

数据文件编程技术

SHU JU WEN JIAN BIAN CHENG JI SHU

梁晋清 杨继刚 编著 上海交通大学出版社

515	07012		+ 100,000	
07205		32		
(290		130,000 +		220,00
		189		338,000
216 × 138mm				
000		+ 248,00	+ 224,	20
17		399,000 +		
				163 + 208
160,000	1856 -		608pp	

数据文件编程技术

梁晋清 杨继刚 编译

上海交通大学出版社

内 容 简 介

本书主要介绍数据文件的使用以及有关数据文件应用的程序设计技术。本书共七章：前三章介绍有关的程序设计方法与技巧；第四章和第五章阐述顺序数据文件及其应用；第六章和第七章阐述随机数据文件及其应用。每章均附有自我测验题和答案，供读者在学习时随时了解自己的学习情况。

本书可作为 BASIC 语言后继课程的教材，也可供有关的大专院校的师生及科技人员参考。

本书还配有一片在 IBM PC 微型计算机上使用的软盘，供读者在学习时使用，或作为应用数据文件的工具。

数据文件编程技术

上海交通大学出版社出版
(淮海中路 1984 弄 19 号)

新华书店上海发行所发行
常熟文化印刷厂 排版印装

开本 787×1092 毫米 1/32 印张 11.75 字数 266000
1986 年 月第 1 版 1986 年 月第 1 次印刷

印数 1—4500

统一书号：15324.203

科技书目：146—277

定价：2.00 元

前 言

本书的目的是介绍数据文件的使用和管理技术,以及有关数据文件的程序设计方法。书中的内容可作为 BASIC 语言的后继课程,供学过基本 BASIC 语言的读者使用。它不仅可作为一本自学教材,而且可以作为各种计算机短期学习班的教材,对大学中开 BASIC 程序设计课程的教师也有一定的参考价值。

全书共分七章。书中通过大量例子,由浅入深地介绍了各种数据文件的应用技术。每章附有自我测验题和答案,读者通过解答这些题目,可以随时掌握自己所学内容的程度。只要读者认真地读了全书,领会了书中的要点,认真完成了每道习题,就可以掌握一整套数据文件的编程和应用技术,也就真正掌握了计算机的实际应用方法。在此基础上,读者就可以编写或修改自己所需的程序,自由地应用到实际工作中去。

书中用到的全部例子均是用 IBM PC 的 DOS 操作系统支持下的 BASIC 语言编写的,这些程序也适用于 IBM PC 的 CP/M 操作系统支持下的 BASIC 语言。其他版本的 BASIC 语言的数据文件程序设计是类似的,仅与本书中使用的部分语句有些差别。如果读者用的是其它种类的 BASIC 语言,可与有关的 BASIC 语言手册配合使用。书中的所有程序只要稍加修改就可在其它微型计算机上使用。

如果读者是第一次使用 IBM PC 及其 BASIC 语言,可先读下面两本书:

1. 《IBM-PC DOS 操作系统使用指南》;

2. 《IBM PC BASIC 语言手册》。

由于我们学识水平有限,工作经验不足,再加上时间的仓足,书中难免有错误和不足之处。敬请读者批评指正。

编 者

1985 年 9 月

目 录

第一章 提高程序的逻辑性和可读性	1
§ 1-1 BASIC 语言	1
§ 1-2 保守的编程方法	2
§ 1-3 程序设计的风格	2
§ 1-4 提高程序的可读性和程序的模块化	3
§ 1-5 自顶向下组织	3
§ 1-6 注释语句	4
§ 1-7 转移语句	5
§ 1-8 引导模块的格式	7
§ 1-9 引导模块之后的模块	10
§ 1-10 子程序	12
§ 1-11 空格的使用	15
§ 1-12 其他提高程序可读性的措施	17
§ 1-13 节约空间和加速程序运行	19
第一章自我测验题	20
答案	21
第二章 BASIC 语言中的重要语句	22
§ 2-1 变量名称	22
§ 2-2 字符串变量	23
§ 2-3 READ 和 DATA 语句	26
§ 2-4 INPUT 语句	28
§ 2-5 LINE INPUT 语句	31

§ 2-6	连接字符串	31
§ 2-7	条件转移语句	32
§ 2-8	ASCII 字符和字符串的比较	35
§ 2-9	LEN 函数	38
§ 2-10	字符串处理的方法	40
§ 2-11	字符串的搜索	47
§ 2-12	多路转移语句	50
§ 2-13	循环语句	52
§ 2-14	多语句行	54
第二章自我测验题		55
答案		56
第三章 建立数据输入和错误检查程序		58
§ 3-1	数据域的长度	58
§ 3-2	检查输入数据的长度	60
§ 3-3	给输入的数据补空格	62
§ 3-4	从一个字符串中去除空格	66
§ 3-5	检查用户是否输入数据	68
§ 3-6	数据项的替换	73
§ 3-7	VAL 函数在数据检查中的作用	75
§ 3-8	用 STR\$ 将数值转换成字符串	78
§ 3-9	检查非法字符	79
§ 3-10	数据输入和检查程序	81
第三章自我测验题		91
答案		96
第四章 顺序文件的读与写		101
§ 4-1	数据存放在磁盘上	101
§ 4-2	打开数据文件	103

§ 4-3	关闭数据文件	105
§ 4-4	将数据送入顺序文件	106
§ 4-5	单个数据文件的写与读	115
§ 4-6	从磁盘中删除一个文件	120
§ 4-7	在一个程序中同时对几个数据文件存取	121
第四章自我测验题		135
答案		145
第五章 顺序文件的实用程序		159
§ 5-1	复制一个数据文件	159
§ 5-2	向顺序数据文件加入数据	162
§ 5-3	修改文件中的数据	163
§ 5-4	编辑、删除和插入数据	175
§ 5-5	合并两个顺序数据文件	187
§ 5-6	与顺序数据文件的有关问题	198
§ 5-7	写英文信件的程序	200
第五章自我测验题		207
答案		216
第六章 随机文件的读与写		227
§ 6-1	随机存取数据文件	227
§ 6-2	随机文件的初始化	229
§ 6-3	缓冲器的域	231
§ 6-4	随机文件的简单读写操作	233
§ 6-5	向随机文件加入数据	244
§ 6-6	用随机文件存放数值型数据	247
§ 6-7	规定记录长度的随机文件	255
§ 6-8	随机文件的拷贝(复制)	265
§ 6-9	编辑随机存取文件的记录	272

§ 6-10 将顺序文件转换成随机文件	277
§ 6-11 显示通用文件内容的程序	282
第六章自我测验题	282
答案	287
第七章 随机文件的应用	294
§ 7-1 随机文件应用于库存管理	294
§ 7-2 随机文件应用于财务管理	303
第七章自我测验题	332
答案	337
总自我测验题	347
答案	351
附录 A ASCII 字符表	363
附录 B 一些语句的说明	365
参考资料	367

第一章 提高程序的 逻辑性和可读性

通过本章学习要做到:

1. 能用自顶向下的程序设计方法编制程序;
2. 能用注释语句写一段程序的说明;
3. 掌握编写打印语句的规律;
4. 掌握节约存贮空间的办法。

§ 1-1 BASIC 语言

BASIC 语言是在六十年代由美国达特姆斯学院 (Dartmouth College) 研究出来的。开始仅包括初学者所需的功能。后来,其他大学、计算机厂商、研究所在实际应用中加入了各自所需的内容,扩大了 BASIC 语言的功能,发展为扩展 BASIC 语言、超级 BASIC 语言、MBASIC 或者 BASIC PLUS 等。一九七八年,美国国家标准学会制订了一个 BASIC 语言的工业标准,此标准也可称为最小 BASIC。尽管有了这个标准,我们现在仍可看到各式各样的 BASIC 语言。它们虽各有特点,但基本上是相象的。

在微型计算机领域广泛使用的 BASIC 语言版本,一般都是由 Microsoft 公司研究出来的,称为 Microsoft BASIC。在各种微型计算机上,有不同版本的 Microsoft BASIC,其中最广泛使用的版本是 BASIC-80 和 BASIC-86。

在本书中,所有的程序均是用 IBM PC 上的磁盘 BASIC 语言写成的。程序的大部分采用适合于所有 Microsoft BASIC 版本的语句写成的,可在各种型号的微型计算机上运行。在本章中,我们仅用 BASIC 语言中最简单的语句编写程序。

§ 1-2 保守的编程方法

与数据文件有关的程序一般较长、较复杂,编写这类程序时,可以使用保守的编程技术。保守的编程技术就是尽量采用熟悉的语句,复杂的语句必须在完全弄清楚的情况下才可使用。特定计算机的打印和图形功能尽量避免使用,以保证程序的通用性,少出差错。

现在计算机型号很多,所用的 BASIC 语言也各有特点。用保守的编程方法,尽量采用各种 BASIC 语言版本中的共同部分,可使程序较方便地从一种计算机移植另一种计算机上运行,从而减少了程序移植时的修改工作。

§ 1-3 程序设计的风格

这里所谈的程序设计风格是指:程序的可读性和程序的模块化结构。

程序的可读性,可以从程序的安排,程序的书写格式,以及程序的模块化结构等多方面考虑。程序的书写格式愈清晰,程序的作用就愈明显,程序的可读性也就能提高。

所谓程序的模块化就是将整个程序分成若干模块,每个模块包括一个主要功能。使程序模块化,实际上就是将问题分解,每个程序模块解决一个问题,实现分而治之,使编程工

作效率提高。程序的模块化同样也使程序的调试变得方便，我们可以先找出程序出错的模块，然后再在模块中查错。

程序设计的风格还包括：自顶向下的组织、自顶向下的程序设计方法，注释语句的使用等等。这些将在后面讨论。

§ 1-4 提高程序的可读性和程序的模块化

要提高程序的可读性，必须事先考虑好要写的程序，用逻辑流程安排好，再用一些特殊的格式把写好后的程序列出。程序书写的格式很重要，书写格式好，可增加程序的可读性，别人一读程序就可知道这个程序的作用。一个现在常常被忽略的问题是尽量不考虑存贮空间的节省和提高程序的执行速度。为了提高程序的可读性，有时不得不牺牲一些存贮空间和执行速度。在本章稍后，我们还要介绍缩短程序和加速程序执行的方法。

§ 1-5 自顶向下组织

在安排程序结构时，首先要考虑程序的主要功能。一般程序的主要功能有：

1. 数据输入；
2. 数据分析；
3. 计算；
4. 文件更新；
5. 编辑；
6. 报表生成。

设计程序时，首先从问题出发，将程序分成若干模块，每

个模块代表一个主要功能。程序由模块组成，模块由子模块组成。每个子模块可以是一个过程或一个计算。子模块的大小由编程者决定。

自顶向下组织，就是从程序的整体出发，按功能分成模块，再将模块根据要完成的任务分成子模块，各块任务明确后就可写出程序。

§ 1-6 注释语句

在一个程序中，模块与模块之间，子模块与子模块之间，可用注释语句或空白行分开。

一般而言，程序的可读性要求对注释语句的使用并无限制，但不能滥用。一个空白行可用单引号‘ ’代替 REMARK 产生，或者在行号之后加一个冒号:，也可以在行号之后加一个 REM。

一个程序模块、子模块或子程序的开头用注释语句作一说明，而在结束时用一个空白行表示结束。如下例：

```
100 REMARK    DATA ENTRY MODULE
110 REMARK **** READ DATA FROM DATA
    STATEMENTS 9000-9090
120  ,
130  :
140 REM
.
.
190  ,
200 REMARK COMPUTATION MODULE
```

统一使用注释语句可提高程序的可读性。注释语句可用 REMARK 开头,也可用 REM 开头,一个程序中要么都使用 REM,要么都使用 REMARK。空白语句也要统一,要么都用一个单引号‘ ’,或者都用冒号:,或者都用空白的 REM 或 REMARK。若须将某些注释语句与别的注释语句区别开来,可在 REM 之后加几个星号*,如上例中 110 行;另一个办法是在 REM 之后空几个空格再加上注释内容,如上例中的 100 行。这些方法使注释语句同其它 BASIC 语句区别开来。

注释语句也可加在其他 BASIC 语句行中,组成一个多语句行。但是,注释语句必须是一行的最后一个语句。这种注释语句用来对某个特定的语句作说明,一般总与前面的语句空开几格,以示区别。如下例:

```
220 LET C(X) = C(X) + U: REM ** COUNT U-  
    NITS IN C ARRAY
```

```
240 LET T(X) = T(X) + C(X): REM**INCREASE  
    TOTALS ARRAY
```

可以自由地使用注释语句来分隔程序的各模块和各个子模块,可以使用注释语句说明程序的作用。但是,注释语句不要滥用。如语句 LET C = A + B 就不需要用注释语句说明。注释语句是用来说明一个模块或一个子模块的功能,而不是对一个模块或一个子模块的详细说明,因此,注释语句要求简单明了,能说明问题即可。

我们这里介绍的只是一种方法,供读者学习参考。

§ 1-7 转移语句

转移语句 GOTO 是 BASIC 语言中最有争议的语句,因

此，纯粹的结构程序设计倡导者认为应禁止使用无条件转移语句。但实际上往往行不通。一个较为可行的办法是限制使用无条件转移语句，规定 GOTO 和 GOSUB 语句只可转向标号比本身标号大的语句行，即只允许向高标号方向转，而不允许向低标号方向转。这种限制也适用于条件转移语句 IF…… THEN EN，但有时有例外情况。如：

```
140 GOTO 210
150 IF X<Y THEN 800
160 GOSUB 8000
```

无条件转移语句 GOTO、条件转移 IF……THEN、转子程序语句 GOSUB，均不要转向仅包含注释语句的行。因为这些注释语句在内存不够时就不能擦掉，而要先修改转移语句中的行号。下面各举一个例子予以说明，

好的例子

```
150 GOTO 300
```

.
.
.
.
.

```
299 REM DATA ENTRY
```

```
300 LINE INPUT "NAME?";N$
```

不好的例子

```
150 GOTO 300
```

.
.

```
300 REM DATA ENTRY
310 LINE INPUT "NAME?";N$
```

§ 1-8 引导模块的格式

BASIC 程序的第一个模块应包含一个程序的简单介绍、用户指导、使用的全部变量的变量表及常量、变量和数组的初始化。这个模块称为引导模块。

一个程序的最前面应该是程序名的注释语句。程序的名字要取得有意义，能够说明程序的作用。程序名之后应是编程序者的姓名和完成此程序的最后日期，最好再注上所用计算机的型号及软件的版本。当一个程序修改后，注释语句中的有关内容亦要作相应的修改。下面是一个程序头的例子：

```
100 REMARK PAYROLL SUBSYSTEM
110 REMARK COPYRIGHT CONSUMER PROG-
    RAMMING CORP 1979
120 REMARK
130 REMARK HP 2000 BASIC
140 REMARK MODIFIED FOR IBM PC BY J.
    BROWN
150 REMARK USING DISK BASIC VERSION 1
    10
160 REMARK
```

程序名说明之后，可用几个打印语句或注释语句，对程序的功能作一个简短的说明，尔后是使用指导。有些程序给用

户一个选择，以决定是否打印使用指导。如果使用指导的文字很长，最好放在程序的最后，用一个子程序专门打印使用说明。这样做的好处是在列表打印程序时，不需每次将长长的说明打出来。

```
170 REMARK THIS PROGRAM WILL COMPUTE
    PAY AND PRODUCE PRINTED PAYROLL
180 REMARK REGISTER USING DATA ENTERED BY OPERATOR
190 REMARK
200 LINE INPUT "DO YOU NEED INSTRUCTIONS?"; R$
210 IF R$ = "YES" THEN GOSUB 800
220 REMARK
```

许多程序员在使用存盘命令 SAVE 时有一个小技巧，即将 SAVE 命令放在一个注释语句中，而该注释语句一般在程序开头，便于查找。例如：

```
2" SAVE "PAYSUB"
```

使用存盘命令 SAVE，可先用编辑命令 EDIT 对该行进行编辑，用 [Del] 键删除行号和单引号，再按结束 [END] 键和 [ENTER] 键，就执行了 SAVE 这条命令。如下图所示：

```
EDIT 2
SAVE "PAYSUB"
OK
```

用这个办法可将程序存入一个固定的文件，不会因打错文件名而破坏其他文件。磁盘中保留的总是最新的程序版本。如要保留以前的程序版本，可在文件名中加一个表示版本的数字，以示区别。