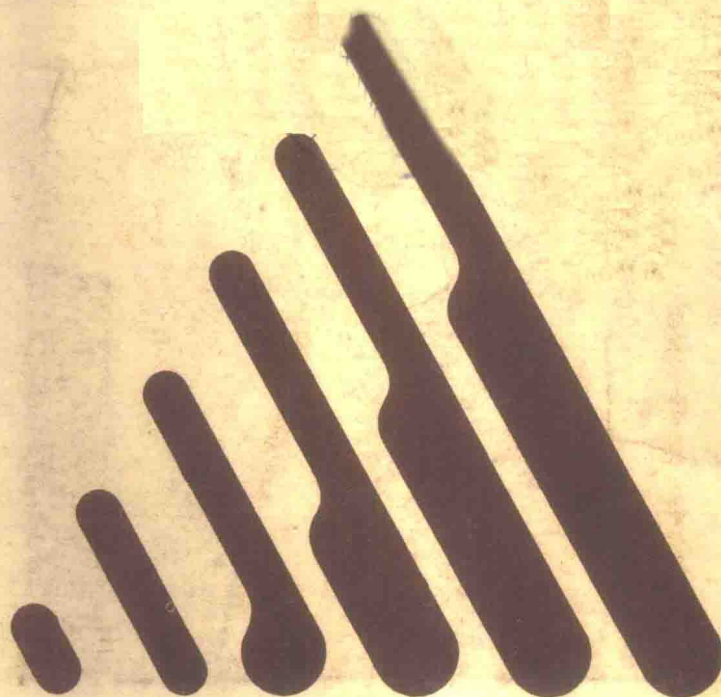


刘向东 牟卫平 译
郑叔良 校
王以和 审阅

dBASEIII PLUS

程序设计及应用



《陕西电子》编辑部

dBASE III⁺ 程序设计及应用

《陕西电子》编辑部

责任编辑：韩毅
封面设计：马军

目 录

引言	1
第一部分: dBASEⅢ ⁺ 程序设计	6
第一章: 程序设计概说	7
1.1 什么是程序设计	7
1.2 程序与应用程序	7
1.3 语言与解释程序	7
1.4 程序设计概念	8
1.5 程序流程	11
1.6 程序的设计与编写	15
第二章: dBASE程序结构与流程	20
2.1 dBASE程序的构成	20
2.2 dBASE程序流程	21
第三章: 内存变量的使用	30
3.1 内存变量的解释	30
3.2 内存变量的建立	30
3.3 内存变量在程序中的使用	33
3.4 全程变量和局部变量	33
3.5 内存变量的释放	35
3.6 内存文件	35
3.7 控制程序流程的逻辑型内存变量	38
第四章: 主程序的建立	40
4.1 主程序功能	40
4.2 设置区	40
4.3 内存变量的建立	45
4.4 连续的循环	45
4.5 主程序模块的其余部分	46
4.6 清除	46
第五章: 域和内存变量的函数	48
5.1 信息显示	48
5.2 类型转换的解释	48
5.3 连接	49
5.4 串的比较	49
5.5 数字函数	50
5.6 串函数	50
5.7 数值到字符的转换	56
5.8 字符串到数字的转换	58

5.9	日期运算	58
5.10	时间的使用	61
5.11	记事型域	62
5.12	逻辑型域	62
第六章:	用户通讯	63
6.1	屏幕格式的选用	63
6.2	屏幕控制	64
6.3	用户输入	67
6.4	用于继续的 $\leftarrow$ 键	69
第七章:	框架与范围的使用	73
7.1	框架的功能	73
7.2	数字和日期型输入范围的限制	76
7.3	显示变换	77
7.4	屏幕设计	77
第八章:	屏幕格式和格式文件	79
8.1	GET 空格区显示变换	79
8.2	相对地址	79
8.3	串的居中	81
8.4	串的右对齐	81
8.5	串的填充	82
8.6	图形和其它特殊字符	82
8.7	屏幕提示的重复使用	83
8.8	字符的重复	84
8.9	密码屏蔽	84
8.10	格式文件	85
8.11	记事型域的应用	87
8.12	快速录入方法	88
第九章:	用户输入时的检查及处理	89
9.1	输入筛选	89
9.2	特殊键	92
9.3	类型测试	95
9.4	ON 命令的使用	95
第十章:	数据库的使用	97
10.1	数据库的设计	97
10.2	数据库文件的开启	98
10.3	磁盘文件管理	101
10.4	记录查找	101
10.5	文件结束条件	104
10.6	程序文件的其它函数	105

10.7 用于筛选的命令	106
第十一章: 数据库中数据的处理	109
11.1 数据的加工	109
11.2 数据库信息的修改	110
11.3 多个工作区的使用	114
第十二章: 打印输出	119
12.1 dBASE III ⁺ 的报表特性	119
12.2 打印机的一般性说明	119
12.3 打印机的连接	119
12.4 向打印机发送输出	120
12.5 打印机坐标	120
12.6 屏幕和打印机间的转换	122
12.7 页面的格式化	123
12.8 容易出问题的位置 (故障点)	124
12.9 特殊效果	125
12.10 相当地址	126
12.11 页面间断的确定	127
第十三章: 内部整理	128
13.1 程序的完成	128
13.2 磁盘的应用	129
13.3 文件维护	133
13.4 数据库结构的修改	134
第十四章: 各程序模块的联结	136
14.1 准备	136
14.2 用户输入	137
14.3 条件测试	137
14.4 提纯部分	139
14.5 小结	140
第十五章: 程序的检测与调试	141
15.1 需寻找的故障	141
15.2 检测和调试步骤	143
15.3 调试命令	143
15.4 其它调试技巧	146
第十六章: 更先进的特性	149
16.1 IF... ENDIF的简化	149
16.2 更有效的程序代码	149
16.3 潜在的错误	153
16.4 使用汇编语言程序	153
16.5 大范围联用的 dBASE 程序	154

16.6	总控钥匙系统	156
16.7	下一步工作的建议	157
第十七章:	如何使用 RUNTIME ⁺	158
17.1	RUNTIME ⁺ 简介	158
17.2	怎样使用 RUNTIME ⁺	158
17.3	RUNTIME ⁺ 的应用	159
17.4	RUNTIME ⁺ 的应用步骤	160
17.5	选择项	160
17.6	详细步骤	161
第二部分:	使用 dBASE III ⁺ 连网	
第一章:	使用 dBASE连网	168
1.1	内容简介	168
1.2	网络环境	169
1.3	dBASE的连网能力	170
1.4	网络需要	171
第二章:	连网操作的考虑	173
2.1	本章概述	173
2.2	应用软件规划	174
2.3	使用 dBASE 命令	175
第三章:	dBASE安全性	179
3.1	本章概述	179
3.2	dBASE安全性概念	179
3.3	注册安全性	179
3.4	访问级安全性	180
3.5	数据加密	181
3.6	DBSYSTEM.DB文件	183
3.7	建立安全系统	183
3.8	操作上的考虑	196
第四章:	dBASE网络程序设计概念	201
4.1	概述	201
4.2	数据保护	201
4.3	使用 dBASE文件	209
4.4	错误和错误校正	209
4.5	使用共享文件, 锁和错误捕捉	210
4.6	程序设计级的安全性	212
4.7	其它的 dBASE 连网命令	212
4.8	应用软件实例	212
第五章:	网络程序设计命令和函数	217

5.1	概述	209
5.2	使用命令	217
5.3	CHANGE / EDIT	219
5.4	DISPLAY STATUS	220
5.5	DISPLAY USERS	221
5.6	LIST STATUS	222
5.7	LOG OUT	222
5.8	RETRY	223
5.9	SET	224
5.10	SET ENCRYPTION	225
5.11	SET EXCLUSIVE	226
5.12	SET PRINTER	227
5.13	UNLOCK	229
5.14	USE EXCLUSIVE	230
5.15	使用函数	231
5.16	ACCESS函数	231
5.17	ERROR函数	233
5.18	FLOCK函数	234
5.19	MESSAGE函数	235
5.20	RLOCK / LOCK函数	236
附录	错误信息	238

引 言

存贮在磁盘内的指导软件《学习dBASE III+》和《使用dBASE III+》，可教会用户如何使用《dBASE III+》去满足其进行数据库管理时的要求。而本书则向用户表明了《dBASE III+》的强大功能。

《dBASE III+》不仅仅是一完整的数据库管理系统，而且还包括了叫作〈dBASE〉的程序设计语言。使用该语言，可针对用户的特殊需要而建立用户化的应用程序，用于控制用户所有的数据库管理任务。还可将其出售给其它用户。

假如初学者对“程序设计”一词有些担忧的话，请放心，因为《dBASE III+》所具有的许多固有特性，使得〈dBASE〉语言编写程序要比使用其它程序设计语言容易得多。而且编写程序实在是一件十分有趣的工作。

本书说明

本书的目的是使初学者学会用〈dBASE〉来编写程序。它提供了学习编程时所必须了解的基本知识。学完本书后，可使初学者通晓《dBASE III+》所具备的全部功能。

如用户已会用〈dBASE〉以前的版本编写程序，则可将本书作为了解《dBASE III+》许多新特性的参考手册。

或许用户在编写程序时已用到了一些〈dBASE〉技术，但却没有意识到。例如，从点状态或ASSIST中发出命令，即选择或输入List命令，查看某个数据库文件内容时，便是在告诉《dBASE III+》去进行某些特定的操作。简言之，以一定的顺序发出一组相关的命令，就是最简单的程序设计。用这种方式编写程序意味着将一系列命令聚集在一个程序文件中，存入磁盘，然后，用运行该文件的方法来执行这些命令。

请看下列这些命令。假设现有一个叫作《Money.dbf》的数据库文件，内所有欠帐人员的信息，需定期检查这些信息以确证谁还未还清帐目。该文件以姓氏检索于索引文件《Last.ndx》。其中，Amount-due 为数字型域，Owing和 Past due为逻辑型域。Gimme 和 Scrooge 则为报表格式。现在报表中的人为仍欠帐者。这些命令行的功能为：

```
USE Money INDEX Last
LIST FOR Owing
CLEAR
LIST FOR Owing .AND. Pastdue
CLEAR
SUM Amount_due FOR Owing
SUM Amount_due FOR Owing .AND. Pastdue
REPORT FORM Gimme FOR Owing PLAIN TO PRINT
REPORT FORM Scrooge FOR Owing .AND. Pastdue PLAIN TO PRINT
USE
```

若以通常方式，则需多次重复输入！但现在则可将这些命令集中，建立一个程序，称其为：Owing.prg。这时所需要做的就是在点状态下打入：Do owing↵，而其结果完全相同。

这只是一个简单例子，但已足以说明学习一点(dBASE)程序设计，则可大大节省时间。

总控

(dBASE)程序设计真正的出类拔萃之处在于编程者对其用户使用(dBASE III+)时的控制能力。这样，就可确保用户不可直接存取重要的数据库信息。而程序的作用只相当于用户与数据库之间的媒介，来保护数据库中最重要的信息免受可能发生事故的破坏。特别是对于那些没有经验的用户来说，可使(dBASE III+)更便于使用。甚至还可使(dBASE III+)每天自动开启程序。这称为“转键系统”。

如像增加新记录或更新数据库中信息等这类日常工作是由办公室中多人分管的，则可设计数个程序，指导大家完成任务。对于初学者，只需花费很短时间就可学会这些程序的使用。这些程序还对数据库文件起保护作用。编程时，取用菜单驱动方式，这样，用户要作的只是根据提示，按几个键就可完成标准的(dBASE III+)操作。诸如：信息显示，数据库文件更新和打印报表。

本书使用说明

在使用本书之前，通过阅读《学习dBASE III+》和《使用dBASE III+》可获得对(dBASE III+)的概括了解。由于本书的目的是学习用(dBASE)来编写程序，所以并不对(dBASE)的每个程序设计特性作详尽的讨论。建议同时使用本书和《汉字dBASE III+使用指南》，后者含有(dBASE III+)所有命令和功能的详细参考资料。

初学用(dBASE)设计程序时，与其从点状态下打入命令，还不如研究一个完整的(dBASE)程序——保存核帐记录的Checkbook管理系统。如需要的话，可用它来保持支票簿的平衡。

Checkbook管理系统由存放在实用程序盘上的一系列文件组成。下面列出了这些文件的名称：

Add.prg	Cursoff.bin
Bank.prg	Curson.bin
Canc1.prg	Deposits.dbf
Cash.prg	Editvoid.prg
Cbmenu.prg	Help.prg
Cbmenu2.prg	Maint.prg
Check.prg	Menu mask.prg
Checks.dbf	Numwords.prg

Chkbook.men	Recon.dbf
Chkmask.prg	Reconcil.prg
Chkno.ndx	Reports.prg
Cleanup.prg	Rprtpro.prg
	Tax.dbf
Clrcash.prg	Taxcodes.prg
Clrdep.prg	Yearend.prg

本书第一章中对 Checkbook 管理系统作了更详尽的介绍。由于下列文件提供了本书中大部分编程实例，因而应按照下列顺序将部分程序打印出来，作为本书的参考资料：

Add.prg	Clrcash.prg
Cancl.prg	Clrdep.prg
Cbmenu.prg	Reconcil.prg
Check.prg	

采用文字处理程序或《dBASE III+》程序，按下列步骤，打印出上述文件：

1. 复制一个实用程序盘的工作拷贝，将原始盘安全存放。
2. 启动《dBASE III+》，并将放置实用工作盘的驱动器设置为预置状态。双盘驱动系统的命令为：SET DEFAULT TO B ↵
3. 启动打印机，并调整打印纸的位置，以便从其顶部开始打印。（以打印纸左侧圆孔为起始点）。
4. 在点状态下输入：TYPE Cbmenu.prg TO PRINT ↵
5. Cbmenu.prg打印完毕后，重复上述步骤，只需要更换文件名称。
6. 关闭打印机。

各章内容简介：

本书前三章主要讲述了程序设计的基本概念和〈dBASE〉语言的编程技巧。应按顺序阅读。其它章节则以定向讲授的方法提供了〈dBASE〉程序设计的细目分类。当然，学习时可根据自己的需要，跳过任何章节或按自己的顺序来阅读。但如果是初学者，最好还是按顺序进行学习。

第一章介绍了程序设计的基本概念和编程方法。

第二章研究了〈dBASE〉程序的结构和组成。

第三章讲述了程序中内存变量的使用。

第四章以Checkbook管理系统中Cbmenu.prg模块为例，描述了主程序的作用。

第五章讨论的是设计屏幕格式时的重点：怎样进行域和内存变量的类型转换，如从字符串型转换为数值型。

第六章的主要内容为如何在程序中建立屏幕格式和用户的输入方法。

第七章通过介绍用于过滤输入的框架和范围，继续讨论屏幕格式的建立。本章的结尾指出了设计屏幕格式时的几个要点。

第八章则主要考虑了屏幕格式问题并提出了许多改进屏幕外观的建议。还解释了有关格式文件的问题和记事型域的工作方法。

第九章介绍了错误查寻和用户输入正确性的检查要点。

第十章讨论了数据库和索引文件的建立，数个工作区的同时使用和信息检索。

第十一章研究了程序的数据库文件中数据的使用和变换方法，还谈及工作区间联系的建立。

第十二章简单地介绍了程序中打印控制问题和特殊打印效果。

第十三章探讨了某些方面的内部整理：程序的完成，数据库文件的关闭和工作环境的恢复，以及程序中文件的维护处理。

第十四章以Checkbook管理系统中的Cancl.prg为例，展示了在实际情况下各编程任务的共同工作关系。

第十五章介绍了程序的检测和调试。

第十六章探讨了更高级的功能，如：过程文件的使用，参数传递，隐含变量和汇编语言程序。同时还可看到《dBASE III+》与操作系统和其它应用程序的相互作用。本章最后介绍了一种“转键系统”。

现一切就绪，可开始学习用〈dBASE〉语言编写程序了。

第一部分 dBASE III⁺ 程序设计

第一章 程序设计概说

程序设计的初学者会发现在实际编写dBASE程序之前,先学习一些一般的程序设计规则和好的程序设计技术将是有益的。然后,就可以此为基础来进行程序设计了。请顺序阅读第一至第三章。每章都提供了重要的内容,并在先前章的基础上阐述了新的概念。如果读者还熟悉其它的程序设计语言,那么还是希望再复习一下前三章的内容。

1.1 什么是程序设计

程序是那些告诉计算机去完成特定任务的指令和命令的集合。程序设计并不仅是和计算机联系在一起的,在某种意义上,任何以一定顺序排列的指令就是一个程序。比如,食谱就是一个程序。当使用食谱的时候,要按照一步接一步的指令序列来烹调食品。每条指令都是一个命令——例如,加半茶匙的盐——就象每条dBASE指令都是一个命令一样。

指示一个人去烹调命令一台计算机去计算之间的区别在于人能够有选择地对食谱作略微的改变,而不必完全按照食谱去做。但是计算机则不行,它是通过识别字母来执行指令的——不是对,就是错。

程序必须包含那些想要计算机以明确次序去执行的每一条指令。正确、有效地组织并且写出这些指令就是程序设计的全部含义。如果是用dBASE写程序,则可以把它存贮到磁盘上的程序文件中,并且可以在任何希望的时候去运行它。

1.2 程序与应用程序

程序可以是完成一件基本任务的命令序列或者是完成所有任务的一个完整的程序项目,后者经常叫作应用程序。可以把一个完整的程序项目分解成模块,一个模块完成一个特殊的任务。通常是用一个主程序来控制这些模块。

这样,一个程序可以是指整个项目或者是指其中的一个部分,例如, Checkbook 管理系统就是由一系列模块程序组成的。

1.3 语言与解释程序

由于dBASE 程序设计语言是用英文单词组成,如IF和DO,所以非常容易使用。但是,计算机并不认识英文单词。dBASE III*解释程序按顺序读入每条命令,并且把它翻译成计算机能够理解的语言。不论命令是在程序里还是从点状态下发出,都将依赖于dBASE III*把每条命令解释给计算机。当用户告诉dBASE III*去建立一个新的数据库时,dBASE III*解释程序首先要在这条命令的各个字母, C—R—E—A—T—E, 去和命令表进行比较,这称之为命令的解析。当解释程序发现所打入的命令和表中的命令完全匹配时,它就会去执行与该命令相应的指令。如果不是完全匹配,它会给出一条出错信息。解释程序不能理解错误的命令,所以必须精确地输入命令。

避免下面四个常见的编程错误可减少许多令人头痛的错误现象。

- 命令拼写错误。
- 命令和其表达式间未设空格

- 语法错误，即命令使用不正确。
- 命令不完整。

1.4 程序设计的概念

本书讨论了一些用于dBASE程序设计的概念。尽管某些特殊的命令有所不同，但总的来讲，这些概念亦适用于其它的程序设计语言。

1.4.1 dBASE程序的编写与编辑

当程序设计者编程时，可用`.prg`作为扩展名把它存贮到一个程序文件中。dBASEⅢ⁺有一个用来建立并且编辑程序文件的文字处理程序，它允许用户处在dBASEⅢ⁺状态就可以对程序进行编辑，然后运行。

进入dBASEⅢ⁺文字处理程序时，应打入 **MODIFY COMMAND**，后面跟上文件名。dBASEⅢ⁺自动提供文件扩展名`.prg`。例如：当用户想建立一个叫作 `Test.prg`的新程序文件时，从点状态下打入命令将是如下的格式（现在不要打入）：

MODIFY COMMAND Test←

如想编辑一个设计好的程序文件时，用户亦可使用带有文件名的 **MODIFY COMMAND** 命令。dBASEⅢ⁺文字处理程序的编辑命令非常类似于Wordstar™的编辑命令。在使用**MODIFY COMMAND**时，可在屏幕的顶部看到最常用的编辑命令菜单。可按F1键来控制菜单的出现和消失。在《Quick Reference Guide》和《using dBASEⅢ Plus》这两本资料中可以看到编辑命令的完整清单。

当完成了对程序的编辑并且希望把它存入磁盘时，请按 **Ctrl—End**。如果不想保存刚修改过的程序，可按 **Esc** 键并且对提示：**Abort editing? (y/n)** 作“y”回答。dBASEⅢ⁺将恢复程序原来的形式并且使用户返回到点状态。

用户可以使用任何文字处理程序或文本编辑程序来建立，并且编辑dBASE程序，只要该文字处理程序或文本编辑程序能够建立ASCII文本文件就可以。美国信息交换标准码ASCII对计算机所使用的大写字母、小写字母、数字、标点符号以及控制字符的内部码作了标准化，这些标准字符称为ASCII码。

程序文件仅能包含命令正文，每条命令占一行并且均以←作为结束。如果使用的不是dBASEⅢ⁺的文字处理程序，那么用户应很好地检查一下，因为程序文件中不能包含任何用于格式化的字符。例如，特殊的页格式，连字符号以及黑体字打印。

多数常用的文字处理程序均可以建立不带有格式化字符的文本文件，当Wordstar使用非文档形式时，Framework II™允许使用DOS文本文件。在使用文字处理程序编写dBASE程序的时候，请记住在文件名中要包括`.prg`扩展名。使用Framework II，总是把文件存贮为磁盘菜单中的一个DOS文本文件，而不是一个通常的Framework II文件。

提示

如果计算机有足够的内存，那么，用户可以设置`Config.db`文件，以便使得所选择的文字处理程序成为预置的文字处理程序。这样，当使用**MODIFY COMMAND**时，dBASEⅢ⁺将自动地装入所选之文字处理程序。参看《Using dBASEⅢ Plus》中有关`Config.db`文件的内容。

尽管程序文件可按用户的需求任意长短,但是dBASE III+的文字处理程序只能处理最多包含5000个字符的文件,约100—200行。Checkbook 管理系统中的某些文件是比较大的,当使用 **MODIFY COMMAND** 命令去打开一个较大的程序文件时,会得到提示: **File too Large, Some data may be Lost** (文件太大,某些数据可能会丢失)。可用其它的文字处理程序来查看这些文件。

提示

如果仅仅是为了查看这些文件的内容,则可使用 **MODIFY COMMAND** 或 **TYPE** 命令。当用 **TYPE** 命令查看某个文件的时候,文件的内容会在屏幕上滚动。按 **Ctrl—S** 停止滚动,按任何键则可继续显示。

1.4.2 程序的运行

当用户希望运行程序的时候,可发出 **DO** 命令,并在后面跟上程序文件的名字。dBASE III+将打开该程序文件并且读入文件的每一行,然后从文件的头上开始,依次执行每条命令。当执行到文件中的最后一条命令,或用户决定提早结束该程序的运行时,dBASE III+将关闭该文件并且返回到点状态。有时程序文件的最后一条命令是 **QUIT**。它将使用户返回到操作系统。

1.4.3 程序环境的建立

如用户在点状态下开始用dBASE III+进行工作时,就处在某种工作“环境”中有许多方法可用来改变这种“环境”以满足用户的特殊要求。例如,所有数据库和索引文件均在B驱动器上,而dBASE III+在A驱动器上,那么,通常在开始使用dBASE III+时,要使用 **SET DEFAULT TO B**。同样,也可以在程序中规定其它的环境条件,如屏幕颜色等。第四章中介绍了怎样来设置这些程序环境。

1.4.4 输入与输出

程序就是接收输入并且产生输出。输入与输出这些术语分别对应于信息的来源和目标。例如,输入可以来自使用中数据库文件的域,或者是用户在程序运行期间从键盘上打入的数据。输出可以是新数据库文件中原来数据的信息,屏幕上域的列表或打印出的报表。

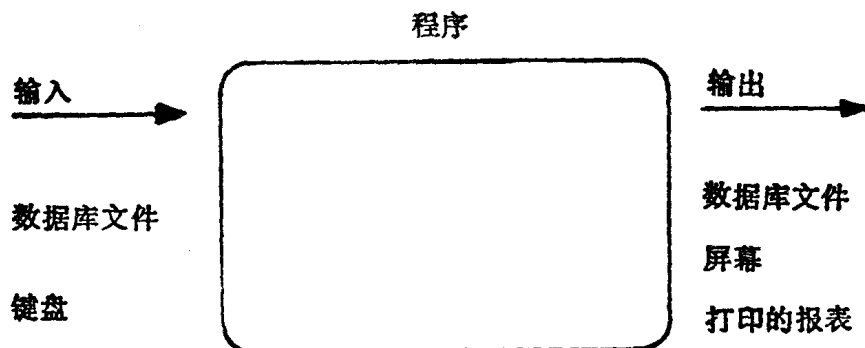


图 1—1 输入和输出的类型