

程序设计 实用技巧

BASIC DBASE

吴志洪 吴洪森 编著

浙江大学出版社

程序设计实用技巧

吴志洪 吴洪森 编著

浙江大学出版社

内 容 简 介

本书系统地论述了BASIC语言和dBASE数据库的程序设计实用技巧。全书共分二十一章,第一章介绍了程序设计的概念和基本方法;第二章至第十章主要介绍了BASIC语言的程序设计技巧,包括程序的调试、“菜单”技术、文件的使用、实时控制及时钟利用、数据和程序的加密保护、音乐和图形、表演程序制作等技巧;第十一章至第十九章主要介绍dBASE II的程序设计技巧,包括数据的输入、维护、统计和检索,打印格式的控制,循环与延时等技巧;第二十章主要介绍了一些其它的常用技巧,包括自启动及其语言的转换,两种语言间的数据交换,常见故障的判别和排除等技巧;第二十一章主要介绍了两个应用系统的程序实例。

本书可作为大、中专院校非计算机专业及各种类型学习班的BASIC语言和数据库课程的教材或教学参考书,也可作为正在学习BASIC语言和dBASE数据库的广大中学生的教学参考书或课外读物。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参考价值。

程序设计实用技巧

吴志洪 吴洪森 编著

责任编辑: 应柏根

浙江大学出版社出版

余杭县三墩印刷厂印刷

浙江省新华书店发行

开本787×1092 1/32 印张: 12 字数: 271千

1986年12月第一版 1986年12月第一次印刷

印数: 1—8 000

统一书号: 15337·028 定价: 1.75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。

但是,应该看到目前许多单位的微型计算机并没能发挥出其应有的作用。那么,为什么今天的微型计算机在速度、容量、外部设备等方面的硬件性能都已超过了以前的大型计算机,而相当多的微型计算机却没有发挥出当时大型计算机所发挥的作用呢?一个很重要的原因就是,当时每一台大型计算机都有一批专家、学者为它编制程序,而今天很多微型计算机却连一个专业的程序设计人员都没有。要使每一台微型计算机真正用起来,必须有一批人为它编写各种各样的应用程序。但是,要想编好应用程序,用计算机来解决具体的实际问题,仅仅靠了解计算机语言的基本语句是很不够的,重要的是掌握程序设计的方法和一些实用技巧,学会怎样结合实际问题使用语句进行程序设计。

近年来许多单位都有了计算机(大部分是微型计算机),而管理、使用这些计算机的人大多数是业余程序员,他们学习过BASIC语言和dBASE II(或dBASE III)数据库语言,也见过不少别人的成功应用,渴望利用自己学过的知识为本单位、本部门做些工作,但是他们会碰到很多困难,而这些困难正是许多从事计算机应用工作的人称之为“技巧”的东西。

让广大初、中级计算机应用人员及学习过BASIC语言和

3407

dBASE数据库语言的学生能得到实实在在的帮助，推动我国计算机应用的发展，这就是我们编写本书的目的。

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。书中介绍的技巧方法和程序以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改动就可用之。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，杭州大学教授朱祖祥，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，中国人民解放军空军工程学院副教授戴震，浙江公安专科学校副校长沈忠民和高级工程师龚达，空军第一研究所计算机室王毅，空军司令部计算站吕梅，浙江省公安厅信息中心应卫平，杭州市人民警察学校梁树芳、杭州轴瓦厂麦铁栋以及顾根贤、段杰、李尚伟给予了热情的帮助和支持；浙江大学韩仁顺老师，高级工程师龚达仔细审阅了全部书稿，提出了不少宝贵的意见，在此由衷地表示感谢。

编写这种综合性的、面向实用技巧的书籍，还是一次尝试，需要在实践中不断加以完善。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来信批评指正。

编 著 者

一九八六年八月

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.....	7
第二章 BASIC语言的编程技巧.....	12
§2-1 常用语句及函数.....	12
§2-2 编程步骤.....	18
§2-3 节省内存的措施.....	21
§2-4 怎样提高程序的运行速度.....	24
第三章 BASIC程序的调试技巧.....	27
§3-1 程序的输入和修改.....	27
§3-2 加快程序输入的速度.....	33
§3-3 程序的保存、合并及打印.....	37
§3-4 动态调试技巧.....	44
第四章 BASIC语言的人机对话	
方式和“菜单”技术.....	51
§4-1 人机对话的方式.....	51
§4-2 “菜单”技术.....	54
第五章 BASIC文件的使用技巧.....	60
§5-1 文件和文件参数的选择.....	60
§5-2 怎样打开和关闭数据文件.....	61
§5-3 顺序文件的使用技巧.....	63
§5-4 随机文件的使用技巧.....	69
§5-5 不同类型文件的转换.....	75
第六章 实时控制和机内时钟的利用.....	78

§ 6-1	日期的输入及合理性判断	78
§ 6-2	时间的输入及合理性判断	81
§ 6-3	周期性定时控制	83
§ 6-4	非周期性定时控制	84
§ 6-5	突发事件的控制	87
第七章	计算机作曲	90
§ 7-1	作曲的方法和技巧	90
§ 7-2	计算机作曲实例	95
第八章	屏幕的美化及图形	99
§ 8-1	如何美化屏幕	99
§ 8-2	作图方法种种	103
§ 8-3	几个实例	111
第九章	BASIC数据和程序的加密保护	118
§ 9-1	数据的保护	118
§ 9-2	程序的保护	120
第十章	表演程序的制作	123
§ 10-1	循环及注意事项	123
§ 10-2	延时方法	126
§ 10-3	表演程序的制作	127
第十一章	dBASE II 的编程技巧	133
§ 11-1	dBASE II 的常用命令及函数	133
§ 11-2	编程五步法	138
§ 11-3	编构程序的设计	140
§ 11-4	常用技巧	147
第十二章	dBASE II 的调试技巧	151
§ 12-1	建立命令文件的多种途径	151
§ 12-2	命令文件的调试技巧	154
第十三章	数据输入的技巧	158
§ 13-1	灵活的输入格式	158

§13-2	有选择的数据输入	160
§13-3	字段数和字符数的扩充	164
第十四章	dBASE II 的交互式命令和	
	“菜单”技术	166
§14-1	用交互式命令实现人机对话	166
§14-2	“菜单”技术的应用	168
§14-3	日期的输入技巧	71
第十五章	循环与延时	177
§15-1	巧用循环语句	177
§15-2	dBASE II 的延时方法	180
第十六章	统计和检索的技巧	181
§16-1	统计的技巧	181
§16-2	检索的技巧	187
第十七章	数据维护的技巧	196
§17-1	数据的修改	196
§17-2	数据的删除	199
§17-3	数据的转移	201
第十八章	屏幕与打印格式的控制	205
§18-1	灵活的屏幕格式	205
§18-2	打印格式的控制	207
第十九章	dBASE II 数据和程序的	
	加密保护	211
§19-1	数据的加密保护	211
§19-2	程序的加密保护	216
第二十章	其它常用技巧	218
§20-1	自启动及其语言转换	218
§20-2	BASIC与 dBASE II 的数据交换	221
§20-3	磁盘文件数的扩充及被删目录的修复	227
§20-4	如何打印区位码和ASCII码	229

§20-5	汉字的输入和输出	232
§20-6	常见故障的判别和排除	236
第二十一章	程序实例	242
§21-1	供、产、销合同微机管理系统	242
§21-2	微机辅助飞行指挥系统	301
附录一	IBM PC/XT微机操作程序图	
附录二	LASER—310微机操作程序图	352
附录三	实现报表输出的常用控制代码 (区位码)	353
附录四	常用计算机英文缩略语(字符) 的全称及译意	358
附录五	程序流程框图的标准画法	365
参考文献		369

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

背景或事实

审 题

确定题材

提 纲

程序设计

课题的提出

需求分析

确定模型

功能设计

腹 稿

拟 稿

修 改

脱 稿

框 图

编写程序

调 试

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，也是一个不易满足的要求。

二、可靠性

对于一个系统来说，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。

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的情况提出来的，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、具有数据安全的措施

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许多麻烦。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。

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进行综合考虑，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。因为要解决的实际问题是很复杂的，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，有时当你一步一步精化后，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容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。

§1-3 模块化结构程序的设计

1969年Dijkstra E. N 首先提出了结构程序设计的概念，强调了从程序的结构和风格上来研究程序设计。目前，结构程序设计的思想得到了广泛的承认，并为广大程序员所普遍采用。

用结构程序设计的方法写出的程序不仅结构良好，易写易读，而且易于修改，易于维护。各模块结构的程序只要模块之间的接口关系不变，每个模块内部的具体细节可以由各自的程序员随意修改。而且，每个模块的意义和功能明确，使得整个程序的阅读、理解和调试极其方便。

模块化结构程序的设计应满足以下两点要求：

一、模块的独立性

即各子模块应具有独立的功能，可独立地运行、调试、修改和维护。

二、模块的相关性

即不需知道各子模块内部的具体细节，就能方便地把它

们搭配成一个大型的系统程序。

这样，一个大型软件系统可以分成若干个不同层次的子模块分而治之，由许多人分头去完成。有些模块还可以套用现成的，经过实践验证的程序，以提高编程的效率和模块自身的可靠性。

对于一个实际的软件系统，不论问题如何复杂，其各个模块间的连接结构不外乎以下四种基本结构的组合：

1. 顺序型连接

这种情况最简单，只要把每个子模块按先后次序依次排列即可，模块之间无需任何控制语句。

2. 分支型连接

这种情况稍微复杂一些，程序的执行事先不能确定，要根据其中间结果（这个中间结果可以是程序自己运行后产生的结果，也可以是运行期间人为输入的结果）判别后才能确定执行哪个子模块。

3. 跳跃型连接

这种情况比较复杂，模块的执行同样不能事先确定，但情况又不同于分支型连接。在这里程序经中间结果判别后，会出现两种不同的情况，一种情况要执行某一模块；而另一种情况则跳过这一模块，最后这两种情况又汇总在一起。

4. 循环型连接

这种情况是：当某一模块由于实际的要求，需反复执行多次，而且执行的次数可根据情况自由控制。

以上四种情况的连接结构如图一所示。