

中等职业学校计算机系列规划教材

QBASIC 语言程序设计

武马群 主编

刘 冰 编著

北京工业大学出版社

内 容 简 介

本书共 10 章，前 8 章主要介绍 QBASIC 语言最基本的编程语句、编程思路和编程方法，这一部分是按知识点的顺序进行一般性能力训练并帮助学生学会举一反三、触类旁通，通过这部分学习，培养学生的综合分析、系统编程和上机操作的应用能力；第 9 章为选择学习部分，介绍了 QBASIC 语言的屏幕控制作图语句、发声与音乐语句，以增加教材使用的弹性；第 10 章介绍 QBASIC 语言的程序文件和数据文件的建立及读写语句。

本书可作为中等职业学校计算机技术专业教材，也可作为计算机爱好者的自学用书。

QBASIC 语言程序设计

武马群 主编

刘 冰 编著

北京工业大学出版社出版发行

邮编：100022 电话：(010) 67392308

各地新华书店总经销

徐水宏运印刷厂印刷

2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

787 mm × 1 092 mm 16 开本 印张 14.5 字数 349 千字

印数：1~5 000 册

ISBN 7-5639-1515-X/T · 249

定价：20.00 元

前 言

程序设计是计算机应用人员的一项基本功，也是计算机专业的学生在校学习期间的一项基本训练。通过对程序设计的学习，学生可以具体地了解计算机的工作原理，知道计算机能做什么和不能做什么，而不是仅仅学会简单的计算机操作。

一直以来，BASIC 语言作为大众化的计算机编程语言在普及计算机知识和应用中起着至关重要的作用。Quick BASIC (简称 QBASIC) 语言是微软公司 1987 年推出的 BASIC 版本。它继承了传统 BASIC 语言易学易用的优点，完全符合结构化程序设计要求，提供了一个开发程序的集成环境，用户在编程、修改、编译、调试及运行时均可通过菜单操作，十分方便。它与 GW BASIC 和 BASIC 高度兼容，提供了全局变量和局部变量，程序模块化；编译后产生一个可执行文件，提高了执行效率。作为第三代 BASIC 语言的代表之一，QBASIC 语言的出现，极大提高了 BASIC 语言的档次与功能，深受用户的欢迎。

本书共分 10 章详细地介绍了 QBASIC 程序设计的方法。第 1 章介绍程序设计的基础知识、第 2 章介绍 QBASIC 语言基础知识、第 3 章介绍顺序结构程序设计、第 4 章介绍选择结构程序设计、第 5 章介绍循环结构程序设计、第 6 章介绍函数与子程序、第 7 章介绍数组、第 8 章介绍字符串处理、第 9 章介绍屏幕图形与声音、第 10 章介绍文件。

本书每章开始都明确指出了本章的学习目标，每章结束都进行本章小结，便于读者有的放矢地学习。每章课后还附有上机指导和大量习题、实验，以加强学生的动手能力，并进一步巩固所学的知识。

本书层次分明、结构合理、知识全面、实用性强，适合作为中等职业学校计算机技术类专业教材，也可作为计算机爱好者的自学用书。

编 者

2005 年 5 月

目 录

第 1 章 程序设计的基础知识	1
1.1 程序设计语言简介	1
1.1.1 计算机指令	1
1.1.2 机器语言	2
1.1.3 汇编语言	2
1.1.4 高级语言	2
1.1.5 编译程序	2
1.2 QBASIC 语言的发展及特点	3
1.2.1 QBASIC 语言的发展	3
1.2.2 QBASIC 语言的特点	4
1.3 QBASIC 语言编程环境	6
1.3.1 运行环境	6
1.3.2 启动 QBASIC	6
1.3.3 QBASIC 语言窗口	7
1.3.4 QBASIC 程序操作	9
【本章小结】	11
【上机指导】	12
【习题】	13
第 2 章 QBASIC 语言基础知识	16
2.1 QBASIC 语言的语法知识	16
2.1.1 字符集	16
2.1.2 模块与程序行	16
2.2 QBASIC 语言的数据类型	18
2.3 常量和变量	18
2.3.1 常量	18
2.3.2 变量	21
2.4 标准函数	23
2.5 运算符和表达式	24
2.5.1 算术运算符与算术表达式	24
2.5.2 关系运算符与关系表达式	25
2.5.3 逻辑运算符和逻辑表达式	25
【本章小结】	26
【上机指导】	27

【习题】	27
第 3 章 顺序结构程序设计	29
3.1 赋值语句 (LET 语句)	29
3.2 变量数据交换语句 (SWAP)	30
3.3 输出语句 (PRINT 语句)	30
3.3.1 PRINT 语句	31
3.3.2 LPRINT 语句	34
3.4 输入语句	34
3.4.1 键盘输入语句 (INPUT 语句)	34
3.4.2 读数、置数语句 (READ/DATA)	35
3.4.3 恢复数据区指针语句 (RESTORE 语句)	36
3.5 暂停和结束语句	37
3.5.1 暂停语句 (STOP 语句)	37
3.5.2 结束语句 (END 语句)	37
3.5.3 注释语句 (REM 语句)	38
【本章小结】	39
【上机指导】	39
【习题】	41
【实验】	42
第 4 章 选择结构程序设计	43
4.1 分支结构语句 (IF 语句)	43
4.1.1 行 IF 语句	43
4.1.2 块 IF 语句	45
4.2 多分支结构语句 (SELECT CASE 语句)	49
4.2.1 多分支结构语句	49
4.2.2 多分支转移语句 (ON GOTO 语句)	52
4.3 无条件转移语句 (GOTO 语句)	52
4.4 选择结构程序设计举例	54
【本章小结】	56
【上机指导】	57
【习题】	58
【实验】	61
第 5 章 循环结构程序设计	62
5.1 计数循环语句 (FOR...NEXT 语句)	62
5.2 条件循环语句 (WHILE 语句)	68
5.3 条件循环语句 (DO...LOOP 语句)	71

【本章小结】	78
【上机指导】	78
【习题】	80
【实验】	82
第 6 章 函数与子程序	84
6.1 模块化结构概述	84
6.2 用户自定义函数 (DEF 函数)	85
6.2.1 单行自定义函数	85
6.2.2 多行自定义函数	86
6.3 子程序的定义和使用	88
6.3.1 块内子程序 GOSUB 语句	88
6.3.2 独立模块的子程序 SUB 语句	91
6.4 模块化函数	96
6.4.1 模块化函数的建立和调用	96
6.4.2 模块化函数举例	97
6.5 全局变量与局部变量	99
6.5.1 全局量与局部量的概念	99
6.5.2 全局变量与全局常量	99
6.5.3 局部变量与局部常量	100
6.6 函数和子程序的嵌套与递归调用	101
6.6.1 过程的嵌套调用	101
6.6.2 过程的递归调用	102
【本章小结】	103
【上机指导】	104
【习 题】	106
【实验】	109
第 7 章 数组	110
7.1 数组简介及建立	110
7.1.1 数组简介	110
7.1.2 数组的建立	111
7.2 数组的应用	113
7.2.1 一维数组的应用	113
7.2.2 二维数组的应用	116
7.3 静态数组和动态数组	120
7.3.1 释放数组语句	120
7.3.2 重定义数组	120
7.3.3 数组上、下界函数	122

7.4 数组的举例	123
【本章小结】	125
【上机指导】	125
【习题】	129
【实验】	129
第8章 字符串处理	130
8.1 字符串的基本概念	130
8.1.1 字符与字符串	130
8.1.2 字符串常量	130
8.1.3 字符串变量	131
8.2 字符串的基本操作	133
8.2.1 字符串变量的赋值	133
8.2.2 字符串长度函数	135
8.2.3 字符串的连接	135
8.2.4 字符串的比较	136
8.2.5 取子字符串	136
8.3 其他字符串函数	138
8.3.1 生成字符串函数	138
8.3.2 从键盘读入字符	139
8.3.3 字符串与数的转换函数	139
8.3.4 大小写字母转换函数	141
8.3.5 字符串序号函数	141
8.3.6 日期和时间函数	142
8.4 字符串处理的举例	142
【本章小结】	144
【上机指导】	144
【习题】	145
【实验】	147
第9章 屏幕图形与声音	148
9.1 屏幕图形的基本知识	148
9.1.1 屏幕显示模式	148
9.1.2 图形屏幕坐标	149
9.2 屏幕控制语句	149
9.2.1 CLS 语句	149
9.2.2 光标定位语句 (LOCATE)	149
9.2.3 WIDTH 语句	150
9.2.4 SCRLIN 和 POS 函数	150

9.2.5	SCREEN 语句	150
9.2.6	屏幕设置语句 COLOR	151
9.3	屏幕作图语句	152
9.3.1	画点语句	152
9.3.2	画线或矩形	154
9.3.3	画圆语句	154
9.3.4	连续画线语句	156
9.4	图形着色语句	157
9.5	发声与音乐	159
9.5.1	发声语句	159
9.5.2	演奏音乐	160
	【本章小结】	165
	【上机指导】	165
	【习题】	167
	【实验】	169
第 10 章	文件	170
10.1	文件的基本概念	170
10.1.1	什么是文件	170
10.1.2	文件的命名	171
10.1.3	文件的分类	171
10.1.4	文件的输入/输出	172
10.1.5	文件长度与文件指针	172
10.2	用户自定义数据类型	172
10.2.1	用户自定义类型的定义	172
10.2.2	记录变量的引用	173
10.3	顺序文件的操作	174
10.3.1	顺序文件的打开	174
10.3.2	顺序文件的写入	175
10.3.3	顺序文件的读出	175
10.3.4	顺序文件的关闭	176
10.3.5	顺序文件的使用	176
10.4	随机文件的操作	179
10.4.1	随机文件的打开	179
10.4.2	随机文件的输入/输出	179
10.4.3	随机文件的建立	180
10.4.4	随机文件的读取	181
10.5	综合应用举例	184
	【本章小结】	194

【上机指导】	194
【习题】	195
附录 A 全国计算机等级考试二级 QBASIC 大纲	197
附录 B 全国计算机等级考试二级 (QBASIC) 笔试试题	200
附录 C 常用字符及其 ASCII 码对照表	212
附录 D QBASIC 语言保留字	213
附录 E QBASIC 语句一览表	214
附录 F QBASIC 函数一览表	218

第 1 章 程序设计的基础知识

【学习目标】

1. 程序设计语言简介。
2. QBASIC 语言的发展及特点。
3. QBASIC 语言编程环境。

1.1 程序设计语言简介

计算机语言，实际上是“人脑”与“电脑”相互联系的纽带，是“人”与“计算机”进行双向信息交流的共同语言桥梁。向计算机提供有关的控制信息，指挥计算机工作的命令称为指令。程序是为了解决某一问题而设计的操作指令的有序集合（又称指令系统）。而指令系统是用某种语言编制的操作指令的有序集合，编制程序的过程称为程序设计。用来编制程序的计算机语言也就是程序设计语言。

人与计算机进行信息交流通常使用程序设计语言。人们将自己的意图用某种程序设计语言编写成程序，然后输入计算机，告诉计算机完成什么任务以及如何完成，达到人对计算机进行控制的目的。随着计算机技术的发展和用户要求的不断提高，编程语言也经历了由简单到复杂、从低级到高级的发展过程。

程序设计语言是软件系统的重要组成部分，一般可以分为机器语言、汇编语言和高级语言三类。

1.1.1 计算机指令

人们利用计算机来解决具体的问题，是通过一连串计算机指令来完成的，这个指令序列就是程序。一条指令规定计算机执行一个最基本的操作，一种计算机所能识别的一组不同指令的集合称为该种计算机的指令或指令系统。指令完全是用二进制数表示的，指令系统包括以下四种类型的指令。

- （1）数据处理指令：用于对数据进行算术运算、逻辑运算、移位和比较操作。
- （2）数据传送指令：用于在存储器、寄存器和微处理器等设备间进行数据传送。
- （3）程序控制指令：用于进行条件转移、无条件转移、转子程序和暂停等操作。
- （4）状态管理指令：用于中断、屏蔽中断等操作。

1.1.2 机器语言

通过二进制代码和计算机的指令符号来编写的程序，叫做机器语言。计算机能够理解并直接执行这种程序，运行速度也快，基本上充分发挥了计算机的速度性能。

但是，机器语言存在两个问题：第一是对于初学者来说不够直观，很难接受和理解，出现错误也难检查和修改，而且不便于交流；第二是对于不同的计算机硬件（主要是 CPU），其机器语言是不同的，因此，针对一种计算机所编写的机器语言程序不能在另一种计算机上运行，所以机器语言的应用有很大的局限性。

1.1.3 汇编语言

正是因为机器语言程序的不易编写与阅读，才促进了汇编语言的发展。为了便于理解和记忆，人们采用能反映指令功能的英文编写助记符来表达计算机语言，这种符号化的机器语言就是汇编语言。

汇编语言采用助记符，比机器语言直观，易于记忆和理解。因此汇编语言程序比机器语言程序易读，易检查，易修改。另外，汇编语言与机器语言一般是一一对应的，其程序的执行效率仍然比较高，但程序可移植性较差。

用汇编语言编写的程序称为汇编语言源程序，机器无法直接执行，要用计算机配置好的汇编程序把它翻译成机器语言表达的目标程序，机器才能执行，这个翻译过程称为汇编。

1.1.4 高级语言

机器语言和汇编语言都是面向机器的语言，虽然其程序的执行效率高，但它们对机器依赖性大，编写程序效率却很低，编制好的程序通用性也差。

高级语言是一种不依赖具体计算机类型，与机器指令系统表面无关，描述方法接近人们对求解问题的表达方式，易于书写与掌握的程序设计语言。

高级语言一经诞生，就得到了迅速的发展。目前广泛应用的高级语言有 BASIC、FORTRAN、PASCAL、C、JAVA 等。所有的高级语言具有以下共同的特点：

(1) 不独立于具体计算机，即使用高级语言完全不必知道所用计算机的机器指令系统。

(2) 高级语言中的一条可执行的语句包含许多条机器指令。

(3) 用高级语言编制的程序可移植性好，不需要经过太大的修改，就可以在其他类型的机器上运行。

(4) 所有高级语言编写的程序（称为源程序）都要通过编译程序翻译成机器语言表达的目标程序才能被计算机执行，或者通过解释程序边解释边执行。

1.1.5 编译程序

在计算机上执行一个高级语言程序一般分为两步：第一步，用一个编译程序把高级语言程序翻译成机器语言程序；第二步，运行所得的机器语言程序求得计算结果。编译程序的作

用是把某一种语言的源程序改造成另一种语言表达的目标程序，而源程序与目标程序在逻辑上是等价的。不同的高级语言都有自己的高级语言编译程序。

一般来说，编译程序均是把源程序改造成用机器语言或汇编语言表达的目标程序。编译程序的工作过程可以划分成词法分析、语法分析、中间代码生成、优化和目标代码生成等阶段。

1.2 QBASIC 语言的发展及特点

1.2.1 QBASIC 语言的发展

1964 年美国两位计算机科学家 G. Kemeny 和 Thomas. E. Kurtz 开发了 BASIC 语言，这是计算机语言发展史上的一件大事，对计算机的推广使用起了重要的作用。BASIC 语言发明者的初衷是“使初学者容易学习”，这正是 BASIC 语言的独到之处和成功之处。由于 BASIC 语言无可比拟的“容易学习”特点，使其从本来供学生使用的一种程序设计语言，很快跨出校园走向社会，传播到全世界，成为初学者学习计算机程序设计的首选语言。

BASIC 语言经过了以下的几个发展阶段。

1. 初期的 BASIC 语言

由于最初计算机内存容量的限制，BASIC 语言只是一种小型语言，是在一些小型机上使用的分时系统，采用的是编译方式，功能相对较弱。但它的出现使计算机由只能被专业人员使用发展到能为一般人使用，开辟了一个“大家都来学习计算机”的新阶段。在这个阶段，BASIC 语言先后在各种计算机上实现了正确运行，而且各个计算机厂家在实现运行时都设法对 BASIC 的功能做了扩充和改进，出现了多种版本，语句数量增加，运行速度提高，使其不仅能用于学习，也能用于解决小型问题。美国标准化协会 ANSI 于 1974 年制定的第一个 BASIC 标准——Minimal BASIC (小型 BASIC) 就是以 1964 年初期的 BASIC 语言为蓝本的。

2. 微机 BASIC

1971 年美国 Intel 公司研制出了微处理器，并由此开辟了微型计算机的时代。由于当时计算机内存较小，其他计算机高级语言难以在微型计算机上实现，而 BASIC 语言对硬件要求比较低，使其得以在微型机上运行和发挥。1975 年美国的比尔·盖茨与他的伙伴们在第一台微型机上将编译型 BASIC 语言压缩成只有 4 KB 的解释型 BASIC 语言，从而实现了 BASIC 语言在微型机上的运行。用解释方式运行程序虽然比编译方式慢得多，但对初学者来说，使用比较方便，而且能够在微型计算机上运行高级语言程序。在比尔·盖茨第一个微机 BASIC 的带动下，以后出现的微机都毫无例外地配置了 BASIC 语言，使得 20 世纪 70 年代成为 BASIC 语言发展过程中一个重要阶段和全盛时期，全世界 BASIC 语言的用户达到一亿之多。

硬件技术的迅速发展也为软件的发展提供了越来越宽广的舞台，各个厂商对 BASIC 语言的功能做了扩展，使其能处理文件、具有绘图和音乐功能。BASIC 语言不仅以它简洁的语法和容易学习的特点得以广泛流行，而且它的实用性使其成为广大专业程序设计人员和普通用户手中的有力工具。20 世纪 70 年代，BASIC 语言已经由小型、简单的学习语言发展成为功能丰富的实用语言。第二代 BASIC 语言的代表是 GW BASIC 和 MS BASIC (即 IBM—PC 的 BASIC)。

它的许多功能已经能与其他优秀的计算机语言相媲美，有的功能如绘图功能甚至超过了其他的语言。当时许多应用软件都是用 BASIC 语言编写的，广大初学者要使用计算机解决问题，也都使用 BASIC 语言来编写程序。我国 20 世纪 80 年代的计算机程序普遍就是以 BASIC 语言来编写的。

但是在 BASIC 语言的发展过程中，不同型号计算机的 BASIC 语言缺乏统一性，各种机器上的 BASIC 语言不能互相兼容，虽然 1978 年通过一个 BASIC 标准，但早已落后于形势没有多少实用的价值，于是人们期待新的 BASIC 标准的诞生。

3. 结构化 BASIC

20 世纪 70 年代，由于软件高速发展而出现了所谓的“软件危机”。著名的荷兰计算机科学家 E.W. Dijkstra 等人提出了结构化程序设计的思想。结构化程序设计要求高级语言有直接实现顺序、选择及循环等三种基本结构的语句，而且要求程序模块化，即主程序和子程序分别属于不同模块，便于按功能设计程序模块。第二代 BASIC 语言显然不能满足结构化程序设计的要求。

20 世纪 80 年代中期，美国国家标准化协会根据结构化程序设计的思想，提出了一个新的 BASIC 标准草案，并于 1987 年正式通过，称为“Full Basic(X3.11-1987)”，意思为“完全的基本”。在此前后，出现了一些结构化的 BASIC 语言，例如 True BASIC、Quick BASIC、Turbo BASIC 等。

BASIC 语言的两位创始人于 1984 年推出的 True BASIC 是对 BASIC 语言做出了重大改进，他严格遵循美国国家标准化协会的规定，完全适应结构化和模块化程序设计的要求，而且保留了 BASIC 语言的好学、易懂、编程、调试简单等优点，还提供了解释和编译两种工作方式。

Quick BASIC 是微软公司 1987 年推出的 BASIC 版本。他提供了一个开发程序的集成环境，用户在编程、修改、编译、调试及运行时均可通过菜单操作，十分方便。它与 GW BASIC 和 BASIC 高度兼容，提供了全局变量和局部变量，程序模块化；编译后产生一个可执行文件，提高了执行效率。

第三代 BASIC 语言的特点是，完全符合结构化程序设计要求，并具有编译工作方式。第三代 BASIC 语言的出现，大大提高了 BASIC 语言的档次与功能，受到用户的欢迎。

4. 在 Windows 环境下运行的 BASIC 语言

20 世纪 80 年代中期，微软公司推出的 Windows 操作系统提供了图形方式的用户界面，通过鼠标、窗口、菜单等操作计算机，使操作变得更加方便，从而受到了广大用户的欢迎。Windows 操作系统的出现要求编制新的软件能直接在其环境下运行，即所谓“for Windows”软件。1991 年微软公司推出了第一个直接运行在 Windows 下的 BASIC 语言是 Visual BASIC，用于开发 Windows 应用软件，用它能设计出具有良好用户界面的应用程序。Visual BASIC 与 Quick BASIC 兼容，用 Quick BASIC 语言编写的程序不加修改就可运行在 Visual BASIC 环境下。

1.2.2 QBASIC 语言的特点

QBASIC 语言由微软公司开发，作为 MS-DOS 5.0 组成部分提供的，因此用户不用专门购买，在 MS-DOS 6.x 中，也同样提供了 QBASIC 语言软件。QBASIC 语言是 BASIC 语言发展到近

阶段的产物，是第二代 BASIC 的现代化超集。QBASIC 语言与第二代 BASIC 语言相兼容，可以认为是第二代 BASIC 语言的最佳继承者，是一种完全结构化和模块化的语言。

QBASIC 语言主要具有以下特点：

(1) QBASIC 语言在实现其先进性的同时，仍保持着易学、易用的特点。

(2) QBASIC 语言是一种现代化程序设计语言，是完全结构化和模块化的语言，与其他高级语言具有一致性。通过对 QBASIC 语言的学习，可以举一反三地学习和掌握其他高级语言。

(3) QBASIC 语言包含了更多的语句和函数，包含的信息量大。

QBASIC 语言扩充了以下一些功能：

(1) 扩充了数据类型，除整型、实型、双精度型、字符型数据外，增加了长整型和 IEEE 格式实型数、定长字符串型变量，同时支持用户定义数据类型。

(2) 提供了新的选择结构“IF...END IF”和多分支选择结构“SELECT CASE...END CASE”。

(3) 改进了循环结构，除了 FOR...NEXT, WHILE...WEND 循环结构外，还增加了 DO 循环结构 (DO WHILE ...LOOP, DO UNTIL...LOOP)。

(4) 支持 EGA 和 VGA 视频图形模式。

(5) 支持递归过程和一个内含的编辑程序。

(6) QBASIC 语言程序最大长度为 160 KB (代码和数据总长)，而 GW BASIC 和 BASIC 程序最大长度只有 64 KB。

(7) QBASIC 语言具有良好的编辑环境和友好的编辑界面，可以使用键盘和鼠标进行编程。

(8) QBASIC 语言具有即时语法检查的功能，包含完善的联机“帮助”系统，在编辑过程中可以随时查询“帮助信息”。

(9) 提供了分步执行和跟踪等集成的调试工具。

(10) QBASIC 语言采用先进的解释程序，使程序运行比 GW BASIC 语言更快。

(11) QBASIC 语言与 GW BASIC 语言完全兼容。QBASIC 语言中，一个语句行可以有或者没有行号或标号。行号或标号只是一个地点标记，与程序的执行顺序无关，只是用于程序流程可以转移到该地点继续执行。用 GW BASIC 或 BASIC 语言编写的源程序可不加修改或做少量修改就可以在 QBASIC 语言环境下运行。

(12) QBASIC 语言是 Quick BASIC 语言和 Visual BASIC 语言的子集，但包含了它们的主要功能，学过 QBASIC 语言以后，可以很容易地学习和掌握如 Quick BASIC、Visual BASIC 等专业的 BASIC 语言。

(13) QBASIC 语言软件只有两个文件，方便上机管理。

总之，QBASIC 语言的这些功能与其他一些高级语言相比并不逊色。它既从其他高级语言中吸取了现代语言的特点，又很好地保持了 BASIC 语言容易学习和使用的传统特点，具有很强的实用价值。

1.3 QBASIC 语言编程环境

1.3.1 运行环境

QBASIC (4.5 版) 是一个用菜单驱动, 集程序的编辑、编译、调试和文件管理于一身的多功能集成编程环境软件。

1. 硬件环境

运行 QBASIC 需要如下的硬件环境:

- (1) 640 KB 以上内存。
- (2) 分辨率为 640 × 350 以上的单色或彩色显示器。
- (3) 如果有一只鼠标 (mouse), 操作起来更为方便。

2. 软件环境

启动 QBASIC 必须有两个文件: QBASIC.EXE (解释程序) 和 QBASIC.HLP (帮助程序)。如果用户在 QBASIC 语言环境中存储了环境配置, 还将自动生成一个配置文件 QBASIC.ini。

QBASIC 是随 MS-DOS 5.0 及以上版本一起发行的, 可以运行在所有直接写屏的汉字系统下。如果用户的机器上安装了 DOS 操作系统, 那么在 DOS 子目录 (一般在 C:\DOS) 下就会有这两个文件; 如果用户的机器中没有这两个文件也没有关系, 只要从其他机器上复制这两个文件到自己的机器中, 或者把它们复制到一张软盘上, 就可以正常运行 QBASIC 了。

1.3.2 启动 QBASIC

进入 QBASIC 语言解释环境的方法如下:

(1) 在 DOS 系统下, QBASIC 语言可以运行在所有直接写屏的汉字系统下, 如 UC DOS、CX DOS 等。在启动中文系统后, 再执行 QBASIC。需要特别注意的是, 在中文系统下, 汉字可以作为字符串使用, 但不能用做变量名。

如果在 Autoexec.bat 批处理文件的环境变量 PATH 中设置了 DOS 文件所在的路径, 那么在任意路径下, 只要在 DOS 的提示符后输入:

```
QBASIC
```

回车后即可进入 QBASIC 语言解释环境中。

(2) 如果使用的是 Windows 系统, 则直接双击 QBASIC.EXE 文件即可进入 QBASIC 的解释环境中。

通过以上任意一种方法启动 QBASIC 成功后, 屏幕上出现如图 1-1 所示的对话框。

屏幕提示信息有两种选择:

< Press Enter to see the QuickBASIC Survival Guide >: 按“Enter”键查看使用指南, 进入联机帮助。

< Press ESC to clear this dialog box >: 按“Esc”键清除本对话框, 进入编辑方式。

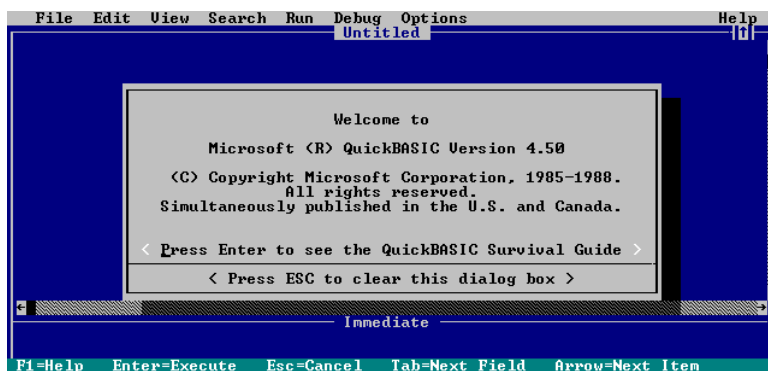


图 1-1 QBASIC 语言的欢迎窗口

1.3.3 QBASIC 语言窗口

1. QBASIC 的工作窗口

现在要做的工作是编辑一个程序，所以按“Esc”键，进入编辑方式，如图 1-2 所示。

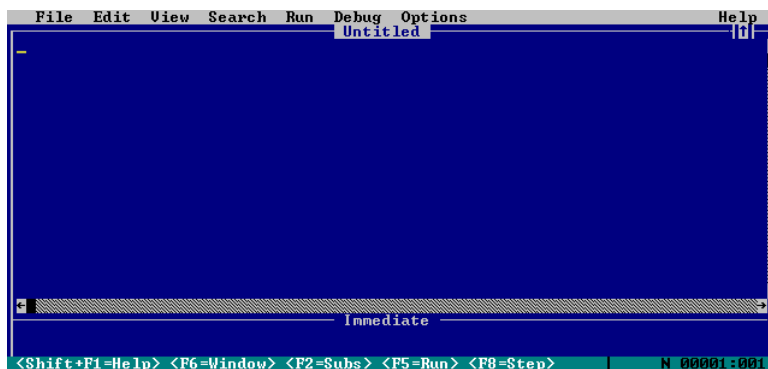


图 1-2 QBASIC 语言的工作窗口

可以看到，QBASIC 语言提供了两个工作窗口。

(1) 上窗口：也叫“View”(观察)窗口或程序窗口，用来输入、编辑程序。此窗口的标题在没有给程序命名并将程序存入磁盘以前总是显示“Untitled”，意为“无标题”，即指新程序还没有命名。若从外存调入一个有文件名的程序，则此处显示该文件名。

(2) 下窗口：也叫“Immediate”(直接)窗口，在将程序语句输入到程序窗口以前，可以在此窗口进行测试。当在此窗口输入一个语句并按回车键后，该语句被立即执行并输出结果。

这两个窗口只能有一个窗口是活动的，其活动窗口的标题是高亮度显示的，用户可以按“F6”键或使用鼠标在两个窗口之间进行切换。

另外，在 QBASIC 工作窗口的最下边是提示栏，它显示了某些功能键的专用功能。

- λ <Shift+F1=Help>：同时按“Shift”和“F1”键进入联机帮助系统，可以得到有关使用 QBASIC 的知识。
- λ <F6=Windows>：按“F6”可以改变活动窗口。
- λ <F2=Subs>：进入子程序。

- λ <F5=Run> : 运行 QBASIC 源程序。
- λ <F8=Step> : 调试源程序时, 进行单步执行。
- λ “ N ”: 表示键盘上的 Num Lock 键已按下。
- λ “ 00001 : 001 ”: 表示当前光标所在位置的行和列。它的值将随光标在窗口中的移动而变化。

2. QBASIC 的菜单

从图 1-2 中可以看到 QBASIC 语言工作窗口的上边有一条菜单栏, 共有 8 个菜单, 下面列出了各菜单的功能说明。

- λ “ File ”: 即文件菜单, 用于处理与文件有关的操作, 如新建、打开、保存等。
- λ “ Edit ”: 即编辑菜单, 用于处理与程序编辑有关的操作, 如剪切、复制等。
- λ “ View ”: 即查看菜单, 用于查看程序的有关部分, 如查看子程序、分割窗口等。
- λ “ Search ”: 即查找菜单, 用于寻找检索所需要的程序内容。
- λ “ Run ”: 即运行菜单, 用于运行程序, 如开始运行。
- λ “ Debug ”: 即调试菜单, 用于调试程序, 如分步运行等。
- λ “ Options ”: 即选择菜单, 用于屏幕的前景、背景颜色等的设置。
- λ “ Help ”: 即帮助菜单, 用于选择帮助信息。

3. QBASIC 菜单操作

QBASIC 提供了集成化的操作环境, 它的大多数功能都可通过屏幕顶行的菜单栏来完成, 当用户打开一个菜单项时, 会弹出一个下拉菜单, 里面放置若干条操作命令。

下面介绍如何打开这些菜单项。

(1) 光标选择法: 按“Alt”键, 菜单栏内第一个菜单项“File”以反白形式出现, 表明菜单栏此时被“激活”。然后按左、右光标键(或)选择用户想要打开的菜单项, 此时该菜单项将以反白形式显示。再按回车键, 就可以打开该菜单项。

(2) 字符键选择法: 按“Alt”键, 激活菜单栏后, 再按用户想要打开的菜单项的第一个高亮度字母(如“File”中的 F, “Run”中的 R 等), 就可以打开该菜单项。

(3) 组合键法: 一手按住“Alt”键, 另一手同时按住用户想要打开的菜单项名字的第一个高亮度字母(如“File”中的 F, “Run”中的 R 等), 就可以打开该菜单项。

(4) 鼠标操作法: 将鼠标指针移到用户想要打开的菜单项上, 单击它就可以打开该菜单项。

当用户打开某一菜单项后, 其下拉菜单中有很多操作命令, 如图 1-3 所示。



图 1-3 打开的下拉菜单

在下拉菜单中的每一条命令中都有一个字母是以高亮度显示的(如“Start”中的 R, “Continue”中的 C 等), 直接按该字母键就可以执行命令(如按“C”键就可以执行“Continue”命令)。另外, 在下拉菜单的一些命令后附上了快捷键(如“Start”命令后的“Shift+F5”等), 用户只需在编辑窗口中按“Shift+F5”就可以直接执行“Start”命令。