

QuickBASIC 4.50 版

程序设计粹要

秦笃烈 童隆正 编写
蔡力群 张 颖

秦笃烈 审校

目 录

序言	(1)
0.1 QuickBASIC语言	(1)
0.2 QuickBASIC环境	(1)
0.3 使用本手册	(2)
0.3.1 程序设计专题选讲	(2)
0.3.2 BASIC 便 览	(2)
0.3.3 附录	(2)
0.4 文档约定	(3)
0.5 本手册程序设计风格	(4)

第一部分 程序设计专题选讲

第一章 流程控制结构	(5)
1.1 改变语句执行的顺序	(5)
1.2 布尔表达式	(6)
1.3 判别结构	(8)
1.3.1 IF...THEN...ELSE语句块	(9)
1.3.2 SELECT CASE 语 句	(11)
1.4 循环结构	(18)
1.4.1 FOR...NEXT 循 环	(16)
1.4.2 WHILE...WEND循环	(21)
1.4.3 DO...LOOP 循 环	(21)
1.5 应用示范	(26)
1.5.1 支票簿结算程序	(26)
1.5.2 回车换行过滤程序	(27)
第二章 SUB过程和FUNCTION过程	(30)
2.1 过程：程序设计的构件	(30)
2.2 SUB过程和FUNCTION 过程的比较	(36)

2.2.1 SUB和GOSUB 的比较	(31)
2.2.1.1 局部变量和全局变量	(31)
2.2.1.2 在多模块程序中的用法	(31)
2.2.1.3 在不同的变量集上的操作	(31)
2.2.2 FUNCTION和DEF FN的比较	(32)
2.2.2.1 局部变量和全局变量	(32)
2.2.2.2 对传送给过程的变量进行修改	(32)
2.2.2.3 在过程定义范围内调用过程本身	(33)
2.2.2.4 在多模块程序中的用法	(33)
2.3 过程的定义	(33)
2.4 过程的调用	(34)
2.4.1 调用FUNCTION 过程	(34)
2.4.2 调用SUB过程	(35)
2.5 把变元传送给过程	(36)
2.5.1 参数和变元	(36)
2.5.2 传送常量和表达式	(37)
2.5.3 传送变量	(38)
2.5.3.1 传送简单变量	(38)
2.5.3.2 传送整个数组	(39)
2.5.3.3 传送数组个别元素	(39)
2.5.3.4 数组界函数的使用	(40)
2.5.3.5 传送一个完整的记录	(40)
2.5.3.6 传送单个记录元素	(41)
2.5.4 用DECLARE 语句检查变元	(41)
2.5.4.1 QuickBASIC不生成 DECLARE 语句的情况	(42)
2.5.4.2 在QuickBASIC环境外部开发程序	(42)
2.5.4.3 使用包括文件进行申明	(43)
2.5.4.4 在Quick库内申明过程	(45)
2.5.5 按参考传送变元	(45)
2.5.6 按值传送变元	(46)
2.6 用 SHARED 实现变量共享	(47)
2.6.1 和模块中的特定过程共享变量	(47)
2.6.2 和模块中的所有过程共享变量	(48)
2.6.3 不同模块的变量共享	(50)
2.6.4 变量串用问题	(52)
2.7 自动变量和STATIC变量	(53)
2.8 用STATIC语句保持局部变量的值	(53)
2.9 递归过程	(54)
2.9.1 阶乘函数	(54)

2.9.2 调整堆栈的大小	(55)
2.10 用CHAIN把控制转到另一个程序	(55)
2.11 应用示范: 递归搜索目录 (WHEREIS.BAS)	(57)
第三章 文件和设备 I/O	(60)
3.1 在屏幕上显示文本	(60)
3.1.1 屏幕上的行与列	(60)
3.1.2 用PRINT 显示文本 和数	(60)
3.1.3 用PRINT USING 显示格式化 输出	(62)
3.1.4 跳过空格和前进到特定列	(62)
3.1.5 改变行数和列数	(63)
3.1.6 建立文本视见窗	(63)
3.2 从键盘输入	(65)
3.2.1 INPUT 语 句	(65)
3.2.2 LINE INPUT 语句	(66)
3.2.3 INPUT\$ 函 数	(66)
3.2.4 INKEY\$ 函 数	(67)
3.3 控制文本光标	(67)
3.3.1 光标定位	(68)
3.3.2 改变光标的形状	(68)
3.3.3 确定光标位置	(69)
3.4 使用数据文件	(69)
3.4.1 数据文件的组织方式	(70)
3.4.2 顺序存取文件和随机存取文件	(70)
3.4.3 打开数据文件	(70)
3.4.3.1 BASIC 的 文 件 号	(71)
3.4.3.2 BASIC 的 文 件 名	(71)
3.4.4 关闭数据文件	(72)
3.4.5 使用顺序文件	(72)
3.4.5.1 顺序文件中的记录	(72)
3.4.5.2 将数据存入新的顺序文件	(74)
3.4.5.3 从顺序文件中读取数据	(75)
3.4.5.4 向顺序文件添加数据	(75)
3.4.5.5 将数据写入顺序文件的其它方式	(75)
3.4.5.6 将数据从顺序文件中读出的其它方式	(77)
3.4.6 使用随机存取文件	(78)
3.4.6.1 随机文件中的记录	(78)
3.4.6.2 向随机文件添加数据	(79)
3.4.6.3 顺序地读取数据	(82)
3.4.6.4 用记录号检索记录	(83)

3.4.7 二进制文件 I/O	(84)
3.4.7.1 二进制存取和随机存取的比较	(84)
3.4.7.2 用SEEK对文件指针定位	(85)
3.5 利用设备进行工作	(87)
3.5.1 设备I/O与文件I/O之间的区别	(87)
3.5.2 通过串行端口进行通讯	(88)
3.6 应用示范	(89)
3.6.1 万年历	(89)
3.6.2 建立随机存取文件的索引 (INDEX.BAS)	(92)
3.6.3 终端仿真程序 (TERMINAL.BAS)	(97)
第四章 字符串处理	(100)
4.1 定义字符串	(100)
4.2 变长和定长字符串	(101)
4.2.1 变长字符串	(101)
4.2.2 定长字符串	(101)
4.3 字符串结合	(102)
4.4 字符串比较	(103)
4.5 字符串查找	(104)
4.6 检索部分字符串	(105)
4.6.1 从字符串左侧检索字符	(105)
4.6.2 从字符串右侧检索字符	(106)
4.6.3 从字符串的任意部位检索字符	(107)
4.7 产生字符串	(107)
4.8 改变字母大小写	(108)
4.9 字符串和数	(108)
4.10 字符串修改	(109)
4.11 应用示范: 把字符串转换成数 (STRTONUM.BAS)	(109)
第五章 图形	(110)
5.1 运行图形程序的要求	(110)
5.2 象素和屏幕坐标	(111)
5.3 画基本图素: 点、线、方框和圆	(112)
5.3.1 用PSET和PRESET画点	(112)
5.3.2 用LINE画直线和方框	(113)
5.3.2.1 使用STEP选项	(113)
5.3.2.2 画方框	(114)
5.3.2.3 画虚线	(115)
5.4 用CIRCLE画圆和椭圆	(116)
5.4.1 画圆	(116)
5.4.2 画椭圆	(116)

5.4.3 画弧	(118)
5.4.4 画饼形图	(119)
5.4.5 利用方位比确定所画的图形的比例	(120)
5.5 定义图形视见窗	(121)
5.6 使用WINDOW重新定义视见窗坐标	(123)
5.6.1 坐标对的次序	(125)
5.6.2 跟踪物理和观察坐标	(125)
5.7 利用颜色	(126)
5.7.1 选择图形输出的颜色	(127)
5.7.2 修改前景和背景颜色	(127)
5.7.3 利用PALETTE和PALETTE USING改变颜色	(129)
5.8 为图形着色	(131)
5.8.1 着色	(131)
5.8.2 用图案着色: 贴面	(133)
5.8.2.1 在不同屏幕模式中贴面图案的大小	(133)
5.8.2.2 在屏幕模式2中建立单色图案	(133)
5.8.2.3 在屏幕模式1中建立多色图案	(135)
5.8.2.4 在屏幕模式8中建立多色图案	(138)
5.9 DRAW: 图形宏语言	(139)
5.10 动画基本技术	(141)
5.10.1 用GET贮存图象	(141)
5.10.2 用PUT使图形移动	(143)
5.10.3 用GET和PUT实现动画	(146)
5.10.4 利用屏幕页面实现动画	(150)
5.11 应用示范	(151)
5.11.1 条线图生成程序 (BAR.BAS)	(151)
5.11.2 用数学方法生成的图形的颜色	(155)
5.11.3 图案编辑程序 (EDPAT.BAS)	(161)
第六章 错误和事件捕捉	(167)
6.1 错误捕捉	(167)
6.1.1 激活错误捕捉	(167)
6.1.2 编写错误处理例行程序	(168)
6.1.2.1 用ERR标识错误	(168)
6.1.2.2 从错误处理例行程序返回	(169)
6.2 事件捕捉	(170)
6.2.1 用轮询方式检测事件	(171)
6.2.2 用陷阱检测事件	(171)
6.2.3 指定要捕捉的事件和激活事件捕捉	(171)
6.2.4 BASIC可捕捉的事件	(172)

6.2.5 暂停或中止事件捕捉.....	(172)
6.2.6 捕捉击键.....	(173)
6.2.6.1 捕捉用户自定义的键.....	(174)
6.2.6.2 捕捉用户定义的组合键.....	(174)
6.2.7 捕捉音乐事件.....	(176)
6.3 在SUB和FUNCTION过程中的错误和事件捕捉.....	(178)
6.4 跨模块捕捉.....	(178)
6.4.1 跨模块事件捕捉.....	(178)
6.4.2 跨模块错误捕捉.....	(179)
6.5 捕捉用BC命令编译的程序中的错误和事件.....	(181)
6.6 应用示范: 捕捉文件访问错误(FIL ERR.BAS)	(182)
第七章 利用模块进行程序设计	(184)
7.1 为何使用模块.....	(184)
7.2 主模块.....	(185)
7.3 仅包含过程的模块.....	(185)
7.4 生成仅含过程的主模块.....	(186)
7.5 装入模块.....	(186)
7.6 多模块中DECLARE语句的用法.....	(186)
7.7 从两个或多个模块中存取变量.....	(187)
7.8 程序开发过程中模块的应用.....	(187)
7.9 模块的编译和连接.....	(187)
7.10 Quick 库	(188)
7.10.1 Quick 库的建立.....	(188)
7.11 做好模块化程序设计工作的提示.....	(189)

第二部分 BASIC便览

第八章 语句和函数摘要	(190)
第九章 快速参照表 :	(209)
9.1 流向控制语句摘要.....	(209)
9.2 BASIC过程中使用的语句摘要.....	(210)
9.3 标准I/O语句摘要.....	(211)
9.4 文件I/O语句摘要.....	(211)
9.5 字符串处理语句和函数摘要.....	(212)
9.6 图形语句和函数摘要.....	(213)
9.7 捕捉语句和函数摘要.....	(214)
附录A 把BASICA程序转换成QuickBASIC	(216)
A.1 源文件格式.....	(216)
A.2 QuickBASIC禁用的语句和函数.....	(216)
A.3 需要修改的语句.....	(216)

A.4 编辑程序在处理制表符 Tab 方面的差别.....	(217)
附录B 和 QuickBASIC 以前版本的差别	(218)
B.1 QuickBASIC 特点.....	(218)
B.1.1 QuickBASIC4.5 的新特点.....	(219)
B.1.2 QuickBASIC4.0 的特点.....	(219)
B.1.2.1 用户定义的类型.....	(219)
B.1.2.2 支持IEEE格式和数学协处理器.....	(219)
B.1.2.3 IEEE格式数的范围.....	(220)
B.1.2.4 PRINT USING 和 IEEE 格式数.....	(220)
B.1.3 用/MBF对老版程序重新编译.....	(220)
B.1.4 对文件和程序进行转换.....	(220)
B.1.5 QuickBASIC 其它特点.....	(222)
B.1.5.1 长整数 (32比特).....	(222)
B.1.5.2 定长字符串.....	(222)
B.1.5.3 对输入的语法检查.....	(222)
B.1.5.4 二进制文件I/O.....	(222)
B.1.5.5 FUNCTION过程.....	(222)
B.1.5.6 对CodeView调试程序的支持.....	(223)
B.1.5.7 与其它语言兼容性.....	(223)
B.1.5.8 内存中的多模块.....	(223)
B.1.5.9 ProKey, SideKick及SuperKey 兼容性.....	(223)
B.1.5.10 插入和重打模式.....	(223)
B.1.5.11 WordStar 式键盘命令.....	(223)
B.1.5.12 递归.....	(223)
B.1.5.13 在分别编译过程中的出错列表.....	(223)
B.1.5.14 在独立编译过程中的汇编语言列表.....	(224)
B.2 环境的差别.....	(224)
B.2.1 选择命令和选项.....	(224)
B.2.2 窗口.....	(224)
B.2.3 新菜单.....	(224)
B.2.4 菜单命令.....	(224)
B.2.5 编辑键修改.....	(225)
B.3 在编译和调试方面的差别.....	(225)
B.3.1 命令行差别.....	(225)
B.3.2 独立编译的差别.....	(226)
B.3.3 用户库和BUILDLIB.....	(226)
B.3.4 关于包括文件的限制.....	(226)
B.3.5 调试.....	(226)
B.4 对 BASIC 语言的修改.....	(227)

B.5 文件兼容性.....	(230)
附录C QuickBASIC 的限制.....	(230)
附录D 键盘扫描代码和 ASCII 代码.....	(232)
D.1 键盘扫描代码.....	(232)
D.2 ASCII字符代码.....	(233)
附录E BASIC 保留字.....	(235)
附录F 元命令.....	(235)
F.1 元命令语法.....	(236)
F.2 处理附加的源文件: \$INCLUDE.....	(236)
F.3 已定维数组内存分配: \$STATIC 和 \$DYNAMIC.....	(236)
附录G 从DOS下编译和连接.....	(237)
G.1 BC, LINK 和 LIB.....	(237)
G.2 编译和连接过程.....	(237)
G.3 用BC命令进行编译.....	(238)
G.3.1 指定文件名.....	(238)
G.3.1.1 大写和小写字母.....	(239)
G.3.1.2 文件扩展名.....	(239)
G.3.1.3 路径名.....	(239)
G.3.2 利用BC命令选项.....	(239)
G.4 连接.....	(240)
G.4.1 LINK的默认值.....	(241)
G.4.2 指定要连接的文件.....	(242)
G.4.3 指定要连接的库.....	(242)
G.4.4 连接命令LINK的内存要求.....	(243)
G.4.5 和混合语言程序的连接.....	(243)
G.4.5.1 QuickBASIC程序中的Pascal和 FORTRAN 模块.....	(244)
G.4.5.2 在汇编语言子程序中的静态数组分配.....	(244)
G.4.5.3 在扩充的运行态模块中对 DGROUP 的参考.....	(244)
G.4.6 利用LINK 选项.....	(244)
G.4.6.1 观察选项表 (/HE).....	(245)
G.4.6.2 在连接过程中的暂停(/PAU).....	(245)
G.4.6.3 显示连接程序处理信息 (/I).....	(245)
G.4.6.4 阻止连接程序给出提示 (/B).....	(245)
G.4.6.5 建立Quick 库 (/Q).....	(246)
G.4.6.6 对可执行文件的包装 (/E).....	(246)
G.4.6.7 使节段包装失效 (/NOP).....	(246)
G.4.6.8 忽视通常的BASIC库 (/NOD).....	(246)
G.4.6.9 忽视词典 (/NOE).....	(246)
G.4.6.10 设置最大节段数 (/SE).....	(246)

G.4.6.11 建立映象文件 (/M)	(247)
G.4.6.12 在映象文件中包括行号 (/LI)	(248)
G.4.6.13 对相邻节段进行包装 (/PAC)	(248)
G.4.6.14 利用Code View调试程序(/CO)	(248)
G.4.6.15 区别大小写 (/NOI)	(248)
G.4.7 其它LINK命令行选项	(248)
G.5 管理独立的库: LIB	(249)
G.5.1 运行LIB	(249)
G.5.2 对LIB常用的响应	(250)
G.5.3 交叉参考列表文件	(251)
G.5.4 命令符号	(251)
G.5.5 LIB选项	(252)
G.5.5.1 忽略符号的大小写	(252)
G.5.5.2 忽略扩充词典	(252)
G.5.5.3 区分符号的大小写	(252)
G.5.5.4 设置页面的大小	(252)
附录H 建立和使用 Quick 库	(254)
H.1 库的类型	(254)
H.2 Quick 库的优点	(254)
H.3 建立 Quick 库	(255)
H.3.1 建立 Quick 库需要的文件	(255)
H.3.2 建立 Quick 库	(256)
H.3.3 从环境内部建立 Quick 库	(256)
H.3.3.1 卸载不需要的文件	(256)
H.3.3.2 装入需要的文件	(256)
H.3.3.3 建立 Quick 库	(256)
H.4 Quick 库的用法	(257)
H.4.1 装Quick库	(257)
H.4.2 Quick库中的浮点算术运算	(258)
H.4.3 考察 Quick 库的内容	(258)
H.5 系统提供的库 (QB.QLB)	(258)
H.6 文件扩展名.QLB	(258)
H.7 从命令行建库	(259)
H.8 在 Quick 库中使用其它语言的例行程序	(259)
H.8.1 建造 Quick 库	(259)
H.8.2 在第一代码节段中有前导零的 Quick 库	(260)
H.8.3 例行程序B_OnExit	(260)
H.9 对 Quick 库的内存考虑	(261)
H.10 建立紧凑的可执行文件	(261)

附录I 出 错 报 文	(261)
I.1 出错报文显示.....	(262)
I.2 调用, 编译态, 运行态出错报文.....	(263)
I.3 连接中的出错报文.....	(280)
I.4 库出错报文.....	(287)

序 言

Microsoft QuickBASIC 4.5在使BASIC更强有力和容易使用方面取得重大进展。它是目前微机上提供的最先进的BASIC。它的操作环境使用户可以集中精力于程序创作,而少考虑书写或调试方面的工作。

0.1 QuickBASIC语言

如果对于用BASIC(或类似的解释性BASIC)编写程序已经有所了解,就会对QuickBASIC提供的使软件容易写和容易维护的增强的语言功能表示欣赏。例如:

●SELECT CASE语句能清晰地把控制转移到任何代码段从而可以不必再使用嵌套的IF-THEN-ELSE语句。SELECT CASE允许对范围极宽的表达式进行测试,所以能确切地建立需要的比较。

●QuickBASIC的SUB和FUNCTION过程允许把多组程序语句放入主程序能反复调用的子程序。QuickBASIC的模块很容易把这些过程贮存起来并在其它程序中加以使用。

●QuickBASIC过程完全支持递归——过程可以不断地调用自身。这一特点使得最好用递归表示的许多数值和排序算法的程序设计工作简化。

●可以定义自己的数据类型,它们由整数,实数和字符串变量组合而成。有关系的一组变量可以结合为一个名称,使得把它们传送给过程或将它们写入文件的过程得以简化。

●QuickBASIC支持二进制文件访问。因为二进制I/O可以直接访问文件中的任何字节,所以程序可以按任何格式读文件并对它们进行操作。

QuickBASIC是供专业人员使用的强有力的开发工具。然而,它也是初中级程序设计人员的理想语言。他们虽然不是专业程序设计人员,但是需要一种语言的帮助有效地实现他们的程序设计目标。

0.2 QuickBASIC环境

QuickBASIC不仅是一种优秀的语言,它也是一种集成化的程序设计环境,能够使软件的书写和调试工作显著简化。

●当用户输入程序时,灵巧的编辑程序检查每一行的语法错误,要执行程序时,按一个键就可以使程序马上运行。如出现问题,可以使用全屏幕编辑程序纠正错误,然后再运行程序。

●能够不退出QuickBASIC对程序进行调试,集成化的调试程序使用户能观察和修改变量,执行程序的任何部分,也可以在遇到特殊情况时中止执行。所有这些事情都是在QuickBASIC环境内部完成,不必去修改程序或增加什么PRINT语句。

●QuickBASIC 4.5有两个新命令使调试程序更为强大:Instant Watch(立即观察)命令和Break on Errors(遇到错误即中断)命令。

●Microsoft QB Advisor也是QuickBASIC 4.5的新特征,它是系统提供的联机屏幕帮助。无论是书写、运行还是在调试程序时, QB Advisor总是随时可用。只要把光标放在需要得到帮助的关键字或用户定义名称上并按F1, QB Advisor就会对BASIC语句和函数进行说明,对使用方法进行解释,并提供可用的程序设计实例。

0.3 使用本手册

本手册共分三部分。第一部分是有关选择的程序设计专题讨论,它提供有关特殊的程序设计技巧和方式的信息。第二部分是BASIC语句和函数的便览,而附录包含了重要的参考材料。

0.3.1 程序设计专题选讲

第一部分的每一章集中讨论一个程序设计领域。学习这部分材料会有助于很快掌握。

- 流程控制结构
- SUB和FUNCTION过程
- 文件和设备输入输出
- 字符串处理
- 图形
- 错误和事件捕捉
- 用模块进行程序设计

每个专题的介绍都是开门见山而且容易理解,并带有能演示BASIC的每一部分工作情况的简短的程序设计实例。专题是按从简单到复杂的方式安排的,所以可以按照自己的速度阅读这一部分材料而不必考虑按什么顺序进行学习。贯穿全书的中心点是如何用QuickBASIC解决常见的程序设计问题的实用方法而不是突出理论。

除了简短的例子以外,各章还包含完整的工作程序,能演示该章介绍的程序设计原则。为方便起见,这些程序已包括在QuickBASIC的发行盘中。

对于已经有经验的程序设计人员,可以浏览目录以便找出感兴趣的专题。对于程序设计初学者,应该从头到尾通读每一章。对于没有用任何语言进行程序设计的读者,应该先阅读“QuickBASIC 4.5环境粹要”一书第4章。

不论兴趣和背景是什么,这七章将帮助读者掌握如何写复杂的BASIC应用程序应该知道的一切东西。

0.3.2 BASIC便览

这一部分手册内容提供对BASIC语句和函数的释要解释以及各种交互参照表。

第8章是按字母顺序排列的每个BASIC关键字的扼要说明,介绍它的作用或用法并给出语法说明。如果读者希望对语句或函数用法作记忆练习可利用这一部分内容。

第9章以表格形式分六节介绍最常用的BASIC语句和函数,每一种语句都有扼要说明。这六节的内容和第1到第6章介绍的内容相匹配。对于希望弄清如何完成特殊的程序设计任务的读者可阅读这部分材料。

0.3.3 附录

本手册的第三部分是包含参考性信息的附录,有关内容是:

- 对BASICA程序的转换

- 和以前版本的差别
- QuickBASIC的限制
- 键盘扫描码和ASCII代码
- BASIC保留字
- 元命令
- 从DOS进行编译和连接
- 建立和使用Quick库
- 出错报文

0.4 文档约定

本手册使用以下印刷约定：

约定的例子	说 明
QB.LIB, ADD.EXE, COPY, LINK, /X SUB, IF, LOOP, PRINT, WHILE, TIMES\$	大写字母表示文件名和DOS级命令。它也用于命令行选项（只接受小写的应 用除外）。 黑体大写字母（注：译文在印刷上难以做到这一点，经常使用普通大写字 母）表示对Microsoft BASIC（有特殊含义的语言未用关键字。除非把关 键字放在下面说明的双方括弧内，它们是语句语法所需要的部分。在所写的 程序中，必须确切地输入关键字，但是大小写可以不拘。
CALL Newproc (arg1, var2%) '\$INCLUDE:'BC.BI' . . . CHAIN"PROG1" END 'Make one pass <i>filespec</i> [<i>optional-item</i>] { <i>choice1</i> <i>choice2</i> }	这种字体为程序实例，程序输出以及出错报文所使用。 一行中的三个圆点表示示范程序的一部分有意省略掉。
ALT+F1	撇号（右单引号）标志示范程序中注解的开始。 斜体字母表示必须提供的信息（如文件名）的位置标志。 双方括弧内的项目是可选的。 花括弧和垂直条表示在两个或多个项目中选一个。除非所有项括在双方括弧 内，必须选择一项。 键名和键序列用大写字母表示，如ENTER和CTRL+R。加号（+）表示 键的组合。例如，CTRL+E表示在按下E键的同时按下CTRL键。回车换 行键有时用“↵”表示，它指的是ENTER键。在数字小键盘上的光标移 动（“箭头”）键称为方向键。它们用键上部的方向（左、右、上、下）或名 称（PGUP, PGDG）标志。本手册所用的键名和IBM个人计算机键名一 致。其它机器可能有不同的名称。

下面的（对“LOCK...UNLOCK”语句）语法代表本手册中的许多印刷约定。

LOCK[#]filenumber[, {record|[start]TO end}]

.
.
.

UNLOCK[[#]]filename[[, {record|[[start]]TO end}]]

注意：在整个手册中，“DOS”是指MS-DOS和IBM个人计算机DOS操作系统。当需要说明系统独特功能时会特别地使用该操作系统的名称，“BASICA”一般是指BASIC的解释版本。

0.5 本手册程序设计风格

在本手册及发行磁盘中书写程序采用以下原则。这些原则仅仅是为了增加程序可读性的建议；读者在写自己的程序时不一定勉强模仿。

●关键字和符号常数以大写字母出现。

```
' PRINT, DO, LOOP, UNTIL are keywords,
PRINT "Title Page"
DO LOOP UNTIL Responses="N"

' FALSE and TRUE are symbolic constants
' equal to 0 and -1, respectively,
CONST FALSE=0, TRUE=NOT FALSE
```

●变量名用小写但首字母大写。有几个小节的变量名可以包含几个大写字母以表示小节的分隔。

```
NumRecords%=45
DateOfBirths="11/24/54"
```

●行标号用于代替行号，行标号的使用限于事件捕捉和错误处理例行程序，以及用RSE TORE参考的DATA语句。

```
' TimerHandler and ScreenTwoData are line labels:
ON TIMER GOSUB TimerHandler
RESTORE ScreenTwoData
```

●上面已提到，撇号代表注解语句的开始：

```
' This is a comment; these two lines
' are ignored when the program is running.
```

●在过程或子程序中的流程控制段和语句从封闭代码中缩入一层：

```
SUB GetInput STATIC
  FOR I%=1 TO 0
    INPUT X
    IF X>0 THEN
      .
      .
      .
    ELSE
      .
      .
      .
    END IF
  NEXT I%
END SUB
```

第一部分 程序设计专题选讲

第一部分介绍用BASIC进行程序设计的基础知识。先介绍较简单的专题。

第一章讨论安排程序执行的流程控制结构。第二章是介绍两种非常有力的程序设计工具：QuickBASIC的SUB和FUNCTION过程。第三章介绍用QuickBASIC和程序接受及产生的数据打交道的方法。第四章包括文本字符串的用法，而第五章介绍 QuickBASIC 的图形功能。

在最后两章介绍更高级的专题。第六章介绍错误和事件捕捉，而第七章则讨论如何用模块进行程序设计而获益。

第一章 流程控制结构

本章说明如何使用流程控制结构，特别是为控制程序执行的流程的循环和判定语句。循环使程序按照用户需要的次数执行语句序列。判定语句让程序决定走若干可供选择路径中的哪一条。

读完这一章，我们可以了解如何在BASIC程序中用循环和判定语句完成下述工作：

- 用关系运算符比较表达式。
- 用逻辑运算符组合字符或数字表达式，以及确定表达式的结果是真是假。
- 用IF...THEN...ELSE和SELECT CASE语句在程序流程中建立分枝结构。
- 写重复执行特定次数语句的循环。
- 写当确定的条件为真或直到满足确定条件重复执行语句的循环。

1.1 改变语句执行的顺序

当不受流程控制结构的控制时，程序的逻辑流向从左到右，从上到下地通过语句。只有些很简单的程序才使用单流向的流程，而程序设计语言大部分的效能和实用性来自于它能用判定和循环结构改变语句执行的顺序。

用判定结构，程序可对表达式求值，然后依据结果转移到几个不同的相关语句组（语句块）之一。用循环结构，程序可重复执行语句块。

如果在接触此BASIC版本之前你曾用BASICA设计程序，那么你将欣赏到这些外加的流程控制结构所增加的功能

- IF...THEN...ELSE语句块
- SELECT CASE语句