

FoxPro

程序设计教程

韩 蒙 主编

Course in
FOXPRO
Programming



电子科技大学出版社

UESTC PUBLISHING HOUSE

12

1/1

TP312
HM

FoxPro 程序设计教程

韩 蒙 主编

电子科技大学出版社

声 明

本书无四川省版权防盗标识,不得销售;版权所有,违者必究,举报有奖,举报电话:(028) 6636481 6241146 3201496

FoxPro 程序设计教程

韩 蒙 主编

出 版: 电子科技大学出版社 (成都建设北路二段四号) 邮编: 610054
责任编辑: 吴艳玲
发 行: 新华书店发行
印 刷: 成都市青羊福利东方彩印厂
开 本: 787×1092 1/16 印张 24 字数 584 千字
版 次: 1998 年5 月第一版
印 次: 1998 年5 月第一次印刷
书 号: ISBN 7—81043—938—3/TP·414
印 数: 1—4000 册
定 价: 26.00 元

内容提要

本书以当前计算机等级考试(数据库部分)的基本要求为起点,由浅入深系统地介绍了FoxPro for DOS 的基础知识和高级应用。第一至五章为基础知识部分,包含概论、基本命令、内存变量和函数、程序设计初步、输入/输出程序设计;第六、七章介绍了FoxPro 的高级应用,包含了屏幕生成器、菜单生成器;第八章简要介绍了FoxPro 的Windows 版本;第九章以一个简化的实例说明应用程序开发的过程。每章后附有习题及部分参考答案。附录中给出了FoxPro 常用的命令、函数及如何生成 APP 和 EXE 文件的方法。对FoxPro 最具特色的复合索引技术、记录的查找、事件捕捉、屏幕生成器、菜单生成器等进行了较深入全面的讨论,并给出了详细的示例程序,所有的示例程序均通过了上机调试。

本书可作为高等学校计算机数据库课程的教材,也可作为计算机的培训和工程技术人员自修计算机课程的教材。

JS364/28

前 言

本书所讲述的 FoxPro 系统主要是基于微软公司推出的 FoxPro 2.5 for DOS,之所以选择 FoxPro 2.5 for DOS,原因是:FoxPro 是当今微机上流行的较成熟的数据库系统,特别适合中小型数据库系统的设计与开发,它以良好的界面,强大的功能,对计算机硬件的要求较低,易学易用而深受软件开发人员的欢迎,从 FoxPro for DOS 开始学习数据库技术,起点适中,可以很快起步,尽快掌握当前流行的开发工具,又可为进一步学习微软的 Windows 下的数据库系统打下良好的基础;目前高校的计算机课程中的数据库部分,放弃陈旧过时的 dBASE、FoxBASE 的内容,转向流行的 FoxPro 的教学,已成为必然的趋势。为此,我们为读者编写了这本书。

本书是在编者讲课手稿的基础上重写而成的,在本书的第一至五章,我们尽量做到以高校的计算机等级(数据库部分)的考试要求为主线,在兼顾目前等级考试要求 FoxBASE 的基础上,加入 FoxPro 更深入的内容,以期能与目前的教学要求和内容做到较好的结合。

对 FoxPro 的最具特色精采的部分,如复合索引技术、屏幕生成器、菜单生成器等,我们分别在第二章、第六章、第七章都给予了深入详细的讨论,并结合具体的图例给出了有实用性完整的例子,以帮助读者理解和掌握这些内容,并能参照给出的例子,设计出自己的模块。

针对本书是一本教材而不是工具书的特点,我们在编写中,采用了传统的命令讲解的方式,对每一个命令,做到讲解尽量详细,并配合例子给予说明。所不同的是,我们尝试着把第四章以后的程序设计的一些内容,如一些简单的程序、循环、条件判断等,有意识地前移。根据已往的教学经验,如果前面仅介绍命令,而不结合具体的程序,学生不易建立整体的概念,对命令的理解也不深,教学的前后容易脱节。因此,我们建议,读者在学习时,遇到不懂的地方,可以提前简单阅读后续章节相应的内容,帮助理解当前的内容,待课程进行到后面时,前后呼应,自然水到渠成。

本书由韩蒙负责确定全书的结构和统稿,并编写了本书的第二章、第三章、第四章、第五章、第七章,朱强编写了第一章、第九章及附录,肖飞编写了第六章、第八章,肖飞选编了全部的习题,电子科技大学计算机学院的黄迪明教授在百忙中审阅了全书的内容,提出了许多宝贵的意见;武英老师、陈娟、吕文玲、张志祥同学对本书的编写做了大量的前期工作,在此表示深深的感谢!由于编者的专业水平有限,加之时间较为仓促,书中一定有不少疏漏和不足之处,我们殷切期望读者给予批评指正!

编 者

1997 年12 月于电子科技大学

目 录

第一章 数据库概论	1
1.1 数据库的理论知识	1
1.1.1 数据库系统	1
1.1.2 关系数据模型	3
1.2 FoxPro 中的基本概念	4
1.2.1 数据类型	4
1.2.2 常量、变量和函数	6
1.2.3 运算符和表达式	9
1.2.4 FoxPro for DOS 的命令	13
1.2.5 FoxPro for DOS 的文件类型	16
1.3 走近FoxPro	17
1.3.1 FoxPro 2.5 for DOS 的特色	17
1.3.2 FoxPro 2.5 for DOS 的初步使用	24
习题一	35
参考答案	37
第二章 FoxPro 基本操作命令	39
2.1 数据库的建立与初步使用	39
2.1.1 创建一个数据库文件	39
2.1.2 数据库文件的打开与关闭	43
2.1.3 数据库文件记录的输出与定位	44
2.2 数据库的维护	50
2.2.1 库结构的修改	50
2.2.2 记录的添加、插入、删除与恢复	52
2.2.3 记录的编辑修改	57
2.3 FoxPro 的磁盘文件操作命令	62
2.3.1 显示文件目录和文件内容	62
2.3.2 文件的复制	63
2.3.3 文件的更名与删除	68
2.3.4 文件管理器(FILER)	69
2.4 文本文件和数据库文件之间的数据交换	73
2.5 数据库的排序	75

2.6 数据库的索引.....	79
2.6.1 创建索引文件.....	80
2.6.2 索引文件的打开、关闭与重建索引	87
2.6.3 设定主控制索引.....	89
2.7 记录的查找.....	93
2.8 多重数据库的操作	101
2.8.1 数据库的互访	102
2.8.2 数据库的关联	104
2.8.3 数据库的联结	111
2.9 数据的统计	113
习题二.....	116
参考答案.....	121
第三章 内存变量、数组、函数	122
3.1 内存变量	122
3.1.1 内存变量的概念	122
3.1.2 内存变量的基本操作	122
3.1.3 内存变量与数据库字段变量之间的数据交换	125
3.1.4 内存变量的作用域	127
3.2 数 组	132
3.2.1 数组与数组元素的概念	132
3.2.2 数组的生成	132
3.2.3 数组与数据库间的数据互换	134
3.3 FoxPro 的函数.....	140
3.3.1 数值函数	140
3.3.2 日期和时间函数	142
3.3.3 字符串函数	143
3.3.4 类型转换函数	145
3.3.5 测试函数	147
习题三.....	149
参考答案.....	152
第四章 程序设计初步	153
4.1 结构化程序设计概述.....	153
4.1.1 程序文件	154
4.1.2 三种基本控制结构的程序设计	156
4.2 多模块程序的设计	166
4.3 调用时的数据传递	175
2	

4.3.1 利用变量的作用域实现数据传递	176
4.3.2 利用带参数的调用实现数据传递	176
4.4 事件捕捉命令	180
习题四	184
参考答案	188
第五章 输入输出程序设计	189
5.1 基本输入输出命令	189
5.1.1 基本输入语句(ACCEPT、INPUT、WAIT)	189
5.1.2 基本输出语句	191
5.2 格式化输入输出命令	195
5.2.1 格式化输入输出命令	195
5.2.2 格式化输出输入命令的应用	206
5.3 窗口设计	216
5.3.1 定义窗口(DEFINE WINDOW)	216
5.3.2 激活与显示窗口	218
5.3.3 挂起与隐藏窗口	220
5.3.4 保存与清除窗口	220
习题五	221
参考答案	224
第六章 屏幕生成器	225
6.1 屏幕生成器简介	225
6.2 桌面与窗口	229
6.3 建立对象	233
6.3.1 文字对象和图形对象	233
6.3.2 字段对象	233
6.4 给屏幕增加一些控制	255
6.4.1 触发式按钮的建立	261
6.4.2 单选钮的建立	263
6.4.3 复选框的建立	264
6.4.4 隐藏弹出式菜单的建立	266
6.4.5 下拉列表的建立	272
6.4.6 隐藏式按钮的建立	275
习题六	277
第七章 菜单程序设计	278
7.1 几种常见菜单系统程序设计举例	278

7.2 与菜单相关的操作	290
7.3 菜单生成器的使用	292
7.3.1 菜单生成器简介	292
7.3.2 菜单生成器的启动	292
7.3.3 建立主菜单条	294
7.3.4 建立菜单基的下拉弹出式菜单	298
7.3.5 菜单程序文件	299
习题七	303
第八章 FoxPro 2.5 for Windows	304
习题八	316
第九章 应用系统示例	317
9.1 系统分析	317
9.2 系统设计	318
习题九	337
附录一 FoxPro 的命令	338
附录二 FoxPro 的函数	361
附录三 生成 .APP 和 .EXE 文件	372
参考文献	375

第一章 数据库概论

1.1 数据库的理论知识

1.1.1 数据库系统

一、信息、数据

信息泛指以文字、符号、图像、声音等各种载体传达的,为人们的感官所感知并被其大脑抽象而得到的有关事物的消息、情报或知识。信息是人脑对客观事物的反映,具体的信息以人的目的性为前提。

数据是表达信息的某种符号,是信息的一种量化表示。由于计算机只能存放和处理二进制数据,因此必须把信息转化成可被计算机接受的二进制形式的数据。今天,计算机可以通过数据来表达、处理绝大多数信息。

二、数据库

对原始数据进行收集、整理、存储、分类、排序、检索、加工、统计和传输等一系列活动总称为数据处理。当今的时代是一个“信息爆炸”的时代,面对宏大的数据流,靠手工和简单工具进行数据处理已无法跟上时代。计算机的出现使数据处理发生了划时代的变革,而计算机的数据管理技术本身也经历了三个阶段:

1. 手工管理方式

早期计算机没有必要的软件支持,用户的程序不仅要设计数据处理的方法,还必须确定数据的物理存放位置即数据在内外存的地址,因而使程序高度依赖于数据存储地址。这样不仅对用户要求较高,使用户负担重,而且数据管理方式缺乏灵活与安全,限制了编程。

2. 文件管理方式

文件管理方式把有关数据组织成文件,这种文件相对独立于应用程序。应用程序与数据文件之间的接口是通过专门的软件系统——文件管理系统来实现的。文件管理系统允许用户在建立数据文件并取名后,直接在其应用程序中实现“按文件名来访问,按记录进行存储”,而不管该文件在内外存的物理地址,因为文件管理系统会帮助其管理这些文件。

由图1-1可见,数据文件与应用程序是一一对应的,这决定了文件管理有如下弱点:

- (1)数据冗余,空间利用率降低;
- (2)同一数据存放于不同的数据文件中,很容易造成数据的不一致性;
- (3)各文件之间缺乏联系,一旦数据文件离开了相应的应用程序,便失去了存在的价值。

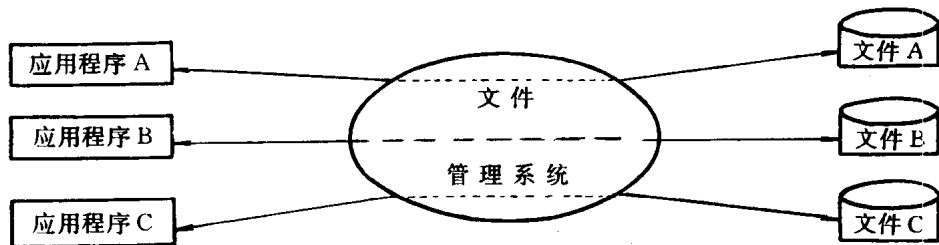


图1-1 文件管理方式示意图

3. 数据库系统方式

数据组织面向整个系统,即用整体观点规划数据,形成一个数据仓库,库中的数据能满足所有用户的不同要求。用户的应用程序不再与单个孤立的数据文件相对应,而是取整体数据库的某个子集作为逻辑文件与应用程序相对应。系统软件——数据库管理系统(DBMS)实现逻辑文件与物理数据之间的映射。

因此,数据库是以一定组织方式存储在一起的、能为多个用户所共享的、与应用程序彼此独立的相互关联的数据的集合。如图1-2 所示。

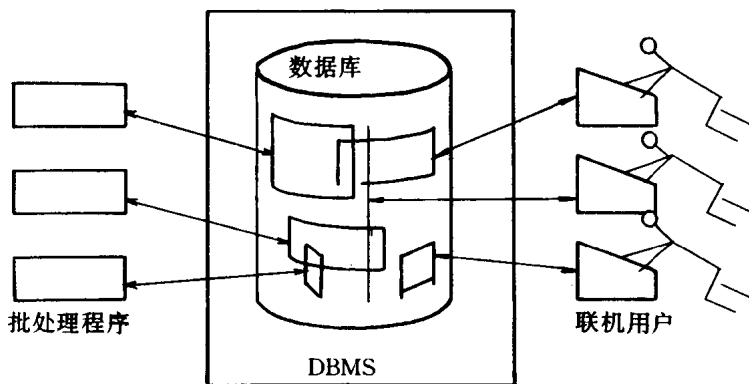


图1-2 数据库系统简图

三、数据库管理系统(Data Base Management System, 即DBMS)

上文已经谈到了DBMS,在此,不妨将图1-2 的数据库称为概念级数据库,将库中的子集称为用户级数据库,而将概念级数据库在存取介质上实际的数据集合称为物理级数据库。简单地讲,DBMS 就是要实现数据库三级之间的转换,把用户对数据库的操作转化到物理级去执行。其功能主要有:提供三级数据库的定义语言,为数据库建立一个框架;装入数据;接收、分析和执行用户的操作请求;控制整个数据库的正常运行;具有恢复和重建数据库的能力。

四、数据库系统

数据库系统指引进数据库技术后的整个计算机系统,它由如下几个部分组成:

1. 硬件及数据库。

2. 软件,包括操作系统、DBMS、编译系统及应用开发工具软件等。

数据库系统的软件主要包括支持DBMS 运行的操作系统及DBMS 本身。此外,为了开发应用系统,还要有各种高级语言及其编译系统,这些高级语言应该有和数据库的接口。FoxPro 就是一种专长于开发数据库应用系统的高级语言,它不仅与数据库有良好的接口,而且还为用户提供高效率、多功能的应用程序开发工具如应用生成器、第四代语言等各种软件系统。

3. 用户,包括数据库管理员、应用程序员和普通用户。数据库管理员(DBA)对整个数据库进行维护;应用程序员根据实际应用对数据库的某一部分进行操作;一般用户通常只需要获取有关综合性的信息,能使用一些简单的操作命令。

1.1.2 关系数据模型

一、数据模型

数据之间的联系有两种:一种是实体内部的联系,反映在数据上是记录内部的联系;另一种是实体与实体之间的联系,反映在数据上是记录与记录之间的联系。数据模型就是反映记录与记录之间的联系的。

数据模型是数据库系统的核心,它是定义数据库模式的根据,数据模型的好坏直接影响着数据库性能,故数据模型的设计方法决定了数据库的设计方法。当前比较流行的数据模型有三种:层次模型、网络模型和关系模型。

层次模型的结构是树形结构,像一棵倒挂的树,它的特征之一是:除根结点外,任何结点只有一个父亲,在这种模型中,一个父记录对应着多个子记录,而一个子记录只对应着一个父记录。如大学行政机构,校部下面有若干系(院)、处,一个系下面有若干教研室,一个教研室下面有若干教研组;反之,一个教学组向上只有一个教研室,一个教研室向上只有一个系,一个系向上只有一个校部。

网络模型是以记录类型为结点的网络结构。它的主要特点是:一个子结点可以有两个或两个以上的父结点;在两个结点之间有两种或多种联系。例如学生与所学习课程的联系,一个学生可以学习多门课程,而一门课程也可以有多个学生来学习。

二、关系模型

1970 年IBM 公司奠定了关系理论的基础。关系模型是以严密的数学理论为基础构造的数据模型。

关系(Relation)看上去是一个二维表。二维表中的列称为关系的属性,一个有 N 个属性的关系称为 N 元关系;关系中的行称为元组,一个关系可以由 M 个元组组成。例如下面就是一个含 9 个元组的 8 元关系:

在关系数据库中,数据是按关系模型组织的。其中,元组又称为记录,属性又称为字段。属性都用一个属性名称(又称字段名)表示,各字段名的集合,构成了关系的框架(又称结构)。一个关系或者说一个数据库文件是由结构和记录构成的。

编号	姓名	性别	出生日期	婚否	职称	工资	备 注
01001	龚剑	男	01/28/56	是	教授	616.20	
01103	张霄	男	02/18/65	否	技术员	283.00	福建漳州市人,1988年大学毕业后到我校工作。
02003	岳小婷	女	10/09/56	是	副教授	490.20	四川成都人,党员,1986年硕士毕业。
02010	龚剑	男	06/16/54	是	工程师	430.60	上海市人,1981年大学毕业。
03005	李啸笛	女	06/14/68	否	助教	300.20	
04020	韩子奇	男	01/28/44	是	教授	624.00	湖北宜昌市人,党员,1970年大学毕业。
05061	王立华	女	11/22/62	是	讲师	386.10	
05005	何泽红	女	08/27/44	是	高工	624.00	获国家级“有杰出贡献工作者”荣誉。
06012	张一维	男	09/07/70	否	助工	283.20	广东中山市人,1991年入党,1992年大学毕业,同年留校工作。

• 二维表作为一个关系应当满足:

1. 表示每一属性的每一列的数据同质,即属于同一数据类型且数据长度一致;
2. 每一列有唯一的字段名;
3. 表中没有完全相同的行;
4. 表中行或列的顺序可以任意交换。

关系模型由于结构简单、易于实现、表达能力强,能方便地处理多种复杂的数据结构,因此被广泛地在数据库系统中采用。目前,微机中的数据库如FoxPro、分布式数据库、智能数据库、知识库都以关系数据库为基础。

能完成关系型数据库的管理系统,必须具有下述的三种基本操作:

1. 选择(Selection)。选择运算是从关系中选出满足指定条件的元组,其运算结果是一个新的关系。具体来说,选择是根据用户要求的范围以及指定的条件对数据库记录进行筛选的横向操作。

2. 投影(Projection)。投影运算是从关系中选出指定的属性,构造其垂直子集的运算。具体来说,投影是根据用户的要求筛选数据库字段作为数据处理的对象。

3. 联结(Join)。联结运算是两个关系的运算,即对多个数据库进行横向拼结。具体来说,是根据给定的条件,将当前数据库的记录同其他数据库的记录按某种规则联结在一起。

以上三种操作可以单独进行,也可以组合进行。例如选择和投影同时进行,就可以从数据库中选出部分记录的部分字段内容作为数据处理的对象。

1.2 FoxPro 中的基本概念

1.2.1 数据类型

数据是实体信息的载体,人们通过计算机处理数据的结果反映实体信息。对于多种多样的信息,可以将具有某些相同属性的归于一类,于是有相应的一类数据表示。例如可以将文字信息用字符型数据表示,将数字信息用数值型数据表示,将图像信息用图像型数据表示,

等等。FoxPro 或其他数据库语言中,所处理的数据对象均隶属于某种分类,每一类称为一种数据类型。这一概念主要特征有:

1. 数据类型确定了常量所属值的范围,变量、表达式所能取值的范围,及运算和函数(过程)所能生成值的范围。

2. 相同类型的数据可以直接进行运算,不同类型的必须变为同一类型方可进行运算,这样可使数据处理具有安全性、有效性。

因此,数据类型实际上是一个三元组:一是值集,即是且仅是该类型的对象所有可能取值的集合;二是运算集,即是且仅是对这些对象所有实施的运算的集合;三是类型的名。FoxPro for DOS 中规定了六种数据类型。下面仅就类型名和值集两方面给予说明,至于运算集在稍后的运算符和表达式中讲述。

1. 字符型(Character)亦称 C 型。由 ASCII 基本集中128 个字符(字母、数字、控制符)、扩充集中一些图形符号、中文方式下汉字及处理汉字的符号而组成的单个或一串任意组合的字符,如'FoxPro2.5'。FoxPro for DOS 规定一个字符型数据宽度最长不超过254 个字符,可为零长度即空字符''。注意不能算两边的定界符" ",这只是字符型数据特殊的表示方法。

2. 数值型(Number)亦称 N 型。这类数据由阿拉伯数字、小数点、正负号组成。数值型数据的最大宽度(含小数点和负号在内)为20 位,正数可以没有符号位,负号占一位。数值精度为16 位。

3. 日期型(Date)亦称 D 型。这类数据只反映日期情况,有美国、欧洲等多种格式。如{05/10/97}既可以表示1997 年5 月10 日(美国格式),又可以表示1997 年10 月5 日(欧洲格式);{14/05/97}和{05/13/97}则分别只表示一种。系统默认为美国格式,要用其他格式必须首先重新设置,否则{14/05/97}便是错的。系统规定日期型数据宽度为 8 个字符。注意不能算两边的定界符"{ }",这只是日期型数据特殊的表示方法。

4. 逻辑型(Logical)亦称 L 型。这类数据用作真假或是否判断,只有真、假两个值。用.T. 或.t.(亦可.Y. 或.y.)表示真,用.F. 或.f.(亦可.N. 或.n.)表示假。系统规定逻辑型数据宽度为 1 个字符。注意不能算两边的定界符".",这只是逻辑型数据特殊的表示方法。

5. 备注型(Memory)亦称 M 型。这类数据仅作为数据库文件的备注型字段变量的值。当数据库记录的字符型字段需要存放的内容超过254 个字符时,便设置 M 型字段突破这一限制。因此 M 型字段可以当作 C 型字段进行操作。M 型字段特殊之处在于:当用户为一数据库文件定义 M 型字段时,系统并不准备在其中存放用户输入的内容,只给它设置10 个字节,而在磁盘上创建一个与该数据库文件同名、扩展名为.FPT 的备注文件用来存放用户输入的内容。每条记录 M 型字段的10 个字节则用来存放一个块号,用以指向用户输入内容在此.FPT 文件中的存放位置。在显示时,M 型字段在数据库中是Memo(有内容)或memo(为空)。

6. 浮点型(Float)亦称 F 型。也是一种数值型数据,可以存储更大范围的数。

FoxPro for Windows 中还另外规定了两种:

7. 通用型(General)。这类数据用来描述关于图像、声音、电子表格、字处理文件或其他类型的数据。该类型信息的连接与嵌入方法要使用Windows 的裁剪与粘贴技术。

8. 图像型(Picture)。这类数据用于存放屏幕上的图像的内存映像。同通用型一样,在记录的字段中存放的不是图像映像本身而是一个指示器,它指向存放该图像映像的位置。

1.2.2 常量、变量和函数

处理一具体数据时,常冠之一个名称(标识符)。这种具有名称和值的数据称为量。量是程序中最基本的要素,根据量的值在程序执行中是否改变,可以将量分为常量和变量。在FoxPro for DOS 中,它们的类型隶属于以上前六种数据类型。

一、常量

顾名思义,常量是其值在程序执行中不改变的量。根据其值所属数据类型可分为:

1. 字符型常量

如"FoxPro"、'168'、[计算机]等。字符型常量两边必须用双引号" "、单引号' '或中括号[]等定界符加以标识。

定界符可以嵌套使用,以对付这些定界符本身就是字符型常量的组成部分之复杂情况,但必须注意匹配。如:[弧度12' 30"]是正确的,而'弧度12' 30"'则错了。

空格符也是常量,如"FOX "与"FOX"就是不同的常量。还有空字符常量""。

2. 数值型常量

如12.6、-168等。数值型常量两边不需要任何定界符标识,只能是由数字0~9、小数点、正负号组合而成。

3. 日期型常量

如{05/13/97}、{13.05.97}等。日期型常量两边必须用大括号{}括起,但其组成部分——中间的分隔符可以设定为不同符号。

4. 逻辑型常量

只有.T.、.t.、.Y.、.y.、.F.、.f.、.N.、.n. 八个逻辑型常量。注意逻辑型常量两边必须有圆点“.”。

量有名和值,而常量通常是名、值合一的。因此,我们通常忽视常量的名而直接使用;两常量的值相等的前提是类型相同,类型不同则肯定不等,如138与"138"是不等的,因前者是数值型而后者为字符型。

5. 符号常量

在程序中,可以使用“编译预处理”命令#DEFINE 为某个常量定义一个标识符,例如:

```
#DEFINE PRICE 30.00
```

以后凡是在程序中出现PRICE的地方都表示30.00。为了与变量标识符区别,符号常量常用大写表示。在符号常量作用期间,不允许改变其值,例如接着执行:

```
PRICE=24.00
```

就是错的。

二、变量

变量是其值在程序执行中可以改变的量。关于变量名的约定:①变量名只能由字母、数字、汉字和下画线构成(但避免用汉字作变量名,因为不仅输入麻烦而且有时可能不受系统支持,最好取含义丰富的英文或拼音简写),长度不得超过10个字符,超过的部分被截除;②

因为系统内存变量的名称以下画线“_”开头，为了区别，其他变量名一般以字母开头但不能以数字开头；③必须注意变量名中不能有空格，还应避免用FoxPro 的命令字或函数名。例如：salary、as sign1 是正确的变量名，而97pic、fld name 则是错误的变量名。

FoxPro 中，变量分为两类：字段变量和内存变量。下面分别作简要说明。

1. 字段变量

字段变量用来描述数据库记录的属性。其名称即二维表中第一行的标题栏目。如：

编 号	书 名	编著者	出版社	单价	册数	总金额
A1001	计算机基础	黄迪明	电子科技大学出版社	12.95	30	388.50
A1002	计算机接口	曾家智	电子科技大学出版社	15.00	20	300.00
A1003	广告学	袁影	教育出版社	5.00	40	200.00
A1004	幽梦影	张潮	文艺出版社	10.00	10	100.00

上例中：编号、书名、编著者、出版社、单价、册数、总金额，都是字段变量。

字段变量的值随着当前记录的改变而改变。如上例中对于“册数”这个字段变量，若当前记录是第一条记录，则其值为30；当前记录是第三条记录，则其值为40。因此，字段变量是一种多值变量，其当前值是当前记录的该字段内容。根据其取值的类型，字段变量可分为字符型、数值型、日期型、逻辑型、浮点数型和备注型六种。如上例中：“编著者”是字符型字段变量，“册数”是数值型变量。

字段变量作用时间随数据库文件打开而开始，随数据库文件关闭而结束，因此脱离数据库文件谈字段变量是没有意义的。

2. 内存变量

内存变量独立于数据库文件而存在。分为两类：系统内存变量和用户定义变量。

(1)系统内存变量(System Memory Variables)

系统内存变量是FoxPro 自动创建和维护的固有内存变量。系统内存变量的名称以下画线“_”开头，其中保存着FoxPro 系统的一些默认设置，例如：屏幕生成程序、菜单生成程序、打印驱动程序等应用程序的名称和路径，当前剪贴板的内容，计算器、日历本等自带桌面程序的状态，编辑器中的TAB 的长度，文本是否缩排、是否回绕，打印机的设置，打印页面的行间距字间距、页的长度，等等。可以返回或重新设置这些系统内存变量的值，其基本用法如下：

- ? <系统内存变量名> →返回系统内存变量的设置
- <系统内存变量名>=<表达式> →重新设置系统内存变量的值

各个系统内存变量的类型是固定的，有些系统内存变量的值也是规定了，使用时一定要注意。关于各个系统内存变量的功能和用法本书省略，有兴趣的读者可以查看<F1>的在线帮助。

(2)用户定义内存变量

用户定义了一个内存变量(给出了名称并赋了初值)后，系统即在内存中为之开辟一工作单元。用户可以在其中存放常量、中间结果，用来得出最终结果或控制程序运行。用户定

义一个内存变量,可以用STORE 命令或赋值运算符 =。例如:pi = 3.1415,STORE "MIKE" TO name 表示把数值3.1415 和字符串"MIKE" 分别存放在内存变量pi 和name 中。

内存变量的值和类型是由最近赋给它的值和此值的类型决定的。例如:

```
m.ok={05/15/97}
```

```
...
```

```
m.ok=.t.
```

那么内存变量m.ok 的值为.t.,类型为逻辑型。这同时也说明:在FoxPro 中,不仅可以按需要随时改变内存变量的值,还可以改变其类型。

还可以将当前屏幕的画面存入一个自定义内存变量,这种变量的类型将是 S 型。例如:

```
SAVE SCREEN TO scr &&. 将当前屏幕的画面存入scr
```

```
? TYPE("scr") &&. 询问变量scr 的类型,返回 S
```

3. 两类变量的差异

表1-1 中列出了字段变量与内存变量的主要差异。

表1-1 字段变量和内存变量的比较

字段变量	内存变量
①是数据库文件的组成部分	①独立于数据库文件而存在
②随数据库文件的定义而建立,随数据库文件的打开而使用。不可用STORE 或= 命令给字段变量赋值、改变类型	②随时建立、使用。可以用STORE 或= 命令给普通内存变量赋值、改变类型
③有 C、N(F)、D、L、M 五种类型	③有 C、N、D、L、S 五种类型
④多值变量	④单值变量
⑤关机后,保存在数据库文件中	⑤除非事先存入内存变量文件,关机不保存

两类变量的命名规则一样,如果内存变量与打开的数据库文件的字段同名,FoxPro 将不加区别地都按字段变量对待。为避免这种情况,常在内存变量名前加上“m.”以示区别。

三、函数

函数是由FoxPro 提供的一些标准子程序,它接受固定数量的参数(自变量),返回一个固定类型的值,因此可以放在程序中作为一个完成一定功能的模块,也可以作为最基本的量放在表达式中(见表达式和运算符)。函数的形式是:

函数名(参数表)

函数名后有一对圆括号,圆括号内是用逗号分开的一系列参数(注意圆括号、逗号都必须是英文形式),参数的个数可有0 个、1 个或多个。一般来讲,函数值是处理自变量后得的结果,自变量不同,得的结果也不同,在这种意义上讲,它是一种特殊的变量。在书写函数时可以只写前四个字母,例如ALLTRIM()可以写成ALLT()、ALLTR()、ALLTRI()。

FoxPro for DOS 提供了200 多个函数。本书对于在前面章节的例子中出现的函数,为避