

成人教育系列丛书

dBASE III 程序设计 实用技巧

●吴存孝 编著

电子工业出版社

TP311.13
WCX/1

成人教育系列丛书

dBASE III 程序设计实用技巧

吴存孝 编著



电子工业出版社

0028797

(京)新登字 055 号

内 容 简 介

本书使用教学实践中所举的各种程序例子及某些管理系统中实际使用的程序实例,系统地介绍了 dBASE III 数据库管理系统的程序设计实用技巧及一些其他特殊用法。全书共分六章,第一章介绍了有关程序设计的概念和基本方法;第二章按命令功能分类对 dBASE III 的命令作复习性的介绍;第三章重点介绍 dBASE III 的编程技巧,以大量的例子分别介绍各种功能模块的编写方法;第四章介绍了系统和程序的加密与解密;第五章主要介绍一些其他常用技巧及特殊用法,包括批处理、自启动、dBASE III 与其他语言的转换;第六章以实用的管理系统为例,讨论管理系统的设计方法。

本书内容实用,便于自学,可作为 dBASE III 程序设计方面的自学用书。尤其可作为成人教学的数据库课程的教材或教学参考书。

JS38/30

dBASE III 程序设计实用技巧

吴存孝 编著

责任编辑:邬志雄(特约) 应月燕

*

电子工业出版社 出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

人民卫生出版社印刷厂

*

开本:850×1168 毫米 1/32 印张:10.375 字数:262 千字

1995 年 3 月第一版 1995 年 3 月第一次印刷

印数:5000 册 定价:13.50 元

ISBN 7-5053-2863-8/TP·938

前 言

随着微型计算机在各行各业、各个领域的广泛应用，dBASE ■ 数据库管理系统也越来越深入地应用于各种事务管理和数据处理之中。dBASE ■ 数据库管理系统的应用不仅大大减轻了广大管理人员的劳动强度，保证了数据的正确和可靠，而且还有效地提高了管理水平，更主要的是充分地体现了管理的科学性和管理的现代化。所以许多人初步学习了 dBASE ■ 基本课程之后，要求进一步提高自己编制程序的能力和掌握一些程序设计的实用技巧。尤其是大量非计算机专业毕业而如今又在从事各种数据管理应用程序开发工作的人们，更迫切地需要一本能以切实可行的实际例子来说明 dBASE ■ 程序设计技巧的书籍。正是为了满足这种需要，在成人教育系列丛中，编写了这本《dBASE ■ 程序设计实用技巧》。

本书的特点是实用，而不是泛泛谈程序的设计。作者根据多年来从事编程实践和成人教学积累的资料，并参考同行们的先进经验，比较系统地介绍了各种功能模块的编写方法及编程时的实用技巧。所举例子具有明显的针对性和普遍性，讲述的方法由浅入深、循序渐进，所遵循的原则是不仅要让读者通过本书能学到编程技巧，更主要的是让读者能通过本书学到一种方法、一种思路，对书中所举的例子能举一反三、触类旁通。所以本书在讲解例子时，强调问题是怎样提出的，解决问题的思路是什么，采用何种方案，最后才落实到具体的程序上。另外，本书在讲解程序例子的同时还十分强调程序的优化，让读者了解任何程序的编写不可能是一劳永逸，只有通过不断的优化，才能使所写的程序既简练又运行迅速。

本书所举的全部程序例子都已形成文件，并且经过上机运行，

还配制了相应软盘,读者可以通过上机实践加深对所举例子的理解,同时也可在这些例子的基础上改写出其他合适的程序。

本书由上海师范大学计算机系张世正主任审阅,华东计算所邬志雄高级工程师组编,上海科普中心电脑天地学校陆根宝先生也参加了组稿工作,在此一并表示感谢。

由于作者水平有限,缺点和错误在所难免,恳切希望同行专家和广大读者批评指正。

编 者

1994 年 10 月

目 录

第一章	怎样进行程序设计	(1)
§ 1.1	程序设计的特点和要求.....	(1)
§ 1.2	系统设计的基本方法.....	(8)
§ 1.3	模块化结构程序的设计.....	(14)
第二章	dBASE Ⅱ 命令分类	(24)
§ 2.1	文件建立与编辑的命令.....	(24)
§ 2.2	文件打开和关闭的命令.....	(33)
§ 2.3	文件其他操作的命令.....	(37)
§ 2.4	数据库关系操作的命令.....	(38)
§ 2.5	数据库记录检索的命令.....	(41)
§ 2.6	数据库记录增加的命令.....	(42)
§ 2.7	数据库记录修改的命令.....	(43)
§ 2.8	数据库记录删除的命令.....	(46)
§ 2.9	数据计算的命令.....	(47)
§ 2.10	数据输出(显示或打印)的命令.....	(48)
§ 2.11	用于内存变量操作的命令.....	(51)
§ 2.12	用于程序设计的命令.....	(53)
§ 2.13	用于程序调试的命令.....	(56)
§ 2.14	用于设置状态或参数的命令.....	(56)
第三章	编程方法与技巧	(60)
§ 3.1	常用编程技巧.....	(60)
§ 3.2	循环与延时.....	(68)

§ 3.3	“菜单”技术·····	(79)
§ 3.4	数据输入的技巧·····	(110)
§ 3.5	统计的技巧·····	(126)
§ 3.6	检索的技巧·····	(151)
§ 3.7	数据维护的技巧·····	(178)
§ 3.8	报表打印程序的编写·····	(196)
第四章	加密与解密·····	(225)
§ 4.1	系统的加密·····	(225)
§ 4.2	文件名的加密·····	(229)
§ 4.3	文件名的解密·····	(245)
第五章	其他技巧及特殊用法·····	(247)
§ 5.1	DOS 的批处理文件 ·····	(247)
§ 5.2	自启动技术·····	(251)
§ 5.3	dBASE ■ 与其他语言的转换 ·····	(262)
第六章	系统设计实例·····	(265)
§ 6.1	过程文件·····	(265)
§ 6.2	录入员训练程序·····	(268)
§ 6.3	论文档案管理系统·····	(280)
附 录		
附录 A	dBASE ■ 命令一览表 ·····	(311)
附录 B	dBASE ■ 函数一览表 ·····	(318)
附录 C	常用符号的区位码表·····	(320)

第一章 怎样进行程序设计

提到“程序设计”，不少初学者会觉得十分神秘深奥。的确，程序设计本身是有一套完整的理论体系，而且还有不少条条框框。但它并非高深莫测。简单地说，所谓“程序设计”，就是按照程序设计语言所规定的命令语法和语句功能，针对程序所要完成的任务，将有关的命令语句进行有机的组合，这种组合的过程就是程序设计过程。这样说，是否又认为“程序设计”太简单了。当然，这种看法也不全面。应该说程序设计是一项技术性很强的工作，是模仿性和创造性的统一。所谓模仿性就是在进行程序设计时所写的语句、变量、格式等等都必须严格符合程序设计语言所规定的语法，不能自由发挥。所谓创造性就是在符合程序设计语言所规定的语法的前提下，程序设计人员可以针对各自的课题尽情发挥，对于同一个目的的要求，不同的编程者可以设计出各不相同的程序，真可谓“八仙过海，各显神通”。其中有大量的技巧和方法的应用，这就是本书所要讨论的内容。

尽管把程序设计说得如此灵活，如此自由，但是，对于如何评价一个程序设计的优劣，在进行程序设计时应该注意到哪些特点和要求等这些问题，必须有一个明确的目标。

§ 1.1 程序设计的特点和要求

众所周知，计算机可广泛地应用于科学运算、数据处理、实时控制等各个领域，而 dBASE Ⅲ 数据库管理系统主要是用来进行数据处理的，就其性质和一般要求而言，与科学运算、实时控制等有共同之处，例如：要求正确、迅速、可靠等。除此之外，数据处理还

有其固有的特点。

一、数据处理的特点

1. 数据量大

不言而喻,数据处理的对象是大量用来描述客观事物的信息数据,例如:在“人事信息管理系统”中,对于每一个人都必须记录下工号、姓名、性别、出生日期、学历、职称、工资等一系列数据,如此大量的原始数据均需通过键盘一个不漏地输入计算机,尤其是系统初创时,这一问题更为突出。这与科学运算有所不同,在科学运算时,所要输入的初始条件是有限的。当然,在运算过程中,可能会派生出大量的中间数据,但这是在计算过程中会自生自灭的,最终得到的可能是一个极简单的结果。所以,数据量大是数据处理的一个显著特点。

针对这个特点,我们在编制输入程序时,应该设计一些直观的输入格式,让操作者能依照原始表格的位置对号入座,另外还可以利用一些技巧,以减少重复数据的输入等等,从而减轻操作者的工作量。

2. 变动频繁

数据处理的频繁变动反映在如下几个方面:

- (1) 随着时间的推移,所描述的客观事物会发生变化,那么相应的数据也应随之变化。例如:在“人事信息管理系统”中,某人的职称、学历、工资等数据都会发生变化,更不用说在“工资管理系统”中,那些扣水费、电费、托儿费以及扣病事假费等等款项的数据,每个月都是不相同的,这就足见变动频繁之程度了。

- (2) 随着操作人员的变更,或者随着主管领导的变更,会对系统提出新的要求,那么需对系统作相应的修改。

- (3) 即使人员没有变更,任何一个系统都有一个不断地自我完善的过程,随着时间的推移,每个系统都必需不断地去适应新情

况,满足新要求。

针对数据的频繁变动,我们必须写出操作灵活、具有多种修改方式、行之有效的数据修改程序;而对于系统本身的完善和修改,必须由程序设计的结构化、程序的易读性和易修改性来作为基本保证。

3. 时间要求

对于数据处理的用户来说,当然是希望系统能及时反应,越快越好。但是,并不能象实时控制那样,要求在零点几秒甚至在零点零几秒内得到响应。我们知道 dBASE III 本身是一种解释执行的程序语言,其执行速度不快。尽管如此,我们还是要求系统的运行应尽可能地迅速。因此,在书写修改、删除、查询等程序的记录定位方法上,应该有多种方案:一种是按照关键字,用 FIND 或 SEEK 命令进行快速定位,这两条命令不管数据库内有多少条记录均可在一两秒钟内找到相应记录,当然其前提是必须有相应的索引文件存在;对于组合条件的记录定位,且这种条件的组合是完全随机的,那么事先不可能有相应的索引文件存在,这就需用另一种方法,用 LOCK FOR 命令后面跟上组合条件,进行记录定位,用这种方法显然要比前一种方法多用一些时间,当数据库内的记录很多时,所用时间可能会更长一些;当然也可以用上述方法的组合,既有快速又有组合条件的记录定位,这需要使用一定的编程技巧。其目的是要使记录定位尽可能地迅速。

对于其他一些功能(例如:运算、统计等)的程序设计,命令语句的选取是否恰当,技巧的运用是否熟练,数据库的结构设计是否合理等等,均会直接影响到程序运行的速度,而且有时运行速度的提高是非常可观的,这就是以后要讨论的内容。

4. 用户特点

我们所开发的应用程序,其操作者不一定是程序员本人,可能是对计算机及程序只有一般了解的企事业管理人员,也可能一点也不懂计算机及程序的原理,因有些系统,如财务管理系统,根据

财务制度要求,明文规定:编程人员不能参加系统的运行;系统操作人员应不懂编程。对于这样的用户,只要求他们根据屏幕上的提示,键入简单的操作即可。这就要求我们所设计的用户界面,要有直观、操作简便及美观等特点。

另外,我们所开发的应用系统,在使用过程中始终是采用人机交互的方式,因此就得考虑到人的有利因素和不利因素。所谓有利因素是:人有主观能动性,对于运行过程中可能出现的各种情况,能做出正确的判断,选择适当的操作等等,当然这还要取决于操作者的知识水平和应变能力。所谓不利因素指的是:人在操作中容易出错,尤其是长时间的操作以后,人易疲劳,这时更会出错。这就要求我们所设计的应用系统具有一定的健壮性,尽量少出故障,甚至不出故障。同时也不能要求,只有当操作者的操作正确时,系统才能正常运行;否则就不能保证系统正常运行。这样的程序不是好程序,好的程序应该是:一旦出现误操作,系统能及时发现错误,并提示操作者重新键入正确的选择,从而保证系统始终能正常运行。这就是必不可少的容错能力。

5. 技巧效果

由于在数据处理过程中是人机直接交互,因此,在程序运行过程中应用的一些技巧效果就能被操作者直接感觉到,体会到。尤其当用户也有这种要求和愿望时,技巧的成效将特别明显。当然,操作者不会去阅读具体的程序而发现其中的编程技巧,但有些技巧的效果却直接反映在人机交互之中。例如:设计一个布局合理的屏幕“菜单”,需要时再加上一些一目了然的提示信息,使操作者看了赏心悦目,觉得操作起来得心应手、简单方便。又如:输入数据时按照数据来源的表格格式,设计一个屏幕格式,使得操作者操作起来能对号入座、顺理成章。再如:对于那些花时间较长的运算过程,最好能隔一定时间,在屏幕上出一条提示信息,以显示运算到什么地方,运算结束能发出“音响”提示,以提醒操作者进行后续操作等等。

值得指出的是：一个程序不是越巧越好，拐弯越多越好。技巧的使用首先要建立在用户有这种要求和愿望的基础上；其次要考虑与原程序的配合，与整个系统浑然一体，起到画龙点睛的作用，严防弄巧成拙。这一点，对于广大渴望学习编程技巧的初学者尤其要注意。

二、对程序设计的要求

针对上述几个特点，用于数据处理的程序设计应该满足哪些要求呢？对于这个问题，不同的人有不同的观点，不同的设计课题所要求的标准也不一样。尽管如此，一个好的程序还是应该满足以下几个方面。

1. 正确性

不言而喻，正确性是我们所设计的系统是否具有生命力的重要前提。正确性反映在如下几个方面：

(1) 正确地完成各种功能

我们所设计的系统，应该正确地完成原设计所要达到的各种功能，既不要去做系统原设计之外的事，又不能缺少系统原设计中应该有的某些功能。这不仅仅取决于程序是否写得正确，还包含有需求分析是否充分，数学模型是否合理，功能设计是否正确等等。这就要求我们切不可在没有对整个作业做深入细致的、合乎逻辑的分析之前，就急于动手编写程序。

(2) 程序流程的正确性

程序的流程是否正确，直接与程序的结构有关。尤其是在遇到分支的情况下，应保证不同的选择，执行不同的分支。在实际编程中往往会由于逻辑关系考虑不周，造成分支走错或不管什么条件都执行同一分支，而某些分支又始终执行不到，形同虚设。这个问题以后将详细讨论。

(3) 具有纠错能力

系统的正确性更多地反映在对操作者的操作是否具有纠错能力,或者称为容错处理,即允许操作者键错,同时给予改错的机会。例如:在“菜单”选择时,我们可用 GET;READ 语句,用范围子句 RANGE 加以限制,当输入超出范围时,会自动报错,并要求重输。对于那些不能限定范围的数据,可以采用如下几种容错处理:

① 平衡校核——对于有平衡关系的数据均可以用这种方法。例如:财务系统的登帐程序,对于每一张记帐凭证,不管是一借多贷,还是一贷多借,必须借贷平衡。所以程序中应具有自动校核的功能,一旦借贷不平衡,即提示该凭证重输。

② 总数控制——对于那些既无平衡关系又无法限定范围的数据,有时在提供原始数据的同时提供了总数,那么就可应用总数控制的方法。在送完每一个分数据以后,即统计出总数并显示之,供操作者校对,发现有错,即回头重送。

③ 二次输入——对于全无上述关系而又比较重要的数据,我们可以采用二次输入或变通的二次输入方法来保证其正确性。前者是第一次输入,第二次是校核,若两次输入不一致,提示重输;后者是第一次输入,然后问操作者:“输入正确吗?(Y/N)”提示操作者检查,若有错,以“N”回答,回头重新输入;若无错,以“Y”回答,继续下一个操作。

2. 可靠性

对一个系统来说,可靠性和正确性是同样重要的。一个程序有可能在用模拟数据调试时是可行的,但在用实际数据反复运行后,可能会出现这样那样的问题。这种情况是常见的,因为模拟数据与实际数据之间总存在着一定的差异,也就是说在模拟运行时对实际情况考虑得不周到、不全面。对这个问题的解决方法,一是从实际数据试运行中发现问题,修正程序;二是在实践过程中不断积累经验,使模拟数据尽量接近实际情况,或编程时就全面考虑可能会发生的情况。总之,要使所编的程序尽可能地贴近实际,这样才能经得起长时间及大批量数据的考验。

3. 安全性

一个数据处理系统中,数据是工作的基础,因此系统中的数据必须绝对安全。数据的安全不仅包括存数据的安全(比如:防止数据丢失、破坏等等),而且还包括取数据的安全(比如:正确的读出、不同层次的人所能检索到的数据权限不同等等问题)。

保证系统数据安全的有效方法是设置口令。进入应用系统必须键入口令,以防止非法用户进入系统,误改数据或窃取数据。有些大的系统又可以分为若干子系统,子系统之间数据可以共享,但使用权限不同。有的子系统对某些数据可以修改;而有的子系统对某些数据只能调用,不能修改。因此,各个子系统还可设二级口令,以保证各个子系统由专人操作,别人不得入内。例如:财务管理系统下可设:工资管理、固定资产、成本核算、出纳、帐务等子系统。工资子系统能修改各个工资项数据;而帐务子系统能调用工资系统的数据,却不能修改等等。这些功能与限制都必须在程序中给予保证。

另外,从财物制度来说,凡登录的帐目,一律不得修改,若发现有错,手工做时用红字冲帐,那么,由计算机做时用负数冲帐。这在程序设计时也应予考虑,对于已登录的帐目只能查询,而不能修改。这也是安全性的一个方面。

4. 易修改性

随着实际情况的变化,要求程序作相应的修改,这是经常会发生的。因此,一个好的程序要经得起修改。易修改既包括程序的易阅读,也包括该程序的某一部分被修改以后,不影响或少影响整个程序。其后者用程序的模块化结构来保证(下面介绍),而程序的易读性除了用模块化结构使语句条理清楚以外,还有一个恰到好处地使用编程技巧的问题,避免弄巧成拙。请看下例:

：

$$A=A+B$$

$$B=A-B$$

A=A-B

⋮

对于这段小程序,如不仔细读一读,恐怕不知其功能。其实,不过是交换两个变量中的值而已。与传统的引入一个中间变量的做法相比,可算用了技巧,节省了一个变量。但是,它破坏了程序的易读性。不但别人一下子看不懂,恐怕过一段时间编程者本人也会不识其真意。若改成如下,则一目了然:

⋮

C=A

A=B

B=C

⋮

5. 面向用户性

由于应用系统的用户是广大的管理、行政人员,因此,编程时就要想到怎样使用户在使用本系统时感到方便。这除了用前面所讲到的那些技巧,提供方便外,还要考虑到程序的流程、操作的步骤应尽量接近沿用至今的手工操作的步骤和习惯。否则,花了大量精力设计出的程序,用户不愿使用,或用一段时间后又废弃不用了。为此,我们在设计程序时应尽量考虑用户的习惯和方便,只有这样才能使我们的系统被用户所接受。

§ 1.2 系统设计的基本方法

上一节我们讨论了用 dBASE Ⅲ 进行数据处理的特点及对程序设计的要求,从中也体会到了程序设计的技术性和灵活性。尽管程序设计很复杂、很灵活,没有一种统一的模式。但是,根据程序设计的基本思路、性质及过程,总还是能总结出一些比较普遍的方法。首先要强调:在进行具体的程序设计之前,先要系统分析及设计。一种常用的系统设计方法是“逐步求精法”——是自顶向下,由

表及里的设计方法,是一个不断地从抽象到具体的细化过程。

“逐步求精法”是一种适应性很强并被广泛采用的系统设计方法;是模块化结构程序设计的核心,具有许多独特的优点。其方法可大致归纳为如下框图:

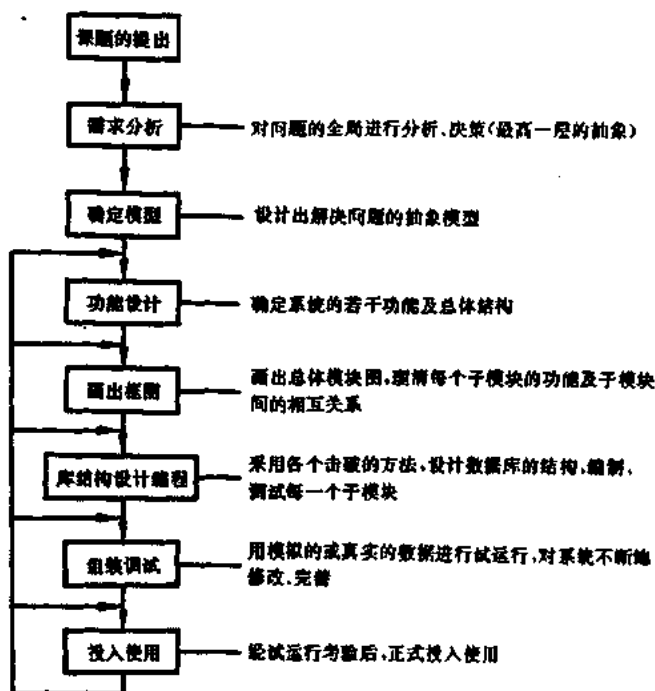


图 1-1 “逐步求精法”框图

1. 课题的提出

作为一个正规的课题,必须由一方提出课题,由另一方承担开发任务。若不是一个正规的课题,当然不需要一本正经地明确甲乙双方,但提出一个设想,还是需要的,这是进行以下步骤的前提。

2. 需求分析

需求分析做得是否充分、细致,是系统功能正确性的基本保证。根据现有的软、硬件条件,提出力所能及的要求,这是考虑问题的出发点,从这一点出发提出待编系统所要完成的各项任务。对于用 dBASE III 开发的数据处理系统,更应进一步考虑到信息量大小与存储空间是否相适应的问题。要运行中西文 dBASE III 对内存的要求至少为 640K,再大一点当然更好;对外存的要求是至少有两个软盘,这是因为单软盘而无硬盘的计算机是无法运行 dBASE III 的,然而软盘的容量十分有限,所以在软盘上开发的系统不能做得很大。如果有了硬盘,存储空间的问题就大为改观了,不过,对于系统的信息量还是应该先有个估计,例如:大致设几个主要的信息库,每个库大概有多少条记录,每条记录有多长,这就可以估算出总的信息量,再打上一定的余量,便可提出对硬盘容量的要求。这些均是在需求分析时所要考虑的具体问题。

3. 确定模型

依据前一步所提出的待编系统所要完成的各项任务,设计出完成这些任务的抽象模型。例如:用人—机对话的方式,通过“菜单”选择,调用子程序模块,执行有关操作,完成相应功能。当然还可以进一步细化到功能实现的细节问题,例如:查询的方案可有哪几种;又可用哪几种不同的方式进行统计;统计的结果怎样输出;有几种不同的表格格式……这些都是必须逐一确定的模型。

4. 功能设计、画出框图

对前一步所确定的抽象模型进一步进行具体化,通过框图语言使得各个子模块之间的相互关系一目了然(见图 1-2)。