在所难免,敬请广大读者不吝指教。

编 者

1987年8月25日

目 录

前言

一、什么是程序设计和技巧	(1)
1. 什么是程序设计	(1)
2. 各种情况	(1)
3. 质量标准	(2)
4. 出发点	(3)
5. 基本方法	(4)
6. 基本技巧	(4)
7. 实例分析	(6)
二、程序设计步骤和方法	(18)
1. 了解问题性质	(18)
2. 绘制程序框图	(19)
3. 强调设计思想	(20)
4. 建立数学模型	(23)
5. 选用计算方法	(24)
6. 注意近似处理	(26)
7. 考虑通用合理	(27)
8. 防止程序出错	(28)
9. 划分功能模块	(29)
10. 讲究设计技巧	(31)
三、程序连接的方法	(34)

1. 如何安排总控.....	(34)
2. 单一程序的连接方法.....	(36)
3. 两个不同程序的连接方法.....	(40)
4. 重订行号的方法.....	(45)
四、节省内存和提高速度的方法.....	(48)
1. 程序存贮的基本知识.....	(48)
2. 节省内存的措施.....	(52)
3. 提高速度的方法.....	(58)
4. 使用机器语言子程序.....	(62)
五、编制程序应注意的问题.....	(67)
1. 关于变量的初始化问题.....	(67)
2. 数组元素的下标要恰当.....	(69)
3. 程序的正确性要经得起时间的考验.....	(70)
4. 防止陷入死循环.....	(71)
5. 矩阵输出注意打印语句的安排.....	(72)
6. 妥善处理保留字.....	(73)
7. 重新读数不要忘记 RESTORE.....	(74)
8. 能用简单变量的就不用下标变量.....	(76)
9. 注意程序的清晰.....	(78)
10. 判断循环变量是否改变.....	(80)
11. 数值计算不要超过机器规定范围.....	(81)
12. 注意使用扩展 BASIC 语句.....	(82)
13. 尽量不用 GOTO 语句.....	(84)
14. 尽量少用乘方运算.....	(86)
六、打印输出的技巧.....	(90)
1. 输出空格函数 (SPC函数) 的使用.....	(90)

2. 输出定位函数 (TAB函数) 的使用	(91)
3. 数据的列印输出	(93)
4. 打印机输出控制	(95)
5. 打印输出结果的对齐处理	(98)
6. 字符的放大打印	(104)
七、程序调试技巧	(107)
1. 不同机器的差异	(107)
2. 尽量避开乘方运算	(110)
3. 作适当的精度调整	(111)
4. 如何处理小数步长	(112)
5. 屏幕抄写的方法	(113)
6. 程序编辑举例	(114)
7. 执行过程的自动跟踪	(116)
8. 输入数据的跟踪检查	(118)
9. 几个常用技术	(120)
八、BASIC 语言的编程技巧	(123)
1. 菜单技术	(123)
2. 人机对话方式	(125)
3. 巧用符号函数 SGN(X)	(128)
4. 巧用逻辑表达式	(131)
5. 循环体中判断语句的巧安排	(133)
6. 多用途的 INT(X)函数	(134)
7. 字符串函数的应用技巧	(139)
8. 绝对值函数的一个应用	(142)
9. 多重循环程序的设计技巧	(145)
参考文献	(156)

一、什么是程序设计和技巧

程序设计是以完成预定任务为目的，这是程序设计最基本要求。然而常常有这样的情况，一个实际问题，不同的人，不同的思路，编制的程序却是多种多样。这就要讨论程序设计的标准是什么？要依据什么原则来制定这些标准？用什么方法使设计的程序达到标准。在程序设计中究竟有哪些基本方法，又有什么技巧等等。

1. 什么是程序设计

人们根据问题的要求，事先安排好的处理方法和步骤，称作程序。一个程序好比是一篇文章，进行程序设计的过程就是人们按照计算机语言所规定的语法和功能书写文章的过程，这个过程一般来说有以下几个阶段：提出问题，需求分析，确定模型，功能设计，绘制框图，编写程序，动态调试，交付使用。

2. 各种情况

考察人们编写的各种程序，会发现程序的编制情况多种多样，归纳起来有：

- 设计思想简单或复杂；
- 程序冗长或简短；
- 运行速度快或慢；

- 精度高或不够；
- 占用内存多或少；
- 计算方法优或劣；
- 使用方便或相反；
- 阅读修改程序难或易；
- 程序移植性好或差；
- 用计算机协助编程或没有。

从上面情况看出，程序设计灵活性很大，技巧性很高。

3. 质量标准

那么，什么是一个好的程序呢？不同的人有不同的观点，不同的课题要求的标准也不一样，所以很难有一个统一的标准。然而，根据计算机用于数据处理的性质和特点，一个好的程序必须满足以下几个方面。

(1) 正确性 编制程序的目的，就是完成解题任务，这是程序设计的最基本要求。一个正确的程序必须保证在任何情况下，都能得到正确的运行结果。正确性还应包括纠错能力，程序结构精巧，使用灵活方便，结果正确并能达到预定的精度要求。

(2) 通用性 通用性包括几方面内容：一是一个程序对同一类问题的不同情况都可适用；二是不能因人、因时、因地而异；三是一个程序稍作修改后，能完成不同功能或满足不同解题要求。

(3) 可靠性 对一个系统来说，可靠性和正确性同样至关重要。一个程序不能运行一次正确，多次运行就失败；或者调试时可行，但进入反复操作后出问题；或者数据量较少

时正确，一旦大批量输入时出现异常情况，所有这些都体现了程序的可靠性。一个好的程序，应该经得起长时间及大批量数据的考验。

(4) 实用性 程序设计应具有实用价值，讲究经济效益。除此以外，程序设计应考虑到如何使别人使用时感到方便、灵活，愿意用自己设计的程序。程序的易读性要好，又便于修改。因此实用和面向用户，是评价一个程序优劣的重要标准。

4. 出发点

有了程序设计的标准，对程序设计的评价就有了依据，一般来说，我们希望编制的程序结构清晰，易读易改，便于移植，结果精确，使用方便。同时在程序设计中还有几点应予注意：

- 解决一个实际问题的方法是多种多样的，设计思路也不尽相同，但我们的立足点还应放在节省机时和减少存贮空间上，只有当两者不能兼顾时，可根据实际需要而偏重一方。

- 不应盲目追求快的响应速度，而损失掉许多别的优点，或者为了提高速度而在其它方面带来不应有的麻烦。要切合实情，量力而行，不要得不偿失。

- 也不要过分追求减少内存空间，使程序易读性很差，带来阅读、调试、修改的困难。我们的出发点是，为确保程序运行的可靠及方便用户，要不惜代价增加一些必要的语句或注释，同时也要尽可能地节省内存。

- 要讲究技巧，注意程序的优化。由于在数据、信息处

理中，是人-机直接交互，因此，应用一些技巧效果会被人直接感受到、体会到，而当人们在实际中切身感到好处时，技巧的成效就越发明显。当然，一个程序也不是越巧越好，拐弯越多越好，严防弄巧成拙，适得其反。

5. 基本方法

程序设计技术性强，也很灵活，有些问题的设计还很复杂，没有统一的模式，但是，根据程序设计的基本思路，性质及过程，总还能总结出一些比较普遍的方法。现简叙如下：

程序设计的基本方法是枚举、归纳和抽象。

建立在抽象原则基础上的一种常用设计方法是逐步求精。

对于规模较大的问题的程序设计，是采用结构模块化。

逐步求精法是一种适应性很强并被广泛采用的程序设计方法，其要点是：

- 对问题全局进行分析；
- 建立抽象模型；
- 确定总体模块图；
- 明确各子模块功能及相互关系；
- 编制、调试每一个子模块；
- 完善整体结构。

6. 基本技巧

程序设计的技巧，包括的内容是十分广泛的，主要有语言编程技巧，程序调试技巧，文件使用技巧，软件加密技

巧，机器使用技巧，功能扩展技巧，数据维护技巧等等。对于这些技巧，将在本书其它章节逐一叙述。

在用BASIC语言进行程序设计时，主要有下面四种技巧：

(1) 简单程序设计 这是初学者程序设计入门的必经之路，它包括数据输入、进行计算、输出结果、结束停机四个内容。由于程序是按行号由小到大顺序执行的，因而比较简单。常用的语句有 LET, INPUT, READ/DATA, PRINT, REM, END 等。

简单程序设计虽然容易，但它能勾画出一个程序的全貌，描述了程序的主要结构，即变量取值、完成计算并保留结果、提供数据、显示结果和结束程序这五个部份。明确这一点，对我们设计一个程序是十分重要的，因为任何事物都是由低级到高级，由简单到复杂。

(2) 分支程序设计 实际问题比较复杂，解决问题的程序常常带有分支，因此，我们十分需要一种具有判断能力的控制转移手段，以便对程序进行有效的控制。这就出现了分支程序设计，而计算机按给定条件进行比较、分析、判断后决定程序的走向。常用的有无条件转向语句 GOTO，条件控制语句 IF/THEN，恢复数据区语句 RESTORE，控制转向语句 ON/GOTO, ON/THEN, ON/GOSUB，以及转子和返回语句 GOSUB/RETURN 等等。

我们可以根据一定的条件，选择不同的语句，控制程序的流向，以解决各种各样的分支问题。必须着重指出，条件语句的使用，在 BASIC 语言中占有很重要的地位，同样，其技巧性也是很强的。

(3) 循环程序的设计 循环程序在我们解决实际问题的

时候，有着极其广泛的应用。循环的技巧可以实现多个数据输入和大量重复的计算。利用循环的方法进行程序设计，既可节省存贮单元又可以提高计算效率，这是程序设计中最主要的和最常用的方法。循环语句比较简单，就是FOR/NEXT，但应用时技巧特强，且十分灵活。

不要忘记，人们利用计算机快速准确且具有逻辑判断的特点，为自己解决那些需要反复进行，而在处理形式上又完全相同的工作时，主要是依靠循环程序来进行。

循环是 BASIC 语言和一切高级语言的核心。

(4) 调用子程序的设计 子程序是程序结构的一种形式，使用十分广泛。子程序在程序设计中常常用来表示那些在功能上具有一定独立性并在程序的不同部位要被经常使用的程序段，用以简化设计并使程序结构清晰。

对于一些重复计算的问题，编制一个相对独立的程序段，在主程序的适当位置安排调用点，GOSUB/RETURN 语句能保证正确调用和自动返回。

对一个规模较大或需要完成多种功能的系统来说，往往需要分段设计，然后连接起来，子程序的技巧可以解决程序段之间的连接问题。使整个程序或系统成为一个有机结合的整体。

程序设计的基本技巧，基本上就是上面四个方面。在这里不拟逐一详述，但须指出一点，即实际问题的设计常常是四种基本技巧的有机组合。

7. 实例分析

一个好的程序应包括下述指标：

- 运行结果正确;
- 有足够的精度;
- 运行速度较快;
- 占用内存较少;
- 简单清晰易懂。

而所有这一切常常取决于好的解题方法。

下面我们通过两个实例的求解，来探讨一下不同的解题方法对程序质量的影响。

例 1，求：

$$Y = 1 - 2^2 + 3^2 - 4^2 + \cdots + 99^2 - 100^2$$

方法 1，改写原式：

$$\begin{aligned} Y &= (1^2 + 3^2 + 5^2 + \cdots + 99^2) - (2^2 + 4^2 + 6^2 + \cdots + 100^2) \\ &= A - B \end{aligned}$$

式中： $A = 1^2 + 3^2 + \cdots + 99^2$

$$B = 2^2 + 4^2 + \cdots + 100^2$$

问题归结为求奇数平方和及偶数平方和的差值。

求和最好用循环语句，循环外设置初值为零，循环体求平方和，即 $S = S + N \wedge 2$ ，最后安排 $Y = A - B$ 并打印，故有程序 No1：

```
10: REM NO. 1
20: A=0
30: FOR N=1 TO 99
   STEP 2
40: A=A+N^2
50: NEXT N
60: B=0
```

```

70:FOR N=2TO 100
  STEP 2
80:B=B+N^2
90:NEXT N
100:Y=A-B
110:LPRINT "Y=";Y
120:END

```

Y=-5050

这个程序结构清楚，层次分明，易读性好。不足之处，占用内存多（120步），速度慢（24秒）。

方法2，将程序 No1 改写成 No1a:

```

10: CLEAR : REM NO.
  1a
20:FOR N=1TO 99
  STEP 2:A=A+N^2
  :NEXT N
30:FOR N=2TO 100
  STEP 2:B=B+N^2
  :NEXT N
40:LPRINT "Y=";A-
  B

```

Y=-5050

程序 No1a 和程序 No1 有以下几点区别:

①两个程序设计思想完全一样，程序 No1a，改用扩展 BASIC 语句，因而比较简短，且节省内存（变成 85 步）。

②运行时间缩短了, 由 24 秒变为 22 秒。

③用 CLEAR 语句, 省去两个循环的初始化条件, 即删去 $A=0, B=0$ 。

④程序 №1 的 100、110 两句, 合并成 №1a 的 40 句。

⑤省去 END 语句, 一般微型计算机都是允许的 (FORTRAN 语言不允许省去结束程序语句 END)。

⑥为了提高速度, 可以把 REM 注释语句放在程序的最后, 因为它不是执行语句, 不影响机器运行速度。

⑦程序 №1a 的第 10 句, 若写成:

10:REM:№1a: CLEAR 则重复运行程序 №1a, 每次结果会不同。因为 REM 是非执行语句, 这样 CLEAR 这一句将不执行, 存储在 Y 中的值没有被清除掉, 重新运行时 Y 值包括了前一次的结果。

由于上述两个程序设计思想完全相同, 后者虽然节省了一些内存, 速度也有微小提高, 但程序质量上并没有实质性的提高。还需要继续改进。

方法 3, 考察原式: $Y = 1 - 2^2 + 3^2 - 4^2 + \cdots + 99^2 - 100^2$

从数值上看, 它们是顺序递增的, 每一项均为 N^2 ($N = 1, 2, 3, \cdots 100$)。

从符号上看, 每一项又是 “+”、“-” 相间的, 其通式为 $(-1)^{n+1}$, n 是奇数时为 +1, n 是偶数时为 -1。

故有以下通项公式:

$$a_n = (-1)^{n+1} \cdot n^2 \quad (n = 1, 2, 3, \cdots 100)$$

$$\therefore \text{原式为: } Y = \sum_{n=1}^{100} (-1)^{n+1} \cdot n^2$$

见程序 №2:

```

10:REM NO.2
20:Y=0
30:FOR N=1TO 100
40:Y=Y+(-1)^(N+1)
   *N^2
50:NEXT N
60:LPRINT "Y=";Y

```

Y=-5049.999999

从上面运行结果可以看出，这个程序除了比较简洁，内存少（72步）外，没有什么优点，相反，缺点倒是明显的，结果不精确，速度慢。同时，本程序中用了负数的乘方运算，有的机器是不执行的，如SHARP PC-1211。

其它机器如SHARP PC-1500，APPLE-II，紫金-II，中华学习机等均能执行本程序，但精度却不够。正确值为-5050。

方法4，为了避开上述解法中负数的乘方运算，对符号和负数乘方采用下面的方法处理，见程序№3清单：

```

10:REM NO.3
20:Y=0:T=-1
30:FOR N=1TO 100
40:T=(-1)*T:Y=Y+T
   *N^2
50:NEXT N
60:LPRINT "Y=";Y

```

Y=-5049.999999