

吴洪森 吴志洪 编著
浙江大学出版社

DBASE II. III

程序设计实用 技巧

DBASE II, III
CHENG XU SHEJI
SHIYONG JI QIAO



dBASE II、III 程序设计实用技巧

吴洪森 吴志洪 编著

浙江大學出版社

内 容 提 要

本书系统地介绍了dBASE数据库的程序设计实用技巧及一些其它特殊用法。全书共分十章,第一章介绍了程序设计的概念和基本方法;第二、三章主要介绍了dBASE II的程序设计技巧,包括编程技巧和调试技巧;第四、五章主要介绍了dBASE III的程序设计技巧,包括dBASE II与III的区别, dBASE II与III的转换及使用技巧;第六章至第九章主要介绍一些其它常用技巧及特殊用法,包括汉字信息处理,数据和程序的加密与解密,实时控制和微型计算机的联机及常见故障的判别、预防 and 排除等。第十章介绍了几个程序实例,包括作曲程序、绘图程序及应用程序。

本书内容由浅入深,便于自学,可作为大、中专院校非计算机专业及各种类型学习班的数据库课程的教材或教学参考书。尤其对于从事微型计算机应用、管理的各级人员具有重要的实用价值。

JS358/31

dBASE II、III 程序设计实用技巧

吴洪森 吴志洪 编著

责任编辑 应伯根

浙江大学出版社出版

浙江大学印刷厂印刷

浙江省新华书店发行

开本850×1168 1/32 印张: 12.25 字数: 300千字

1988年2月第1版 1988年2月第1次印刷

印数: 1—15 000

ISBN 7-308-00064-1

TP·006 定价: 3.15元

前 言

自七十年代以来,随着微型计算机的问世,给计算机的应用带来了新的飞跃。微型计算机的出现真正把计算机从科学家、工程师们手中解放出来了,微型计算机的出现使得各行各业应用计算机成为可能,而 dBASE 数据库语言的开发又促使了这种可能成为事实。实践证明 dBASE 数据库语言不仅简单易学、而且具有实用价值。不少单位和个人利用所学的 dBASE 知识编写程序解决了许多实际问题,取得了很好的效果。

在初步学习了 dBASE 数据库语言之后,广大读者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,希望获得更多同行的经验供自己学习时参考。考虑到目前能适合广大初、中级计算机应用人员及学习过 dBASE 数据库语言的学生进一步提高的书籍、资料确不多见,我们编写了这本“dBASE II、III 程序设计实用技巧”。

本书的特点是实用,而不是泛泛其谈语言和程序设计;本书的资料来自作者多年亲自编程实践和教学,同时还吸收了国内外、军内外的先进经验,具有显明的针对性和普遍性。书中介绍的技巧和用法以 IBM PC (PC/XT) 和 APPLE II 微型计算机为主,同时兼顾到北极星微型计算机和 LASER-310 微型计算机。对于其它机型只需根据具体机种的差异作相应的变动就可适用之。本书中所阐述的技巧方法、思路对其它程序设计语言都有重要的参考价值。

在本书的编写过程中,浙江省计算技术研究所黄龙秋高级工程师,杭州大学计算机系副主任、浙江省计算机教育委员会主任吴美朝副教授,浙江公安专科学校龚达高级工程师,杭州电子工学院计算机系张金德老师,浙江丝绸工学院计算中心金永康老师,浙江省公安厅信息中心应卫平,杭州市计算机职教中心郑子罕老师及杭州民生药厂电脑室鲍道胤,浙江省计算机用户协会袁建国等给予了热情的帮助和支持,提出了不少宝贵意见,在此由衷地表示感谢。

由于我们水平有限,缺点和错误一定是难免的,恳切希望同行专家和广大读者批评指正。

目 录

第一章 怎样进行程序设计	1
§1-1 什么是程序设计.....	1
§1-2 程序设计的基本方法.....	5
§1-3 模块化结构程序的设计.....	13
第二章 dBASE II 的编程技巧	19
§2-1 常用命令及函数.....	19
§2-2 编程五步法.....	24
§2-3 结构程序的设计.....	25
§2-4 常用技巧.....	31
§2-5 数据输入的 技 巧.....	34
§2-6 交互式命令和“菜单”技术.....	42
§2-7 循环与延时.....	52
§2-8 统计和检索的技巧.....	55
§2-9 数据维护的技巧.....	68
§2-10 屏幕与打印格式的控制.....	75
第三章 dBASE II 的调试技巧	81
§3-1 建立命令文件的多种途径.....	81
§3-2 命令文件的调试技巧.....	84
§3-3 数据库的建立和修改.....	87
第四章 dBASE III 与 dBASE II 的区别	92
§4-1 dBASE III 与 dBASE II 的主要 区别.....	92
§4-2 dBASE III 中新增加的命令及其 使用	96
§4-3 dBASE III 中新增加的函数及其 使用.....	119
第五章 dBASE III 的编程及使用 技巧.....	125
§5-1 dBASE II 与 dBASE III 的 转换.....	125
§5-2 dBASE III 的调 试 技 巧.....	132
§5-3 dBASE III 的编程及使用 技巧.....	145
第六章 汉字信息处理	172

§6-1	汉字信息处理概况	172
§6-2	汉字的输入和输出	177
§6-3	微机汉字信息处理系统的配置	187
第七章	数据和程序的加密与解密	194
§7-1	数据的加密保护	194
§7-2	程序的加密保护	201
§7-3	程序的解密	204
第八章	实时控制和微型计算机的联机	211
§8-1	日期和时间的输入及合理性判断	211
§8-2	周期性定时控制	216
§8-3	非周期性定时控制	217
§8-4	突发事件的控制	220
§8-5	微型计算机的联机	222
§8-6	BASIC的通讯文件	224
第九章	其它常用技巧及特殊用法	230
§9-1	自启动及其语言转换	230
§9-2	BASIC与dBASE的数据交换	241
§9-3	磁盘文件数的扩充及被删目录的修复	246
§9-4	常见故障的判别、预防和排除	257
第十章	程序实例	270
§10-1	计算机作曲的方法和实例	270
§10-2	IBM PC屏幕美化的方法和实例	284
§10-3	APPLE II及LASER-310的图形	301
§10-4	应用程序实例	310

附 录

附录 I	常用微机操作程序图
附录 II	dBASE III命令、函数一览表
附录 III	实现报表输出的常用控制代码(区位码)
附录 IV	国内常见微机配置及局部网络性能参数表
附录 V	程序流程框图的标准画法
附录 VI	常见计算机英文缩略语(字符)的全称及译意

第一章 怎样进行程序设计

作为一名计算机应用人员，必须明确自己所从事工作的性质和意义，以及为了把自己所从事的工作做好，应该朝什么方向去努力，达到什么样的标准。本章作为全书的引导，首先向读者介绍有关什么是程序设计，程序设计的标准以及程序设计的基本方法等内容。

§1-1 什么是程序设计

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的语言所规定的语法。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，程序设计人员都可以针对各自的课题尽情发挥。一个程序好比是一篇文章，进行程序设计的过程就是程序设计人员按照计算机语言所规定的语法和功能写文章的过程。如果把传统的写文章与进行程序设计相比，一般来说应具有以下几个阶段：

作 文	程序设计
背景或事实	课题的提出
审 题	需求分析
确定题材	确定模型
提 纲	功能设计
腹 稿	框 图
拟 稿	编写程序
修 改	调 试
脱 稿	使 用

当然，学了中文的词法、句法等语法后不一定就会写文章，一个语法掌握得很好的人，不一定就能写出好文章。同样，一个对计算机语言的语句、命令学得很好的人，也不等于就能设计出好的程序。对于同一个题目，十个人写出的文章会十个样子；程序设计也一样，对同一个课题十个人设计出的程序也会不同。而且程序设计不但要满足人的要求，还要符合机器所规定的要求，因此程序设计是一项极其复杂又十分灵活的工作。评价一个程序的好坏也是由许多因素决定的。

那么，什么是一个好的程序呢？不同的人有不同的观点，不同的设计课题所要求的标准也不一样，所以，这个问题很难有一个统一的标准答案，但是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。

一、正确性

编制程序的目的是为了完成某些功能，如果一个程序不全具有所要求的功能，则这个程序将毫无价值。应该提出的是程序的正确性并不仅仅指程序的使用正确，它应该贯穿于程序设计的整个过程中，包括需求分析的正确、数学模型的正确、功能设计的正确等等。除此以外，程序的正确性还应包括纠错能力，因为对一个程序来说，仅仅在提供正确的输入之后，才执行、产生正确的输出是不够的。一个正确的程序必须保证在任何情况下，都能进行正确的操作。因此，程序的正确性是一个最基本的，也是一个不容易满足的要求。

防止下列几类错误，有助于实现程序的正确性：

1. 在没有对整个作业做深入细致合乎逻辑的分析的情况下，就急于动手写程序。
2. 让程序去干那些并不是程序应该干的事情。
3. 程序应该干的事情，程序员没让程序去干。

4. 程序员想让程序干的事情，程序并没有去干。

5. 程序干错了程序员让它干的事情。

二、可靠性

对一个系统来说，可靠性和正确性是同样重要的。一个程序有可能在调试时是可行的，但在进入实际反复运行后会出问题，或当数据量较少时是可行的，一旦数据量增加到几千或几万则又会出现一些想象不到的问题，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。

可靠性本身也是度量程序正确性的一项指标，不可靠实际是一种隐含错误，是在调试、管理阶段没有发现的错误。

三、面向用户性

一般情况下，程序是从事计算机应用的程序员设计的，而使用是由广大的管理、行政人员执行的。因此，程序员设计程序时就要想到怎样使得别人使用时感到方便，愿意用自己设计的程序。常常有这样的情况，程序员化了大量精力设计出的程序，许多用户不愿用，或用一段时间后又不想用了。因此，程序的面向用户性是评价一个好程序的很重要的标准，尤其在目前计算机的应用还不够普及，使用、管理水平比较低的情况下更值得引起广大程序员的重视。有时为了保证这一标准的实现，就是要打破一些传统的程序设计要求，牺牲一点其它方面的标准。

四、易修改性

程序的设计要求是根据实际工作情况提出来的，而实际工作的情况是因人、因时、因地而异的。因此，一个好的程序要经得起修改。当然，易修改既包括该程序易阅读，也包括该程

序的某一部分被修改后，不影响或少影响整个程序。

五、具有数据的安全措施

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）。

对应用于数据处理的程序的上述五项必须满足的评价标准是根据数据处理的性质和特点提出来的。那么数据处理与科学计算、实时控制相比有何特点呢？

1. 使用程序的不是程序员本人，而是对计算机及程序只有一般了解的企事业管理人员。

2. 数据量大。一个系统的完成其中大量的工作是输入原始数据，尤其是当第一次建立系统数据库时，这一问题更为突出。

3. 处理的对象——数据，经常因人、因时、因地而异，包括原始数据的变动，输出格式的变动等等。

4. 使用过程中是人——机直接交互，因此就得考虑到人的有利因素和不利因素。有利因素有人能进行灵活的判断、选择等等；不利因素有人易疲劳、开小差、易出错等等。

5. 就目前实际情况来看，对数据处理的用户不象实时控制那样，要求得到即时的反映。比如，当用户键入一个数据要求检索某条信息，原来五秒钟就出来的结果，而现在要等10秒钟才出来，实际上这也是可接受的。提出这个论点的目的，不是希望广大程序员不要去追求快的响应速度，而是提醒大家不要一味地为了追求高速度而牺牲掉许多别的优点；或为了提高速度而在其它方面带来麻烦。要切合实际量力而行，否则会得不

偿失。

6. 由于在数据处理中是人——机直接交互，因此，在这个领域中应用的一些技巧效果就能被人直接感觉到、体会到。用户也有这种要求和愿望，技巧的成效特别地明显。当然，值得提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前提下；其次要考虑与原程序的配合，起到画龙点睛的作用，严防弄巧成拙。这一点，对于广大渴望学技巧的初学者尤其要注意，下面举一个例子，以加深印象：

```
10  A = A + B
20  B = A - B
30  A = A - B
```

这段小程序短短三行，读者如不仔细读一读，恐怕不知其功能吧。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办法相比可算是巧了吧。但是，它破坏了程序的易读性、易修改性，不但别人一下子看不懂，恐怕过一段时间你自己也会不认识，如改成如下，则一目了然：

```
10  C = A
20  A = B
30  B = C
```

尽管本书是一本讲编程技巧方面的书。但却主张使用第二种方法来交换两个变量中的值。

对于以上提出的要求和特点，在设计程序时应该根据具体情况综合考虑，以保证系统获得最佳效果。

§1-2 程序设计的基本方法

上一节我们已经谈了程序设计的一些概念，从中也体会到

了程序设计的技术性和灵活性。在进行程序设计时，每个人可以根据各自的课题“八仙过海，各显神通”。

尽管程序设计很复杂，很灵活，没有一种统一的模式，程序员可以尽情发挥，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。

程序设计方法中的一个基本原则就是抽象原则。因为要解决的现实问题是很复杂的，人的智力往往不可能一下子涉及到问题的全部细节。在进行了需求分析之后，我们总是先设计出一个抽象的模型，用抽象的数据进行一系列抽象的操作，这时的数据和操作只反映问题的本质属性，而将所有细节都抽象掉了。然后，作为下一步，再考虑这些抽象数据和操作的具体实现。这属于抽象阶段，在这个阶段只考虑“做什么”，而在实现阶段才考虑“如何做”……。这样可把一个很复杂的问题一步一步地解决，最后达到实用。因此，抽象原则是程序设计方法中的一个基本原则。

建立在抽象原则基础上的一种常用的程序设计方法是：逐步求精法。前面我们已经谈过，处理一个复杂问题的原则是抽象，在有了抽象的模型，抽象的数据及程序以后，怎样完善这个模型和程序呢？这就是逐步求精法所要完成的任务。逐步求精的程序设计方法的基本思路是：对抽象的模型，数据及程序作进一步的求精，而进入下一层的抽象。在求精过程中的每一步，抽象的语句和抽象的数据都将进一步分解和精细化，直至每一个步骤都可以用程序设计语言来描述。如此继续，一直到最后系统能满足实际要求为止。

由此可见，逐步求精法是一种自顶向下，由表及里的程序设计方法。但是，这种自顶向下，由表及里的过程也不是完全绝对化的，因为我们所采用的逐步求精法也是在抽象这个基本原则上进行的。事实上这种过程有时常常要进行多次尝试才能

比以前更精化一步。因为，有时当你一步一般精化后，在以后的步骤中会发现原来这样的一种精化设想并不好，或者有错，或者对进一步精化不便。这样，就必须自底向上，由里向外对原来已决定的某些步骤进行修改。如果没有这样一种修改，就要求上层所作的每一步都是正确的，而且是最优化的，实际上这也是不可能的。因此，逐步求精的过程应理解为螺旋式的自顶向下，由表及里的过程。

逐步求精法是一种适应性很强并广泛采用的程序设计方法，是模块化结构程序设计的核心，有其许多独特的优点。其方法可大致归纳如下：

一、对问题的全局进行分析、决策。

二、设计出解决问题较为自然、很可能是用自然语言描述的抽象模型。

三、确定程序的总体结构，即总体模块图。

四、理清每个子模块的功能及子模块间的相互关系。

五、在抽象的基础上，采用个个击破的方法，编制、调试每一个子模块，

六、回到实际中去检验，不断修改，完善各子模块，使之进一步的精化。

一般说来，用逐步求精法最终得到的程序的结构是良好的。整个程序是由一些相对小的子程序构成，每个子程序都具有一定的相对独立的意义。改变某一子程序的内部不会影响整个程序和程序的结构，形成了一种模块化结构的程序。

下面以建立一个“图书馆辅助管理程序”为例来说明一下逐步求精法的应用

一、问题的提出及需求分析

编写一个“图书馆辅助管理程序”分别完成下述任务：

- 记录有关每天采购进来的图书、报刊的单价、品种、数量、书名等信息。

- 能根据需要对库存资料进行查询、统计、排序等操作。

- 具有自动报警功能，比如某一种书有重复进货或编目不合理等。

- 具有紧急情况的应急处理功能。

这一步要与图书馆工作人员一起制定，并充分了解他们在没使用计算机以前各项工作是怎样开展的，以及用计算机来代替的可能性和代替后的效果。

二、进一步分析上功能的实现，用自然语言来描述一个抽象的模型

问：完成上述功能以什么方式工作？

答：以人一机对话方式工作。

问：如何用人—机对话方式实现所需功能？

答：利用键盘输入命令，程序识别并执行，再输出回答信息。

⋮

经过这样的分析，还可对提出的每一个问题进一步求精，

问：怎样执行各种功能的命令？

答：由主控模块根据用户输入的信息调用相应的子程序模块，执行有关操作完成相应功能。

问：命令结束后如何回答？

答：命令执行结束后，在屏幕上或打印机上输出有关信息。

问：查询分哪几种不同功能的查询？

答：要完成按书名(主题词)查询、按出版社查询、按作者

名查询、按出版时间查询等等。

问：统计分哪几种不同功能的统计？

答：按出版时间统计、按出版社统计、按某一种书在某一时间内被借用次数及借用时间的长短统计等等。

⋮

经过这样一步一步的分析求精，就可大致确定程序的总体结构。

三、程序总体结构，即总体模块图的确定

程序总体模块图如图 1-1

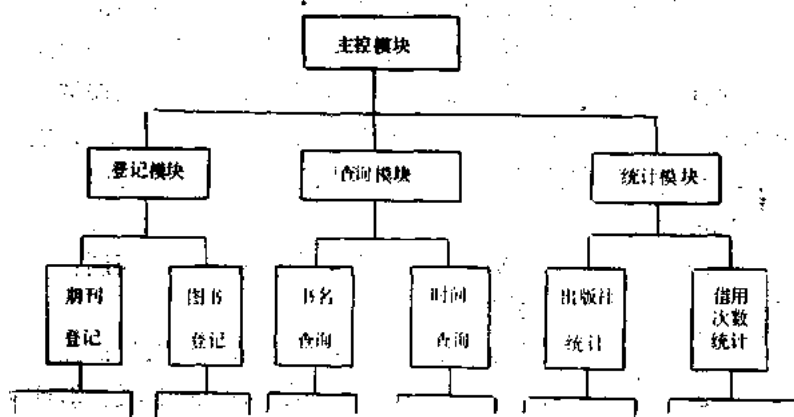


图1-1

所谓结构化程序就是按照结构程序设计思想而设计出的具有合理结构的程序。一个结构化程序应具有下述特点：

- 全部程序均由下述的顺序型、条件型、跳跃型和循环型四种基本结构组成，不出现除这四种基本结构以外的其它类型的结构。

• 程序中的所有语句均属于各自的功能模块(即子程序),只要模块间接口关系不变,则修改某一模块中的语句时,不会影响别的模块的功能。

• 程序的整个结构体现出顺序型,即程序中的所有语句都有被执行的机会,程序从头至尾执行时间是有限的。

• 不管程序多么长、多么复杂,但结构始终清晰、层次分明,始终保持程序的易读性、可靠性。

• 程序的内部有恰当的说明。说明主要有两种,一种是在每个子程序的前面,另一种是插在相应的语句中间。说明的内容主要是该程序的作用,调用方法以及与其它程序的关系、注意事项等等。采用说明是提高程序的易读性、易理解性和易维护性的一种有力措施。而好的说明是结构化程序的特点之一。

由此可见,结构化程序的设计方法,实质上是在保证程序的正确性、可靠性的前提下,使程序更清晰、易读、易理解、易维护,具有良好的结构。而良好的程序结构正是程序正确性、可靠性的有力保证。两者是辩证的统一,结构化程序设计是实现这种统一的方法。

下面再谈谈进行结构化程序设计时应注意的几个问题:

1. 源程序书写格式的结构化

源程序的书写格式对程序的易读性、理解性及易维护性有重要的影响。因此,在书写程序时可采用退格、进格或换行、加辅助线等办法。例对多重循环或条件嵌套可采用如下格式书写:

```
10  FOR I=1 TO 10
20  PRINT TAB(10-I);
30  FOR J=1 TO 2*I-1
40  PRINT ":",
50  NEXT J
60  PRINT
70  NEXT I
```

或者:

```
10  FOR I=1 TO 10
20  PRINT TAB(10-I),
30    FOR J=1 TO 2*I-1
40    PRINT " ",
50    NEXT J
60  PRINT
70  NEXT I
```

2. 嵌套不要加深

过深的嵌套会使程序变得臃肿难读。实际上这是设计者思路模糊不清的反映。例如，在下面程序中，由于条件语句的嵌套太深，已经很难分析出执行命令文件：DO D的条件了：

```
IF A1
  DO A
ELSE
  IF A2
    DO B
  ELSE
    IF A3
      DO C
    ELSE
      IF A4
        DO D
      ELSE
        :
      ENDIF
    ENDIF
  ENDIF
ENDIF
```

3. 功能与子程序一、一对应