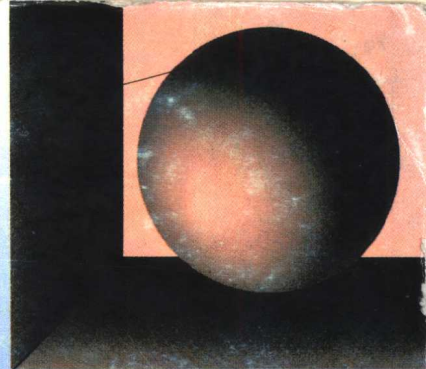


FOXBASE

关系数据库



通用程序百例详解

牛树长 编著



- 多种菜单管理方案
- 功能完备的字典库
- 自动适应数据的表格

- 自由表格与数据对接
- 自适应的数据库结构

- 文档文件管理

北京理工大学出版社

FOXBASE 关系数据库 通用程序百例详解

牛树长 编著

北京理工大学出版社

内 容 简 介

本书以介绍 FOXBASE 编程经验为主。通过剖析具有普通意义的程序实例,由浅入深的探讨了程序构思和实现技巧,同时介绍了程序设计基本原则和方法。书中有许多经过优化的实用程序:有简单的数据定位、屏幕格式设计、字典应用,有通用菜单、行列菜单及文档管理,也有较高难度的随机条件合成、格式输出、自由报表与数据库对接等程序。这些高性能的通用源程序可直接用于各种软件开发,能起到缩短周期和提高性能的双重效果。本书是学习编程的好教材,也是软件设计人员的得力工具。

读者对象:关系数据库的程序设计人员、使用数据库的有关工作人员及任何有意学习数据库应用的人员。

本书所有源程序及运行环境有配套磁盘同时发行。

图书在版编目(CIP)数据

FOXBASE 关系数据库通用程序百例详解/牛树长编著. —北京:北京理工大学出版社,1995

ISBN 7—81045—023—9

I. F… II. 牛… III. 关系型数据库, FoxBASE—程序设计 IV. TP311.13

中国版本图书馆 CIP 数据核字(95)第 06288 号

000001

北京理工大学出版社出版发行

(北京市海淀区白石桥路 7 号)

(邮政编码 100081)

各地新华书店经售

北京房山先锋印刷厂印刷

*

787×1092 毫米 16 开本 22.5 印张 567 千字

1995 年 11 月第一版 1995 年 11 月第一次印刷

印数:1—5000 册 定价:23.00 元

※ 图书印装有误,可随时与我社退换 ※

前 言

FOXBASE 是一种掌握容易、使用方便、功能完善、运行快、占用资源少的关系数据库语言,适合于管理人员的知识层次并能满足一般管理工作的实际需要。虽然更先进的 FOXPRO 系统已经问世,但出于资源和应用水平的关系,FOXBASE 仍然占有不可取代的重要地位,在管理应用及家庭普及方面,它仍是目前用户最多的数据库系统之一。

本书以使用 FOXBASE 语言的计算机用户为基本对象。以介绍典型应用程序为主要内容,以解决应用中的实际问题为主要目的。本书介绍的程序来自管理应用中一些带有普遍意义的实际需要。涉及到程序控制、菜单管理、记录定位、数据查询与维护、计算汇总及统计分析、表格形成、打印处理及文档资料等多方面。本书从实用经验的角度,以各类命题产生和一般实现为出发点,通过分析归纳得出具有规律性的程序模型,再经过逐步的演绎优化,形成性能较好的实用模型或通用程序。这些模型和程序在多种管理软件中使用并在多种语言版本及各种档次的机型上长期运行,实践证明了它们具备较好的性能和比较广泛的适应性。即便是一些用户已经开始使用 FOXPRO 系统,这些程序仍然能发挥其作用。

本书侧重于对程序形成过程的分析和对编程技巧方面的介绍,对所有涉及到的程序不但给出了经过验证的源程序清单,还用大量的实际演示说明了源程序的基本环境、接口参数、服务对象、调用方法及运行结果。通过对这些典型实例的剖析,不但能够迅速掌握编程的基本方法和技巧,也可以方便准确的将这些程序直接引用到自己开发的软件系统之中。在这些程序中,有些可以适合任何软件系统,有些能够辅助软件开发,也有些可以在交互状态下直接作为工具程序使用。引用这些程序和编程方法,不但能缩短开发周期和提高工作效率,对系统的整体功能也会有些改善。凡是对编程感兴趣的读者,或多或少的会从这些设计中得到一些有益的启发。

本书认为读者已经基本掌握 FOXBASE 系统基础命令,至少是已经具备了学习该系统的能力。为了节省篇幅和突出主题,本书没有重复对系统命令一般性的说明,突出了对命令内含和基本概念方面的解释,把重点完全放到对程序推理构思等主观思维方面的阐述。为了学习方便。建议读者准备一册有关的命令集,以备在学习过程中随时参考。

本书以个人见解和经验为主,在于与读者共同交流,由于水平有限,不免有偏见和不妥之处,或有技术和方法方面的错误和不足,诚请读者批评指正并谅解。

李平贤参加了本书的编写,王文成、栾文胜为本书审稿,在此表示感谢。

作者 1995. 3

目 录

第一章 编程入门

§ 1 系统的启动	(1)
§ 2 编程的概念	(1)
§ 3 字符串显示程序	(3)
1. 命题的提出	(3)
2. 实现一个规范	(3)
3. 找到一个模型	(4)
4. 程序的书写	(5)
5. 使用接口参数	(6)
6. 改善程序的功能	(7)
7. 完整的程序	(9)
8. 概念的补充	(10)
§ 4 控制变量接收	(14)
1. 一般形式	(14)
2. 接口参数	(15)
3. 坐标计算	(15)
4. 初步运行	(16)
5. 功能的改进	(17)
6. 完整程序	(20)
7. 功能归纳	(21)
8. 调用举例	(21)

第二章 几个简单程序

§ 1 日期转换	(25)
§ 2 金额转换	(28)
§ 3 颜色选择	(31)
§ 4 边框程序	(33)
1. 绝对坐标	(33)
2. 中间位置	(35)
3. 西文边框	(36)
4. 西文边框改进	(38)
§ 5 警告与提示	(40)
§ 6 书写风格	(41)

第三章 数据库与定位

§ 1 数据库的基本概念	(43)
--------------------	--------

1. 数据库结构	(43)
2. 索引文件	(45)
§ 2 记录定位	(46)
1. 定位方式的差别	(46)
2. 程序的原始模型	(47)
3. 实用定位程序	(49)
4. 多索引智能转换	(52)

第四章 矩形菜单管理

§ 1 系统菜单的缺陷	(56)
§ 2 菜单的作用	(56)
§ 3 键盘选择式菜单	(57)
1. 菜单库的结构	(57)
2. 菜单库内容	(58)
3. 菜单管理程序	(59)
4. 运行结果	(60)
5. 程序分析	(64)
§ 4 光标选择式菜单	(65)
§ 5 界面库的维护	(68)
§ 6 文档资料的生成	(70)
§ 7 辅助菜单	(71)
1. 行向菜单	(71)
2. 标题及行向菜单	(73)
3. 列向菜单	(75)
4. 数组菜单	(76)

第五章 模拟系统菜单

§ 1 系统菜单的状态分析	(81)
§ 2 标题点的设计	(82)
§ 3 菜单管理数据库	(82)
§ 4 模拟菜单的演示	(83)
§ 5 菜单管理主程序	(85)
§ 6 页面控制程序(局部数组)	(87)
§ 7 页面控制程序(数据库)	(89)
§ 8 数组菜单管理	(91)
§ 9 维护及文档文件	(96)

第六章 字段选择

§ 1 字段表的用途	(98)
§ 2 字段名显示	(99)
§ 3 字符式选择	(101)
1. 程序清单	(102)

2. 程序分析	(103)
3. 实用的举例	(103)
§ 4 光标式选择	(105)
§ 5 参数式字段表	(107)
1. 字段表提取	(107)
2. 序号表提取	(108)
3. 兼容式提取	(109)
4. 特征字提取	(111)

第七章 记录显示格式

§ 1 格式处理的必要性	(113)
§ 2 处理格式的模型	(114)
1. 端口参数	(114)
2. 模型及分析	(115)
3. 程序分析	(115)
4. 演示结果	(117)
§ 3 实用程序	(118)
1. 程序及分析	(118)
2. 演示及变化	(120)
§ 4 自动生成 PRG 文件	(122)
1. 程序及分析	(122)
2. 演示一结果	(124)
§ 5 选择字段	(126)
1. 必要性	(127)
2. 可能性	(127)
3. 程序及分析	(128)
4. 演示及结果	(129)
§ 6 再次实现自动编程	(130)
§ 7 程序端口设计	(134)

第八章 数据库访问

§ 1 指针控制	(136)
§ 2 数据库访问	(138)
1. 实验数据	(138)
2. 功能控制	(139)
3. 执行状态	(140)
4. 功能扩充	(141)
§ 3 通行字	(142)
1. 通行字程序	(142)
2. 程序分析	(144)
3. 程序状态	(145)
§ 4 数据库排序	(146)

§ 5 库结构生成	(148)
-----------------	-------

第九章 字典库的应用

§ 1 使用的必要性	(151)
1. 简化录入操作	(151)
2. 可靠性和一致性	(151)
3. 重新安排使用顺序	(151)
4. 形成统计规范	(152)
§ 2 字典库的建立	(152)
1. 代码的唯一性	(152)
2. 内容的全面性	(152)
3. 含义要清楚	(153)
4. 符合使用习惯	(153)
5. 统计特征明确	(153)
6. 组合程度合理	(153)
7. 实现其它控制	(153)
§ 3 访问字典库	(154)
1. 平面查询方式	(154)
2. 通用数据查询	(157)
3. 列查询方式	(159)
4. 行查询方式	(164)
§ 4 代码字段	(167)
1. 使用字符代码	(168)
2. 建立编码规范	(168)
3. 编码程序	(170)
4. 汉字串的双字节化	(171)
5. 混排字符串	(173)
§ 5 数据排版	(174)

第十章 数据计算

§ 1 关联与结构库	(177)
1. 工作区关联	(177)
2. 结构数据库	(178)
§ 2 库一般计算	(181)
1. 行间计算	(181)
2. 列间计算	(184)
3. 报表计算	(185)
4. 计算演示	(186)
§ 3 报表的平面汇总	(187)
§ 4 定字段汇总	(189)
1. 同结构定字段汇总	(189)
2. 异结构间的列汇总	(190)

§ 5 钞票配款	(191)
1. 配款分析	(191)
2. 配款汇总	(193)
3. 配款主程	(195)
4. 演示结果	(196)

第十一章 数据分析

§ 1 一维分析	(198)
1. 实现程序	(199)
2. 程序分析	(200)
3. 实用举例	(201)
4. 分析的应用	(202)
§ 2 二维分析	(204)
1. 数据坐标系	(204)
2. 字段列坐标系	(208)
§ 3 二维求和	(212)
§ 4 随机归类	(215)

第十二章 数据打印

§ 1 任务分析	(217)
§ 2 列表打印	(217)
1. 确定参数	(218)
2. 定义参数库	(218)
3. 数据录入	(219)
4. 记录管理	(220)
5. 过渡变量组	(221)
6. 实施程序	(222)
7. 程序分析	(224)
8. 打印实例	(225)
§ 3 套表打印	(226)
1. 套表程序	(226)
2. 程序分析	(231)
3. 套表实例	(232)
4. 调用方法	(234)
§ 4 求和打印	(235)
1. 技术要求	(235)
2. 参数管理	(236)
3. 程序设计	(237)
4. 程序分析	(243)
5. 应用变通	(244)
6. 操作说明	(245)
7. 应用效果	(246)

§ 5	数据填表	(247)
1.	基本设想	(247)
2.	填表参数	(248)
3.	填表程序	(249)
4.	程序分析	(252)
§ 6	填表演示	(253)
1.	基础格式	(253)
2.	默认字段	(255)
3.	指定字段	(258)
4.	重复填写	(260)
§ 7	附属数据	(261)

第十三章 数据库检索

§ 1	再谈字段表	(265)
§ 2	字段算式	(266)
§ 3	算式合成	(267)
1.	算式接收	(268)
2.	接收演示	(268)
§ 4	条件合成	(270)
1.	关系算式	(271)
2.	逻辑算式	(273)
3.	条件合成	(275)
4.	应用范例	(276)
5.	实现的特点	(283)
§ 5	综合实现	(284)

第十四章 文档工具程序

§ 1	文件清单生成	(288)
§ 2	过程文件清单生成	(291)
§ 3	数据库结构清单	(294)
§ 4	数据库结构表	(296)
§ 5	文本文件分页打印	(298)

第十五章 数据录入

§ 1	录入方式	(301)
1.	直接录入	(302)
2.	间接录入	(302)
3.	控制库	(304)
4.	主用库	(305)
§ 2	录入程序	(308)
1.	控制数据	(308)
2.	录入程序	(309)

3. 简单录入	(310)
§ 3 辅助汉字库	(311)
§ 4 代码翻译	(313)
1. 数据库环境	(313)
2. 翻译程序	(314)
3. 程序分析	(317)
4. 追加字典数据	(318)
5. 控制库数据	(319)
6. 字典库数据	(319)
7. 运行情况	(320)
8. 遗留问题	(322)
§ 5 总体实现	(322)
1. 主程序	(322)
2. 运行结果	(323)
3. 功能扩充	(324)

第十六章 系统设计

§ 1 路径管理	(328)
1. 缺席路径	(328)
2. 任意路径	(330)
§ 2 颜色方案	(331)
§ 3 程序设计	(332)
1. 工作区冲突	(332)
2. 结构性错误	(333)
3. 字段优先	(334)
4. 变量名冲突	(334)
5. 本书程序	(335)
§ 4 程序组织	(335)
1. 命名方案	(335)
2. 过程文件	(335)
3. 文件装订	(336)
4. 编译运行	(337)
§ 5 库结构设计	(337)
§ 6 关系设计	(338)
§ 7 辅助库	(339)
1. 结构库	(340)
2. 字典库	(340)
3. 菜单库	(340)
4. 录入控制	(341)
5. 打印控制	(341)
6. 外部参数	(341)
§ 8 维护模式	(342)
1. 定位维护	(342)

程序目录索引

01 01 -1. TU. PRG	在默认或指定行中间位置反相显示提示信息 (9)
02 01 -2. KV. PRG	接收信息(鸣铃、坐标)并进行合法检查 (20)
03 02 -1. RQVH. PRG	将字符数字日期转换成汉字日期(供打印) (25)
04 02 -2. JEVH. PRG	将金额转换成符合票据打印的大写汉字 (29)
05 02 -3. YSZX. PRG	在屏幕上显示所有颜色搭配供程序员挑选 (32)
06 02 -4. BKT. PRG	在屏幕指定区域划边框并可分别着色 (33)
07 02 -5. BKV. PRG	在屏幕中间画边框并可使用标题和颜色 (35)
08 02 -6. BKX. PRG	在屏幕中间画边框并可使用标题和颜色 (36)
09 02 -7. BKY. PRG	中间边框标题隔离并可各自独立使用颜色 (38)
10 02 -8. TUJG. PRG	用五行的矩形边框显示提示/警告信息 (40)
11 03 -1. UDW _J. PRG	用索引或记录号对记录定位(记录号优先) (48)
12 03 -2. UDW _S. PRG	用索引或记录号对记录定位(记录号优先) (49)
13 03 -3. UDW _Q. PRG	用索引或记录号对记录定位(索引优先) (50)
14 03 -4. UDW _X. PRG	多索引智能转换定位程序参考模型 (53)
15 04 -1. CD _ZF. PRG	菜单及程序命名规范管理(字符键选择) (59)
16 04 -2. CD _GB. PRG	菜单及程序命名规范管理(光标键选择) (66)
17 04 -3. CD _WH. PRG	用于对 CDF. DBF 进行现场维护 (68)
18 04 -4. CD _WD. PRG	将界面库以表格形式写到指定的磁盘文件 (70)
19 04 -5. CD _HX. PRG	中间菜单在程序中进行功能或状态选择 (72)
20 04 -6. CD _H2. PRG	行向菜单 在程序中进行功能和状态选择 (74)
21 04 -7. CD _LX. PRG	列向菜单 在程序中进行功能和状态选择 (75)
22 04 -8. CD _UZ. PRG	数组菜单 用光标键选择数组对应元素号 (77)
23 05 -1. CD _MN. PRG	模拟系统菜单管理主程序(页面式下拉) (85)
24 05 -2. CD _M1. PRG	局部数组→模拟系统菜单页面显示及控制 (87)
25 05 -3. CD _M2. PRG	数据库→模拟系统菜单页面显示及控制 (90)
26 05 -4. CD _ME. PRG	由数据库→菜单数组→变量文件 CDK. MEM (92)
27 05 -5. CD _MU. PRG	用数据库→数组→模拟系统菜单 (94)
28 05 -6. CD _WU. PRG	生成菜单库的文档文件(模拟系统菜单) (96)
29 06 -1. ZD _XU. PRG	用端口数据格式化显示数据库字段名称表 (100)
30 06 -2. ZD _ZF. PRG	用字段序号或字段名随机合成字段表(名) (102)
31 06 -3. ZD _GB. PRG	端口数据显示结构与光标控制选择字段表 (105)
32 06 -4. ZDB _V. PRG	用字段序号表取回指定的字段名表列 (109)
33 06 -5. ZDB _Z. PRG	用字段序号表取回字段名序列(或全部) (110)
34 06 -6. ZDB _T. PRG	用特征字取自段表列 (111)
35 07 -1. GUXU _Y. PRG	用端口数据格式化显示或编辑记录数据 (115)
36 07 -2. GUXU _V. PRG	用端口数据格式化显示或编辑数据库记录 (118)
37 07 -3. GUXU _P. PRG	通过端口调整数据格式后并自动生成 . PRG (122)
38 07 -4. GUXU _X. PRG	选择字段格式化显示或编辑库记录数据 (128)
39 07 -5. GUXU _XP. PRG	选择字段实现格式显示或编辑并生成源文件 (130)
40 08 -1. VVKV. PRG	用光标、翻页等键实现多种方式指针控制 (136)
41 08 -2. JLFW. PRG	数据访问主程序显示、增加、修改及删除 (139)

42 08 - 3. TXZ. PRG	鉴定及建立、修改系统及操作员的通行字	(143)
43 08 - 5. UJPX. PRG	将当前数据库按制指定索引表达式排序	(147)
44 08 - 4. JGUI. PRG	数据库结构生成(由参数定义数据库结构)	(148)
45 09 - 1. ZD_P. PRG	字典库平面显示,带边框按特征分翻页排版	(154)
46 09 - 2. PMXU. PRG	用平面显示自动分页的方式访问任意数据库	(157)
47 09 - 3. ZD_L. PRG	数据库列显示仿 BROW 状态并返回定位值	(159)
48 09 - 4. ZD_H. PRG	字典库行查询显示、执行程序	(164)
49 09 - 5. ZD_C. PRG	字典库行查询控制及代码翻译	(165)
50 09 - 6. HZ_D_1. PRG	依 HZK_1 中规范将汉字串翻译成字符编码	(170)
51 09 - 7. HZ_UZJ. PRG	将中西文混排字符串处理成双字节形式	(172)
52 09 - 8. HZ_D_2. PRG	将中西文混排字符串翻译成字符编码	(173)
53 09 - 9. UJPB. PRG	把纵列数据排版成平面的 PB?? .TXT 支持打印	(175)
54 09 - 10. DYPB. PRG	把纵列数据排版后的文件 PB?? .TXT 送打印机	(176)
55 10 - 1. B_JS_H. PRG	表行计算 运算记录组运指向目的记录	(182)
56 10 - 2. B_JS_L. PRG	表列计算 计算字段组并向目的字段传递	(184)
57 10 - 3. B_JSKV. PRG	通过数据库组织报表行列的计算过程	(186)
58 10 - 4. B_JS_PM. PRG	表计算平面汇总 汇总相同结构的数据库	(187)
59 10 - 5. B_PM_VD. PRG	数据库平面(指定字段)汇总程序	(189)
60 10 - 6. B_JS_LJ. PRG	表计算、列间、汇总不相同库结构的数据列	(190)
61 10 - 7. PKFX. PRG	按钞票面值统计金额字段中各类钞票数量	(192)
62 10 - 8. PKHZ. PRG	按给定字段分类对各种钞票数量进行汇总	(193)
63 10 - 9. PKVI. PRG	钞票配款主程	(195)
64 11 - 1. YWFX. PRG	对表达式值进行分类及分布分析	(199)
65 11 - 2. FXEI. PRG	数据坐标二维分析示意程序 执行部分	(205)
66 11 - 3. FXE0. PRG	数据坐标二维分析示意程序 初始化部分	(206)
67 11 - 4. EWFX. PRG	任意字段在一个字符字段上的二维分析	(208)
68 11 - 5. EWQH. PRG	任意字段在一个字符字段上的二维求和	(213)
69 12 - 1. DY_KV. PRG	显示或编辑一个打印任务的全部控制参数	(219)
70 12 - 2. DY_WH. PRG	管理打印参数数据库,连接维护与打印	(220)
71 12 - 3. DY_CU. PRG	取指定的打印任务控制参数到变量组	(222)
72 12 - 4. DY_LB. PRG	取指定的打印任务控制参数到变量组	(223)
73 12 - 5. DY_TB. PRG	用表格形式实现打印任务(套表打印)	(226)
74 12 - 6. QH_DY. PRG	与套表同时打印出指定字段的小计及总计	(237)
75 12 - 7. UJ_TB. PRG	将数据与表格按指定的方式对接填表	(249)
76 12 - 8. UJ_F0. PRG	配合数据套表程序 UJ-TB. PRG 清理无效数据	(252)
77 12 - 9. UJ_F1. PRG	配合数据套表程序 UJ-TB. PRG 填写附属信息	(262)
78 13 - 1. ZD_JS. PRG	将字段序号表示的算式翻译成参数形式	(266)
79 13 - 2. ZD_BJ. PRG	从键盘接收字段算式并完成合法性检测	(268)
80 13 - 3. JS_TV. PRG	支持操作员完成对一个关系算式的合成	(271)
81 13 - 4. JS_TJ. PRG	在屏幕的揭示下完成逻辑算式的合成	(274)
82 13 - 5. JS_ZH. PRG	数据检索的综合实现 程序框架	(284)
83 14 - 1. WJQD. PRG	将通配符指定的文件整理成数据库形式	(289)
84 14 - 2. GIQD. PRG	将过程中的子程序整理成数据库形式	(292)
85 14 - 3. JGQD. PRG	将文件清单中的数据库结构生成库形式	(294)
86 14 - 4. JGWB. PRG	将库结构清单以表格形式输出到磁盘文件	(296)

87 14 - 5. WBDY. PRG	文本或过程文件分页打印(可分单双面)	(298)
88 15 - 1. GDK - J1. PRG	由录入控制库 LRKV. DBF 生成主库过渡结构	(306)
89 15 - 2. VYK - BG. PRG	处理结构维护时主用库数据安全过渡	(307)
90 15 - 3. GDK - J2. PRG	由录入控制库 LRKV. DBF 生成汉库过渡结构	(312)
91 15 - 4. LR - FY. PRG	字典库增加、行查询、翻译及录入控制	(314)
92 15 - 5. ZD - ZJ. PRG	用于支持数据录入时在线补充字典库数据	(318)

第一章 编程入门

为了保持全书的系统性,本章内容以初学者为对象,以简单的程序构思过程为主线索。一些对于编程非常重要的基础知识和基本概念在有关语言介绍的书籍中很少涉及,这里则侧重这些方面的内容。如果你已经具备了编程的一般基础,可以跳过本章的叙述部分,直接查看一下本章的两个小程序,这两个程序在以后要经常使用。

在语言系统启动后,用简单的交互命令尝试对语言的应用是编程的开始。本章从这个基础起点出发,介绍从交互操作过渡到程序设计的全部过程。从命题立意开始,以简单的程序为依托,详细的介绍了基本编程的基本命令和简单控制的实现方法。

§ 1 系统的启动

FOXBASE 语言支持用户在交互状态下使用。这时,计算机能够对输入的指令作出及时反映并可随时报告执行结果。这样的状态有助于用户学习和掌握各种命令、函数和状态的使用方法,也是学习本语言的有效途径。只要将包括本语言的磁盘文件拷贝到硬盘的一个子目录,然后键入启动命令就可以进入到交互状态。以 FOXPLUE 2.0 版为例,最基本的系统文件为 MFOXPLUS.EXE 和 MFOXPLUS.OVL,只要这两个文件在磁盘的当前目录中(例如 C:盘的 FOX 目录),发出命令:

```
C:\FOX>MFOXPLUS
```

就可启动该语言并进入交互状态,系统启动后会使用默认的提示符".",它表示计算机此时在 FOXBASE 系统的监控下工作。此后可以使用该语言中规定的所有命令、函数和状态,这是学习的开始。按有关书籍的介绍输入命令并回车就可以得到命令执行的结果,了解命令在交互状态下的应用是以后编程的基础。我们可以从建立一个数据库开始,逐步的学习如何对数据库进行各种操作。如果能够坚持上机练习,就能做到对本语言的初步了解和入门。

§ 2 编程的概念

在掌握了一些常用命令后,一些初学者对如何进入编程阶段总是感到困惑,较长时间的停留在初级应用阶段。这是由于对程序概念和系统状态缺乏必要的认识。本语言的程序与交互状态有着良好的兼容性,能够在交互状态下使用的命令在程序中仍然相同。

我们在交互状态下完成一项工作,需要依次发出不同的命令,直到任务完成。遗憾的是使用过的命令在退出系统时就被丢失了。如果经常需要重复某项工作,每次都要使用一个相同的命令组。用户希望能将使用过的命令组存储在磁盘上,下次就可以通过文件调用的形式完成工作,这样的想法就是程序产生的起源。记录这种操作命令的磁盘文件就是最简单的程序。

在限定语言的基础上,根据自己要解决的问题顺序组合系统命令,最终完成给定的命题,

这是程序设计最原始的思路,也是学习编程起点。作为编程的初级阶段,这种程序有很强的目的性和针对性,离真正的程序还有一些距离,但只要逐步探索,就会不断的有所提高。从这样的角度来讲,编程并不是多么神秘的事情。只要能够掌握一些常用的命令,就可以对一些简单的应用进行编程。程序不过是个能够重复使用的命令集,这样认识问题将有助于迅速的从交互应用过渡到编程角色。

对编程的学习不能以一个命题的简单实现为最终目的,更多和更重要的工作是根据应用的需要不断的对原始实现进行优化和完善。学会程序的编写并不是件困难的事,逐步的提高经过努力也可以办到,但真正达到对语言应用和编程思路的运用自如,只有经过勤奋的实践之后才能实现。

初学者在程序设计中经常会重复使用非常相似的程序段落,在这些基本相同的段落之间,绝大多数语句基本相同,只在个别语句上有些细微的差别,命题越是接近,这种程序雷同的现象就越严重,这是缺乏编程修养的表现。只要能够对语言再多一些理解或对命题再多一些概括,这种局面就会大为改观。

雷同的程序过多会使软件规模急剧膨胀,完成一个不大的课题可能会产生成百上千甚至上万条源程序。书写量的增加使出现错误的概率成倍提高,可靠性随着程序行的增加直线下降。这样的程序调整实验时令人急不可待,维护起来困难重重。由于程序与对象之间直接相关,一旦程序的对象或目标有变化,对程序进行改造将不可避免。

在一个软件产品交付使用之后,使用的环境很难保持一成不变,出于各方面的原因,要求改变程序功能和数据结构的情况时常有所发生,这些现实的要求会影响到程序的运行状态。如果程序的针对性很强,程序员就不得不频繁的维护和改造自己设计的软件产品,这是一些程序员感到十分恼火又无可奈何的局面。这种把用户和程序员紧紧的联系方式起来的软件产品,时刻面临着寿命周期的威胁,也在很大程度上牵制了程序员的精力,阻碍了程序员水平的继续提高。

这些是一些初学者容易出现的倾向,也是在学习过程中应当尽力避免的问题。实践已经证明了这样一个结论:只满足于命题实现的程序员永远不会轻松。

要改变这种被动局面的唯一途径,就是提高程序员的自身素质。不能满足于命题的一般化及基本实现,而是要在这样的基础之上不断的探索和编写一些功能性和适应性较强的通用程序。如果是一个通用性良好的程序,就会在日后的许多设计中发挥重要的作用。通用程序的积累是经验的积累,也是智慧的积累。这种积累越多,软件设计就会变得越简单,其功能和可靠性就会越高。

要以这样的追求为设计程序的目标,就要经常考虑到一个程序如何与命题可能出现的各种变化相适应,如何保证在各种变化的状态中实现可靠的控制和正确的运行。与直接面向对象的程序相比,这要投入更多的智力和时间,也会增加程序的复杂性,但从程序的寿命和长远的利益来看,这些投入应当是值得的。

习惯于简单实现的程序员在开始阶段可能会对本书中的编程不能适应,往往会产生一种错觉,认为本书将一些简单的命题设计得过于复杂化,但只要能将本书中的一些实例应用到自己的系统,就会体会到这种做法的实际价值和现实意义。

特别强调程序的通用性和适应性是本书的基本特点之一,但这并不是一个容易达到的境界,对于一些复杂的命题更是如此。由于水平的局限,这里只是就一些带有普遍性的问题做些有益的探讨,在能力许可的情况下,力求使程序比较完美。

§ 3 字符串显示程序

如何才能迅速的从交互操作过渡到实际编程是初学者普遍关心的问题,也是个很难用一两句话就说清楚的问题。如果没有具体对象,对编程的讨论只能是空洞的说教。我们还是从一个简单的实际问题开始,依照问题的提出和解决的步骤,逐步深入的讨论程序设计的构思方式和实现方法。通过一个简单的程序,介绍在编程的过程中如何使用一些基本命令和如何对具体问题进行归纳和抽象。

1. 命题的提出

这个简单的例子起源于在屏幕上显示一个提示信息。在屏幕上显示一个数据和一条说明是计算机语言中最基础的状态之一,也是任何一个软件设计所不可避免的课题之一。这个命题最简单,也最有普遍性。在程序中可以完成显示的命令有几种:

? /?? [表达式 1[, 表达式 2……]]

WAIT [提示表达式] [TO 变量]

@ 行,列 SAY <数据或提示信息表达式> [PICT <格式字符表达式>]

它们分别是:数据输出命令、等待命令和格式化输入、输出命令。

本语言规定每个命令必须由一个动词开始,后面可以有限定命令功能的参数和字句,其中用尖括号表示用户必须提供的参数部分,用方括号表示可用可不用的部分,用斜杠分隔表示在两者之间选一。这是系统命令的书写和阅读规范。

以上是三种常用的显示命令,可以根据数据类型和用途的区别选用,不同的程序员对使用这些命令时可以有自己的偏爱。如果我们对显示位置没有任何要求,使用 "?" 完成显示是最简单的方式,也确实有许多程序在使用这种方法。这样随意安排显示信息的方式,会使屏幕环境缺乏规范性,经常会令人感到很不舒服。

虽然这三种方式都可达到显示信息的目的,但出于对格式、位置的要求,各自还有一些不同的区别。其中的 "@... SAY ..." 命令可以准确控制输出信息在屏幕上坐标位置,容易形成比较规范的屏幕效果。在设计一个软件系统时,如果能够在控制显示位置方面制定一个小小的规范,屏幕效果就会让人感到规律和稳定。比如说在系统中所有的提示被安排到屏幕的一个特定行上;不论信息长短,都能被放到屏幕的中间位置,这样符合人们日常观察的习惯。

在具备了这些初步知识之后,我们试图对屏幕信息显示设计一个简单实用的程序。

2. 实现一个规范

这个小小规范看来简单,通过实践后就会发现,实现这个规范并不是件容易的事情。因为,在所有信息显示之前,计算这些符合规范的坐标是十分繁琐的事情,尤其是在事前不可预知的信息长度时,控制显示位置的想法更难实现。

如果要将这样的一行命令设计成一个简单的程序段落,很多人会认为是多此一举。本来只有一个命令行,已经简单得不能再简单了,如果要把一个程序行改写成一个程序段落,岂不是把本来简单的问题搞得复杂化了吗? 这种看法虽然有一定的理由,但不一定就正确。命令行数的多少并不是考察程序是否合理的主要依据,主要应当考察这些命令使用得是否合理和必要。