

QBasic 语言程序 设计

金怀群 编著

DBA

华南理工大学出版社

QBasic 语言程序设计

金怀群 编著

华南理工大学出版社
· 广州 ·

内 容 简 介

本书由浅入深、循序渐进、全面系统地介绍了 QBasic 语言的基本思想、概念、方法、功能和环境设施,主要包括程序设计与 QBasic 语言、数据类型与运算、标准输入输出、控制程序流程、过程、数组、字符处理、文件、DOS 和程序管理、陷阱技术、图形、声音和音乐以及程序调试等 14 章和 5 个附录,每章配有习题可作为普通高校非计算机专业的学生,或中专学校作为学习计算机课程的教材,也可作具有中学文化程度者自学之用。

图书在版编目(CIP)数据

QBasic 语言程序设计/金怀群 编著. —广州:华南理工大学出版社,1996. 9
ISBN 7-5623-1063-7

I. Q…

I. 金…

Ⅱ. QBasic 语言—程序设计—图形—声音

N. TP. 3

华南理工大学出版社出版发行

(广州·五山 邮编 510641)

*

中山大学印刷厂印装

开本:787×1092 1/16 印张:19.125 字数 442 千

1996 年 9 月第 1 版 1996 年 9 月第 1 次印刷

印数 1—5000 册

定价:25.00 元

前 言

在全球加速向信息社会发展的今天,计算机知识已经成为大学生知识结构中的重要组成部分,计算机基础教育已显得日益重要和紧迫。全国高等院校计算机基础教育研究会制订了高校非计算机专业的计算机基础教育计划,目的是培养学生的计算机意识,知道计算机能做什么,不能做什么,同时使学生具有利用计算机解决实际问题的愿望和能力。该计划实际上已在各高校中不同程度地实施,其中第二层次是学习计算机程序设计语言,要求学生能够用一种高级语言(例如 BASIC, FORTRAN, PASCAL, C 等语言)或一种数据库语言熟练地编写程序。

高校非计算机专业进行计算机基础教育是要培养各行各业的计算机应用人才,而不是计算机专门人才,选择教学用的程序设计语言应该满足“学以致用”的原则,即学生在学习之后能够用这种语言编写各种应用程序。在非计算机专业开设程序设计语言课程的目的是,让学生学会该语言本身,掌握其全部内容,使学生能够辨别该语言的主要优缺点,并熟练地编写程序。因而要求一般的机器上能配有这种语言系统,也就是应当考虑这种语言的软件和硬件要求,适合一般学校的实验条件;所教的语言应该成为学生以后学习其他程序设计语言的基础。显然,学生现在学习什么语言并不一定以后就要用什么语言,因为语言只是工具、是为应用服务的,而且程序设计语言本身的发展速度很快,今天是很好的语言,几年后拿新的标准和技术进行衡量,就可能不是一个好的语言了。因此,教第一个程序设计语言应该着重于这种语言的基本思想、概念、设施,培养学生的程序设计基本功,为今后进一步学习软件技术基础知识或软件开发技术,提高程序设计能力打下基础。

BASIC 是 Beginner's All-purpose Symbolic Instruction Code(初学者通用符号指令代码)一词的缩写。1962 年,美国 Dartmouth 学院数学系 Thoms Kurtz 教授向当时的系主任 John Kemeny 教授建议:让学生在校 4 年学习使用计算机不断线。两位教授结合当时的计算机技术,吸收 FORTRAN 和 ALGOL 的精髓,专为初学者设计一种能够与计算机进行“交谈”的简单会话式程序设计语言——BASIC。该院学生终于在 1965 年 5 月 1 日开始学习使用 BASIC 语言。

1975 年, Bill Gates 和 Paul Allen 为 Altair 微机设计了一个只占 4KB 的 BASIC 解释程序,结果得到用户的好评。随后,“蓝色巨人”IBM 公司也在它的第一代 PC 机中配备了 BASIC 语言解释器。30 年来, BASIC 语言从一种功能简单、灵活易用的教学工具经过不断地推陈出新,功能越变越强,在软件世界中的地位越变越重要。Bill Gates 在 1989 年 10 月庆祝 BASIC 语言 25 岁生日时说:“我至今仍是 BASIC 的狂热爱好者。它能处理其他语言迫使你不得不去应付的许多工作,从而解放出你的编程创造力。我已经向程序员们挑战,为解决任一问题他们可以选用任何工具编程,我打赌,我用 Quick BASIC 会更快地编出同样的程序。”他还说:“我相信,就 BASIC 而论,这种语言正在欣欣向荣地成长,一旦 BASIC 变成了一种通用的宏语言(依我看它正向此方向发展),那么,它将比其他任何语言都长寿。”其实,1985 年之后, BASIC 已经发展成为一种功能齐全的集编辑、调试、编译、连接于一身的一体化集成软件开发环境。它具有时钟、日期、数学运算、字符串处理、绘图、动画、音乐、读写内存、文件处理、菜单、多窗口及多模块调用等等功能,现在与任何一种程序设计语言相比毫不逊色,应用广泛。

BASIC 几乎是微机的通用语言,它简单易学,很受欢迎,非常普及,就像现实世界中的英语一样。大多数微机都配有 BASIC;谭浩强先生的《BASIC 语言》一书累计发行 1000 万册,创世界纪录,就是一个很好的证明。《计算机世界》周报(1995 年 3 月 22 日 197 版)上沈林兴的文章“1994 年软件

资格和水平考试分析”也称：国内初级程序员首选 BASIC 的比例高达 71%，雄踞榜首，与次选语种间相差 51 个百分点，可见 BASIC 还是广大程序员喜爱的、用得很多的、他们自认为掌握得最好的一种程序设计语言。1981 年美国 IDG 公司对 995 位管理信息系统经理作过一次使用程序设计语言的调查，使用 BASIC 语言仅占 5.4%；六年后，美国波士顿计算机学会对其成员作了类似的调查，BASIC 的使用者上升到 80%~90%。到 1994 年下半年为止，在美国尚存一举足轻重的 Quick BASIC 大用户群。1994 年出版的《Moving from QBasic to C》和《Leaping from BASIC to C++》的作者 Grey Perry 和 Robert J. Traister 也一致认为：BASIC 语言是最好的初学者语言；他们说：“你的 BASIC 程序设计知识在你学习 C++ 和其他任何一门计算机程序设计语言时，会给你无法估量的帮助。”

现在，一般可以把 BASIC 语言家族划分为下列三代：

80 年代中期以前为第一代，BASIC 是一种解释型的带行号的非结构化语言。例如，BASICA 或 GWBASIC。

1985 年，Kemeny 和 Kurtz 对 BASIC 动“大手术”，装上编译器，使之成为既有解释器又有编译器的“两栖型”语言，并进行严格的结构化改造，抛弃行号，废除 GOTO 语句，从根本上改变了 BASIC 的面貌，推出了 True BASIC。而 Quick BASIC 通过对 Quick 库和独立库的调用，已经变成了多种语言混合编程的理想环境。这就是第二代，称为结构化的编译 BASIC。

1990 年以来，Windows 和 OOP 技术飞速发展，BASIC 也日新月异，形成了第三代 BASIC，即所谓 Windows BASIC，如 Visual BASIC 等。

目前，国内各高校的 BASIC 语言教科书或 BASIC 语言教学大多数还是采用早期的非结构化的带行号的解释 BASIC，第一代 BASIC 版本不完全是一个好的教学语言，有“玩具”语言之讥，现在看来十分陈旧，严重落后于 BASIC 语言本身的发展，急需更新换代，以便较好地适应信息社会的发展。当然，单纯地“喜新厌旧”，追逐新潮也是偏见，而且不切实际，因为国内各高校进行计算机基础教育的条件差别很大，至少在设备、教材、师资及其知识更新等方面，就不能保证大家在短期内都进入 Windows BASIC 时代，实际上，我们的程序设计语言教学不得不适当地滞后于程序设计语言本身的发展。1993 年，我们在承担国家教委师范司委托的世界银行贷款“师范教育发展”项目教改课题之后，经过详细的调查研究和分析，决定推荐 QBasic 语言作为高校非计算机专业学生学习计算机程序设计的入门课。今年，全国计算机软件专业技术资格和水平考试新版考试大纲也把 QBasic 语言作为初级程序员的一种考试语种。

QBasic 是从 Quick BASIC 语言演变而来的，它简明易学，新颖实用，软硬件要求也不高，且垂手可得（有 MS-DOS 5.0 以上，也就有 QBasic），易于普及。它与老的 BASIC 兼容，但其功能却大大超过了原来的 BASIC 语言，而且具有当前一般程序设计语言所应有的特征与设施，编程开发环境也十分良好，功能强大，能很好体现结构化程序设计的良好风格。对今后进一步学习像 Visual BASIC 或 C++ 等其他程序设计语言有极大的帮助。

本教材曾在我校数学系和化生系试用，进展顺利，效果良好。根据我们的经验，本课程的教学与实验需要约 80 小时。

中山大学黄友谦教授和韩山师范学院林立聪教授对本书的编写和出版给予了热情的鼓励和指导，中山大学吴湘辉老师对本书的出版也给予了很大的帮助，我在此对他们表示衷心的感谢。

由于时间匆促，编者水平所限，欠妥和不足之处，敬请行家和读者指正。

作 者

1996 年 7 月 30 日

目 录

前言

第一章 程序设计与 QBasic 语言	(1)
1.1 程序设计语言	(1)
1.2 QBasic 语言简介	(2)
1.3 QBasic 的启动与退出	(2)
1.3.1 启动 QBasic	(2)
1.3.2 QBasic 屏幕	(4)
1.3.3 退出 QBasic	(6)
1.4 QBasic 程序的建立与运行	(7)
1.4.1 建立源程序	(7)
1.4.2 程序的储存	(7)
1.4.3 运行程序	(7)
1.4.4 QBasic 程序的语法	(9)
1.5 QBasic 程序的编辑	(10)
1.5.1 自动编辑	(10)
1.5.2 Edit 菜单	(12)
1.6 获取 QBasic 的帮助	(12)
1.6.1 HELP 窗口	(13)
1.6.2 HELP 菜单	(13)
习题 1	(13)
第二章 数据的类型与运算	(14)
2.1 语言元素	(14)
2.1.1 字符集	(14)
2.1.2 标识符	(14)
2.1.3 汉字	(15)
2.2 数据类型	(15)
2.2.1 基本数据类型	(15)
2.2.2 用户定义的数据类型	(16)
2.3 常量与变量	(17)
2.3.1 常量	(17)
2.3.2 变量	(18)
2.3.3 常量和变量的作用域	(21)
2.3.4 数据类型转换	(21)
2.4 运算符与表达式	(22)
2.4.1 算术运算	(23)

2.4.2	表达式的执行顺序	(24)
2.5	函数	(24)
2.5.1	函数的概念	(24)
2.5.2	标准函数	(25)
2.5.3	日期与时间	(25)
2.5.4	随机函数	(27)
习题 2		(29)
第三章	标准输入与输出	(31)
3.1	数据的输入	(31)
3.1.1	键盘输入	(31)
3.1.2	READ 语句和 DATA 语句	(34)
3.1.3	RESTORE 语句	(36)
3.2	屏幕显示与打印	(37)
3.2.1	PRINT 语句	(37)
3.2.2	TAB 函数和 SPC 函数	(39)
3.2.3	WRITE 语句	(39)
3.2.4	PRINT USING 语句	(40)
3.2.5	LPRINT/LPRINT USING 语句	(42)
3.3	文本光标控制	(42)
3.3.1	LOCATE 语句	(42)
3.3.2	检查光标位置	(44)
3.4	文本屏幕尺寸	(44)
3.5	文本视见区	(46)
3.6	清除屏幕	(46)
3.7	软键	(47)
习题 3		(49)
第四章	控制程序流程之一:选择型程序设计	(52)
4.1	判断与条件表达式	(52)
4.1.1	关系表达式	(52)
4.1.2	逻辑表达式	(53)
4.2	IF(条件)语句	(54)
4.2.1	单行结构条件语句	(55)
4.2.2	块(多行)结构条件语句	(55)
4.3	SELECT CASE(情况)语句	(58)
习题 4		(62)
第五章	控制程序流程之二:循环型程序设计	(66)
5.1	FOR 循环	(66)
5.2	WHILE 循环	(69)
5.3	DO 循环	(69)
5.4	多重循环与 EXIT 语句	(71)

5.4.1 多重循环	(71)
5.4.2 EXIT 语句	(74)
5.5 程序举例	(75)
习题 5	(81)
第六章 过程	(84)
6.1 过程的定义与调用	(84)
6.1.1 SUB 过程的定义	(84)
6.1.2 FUNCTION 过程的定义	(85)
6.1.3 过程的调用	(86)
6.2 过程声明	(91)
6.3 过程中的变量及其用法	(91)
6.3.1 变量的作用域:全局与局部	(91)
6.3.2 变元传送	(93)
6.3.3 变量的共享	(94)
6.3.4 STATIC 变量	(95)
6.3.5 自动变量	(96)
6.3.6 变量重名	(97)
6.4 递归	(98)
6.5 程序举例	(99)
习题 6	(103)
第七章 数组	(107)
7.1 数组定义	(107)
7.1.1 DIM 语句	(107)
7.1.2 OPTION BASE 语句	(108)
7.2 数组的基本操作	(110)
7.2.1 数组元素的输入	(110)
7.2.2 数组元素的输出	(111)
7.2.3 数组元素的复制	(112)
7.3 静态数组与动态数组	(113)
7.3.1 静态数组与动态数组	(113)
7.3.2 ERASE 语句	(114)
7.3.3 REDIM 语句	(115)
7.4 在过程中使用数组	(116)
7.5 程序举例	(118)
习题 7	(124)
第八章 字符处理	(130)
8.1 字符串的概念	(130)
8.1.1 两种字符串	(130)
8.1.2 字符串数组	(132)
8.2 字符串的输入	(133)

8.2.1	用 READ/DATA 语句向字符串变量赋值	(133)
8.2.2	用 INPUT 语句给字符串变量赋值	(134)
8.3	字符串的运算	(134)
8.3.1	字符串的并置	(135)
8.3.2	字符串的比较	(135)
8.4	有关字符串运算的函数	(137)
8.4.1	确定字符串的长度	(137)
8.4.2	改变字符串的大小写	(137)
8.4.3	子字符串	(138)
8.4.4	其他函数	(142)
8.5	程序举例	(145)
习题 8		(148)
第九章	文件	(153)
9.1	文件的概念	(153)
9.1.1	文件的分类	(153)
9.1.2	文件说明	(153)
9.1.3	文件与记录	(154)
9.2	与文件处理有关的语句和函数	(155)
9.2.1	打开文件语句 (OPEN 语句)	(155)
9.2.2	关闭文件语句 (CLOSE 语句)	(157)
9.3.3	其他语句与函数	(158)
9.3	顺序文件	(162)
9.3.1	顺序文件的写操作	(162)
9.3.2	顺序文件的读操作	(164)
9.4	随机文件	(169)
9.4.1	FILD 方法	(169)
9.4.2	TYPE...END TYPE 方法	(169)
9.5	二进制文件	(173)
9.6	设备文件	(174)
习题 9		(176)
第十章	DOS 和程序管理	(179)
10.1	语句的分隔与注释	(179)
10.1.1	语句分隔符	(179)
10.1.2	注释语句	(180)
10.2	程序的暂停与结束	(181)
10.2.1	STOP 语句	(181)
10.2.2	END 语句	(181)
10.2.3	SYSTEM 语句	(182)
10.3	QBasic 的 DOS 环境功能	(183)
10.3.1	目录管理	(183)

10.3.2	文件命令语句	(183)
10.3.3	DOS 环境操作	(185)
10.4	程序的链接	(187)
10.4.1	CHAIN 语句	(187)
10.4.2	RUN 语句	(188)
10.5	CLEAR 语句	(189)
10.6	与设备驱动程序通信	(191)
习题 10		(191)
第十一章	陷阱技术	(193)
11.1	错误陷阱	(193)
11.1.1	设置错误陷阱	(193)
11.1.2	错误的模拟与识别	(194)
11.1.3	从错误处理子程序返回	(195)
11.2	事件陷阱	(198)
11.2.1	功能键陷阱	(198)
11.2.2	用户定义键陷阱	(200)
11.2.3	时钟陷阱	(202)
11.3	过程中的陷阱	(203)
习题 11		(204)
第十二章	图形	(205)
12.1	文本模式	(206)
12.1.1	文本模式与字符坐标系	(206)
12.1.2	用 LOCATE 语句制作字符动画	(206)
12.2	图形模式与屏幕	(207)
12.2.1	图形模式	(207)
12.2.2	点坐标系	(207)
12.2.3	SCREEN 函数	(208)
12.2.4	SCREEN 语句	(209)
12.3	基本绘图语句	(212)
12.3.1	画点	(212)
12.3.2	STEP 关键字	(214)
12.3.3	LINE 语句	(214)
12.3.4	DRAW 语句	(215)
12.3.5	CIRCLE 语句	(217)
12.4	颜色	(218)
12.4.1	颜色的设置	(218)
12.4.2	图形的着色	(221)
12.4.3	填充图案	(223)
12.5	视见区与窗口	(225)
12.5.1	VIEW 语句	(225)

12.5.2	WINDOW 语句	(226)
12.5.3	PMAP 函数	(229)
12.6	简单动画	(230)
12.6.1	图形动画	(230)
12.6.2	GET 与 PUT 语句	(231)
12.6.3	分页动画	(235)
12.7	程序举例	(237)
习题 12		(240)
第十三章	声音与音乐	(243)
13.1	BEEP 语句	(243)
13.2	SOUND 语句	(243)
13.3	PLAY 语句(音乐)	(245)
13.4	音乐事件陷阱	(247)
13.4.1	PLAY 与 ON PLAY 语句(事件陷阱)	(248)
13.4.2	PLAY 函数	(249)
13.5	程序举例	(249)
习题 13		(254)
第十四章	QBasic 程序调试	(255)
14.1	常见的程序错误与程序调试的一般方法	(255)
14.1.1	语法错误	(255)
14.1.2	算法错误	(256)
14.1.3	逻辑错误	(256)
14.1.4	程序调试的一般方法	(256)
14.2	程序调试实例	(257)
14.2.1	例一:仅用 PRINT 语句	(257)
14.2.2	例二:使用调试器	(259)
14.3	编程要点	(262)
14.3.1	避免错误	(262)
14.3.2	写一个好的程序	(263)
习题 14		(264)
附录 I	QBasic 关键字	(266)
附录 II	QBasic 菜单与键盘的使用	(268)
附录 III	QBasic 的限制与其他版本的 BASIC	(280)
附录 IV	QBasic 的出错信息	(284)
附录 V	ASCII 字符代码集与键盘扫描码	(294)
参考文献		(296)

第一章 程序设计与 QBasic 语言

1.1 程序设计语言

迄今为止,计算机还不懂得任何一种人类语言,它只懂也只能执行机器语言形式的程序指令。从计算机一诞生,人们就使用“计算机语言”(Computer Language)这一名词。50 年来,该专有名词一直与“机器语言”(Machine Language)具有相同的含义,它们同指一种事物,即计算机或机器的指令系统,或者说,一种计算机能够直接使用的、不须经过翻译的指令代码。计算机语言就是机器语言。不同型号的计算机具有不同的指令系统,也即具有不同的机器语言。

但是,计算机语言不是程序设计语言。程序设计语言(Programming Language)是人们用于编写计算机程序的语言,也是人与计算机进行信息交流的工具。程序设计语言可以是面向计算机或机器的语言,即机器语言和汇编语言等低级语言,也可以是独立于机器的语言,即高级语言,例如 COBOL、FORTRAN、BASIC、C 和 PASCAL 等等,而面向对象语言正在成为主流。但是,目前学习使用高级语言进行程序设计仍然是各行各业计算机应用人员的一项基本训练。

使用高级语言编写的程序,如 BASIC 程序,必须经过翻译才能被计算机接受。翻译的方法主要有三种:解释、汇编和编译。

解释的工作是渐进式的,程序动态地翻译成机器语言,即一边运行一边翻译,有如国际会议上的同声翻译。将被解释的程序(程序 A)由一个解释程序(程序 B)来处理。当要使用程序 A 时,计算机实际上要运行程序 B 并由它来执行程序 A 的每一步,程序 B 扫描程序 A 的文本,逐字逐字地解释程序 A,然后逐条逐条指令动态地执行,一步一步地完成 A 的任务。

由于计算机同时承担了翻译程序与执行程序两项工作,所以解释的速度很慢;解释不是每运行一次翻译一次,而是每执行一步就翻译一次,由于计算机程序的能力主要在于重复(即程序循环),这样,程序的许多指令在解释过程中都将被翻译多次,因此效率也低。另一方面,由于被解释程序能被动态地调整、修改和更正,所以解释过程比较灵活,而且翻译是持续进行的,修改可以随时进行并立即可见。

当你看到一个解释型源程序时,比如本书后面的 QBasic 例程,可以认为它就是在机器中运行的东西。但严格地说,这种看法是不正确的。从计算机和操作系统的角度看,真正在运行的是解释器,而所见到的程序只是它处理的数据。例如,对于一个 QBasic 程序,真正在运行的是 QBASIC.EXE,而“程序”只是它的数据。当然,这种数据是非常特殊的,它描述了用户想让计算机执行的工作,也即是通常说的程序。

解释型程序在运行时动态地翻译,而汇编和编译却不同,它们被事先翻译成机器语言并能独立运行(运行时不需要汇编器或编译器),永久保存。关于汇编与编译的详细讨论,请参考其他书籍。解释型程序不能创建独立的可执行程序,必须有如 QBASIC.EXE 一样的解释器才行,用户编写的 QBasic 程序只能在 QBasic 环境下运行。但是,QBasic 将编译的速度和解释的灵活性结合起来。当用户第一次装入程序时,解释程序 QBasic 立即将它转换成部分编译的中间形式,在运行时 QBasic 完成剩下部分的翻译工作。因为此时大部分工作已做了,程序几乎可以马上执行。如果用户作了些

修改,QBasic 只是重新处理修改所涉及到的部分,通常是一个子程序或更少。所有这些都使其程序开发环境的速度很快。

1.2 QBasic 语言简介

过去 BASIC(如 BASICA 与 GW-BASIC)的解释器都是简单的行编程环境,与目前流行的高级语言编译器相比,功能和复杂性都相去甚远,有很大的局限性。运行速度远远慢于其他高级语言,而且程序大小及其控制的数据量都有限。程序员认为这些老式的 BASIC 十分笨拙。幸运的是,从 MS-DOS 5.0 开始,Microsoft 公司在 DOS 中提供了一种类似于 Quick BASIC 的新型 BASIC 语言:QBasic。它能够理解以前的 BASIC 的关键字和语法,即兼顾解释 BASIC 老用户的习惯,又对准了编译 BASIC 的趋向,即 QBasic 解释器将编译的速度和解释的灵活性结合起来。它还包含其他流行的程序设计语言的关键字和功能,从而成为你编写程序的完备系统。

QBasic 最突出的优点是为用户提供了一个完整的结构化程序设计环境,能直接编写和运行程序,可将程序存入磁盘、在打印机上打印输出。QBasic 环境还包括许多文字处理器的功能,对输入的每个语句同时进行检查,保证你不出现打错字、误用关键字或语法规则等情况,使编写和修改程序更加方便。总之,QBasic 具有的功能强大的编辑环境、联机帮助功能以及增强的语言功能都会提高你现有程序的质量并帮助你更快地建立程序。主要功能如下:

- ①良好的开发环境:全屏幕编辑、调试器、程序文件管理系统和下拉式菜单;
- ②保留解释器,与 IBM 的 BASICA、Microsoft 的 GW-BASIC 兼容,便于使用过去的解释 BASIC 程序;
- ③改进了语言功能:增加自定义过程,局部变量,模块结构、递归过程以及与解释 BASIC 相容的数字标号;
- ④具有 Smart Editor,自动检查每一行的语法并标志错误,编辑程序时即可改正错误;
- ⑤支持大多数外部设备:鼠标器、协处理器和彩显;
- ⑥程序长度不受 64KB 限制,最大 160KB;
- ⑦支持 IEEE(美国电工程师协会)标准化浮点格式的数值。

QBasic 的局限主要是不能产生能够独立运行的可执行程序,余者与 Quick BASIC 几乎完全相同。本书附录 I 提供了 QBasic 与其他版本 BASIC 的一些差别,以及有关 QBasic 本身的一些其他信息。

1.3 QBasic 的启动与退出

我们从 QBasic 的编程环境(调入、编辑、调试和执行 QBasic 程序的外壳框架)入手,等你对它初步熟悉之后,才开始学习程序设计不迟,即使你是一个新手。

1.3.1 启动 QBasic

有两种不同的启动方法:

- 1. 从 MS-DOS 的外层 DOSSHELL 中启动 QBasic。请参考 MS-DOS 手册或有关图书。
- 2. 从 DOS 提示符状态下启动 QBasic,其完整的命令格式(在 DOS 状态下输入命令 qbasic/? 即显示)如下:

[D:PATH] QBASIC [/B] [/EDITOR] [/G] [/H] [/MBF] [/NOHI] [[/RUN]sourcefile]

【说明】

①D:PATH:指定 QBASIC 所在的驱动器和路径名。当 QBASIC 不在当前目录或不在指定搜索路径上时,必须指定这个参数;

②/B:允许使用带彩色图形卡的混合单色显示器。彩色显示器按单色显示;

③/EDITOR:调用 MS-DOS 的 EDIT 文本编辑程序。可以省略为:/ED;

④/G:指定在 CGA 上以尽可能快的方式启动 QBASIC;

⑤/H:按最多行数显示 QBASIC;

⑥/MBF:指定 QBASIC 的转换函数 CVS、CVD、MKS\$、MKD\$ 等把 IEEE 格式数处理成 Microsoft 二进制格式数;

⑦/NOHI:指定在不支持高分辨率的显示器上运行 QBASIC;

⑧/sourcefile:指定 QBASIC 启动时装入的源程序文件名。若源程序是用 BASICA 或 GWBASICA 建立的,则须按其中的“A”方式存盘;

⑨/RUN sourcefile:指定 QBASIC 装入源程序并立即运行,然后才显示程序。

现在,我们只考虑在 DOS 提示符下不带参数来启动 QBASIC。这也有两种不同的方法,即从硬盘或软盘启动 QBASIC。假设硬盘上已经安装 MS-DOS 5 或 6.X, QBASIC 程序及相关文件位于 C 盘的 DOS 目录中。这时,先打开机器,然后输入以下命令(带下划线的小写字母),并在每个命令后按回车键 Enter:

C:\>cd dos

C:\DOS>qbasic

很快就会看到如下包含 Welcome 欢迎信息的 QBASIC 屏幕:

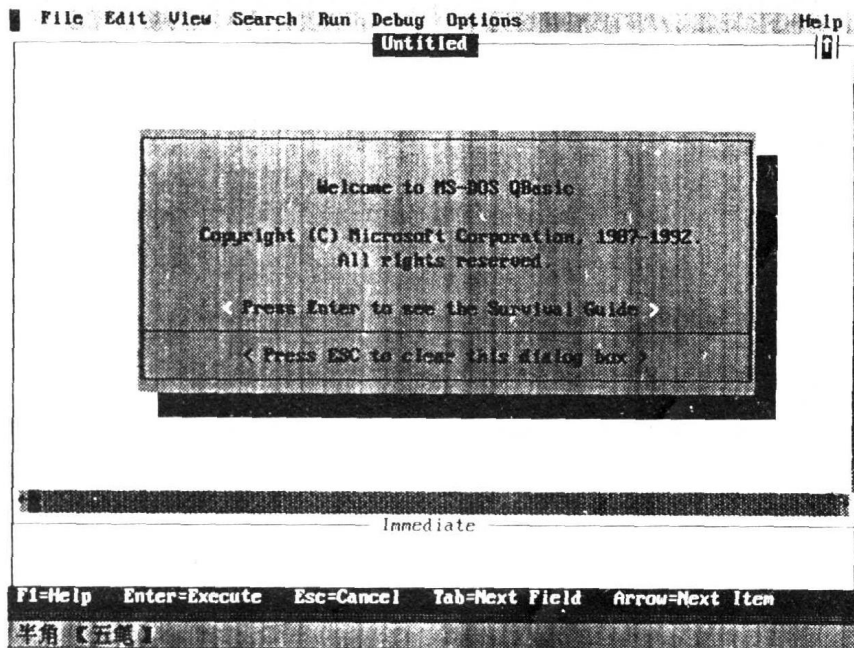


图 2-1 QBASIC 启动屏幕

以上屏幕提供两种选择:

- ①按 Enter 键浏览 QBasic 的 Survival Guide 即所谓生存指南帮助系统;
- ②按 Esc 键清除屏幕中的 Welcome 欢迎信息进入 View 窗口,这时可以开始编写程序。

现在请按 Esc 键,屏幕显示如图 2-2。

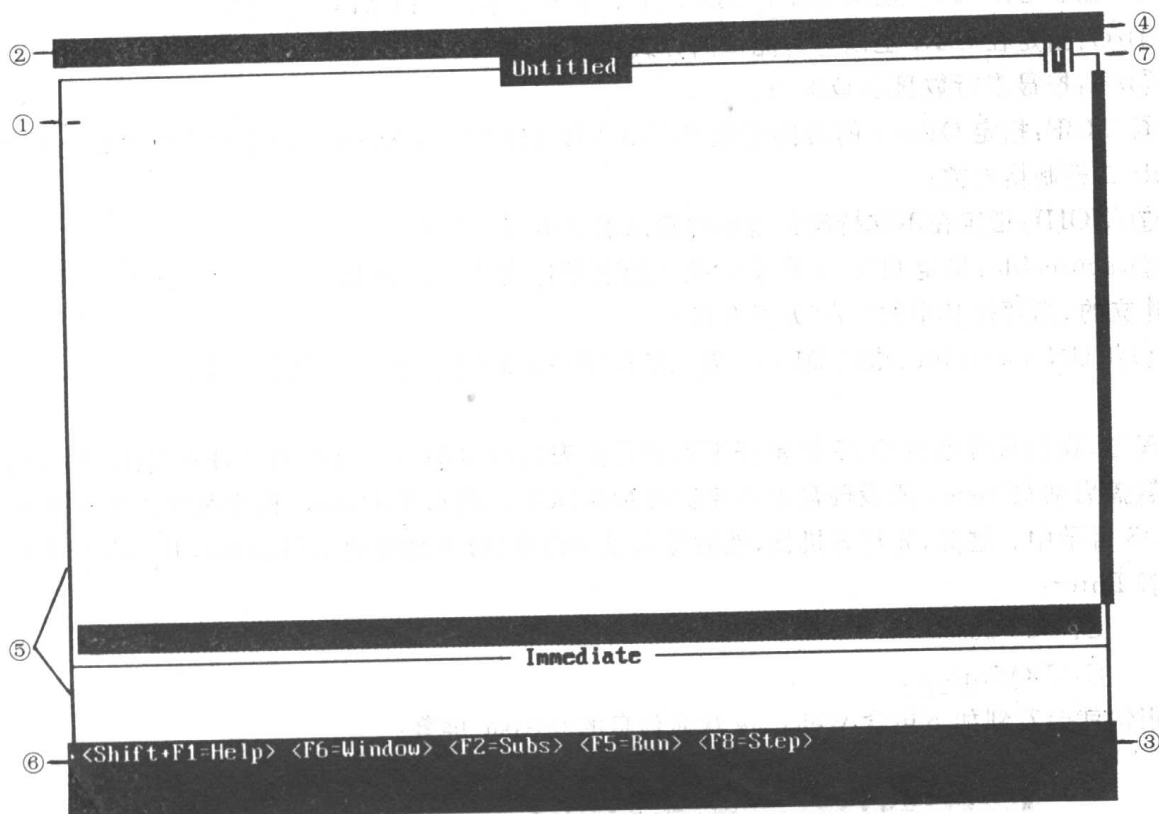


图 2-2 QBasic 的 View 窗口

1.3.2 QBasic 屏幕

从现在开始,我们的学习和程序设计将与 QBasic 语言环境分不开,也就是说,我们将围绕 QBasic 屏幕进行学习,因此,仔细考察 QBasic 屏幕的各个部分是必要的,这将有助于你建立程序和处理文本。以下介绍图 2-2:

①**光标**:屏幕左上角的闪烁下划线就是光标,它指示下一个输入的字符在屏幕上出现的位置。你会立即注意到光标的。

②**鼠标指针**:如果你的计算机已经安装鼠标器并运行鼠标驱动程序,那么,一个矩形的鼠标指针将出现在屏幕左上角,它使你能够更加容易地选择菜单项,同时在编辑、修改程序时也能迅速移动(光标)位置。

③**行列位置编号**:位于屏幕右下角的一对数字表示光标的当前位置,第一个数字表示光标所在的行号,冒号后面的数字是光标的列号。光标一移动,这对数字就改变。

④**菜单条和菜单名称**:菜单条横贯屏幕顶部,包含 QBasic 的所有 8 个下拉式菜单的名称。详见附录。

⑤**View 窗口和 Immediate 窗口**:QBasic 一开始就提供两个工作窗口——

- 可在其中输入和运行程序的上窗口称为 View(观察)窗口。
- 将程序指令实际输入到程序之前并可在其中进行测试的下窗口称为 Immediate(直接)窗口。

每次只能在这两个窗口中的一个工作,可按功能键 F6 在这两个窗口之间切换;如果使用鼠标器,只需将鼠标指针移到需要激活的窗口内部后单击。目前工作的窗口称为活动窗口,光标位于其中,而且窗口标题变亮。图 2-3 表示 Immediate 窗口为活动窗口,它是一个可扩展的窗口。用户只要在这里键入要试验的单独的一个 QBasic 命令或语句并按回车键,命令就能立即执行,并且立即就能看到命令的结果。对这里的语句作精心的修改或调整之后,利用 QBasic 的剪贴技术,便可将它插入 View 窗口的程序中。用户也可以键入一系列单独的命令到 Immediate 窗口中,然后通过移动光标到每一行并按 Enter 键进行逐个检验。本书第十四章将举例使用该窗口调试程序。

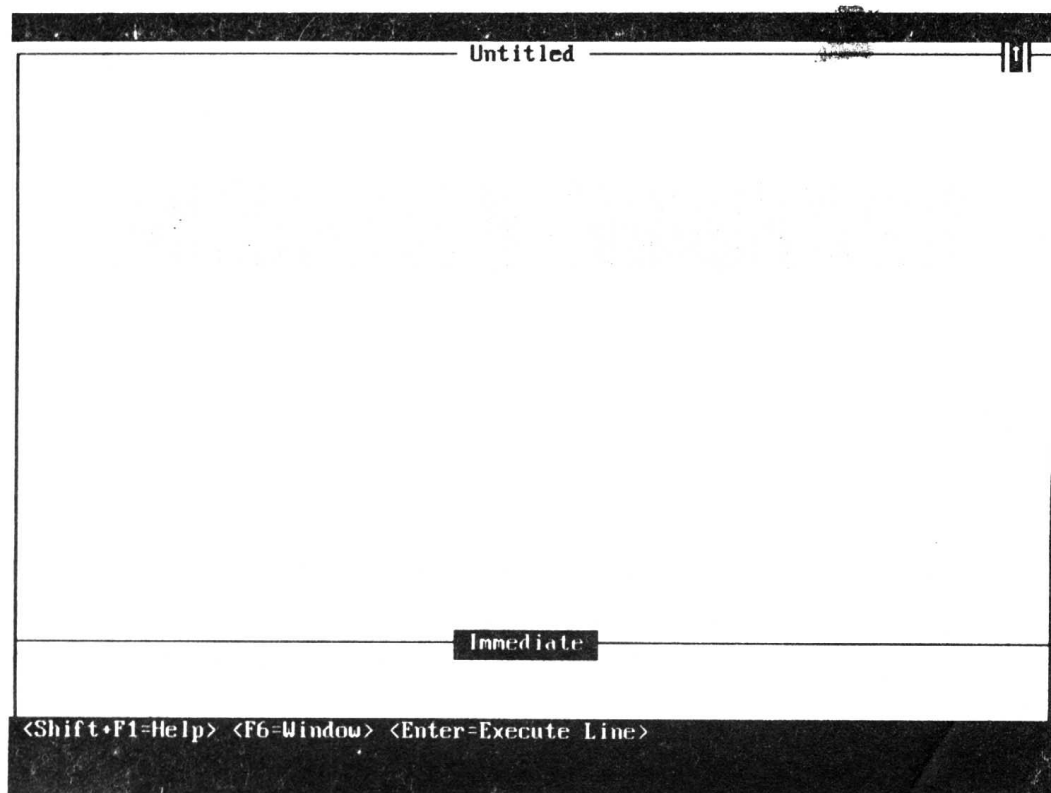


图 2-3 活动的 Immediate 窗口

⑥参考条:屏幕底行为参考条。参考条包含可在活动窗口使用的控制键集合,这个集合随着窗口切换和菜单项的变化而变化。QBasic 中所有参考条的中英文对照参阅附录Ⅱ。

⑦放大与缩小控制:位于屏幕右上角 Help 菜单名下包含上箭头的矩形就是缩放控制。它可使 View 窗口扩大并覆盖 Immediate 窗口而使后者看不见。使用缩放控制之后,上箭头变成双向箭头,这时 View 窗口最大。可用鼠标单击缩放控制进行放大和缩小;也可用键盘放大缩小,只要在按住 Ctrl 键的同时按 F10 键即可。图 2-4 表示 View 窗口最大。

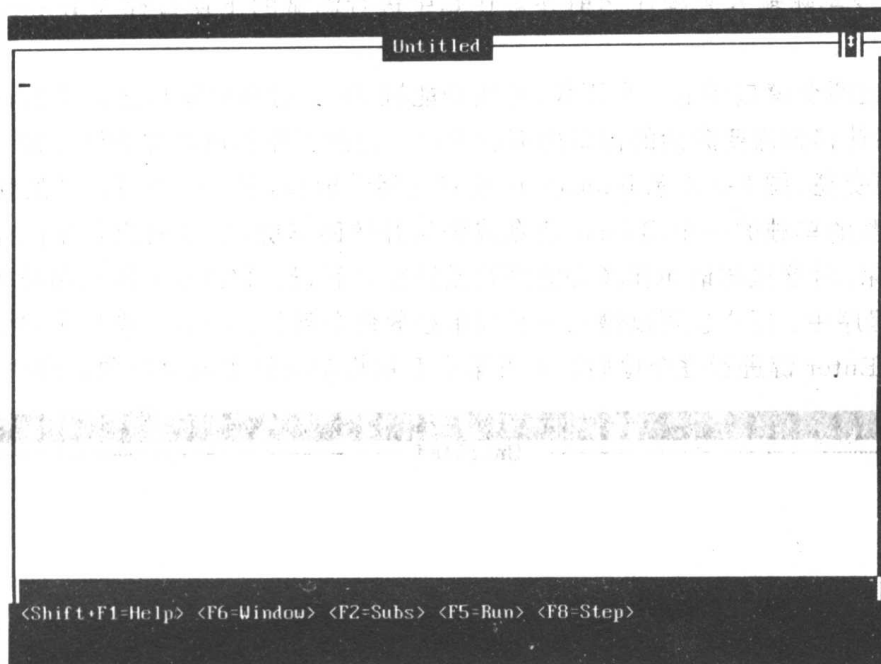


图 2-4 最大的 View 窗口

1.3.3 退出 QBASIC

使用鼠标器或者按 Alt 键激活菜单,选择 File,然后选择 Exit 退出命令。这将从内存中清除 QBASIC 并立即返回 DOS。

在关闭计算机之前或者结束 QBASIC 编程转而为计算机做别的事情时,应当先退出 QBASIC。因为如果在 View 窗口中尚有未存盘的程序,或者以前曾存盘但已作了修改的程序,QBASIC 将会问你是否要将程序存盘。这时,在 View 窗口中出现如下所示的对话框:(对话框的用法请参考附录 I)

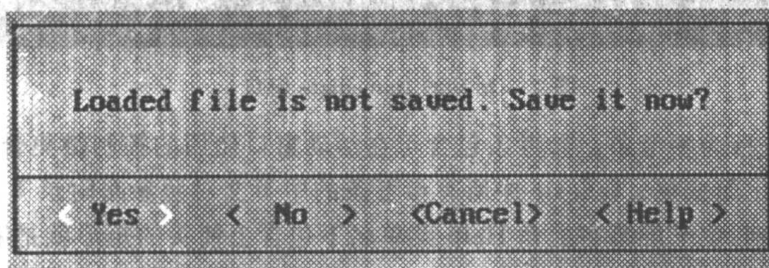


图 2-5 Exit 命令对话框

这里,选择 Yes 则按 View 窗口标题位置当前显示的名称将新的未曾存盘的程序存盘或者将程序的修改版本存盘,已在磁盘上的前一版本被覆盖,如果 View 窗口标题显示“Untitled”时,QBASIC 会要求用户输入程序文件名称;选择 No 则不存盘,请不要匆忙选择 No,因为程序或者程序的修改将丢失而不被保留;选择 Cancel 则取消 Exit 命令,返回 View 窗口中的程序;选择 Help 则可