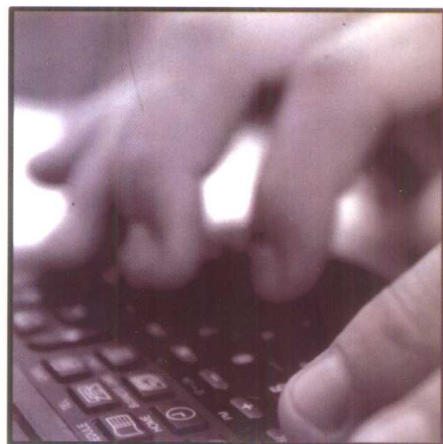


Quick BASIC

程序设计

(修订版)

刘炳文 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.co.cn>

Quick BASIC 程序设计

(修订版)

刘炳文 编著

电子工业出版社
Publishing House of Electronics Industry

内 容 简 介

Quick BASIC 是在解释型 BASIC 基础上研制而成的, 是一种完全结构化和模块化的程序设计语言。它把解释型 BASIC 的方便性和编译 BASIC 的执行速度有机地结合在一起, 功能强、界面友好、易学习, 并具有当代计算机语言的基本特征。

本书全面、系统地介绍了 Quick BASIC 编程环境和程序设计方法。可作为大专院校教材, 可供电脑爱好者自学, 也可供计算机应用开发人员参考。

书 名: Quick BASIC 程序设计 (修订版)

编 著: 刘炳文

责任编辑: 吴剑锋

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河市金马印装有限公司

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话: 68214070

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 25.125 字数: 640 千字

版 次: 1998 年 3 月第 1 版 1998 年 3 月第 1 次印刷

书 号: ISBN 7-5053-4494-3
TP·2094

定 价: 32.00 元

凡购买电子工业出版社的图书, 如有缺页、倒页、脱页者, 本社发行部负责调换
版权所有·翻印必究

前 言

BASIC 是一种国内外广泛使用的、适合于初学者的程序设计语言，它简单而不贫乏，易学而实用，适用于数值计算、事务处理、绘图、游戏等。70 年代中叶，Microsoft 公司以 BASIC 起家，逐渐发展成为世界上首屈一指的软件公司。20 多年来，随着计算机软、硬件环境的变化，BASIC 语言的版本也在不断更新。Microsoft 公司继 BASICA 之后，于 1987 年推出了 Quick BASIC，在此基础上，增加了“可视化”功能，于 1991~1993 年推出了在 Windows 3.x 下运行的 Visual Basic (1.0~3.0 版)，1995 年和 1997 年又推出了可在 Windows 95 环境下运行的 32 位 Visual Basic 版本，即 Visual Basic 4.0 和 5.0，使得 BASIC 成为可用来进行大型程序设计的计算机语言。

Quick BASIC 是在早期 BASIC 的基础上研制而成的，但对原来 BASIC 的改进是革命性的。它吸取了 PASCAL、C、FORTRAN 以及 Ada 等语言的优点，克服了原来 BASIC 结构化程度差和运算速度慢的缺点，是一种功能强、易学习的程序设计语言。从难度上说，它与 BASICA 差不多，而功能则大为扩充，其完善程度可以和 PASCAL、FORTRAN、C 相媲美。

Quick BASIC 使用下拉式菜单，对话方式操作，解释与编译并行。它把解释型 BASIC 及 LINK、DEBUG 等软件加以合并，使用户可以在同一环境下对源程序进行处理，把程序的编辑、编译、链接、调试等步骤集成为一个单一的过程，把解释型 BASIC 的方便性和编译 BASIC 的执行速度有机地结合在一起，并能快速生成可以直接在 DOS 下运行的 .EXE 文件。

Quick BASIC 能够十分有效地组织程序，具有很强的库管理功能。使用 Quick 库，可以把用户编写的过程和例行程序作为 Quick BASIC 的虚拟扩充部分。与原来的 BASIC 版本不同，Quick BASIC 可以使用全部内存，并支持多个模块同时驻留内存，通过十分简单的操作即可将多个模块编译、链接，生成一个可执行文件。

此外，在数据类型、流程控制、绘图、模块化结构及文件输入输出等方面，Quick BASIC 都提供了令人满意的功能。它可以在较大范围内选择数据类型和数据结构，除标准 BASIC 的数据类型外，还增加了长整型和定长字符串类型，并接受标准 IEEE 格式的实型数据。在复合数据类型方面，Quick BASIC 允许定义双精度数组，数组的下标可以为任意整数（包括负数），而且支持动态数组。同时，Quick BASIC 还设立了 TYPE...END TYPE 语句，它类似于 C 语言中的“结构”和 PASCAL 语言中的“记录”。利用这个语句，用户可以定义自己的数据类型，存放不同的数值和字符串的任意组合，从而使随机存取文件的输入输出得以简化。

在流程控制方面，Quick BASIC 提供了 SELECT CASE 之类的开关语句和 DO...LOOP 循环，以及块结构的条件语句，使得程序逻辑结构清晰，且易于修改。同时，Quick BASIC 具有独立的子程序块，可以把一个程序分解成逻辑上相对独立的部分，从而能方便地实现结构化程序设计，其子程序（过程）具有 PASCAL 语言的特征。

在 Quick BASIC 中可以方便地调用其它语言（如 C、PASCAL、宏汇编等）的子例程，并可执行 DOS 中断和功能调用。Quick BASIC 代码的执行速度大大高于编译 BASIC，接近

于 FORTRAN, 而编译、链接比 FORTRAN 要方便得多。

Quick BASIC 具有较强的绘图功能, 支持多种适配器。

总的来看, Quick BASIC 具有其它高级语言的许多优点, 但又不像其它高级语言有诸多限制, 使用起来和普通 BASIC 一样方便, 十分适合于开发各种应用软件。

《Quick BASIC 程序设计》全面系统地介绍了 Quick BASIC 的程序设计功能。本书于 1991 年发行初版, 受到广大读者欢迎, 先后多次重印。几年来, 微机的机型不断变化, 配置越来越高, 为了适应计算机发展的需要, 笔者对本书进行了修订。本次修订对初版的内容作了较大的增删和调整, 改变了原来的章节结构, 使各部分内容相对集中, 更利于读者使用。

全书共 14 章。第一到四章介绍了 Quick BASIC 的特点、安装、程序开发环境、源程序的建立、编译、运行及库管理功能; 第五到十章介绍了 Quick BASIC 的基本程序设计操作, 包括数据类型、运算符、表达式、赋值、输入输出、流程控制、数组、过程和字符处理; 第十一、十二章分别介绍了 Quick BASIC 的文件处理和图形功能; 第十三章介绍了 Quick BASIC 的程序调试、错误处理及陷阱技术; 第十四章介绍了 Quick BASIC 的 DOS 环境操作及如何在 Quick BASIC 中实现 DOS 中断和功能调用。

Quick BASIC 有多种版本, 包括 2.0、3.0、4.0、4.5 和编译 6.0, 在 Quick BASIC 的基础上, Microsoft 公司随 DOS 5.0 推出了 Quick BASIC 的简化版本, 即 QBASIC。本书以 4.0 版本为基础进行介绍, 所有程序都已在 4.0 版本中运行通过, 可在 4.5 及编译 6.0 版本中运行, 大多数程序可在 QBASIC 中运行。

在修订本书的过程中, 笔者注意了内容的科学性、实用性、通俗性和良好程序设计风格的训练, 希望它能在推广 Quick BASIC 方面起到一些作用。但是, 笔者水平有限, 而且计算机科学技术发展迅速, 相信本书难免会有不少缺点甚至错误, 恳请专家、读者不吝赐教, 提出宝贵意见。

作者

1997 年

目 录

前言	I
第一章 概述	(1)
1.1 早期 BASIC 语言与 Quick BASIC	(1)
1.2 Quick BASIC 的特点	(2)
1.3 Quick BASIC 的版本	(4)
1.3.1 新的特性	(4)
1.3.2 编程环境	(5)
1.3.3 语言功能	(6)
1.4 Quick BASIC 的安装	(7)
1.4.1 Quick BASIC 4.0 系统文件	(7)
1.4.2 Quick BASIC 4.0 的安装	(8)
第二章 Quick BASIC 4.0 程序设计环境	(11)
2.1 启动与退出 Quick BASIC	(11)
2.1.1 Quick BASIC 的启动	(11)
2.1.2 Quick BASIC 工作界面	(12)
2.1.3 退出 Quick BASIC	(14)
2.2 菜单与对话框	(15)
2.2.1 菜单	(15)
2.2.2 对话框	(17)
2.3 窗口	(20)
2.3.1 观察窗口与直接窗口	(20)
2.3.2 窗口的控制	(21)
2.4 文本编辑	(22)
2.4.1 文本输入和简单编辑操作	(22)
2.4.2 选择文本	(24)
2.4.3 删除与插入	(25)
2.4.4 移动与拷贝	(27)
2.4.5 查找与替换	(27)
2.4.6 缩进	(30)
2.4.7 其它编辑操作	(31)
2.5 Quick BASIC 源文件管理	(32)
2.5.1 程序文件	(33)
2.5.2 模块文件	(35)
2.5.3 蕴含文件	(38)
2.5.4 文档文件	(40)

2.6 文件的合并与打印	(40)
2.6.1 文件合并	(40)
2.6.2 文件打印	(41)
第三章 Quick BASIC 程序的建立和执行	(43)
3.1 程序的建立和运行	(43)
3.1.1 建立主模块和输入程序语句	(43)
3.1.2 运行 Quick BASIC 程序	(45)
3.2 在 Quick BASIC 环境下建立可执行文件	(45)
3.2.1 建立可执行文件的一般过程	(45)
3.2.2 建立依赖于运行时模块的可执行程序	(46)
3.2.3 建立独立程序	(47)
3.2.4 建立多模块程序	(47)
3.3 过程的使用	(48)
3.3.1 过程的建立	(48)
3.3.2 与过程有关的操作	(49)
3.4 直接窗口	(50)
3.4.1 直接窗口的特点	(50)
3.4.2 直接窗口的操作	(51)
3.5 在 DOS 下建立可执行程序	(52)
3.5.1 在 DOS 下生成可执行文件的必要性	(52)
3.5.2 在 DOS 下编译、链接的一般过程	(53)
3.6 编译命令 BC	(53)
3.6.1 BC 命令的执行	(53)
3.6.2 文件名约定	(54)
3.6.3 BC 命令选择项	(55)
3.7 链接命令 LINK	(56)
3.7.1 LINK 命令的执行	(56)
3.7.2 LINK 命令的选择项	(59)
第四章 Quick BASIC 库管理	(64)
4.1 Quick BASIC 中的库	(64)
4.2 建立 Quick 库	(65)
4.2.1 Quick 库的建立	(65)
4.2.2 在 Quick 库中使用其它语言的例行程序	(67)
4.3 使用 Quick 库	(67)
4.3.1 Quick 库的装入和查看	(67)
4.3.2 几点说明	(69)
4.4 独立库	(70)
4.4.1 LIB 命令的执行	(70)
4.4.2 命令符号和库页长度	(71)
第五章 Quick BASIC 基本词法	(74)

5.1 Quick BASIC 程序的构成	(74)
5.1.1 程序行	(74)
5.1.2 字符集	(76)
5.2 数据类型	(76)
5.2.1 基本数据类型	(76)
5.2.2 用户定义的数据类型	(78)
5.3 常量和变量	(79)
5.3.1 常量	(79)
5.3.2 变量	(81)
5.4 常用内部函数	(84)
5.5 运算符与表达式	(86)
5.5.1 算术运算符	(86)
5.5.2 关系运算符与逻辑运算符	(88)
5.5.3 表达式的执行顺序	(90)
5.5.4 混合运算与类型转换	(90)
习题	(91)
第六章 赋值与输入输出	(93)
6.1 赋值语句	(93)
6.2 键盘输入语句	(94)
6.2.1 INPUT 语句	(94)
6.2.2 行输入语句 (LINE INPUT)	(96)
6.3 读数据语句	(97)
6.3.1 READ-DATA 语句	(97)
6.3.2 恢复读数据语句 (RESTORE)	(98)
6.4 屏幕输出	(99)
6.4.1 PRINT 语句的基本功能	(99)
6.4.2 PRINT 语句的输出格式	(100)
6.4.3 WRITE 语句	(103)
6.4.4 PRINT USING 语句	(104)
6.4.5 LPRINT 和 LPRINT USING 语句	(106)
6.5 特殊输入操作	(107)
6.5.1 INKEY\$ 函数	(107)
6.5.2 KEY 语句	(108)
6.6 文本光标控制	(109)
6.6.1 光标定位	(110)
6.6.2 光标大小和位置	(112)
6.7 注释、暂停与程序结束	(113)
习题	(114)
第七章 流程控制	(116)
7.1 条件语句	(116)

7.1.1 单行结构条件语句	(116)
7.1.2 块结构条件语句	(117)
7.2 情况语句	(120)
7.3 循环语句	(124)
7.3.1 计数循环 (FOR-NEXT)	(124)
7.3.2 当循环 (WHILE-WEND)	(129)
7.3.3 DO 循环 (DO-LOOP)	(131)
7.3.4 多重循环与退出循环	(135)
7.4 GOTO 型控制	(138)
7.4.1 GOTO 语句	(138)
7.4.2 ON-GOTO 语句	(139)
7.5 程序举例	(140)
习题	(148)
第八章 数组	(150)
8.1 数组的概念	(150)
8.1.1 数组的定义	(150)
8.1.2 静态数组与动态数组	(152)
8.2 数组的重定义	(153)
8.2.1 ERASE 语句	(153)
8.2.2 REDIM 语句	(154)
8.3 数组的基本操作	(155)
8.4 程序举例	(159)
习题	(167)
第九章 过程	(169)
9.1 转子与返回	(169)
9.2 用户自定义函数	(173)
9.3 过程	(174)
9.3.1 SUB 过程与 GOSUB 子程序	(175)
9.3.2 FUNCTION 过程与 DEF FN 函数	(176)
9.4 过程的定义	(178)
9.4.1 SUB 过程的定义	(178)
9.4.2 FUNCTION 过程的定义	(180)
9.5 过程的调用	(181)
9.5.1 调用 FUNCTION 过程	(181)
9.5.2 调用 SUB 过程	(184)
9.6 参数传送	(187)
9.6.1 形参与实参	(187)
9.6.2 参数的传送	(188)
9.6.3 引用与传值	(190)
9.6.4 数组参数的传送	(193)

9.7 过程说明	(195)
9.7.1 DECLARE 语句	(195)
9.7.2 使用蕴含文件说明	(196)
9.8 变量的作用域	(198)
9.8.1 局部变量	(198)
9.8.2 全局变量	(199)
9.8.3 模块与过程共享变量	(203)
9.9 局部内存分配	(206)
9.10 递归	(209)
9.10.1 过程的递归调用	(209)
9.10.2 栈区空间设置	(211)
9.11 程序的链接	(212)
9.11.1 CHAIN 语句	(212)
9.11.2 RUN 语句	(215)
9.12 程序举例	(216)
习题	(225)
第十章 字符串	(226)
10.1 字符串的定义与分类	(226)
10.1.1 字符串的定义	(226)
10.1.2 字符串的分类	(227)
10.2 字符串基本操作	(227)
10.2.1 字符串连接	(227)
10.2.2 字符串比较	(228)
10.3 字符串函数	(229)
10.3.1 子字符串函数	(229)
10.3.2 转换函数	(233)
10.3.3 其它字符串函数	(235)
10.4 字符串数组	(239)
10.5 程序举例	(240)
习题	(243)
第十一章 数据文件	(244)
11.1 文件说明与文件结构	(244)
11.1.1 文件说明	(244)
11.1.2 文件结构和分类	(246)
11.2 文件的打开与关闭	(247)
11.2.1 文件的打开	(248)
11.2.2 文件的关闭	(250)
11.3 文件操作函数和语句	(251)
11.3.1 文件操作函数	(251)
11.3.2 LOCK 和 UNLOCK 语句	(253)

11.4 顺序文件	(254)
11.4.1 顺序文件的写操作	(254)
11.4.2 顺序文件的读操作	(258)
11.5 随机文件	(265)
11.5.1 随机文件的读写操作	(265)
11.5.2 随机文件中记录的增加与删除	(275)
11.6 二进制文件	(277)
11.6.1 二进制存取与随机存取	(277)
11.6.2 文件指针	(277)
11.7 设备文件	(280)
习题	(282)
第十二章 图形程序设计	(284)
12.1 图形显示与视频设备	(284)
12.2 字符坐标系与点坐标系	(285)
12.2.1 文本方式与字符坐标系	(285)
12.2.2 图形方式与点坐标系	(285)
12.3 屏幕控制	(286)
12.3.1 清屏与改变行宽	(287)
12.3.2 SCREEN 函数与 SCREEN 语句	(288)
12.4 点、直线和矩形	(293)
12.4.1 画点	(293)
12.4.2 画直线	(294)
12.4.3 画矩形	(295)
12.5 连续画线	(297)
12.6 圆、椭圆和弧	(299)
12.6.1 画圆	(300)
12.6.2 画椭圆	(301)
12.6.3 画弧	(302)
12.7 颜色	(304)
12.7.1 屏幕颜色设置 (COLOR 语句)	(304)
12.7.2 测试屏幕上某点的颜色 (POINT 函数)	(306)
12.8 填充	(307)
12.8.1 PAINT 语句	(307)
12.8.2 填充颜色	(308)
12.8.3 填充图案	(309)
12.9 视见区与窗口	(315)
12.9.1 视见区语句 (VIEW)	(315)
12.9.2 建立文本视见区 (VIEW PRINT 语句)	(316)
12.9.3 窗口语句 (WINDOW)	(318)
12.10 简单动画技术	(322)

12. 10. 1 字符动画	(322)
12. 10. 2 图形方式动画	(324)
12. 10. 3 快速动画	(325)
12. 10. 4 分页动画	(331)
12. 11 程序举例	(332)
第十三章 程序调试与陷阱技术	(337)
13. 1 Quick BASIC 的程序调试	(337)
13. 1. 1 Quick BASIC 的错误类型	(337)
13. 1. 2 Quick BASIC 的程序调试手段	(338)
13. 1. 3 调试术语与概念	(339)
13. 2 Debug 菜单调试命令	(340)
13. 2. 1 程序跟踪	(340)
13. 2. 2 断点、观察点和观察表达式	(341)
13. 2. 3 控制执行	(343)
13. 3 其它调试功能	(344)
13. 3. 1 Calls 菜单	(345)
13. 3. 2 与 Code View 调试程序的兼容性	(345)
13. 4 错误处理	(346)
13. 4. 1 错误处理子程序	(346)
13. 4. 2 错误的模拟与识别	(350)
13. 5 事件陷阱	(353)
13. 5. 1 设置事件陷阱	(353)
13. 5. 2 事件处理	(354)
13. 5. 3 功能键陷阱	(355)
13. 5. 4 用户定义键陷阱	(356)
13. 5. 5 时钟陷阱	(359)
13. 6 过程及多模块中的陷阱	(360)
13. 7 用 BC 命令编译含有错误陷阱和事件陷阱的程序	(362)
第十四章 DOS 环境与中断调用	(363)
14. 1 Quick BASIC 的 DOS 环境功能	(363)
14. 1. 1 目录操作	(363)
14. 1. 2 文件命令语句	(364)
14. 1. 3 DOS 环境操作	(366)
14. 2 SHELL 语句	(367)
14. 3 DOS 中断与功能调用	(371)
14. 3. 1 中断概述	(371)
14. 3. 2 与中断调用有关的语句和函数	(373)
14. 3. 3 Quick BASIC 中的 DOS 功能调用	(374)
附录 1 ASCII 字符代码	(383)
附录 2 Quick BASIC 保留字	(386)

第一章 概述

Quick BASIC 是 Microsoft 公司于 1987 年推出的结构化 BASIC 语言，它与以前的 BASIC 语言（GW BASIC 和 BASICA）高度兼容，但功能更强，使用更方便。在这一章中，我们将介绍 BASIC 语言的发展概况、Quick BASIC 的特点及安装方法，为进一步学习 Quick BASIC 程序设计做好准备。

1.1 早期 BASIC 语言与 Quick BASIC

Quick BASIC 是在原来的 BASIC 语言的基础上研制而成的，它具有 BASIC 语言简单而不贫乏的优点，同时增加了结构化程序设计语言的特点。

60 年代初，美国 Dartmouth 学院的两位学者，G.Kemeny 和 Thomas E.Kurty 发明了一种称为“BASIC”的语言，其含义为“初学者通用的符号指令代码”。BASIC 不是世界上第一个计算机高级语言，但它的诞生称得上是计算机语言发展史上的一件大事，对计算机的推广应用起到了重要的作用。

70 年代初，美国英特尔（Intel）公司研制出世界上第一个微处理器，开辟了微型计算机时代。70 年代中叶，比尔·盖茨（Bill Gates）为微型机配置了 BASIC 语言，并在此基础上成立了世界上第一个微型计算机软件公司，即 Microsoft 公司。此后，BASIC 语言被配置在所有新推出的微型计算机上，而各个微型计算机公司也竞相研制具有自己特色的 BASIC 语言。在此期间，较有代表性的 BASIC 语言是 GW BASIC 和 Microsoft BASIC（即 IBM PC 上的 BASICA），这两种 BASIC 语言都是解释型的。

解释型 BASIC 简单易学，使用方便，功能也比较强。但是，这种 BASIC 解释程序在执行时需要通过译码，执行速度较慢，而且所编写的程序只能在特定的环境中运行（不能直接在 DOS 下运行）。

为了提高执行速度，软件工作者研制出了 BASIC 语言的编译软件，其中较为流行的是 BASCOM。用该软件可以对 BASIC 源程序进行编译、链接，生成可执行文件（.EXE 文件），其运行速度有明显提高。但是，在这种软件中，某些 BASIC 语句无法使用或受到一定的限制，而且软件本身的使用也很不方便，需要通过繁琐的步骤编译一个源程序（即使是很小的程序）。

解释型 BASIC 编程、调试比较方便，但执行速度慢；用编译 BASIC 可以生成可执行文件，速度较快，可直接在 DOS 下运行，但需要经过编辑、编译、链接、修改和重新编辑、编译、链接等繁琐步骤。

Quick BASIC 把解释型 BASIC 的方便性和编译 BASIC 的执行速度有机地结合在一起，既可以在同一个环境下完成编辑、调试和运行，又提高了执行速度，同时对 BASIC 语言的功能进行了较大的扩充。此外，在 Quick BASIC 环境下，所有的操作都在内存中完成，而不需要对盘上的文件进行读写，从而使应用程序的研制开发更快、更容易。Quick BASIC 把 BASICA、BASCOM、LINK 以及 DEBUG 等软件合并在一起，在同一环境下对用户的源程

序自动进行处理,包括编辑、语法检查、编译、链接、执行、调试等,大大节省了用户编写和调试程序的时间,提高了软件开发效率。

Quick BASIC 与 BASIC 基本兼容,大多数用 BASICA 开发的程序不用修改即可在 Quick BASIC 下运行或编译、链接;有些程序则可能需要进行简单修改,主要是关于以下几个方面:

1. 源文件格式

Quick BASIC 的源文件是 ASCII 格式或 Quick BASIC 格式,因此,对于用 BASICA 建立的程序,如果想在 Quick BASIC 环境下运行,则必须以 ASCII 码文件存盘(使用“A”参数)。

2. 禁止使用的语句、命令和函数

BASICA 中有些语句、命令和函数不能在 Quick BASIC 中使用。这些语句、命令和函数涉及源文件的编译操作、程序执行接口、磁带设备的操作及与 Quick BASIC 环境所提供的支持相重复的操作,包括:

AUTO	LIST	NEW
CONT	LLIST	RENUM
DEFUSR	LOAD	SAVE
DELETE	MERGE	USR
EDIT	MOTOR	

3. 需要修改的语句

如果 BASICA 程序中有下列语句,则在用 Quick BASIC 对其进行编译之前必须对源程序进行修改:

BLOAD/BSAVE	Quick BASIC 中存储位置不同
CALL 名字	这里的“名字”是被调用的 FUNCTION 或 SUB 过程的名字
CHAIN	Quick BASIC 不支持 ALL、MERGE、DELETE 或行号选择
COMMON	COMMON 语句必须出现在所有可执行语句之前
DEF TYPE	DEF TYPE 语句应出现在源文件的首部
DIM	所有声明静态数组的语句必须出现在程序的首部
DRAW、PLAY	Quick BASIC 要求嵌套变量要使用 VARPTR \$ 函数
RESUME	如果在某行中出现错误,Quick BASIC 试图从行开头重新执行程序
RUN	在 BASICA 中运行时, RUN 的目标是 .BAS 文件;而在 Quick BASIC 环境下,格式为: RUN 行号 文件说明 ,从指定行启动程序运行,或执行指定的程序,而且所执行的程序也可以是 .EXE 文件

1.2 Quick BASIC 的特点

与 BASICA 等早期的 BASIC 版本相比,Quick BASIC 的改进是革命性的,其主要特色表现在两个方面:一是编程环境,二是对语言功能的扩充。

1. 编程环境

在编程环境方面, Quick BASIC 有以下特点:

(1) **交互编辑:** Quick BASIC 环境的核心部分是一个性能优越的编辑程序。在这个环境中, 输入一程序后, 立即就能对其进行语法检查, 如果没有语法错误, 则该程序即被翻译成可执行代码; 否则给出适当的错误描述信息。该编辑程序还能自动地将 Quick BASIC 的关键字转换为大写, 校正遗漏和错误, 并可显示程序的逻辑结构。

(2) **交互式调试:** Quick BASIC 把编辑、运行和调试有机地结合起来, 不必来回切换。在调试程序时, 可以将一个正在运行的程序在任何地方暂停, 对它进行编辑、修改, 然后从暂停处继续执行。用户可利用屏幕下部的“直接窗口”测试程序中的语句, 甚至可以修改正在运行的程序中的数据, 以观察修改后的效果, 而不必重新启动程序。

(3) **快速生成可执行文件:** 程序调试通过后, 如果需要, 则只要进行很简单的操作, 就能生成可直接在 DOS 下运行的文件 (.EXE 文件)。

(4) **多个模块同驻内存:** 4.0 以上版本的 Quick BASIC 支持多个模块同时驻留内存, 如果需要将内存中的全部模块编译、链接生成一个可执行文件, 则只用一步操作即可实现。

(5) **库的生成与管理:** Quick 库可以将用户编写的过程和例行程序变为 BASIC 语言的虚拟扩充部分。只要执行一个菜单命令即可构造一个 Quick 库; 而且每当建立或修改一个 Quick 库时, Quick BASIC 将自动地建立或更新与之相应的独立库。

(6) **灵活的联机帮助功能:** Quick BASIC 提供了灵活的联机帮助功能, 它既为每个语句提供语法检查, 同时也为菜单命令、热键或关键字提供帮助信息。

(7) **完全图形支持:** Quick BASIC 支持 CGA、EGA、VGA、MCGA 等图形适配卡及 PS/2 的图形方式和单色方式, 同时支持增强图形卡。

2. 语言功能

在语言功能方面, Quick BASIC 兼容 BASICA, 同时增加了一些新的功能, 主要有:

(1) Quick BASIC 把微机内存看作是一张空白程序纸, 在这张纸上的每一行可含有一个或多个 Quick BASIC 语句 (各语句之间用冒号隔开), 程序按物理行的顺序执行, 因而源程序可以不写行号。当在 Quick BASIC 中执行原来用 BASICA 编写的程序时, 其行号作为标号来处理。标号既可用数字表示, 也可以用字符串表示。例如:

```
INPUT I
IF I > 0 THEN GOTO SSS
PRINT "I <= 0": GOTO WW
SSS:
PRINT "I > 0"
WW:
END
```

上面程序中的“SSS:”和“WW:”是标号。

(2) Quick BASIC 程序由模块组成, 每个模块可以独立保存。这样就可以把一个程序分解为逻辑上不同的若干个部分, 从而实现结构化程序设计。此外, 在 Quick BASIC 中还可以编写与 PASCAL、C 语言中类似的函数和子程序。

(3) 增加了 TYPE...END TYPE 语句, 可以用来定义用户自己的数据类型。它类似于 C 语言中的“结构”和 PASCAL、Ada 语言中的“记录”。利用这个语句, 可以把若干个不同类型的变量组成一种用户定义的类型, 从而使随机文件的输入输出得以简化。

(4) 具有 *SELECT CASE* 之类的开关语句和十分灵活的 *DO...LOOP* 循环, 并有与 C、Ada 等语言中功能相同的块结构的条件语句。这些控制结构使得 Quick BASIC 程序的逻辑结构清晰, 且易于修改。

(5) Quick BASIC 增加了新的数据类型, 其中包括 32 位的长整数类型。此外, Quick BASIC 支持 IEEE 格式的实型数据, 使所表示的数据的范围更大, 精度更高。

(6) 对磁盘文件能进行二进制存取, 因而可以读取和修改非 ASCII 磁盘文件。

1.3 Quick BASIC 的版本

Quick BASIC 问世后, 先后推出了几个版本, 包括 2.0、3.0、4.0 和 4.5, 此外还有 Quick BASIC 编译 6.0 (其编程环境仍为 4.0)。本书介绍的是目前较为流行的版本, 即 Quick BASIC 4.0, 所有的程序都可以在 Quick BASIC 4.5 及 Quick BASIC 编译 6.0 中运行。

Quick BASIC 4.0 与 4.5 及编译 6.0 相比, 差别并不大, 但与 2.0、3.0 却有较大的不同, 下面介绍 4.0 与 2.0、3.0 版的主要区别。

1.3.1 新的特性

与 2.0、3.0 相比, Quick BASIC 4.0 增加了一些新特性, 使得语言功能大为增强 (见表 1.1)。

表 1.1 Quick BASIC 4.0 新增特性

特 性	2.0	3.0	4.0
用户定义类型	无	无	有
IEEE 格式与数学协处理器支持	无	有	有
联机帮助	无	无	有
长整数 (32 位)	无	无	有
定长字符串	无	无	有
语法检查	无	无	有
二进制文件 I/O	无	无	有
FUNCTION 过程	无	无	有
Code View 调试程序支持	无	无	有
与其它语言的兼容性	无	无	有
多模块驻留内存	无	无	有
Prokey、Sidekick、Superkey 兼容性	无	有	有
插入/重写模式	无	有	有
Wordstar 键盘命令	无	无	有
递归	无	无	有
分别编译期间错误列表	无	有	有
分别编译中汇编语言	无	有	有

1.3.2 编程环境

Quick BASIC 4.0 比早期版本提供的命令选择更灵活, 附加的窗口选择项更多, 菜单命令也更丰富。例如, 在 4.0 版中, 可以选择任何菜单, 按下 *Alt* 键再按一个指定的键, 就能打开一个菜单, 接着再按一个字母键, 就可执行一个菜单命令。此外, 在 4.0 版中可以使用两个工作窗口。

4.0 版新增加了 3 个菜单项, 即 **Debug** (调试)、**Calls** (调用) 和 **Help** (帮助), 保留了原有的 4 个菜单, 即 **File** (文件)、**Edit** (编辑)、**View** (观察) 和 **Run** (运行), 但增加了新的命令, 见表 1.2。

表 1.2 Quick BASIC 4.0 新增菜单和命令

菜 单	命 令	功 能
File	New Program	与 2.0 中的 New 相同
	Open Program	与 2.0 中的 Load 相同
	Merge	合并文件
	Save As	命名并保存文件
	Save All	保存当前装入的全部文件
	Create File	建立新的模块、蕴含文件或文档
	Load File	装入模块、蕴含文件或文档
	Unload File	清除内存中的模块、蕴含文件或文档
	DOS Shell	与 2.0 中的 Shell 相同
	Exit	与 2.0 中的 Quit 相同
Edit	Undo	热键为 Alt + BKSP
	Cut	热键为 Shift + Del
	Copy	热键为 Ctrl + Ins
	Paste	热键为 Shift + Ins
	Clear	删除而不拷贝剪贴板
	New SUB	建立新的 SUB 过程
	New FUNCTION	建立新的 FUNCTION 过程
	Syntax Checking	语法检查
View	SUBs	查看当前装入的模块与过程
	Next SUB	查看当前模块中装入的过程
	Split	打开或关闭窗口
	Next Statement	将光标移到下面要执行的语句
	Output Screen	查看程序输出
	Include File	编辑蕴含文件
	Options	设置屏幕显示
Run	Start	同 2.0 版
	Restart	重置各变量值
	Continue	继续执行
	Modify Command \$	设置由 COMMAND \$ 读出的字符串
	Make EXE File	建立可执行文件
	Make Library	建立 Quick 库
	Set Main Module	指定新的主模块