

Quick BASIC

语言程序设计

Quick BASIC
Quick BASIC
Quick BASIC
Quick BASIC

王殿玉 编著

印刷工业出版社

Quick BASIC 语言程序设计

王殿玉 编著

印刷工业出版社

内 容 提 要

Quick BASIC 是目前国内外开始流行的一种程序设计语言,它即适用于初学者,也可以用它编写大型计算机软件。本书从基本概念出发,循序渐进的对 Quick BASIC 语言的数据及语句功能做了全面的讲解,配合一些例题介绍了如何利用 Quick BASIC 语言进行程序设计。本书每章都配有适量的例题及习题,这些例题都可以在一般的 IBM-PC 机上运行。

本书适于作高等院校计算机语言课程的教材或参考书,也可供科学技术人员及其他从事计算机程序设计的人员参考。

图书在版编目(CIP)数据

Quick BASIC 语言程序设计/王殿玉编著. —北京:印刷工业出版社, 1995. 8

ISBN 7-80000-198-9

I. Q… II. 王… III. BASIC 语言-程序设计 N. TP312BA

中国版本图书馆 CIP 数据核字(95)第 11976 号

Quick BASIC 语言程序设计

王殿玉 编著

*

印刷工业出版社出版发行

(北京复外翠微路 2 号)

顺义振华印刷厂印刷

各地新华书店经售

*

850×1168mm 1/32 印张:8.875 字数:234 千字

1995 年 8 月第一版第一次印刷 定价:13.50 元

前 言

1987年,Microsoft 公司推出了 Quick BASIC 的 4.0 版本,接着在 1988 年该公司又推出了 Quick BASIC 的 4.5 版本。这两个版本的 Quick BASIC 全面的继承了近些年来 BASIC 的发展,吸收了 True BASIC 及 Borland 公司的 Turbo BASIC 的优点,使其成为一个全新的、功能强的、简单易学的结构化的高级语言。

Quick BASIC 与通常的 BASIC 相比,取消了行号,同时又在数据类型、流程控制、绘图、模块化结构及文件的输入输出等方面都提供了令人满意的功能。它具有高级语言的优点,但没有其他高级语言的诸多限制,使用起来同普通 BASIC 一样方便,十分适合开发各种应用软件。

Quick BASIC 即可以解释运行,又可以编译运行,这给编程带来很大方便。它有高度集成的编程环境,有灵活方便、功能极强的编辑器,编辑时结合编译逐行进行,发现语句错误立即给出信息,同时,它还配有功能性很强的调试手段,这些都是吸引用户的极有力的地方。

Quick BASIC 是目前国内外广泛流行的一种程序设计语言,从 DOS 6.0 版本以后,Microsoft 公司将免费随 DOS 操作系统赠送一个简易型版本的 Quick BASIC。

本书的特点是把多模块的程序设计单独作为一章,并且放到本书的较后的地方讲解,使读者在掌握了单模块的程序设计之后,自然地转入到多模块结构的程序设计,从而避免了一开始就要记住很多多模块程序设计的变量说明,变量传递及多模块设计规则等问题。本书的最后一章讲了 Quick BASIC 的环境,阅读了这一章之后就可以上机编程,而不必去看很多使用手册。

由于本书编写时间仓促,难免有错误之处,恳请广大读者指正。

编者

1994 年

目 录

前言

第一章 计算机系统及程序概念	(1)
1.1 计算机系统	(1)
1.2 源程序的编译和程序的运行	(2)
1.3 计算机中的数制	(3)
1.3.1 数制及数值的表示	(3)
1.3.2 数值在不同数制中的转换	(4)
1.3.3 转换函数	(6)
1.4 标识符和 Quick BASIC 字符集	(6)
1.4.1 QB 字符集	(6)
1.4.2 标识符	(7)
1.5 Quick BASIC 程序结构	(8)
1.6 流程图.....	(10)
1.7 程序行.....	(10)
习题	(12)
第二章 数据和表达式	(13)
2.1 数据类型.....	(13)
2.1.1 数值型数据.....	(13)
2.1.2 字符串类型数据.....	(14)
2.1.3 数组.....	(15)
2.1.4 用户定义类型(记录类型).....	(16)
2.2 常量.....	(16)
2.2.1 直接常量.....	(16)
2.2.2 符号常量.....	(18)
2.3 变量.....	(19)

2.3.1	变量的构成	(19)
2.3.2	变量的类型说明	(19)
2.4	表达式与运算符	(21)
2.4.1	数值表达式	(21)
2.4.2	字符串表达式	(23)
2.4.3	逻辑表达式	(23)
2.5	内部函数	(28)
2.6	表达式的书写格式	(32)
2.7	类型转换	(32)
2.7.1	强制转换	(33)
2.7.2	隐含转换	(34)
2.8	字符串运算	(34)
2.8.1	字符串的合并运算	(34)
2.8.2	字符串的比较运算	(35)
	习题	(36)
第三章	顺序结构程序设计	(38)
3.1	赋值语句	(38)
3.2	注释、暂停与程序结束语句	(39)
3.2.1	注释语句	(39)
3.2.2	暂停语句	(40)
3.2.3	程序结束语句	(40)
3.3	输入语句	(41)
3.3.1	键盘输入语句	(41)
3.3.2	行输入语句	(41)
3.3.3	读数据语句	(42)
3.3.4	恢复读数据语句	(44)
3.4	输出语句	(45)
3.4.1	PRINT 语句	(45)
3.4.2	与 PRINT 语句有关的函数	(46)

3.4.3	WRITE 语句	(48)
3.4.4	带有格式的输出生句	(48)
3.4.5	LPRINT 和 LPRINT USING 语句	(51)
3.5	特殊输入输出操作	(51)
3.5.1	INKEY\$ 函数	(51)
3.5.2	INPUT\$ 函数	(52)
3.5.3	访问内存的语句和函数	(52)
3.6	文本光标控制语句	(53)
3.7	SWAP 语句	(54)
3.8	顺序结构程序设计举例	(54)
	习题	(59)
第四章	分支结构程序设计	(61)
4.1	条件语句	(61)
4.1.1	单行结构条件语句	(61)
4.1.2	块结构条件语句	(63)
4.2	GOTO 语句	(65)
4.2.1	GOTO 语句	(65)
4.2.2	ON...GOTO 语句	(65)
4.3	情况语句	(66)
4.4	分支结构的嵌套	(69)
4.4.1	单行 IF 语句的嵌套	(69)
4.4.2	块 IF...THEN...ELSE 语句的嵌套	(70)
4.4.3	SELECT CASE 语句的嵌套	(71)
4.5	综合举例	(72)
	习题	(75)
第五章	循环结构程序设计	(78)
5.1	FOR-NEXT 循环	(78)
5.2	WHILE-WEND 循环	(83)
5.3	DO-LOOP 循环	(85)

5.4	多重循环	(90)
-----	------	------

5.5	程序综合示例	(92)
-----	--------	------

	习题	(96)
--	----	------

第六章	数组	(98)
------------	-----------	------

6.1	数组的概念	(98)
-----	-------	------

6.2	数组变量的说明	(98)
-----	---------	------

6.2.1	DIM 说明数组语句	(98)
-------	------------	------

6.2.2	测试数组下标上、下界的函数	(101)
-------	---------------	-------

6.3	静态数组与动态数组	(102)
-----	-----------	-------

6.3.1	静态数组	(102)
-------	------	-------

6.3.2	动态数组	(103)
-------	------	-------

6.4	数组删除语句	(103)
-----	--------	-------

6.5	数组重新定义语句	(104)
-----	----------	-------

6.6	数组的基本操作	(106)
-----	---------	-------

6.6.1	数组的输入与输出	(107)
-------	----------	-------

6.6.2	数组元素的复制	(108)
-------	---------	-------

6.7	程序综合举例	(109)
-----	--------	-------

	习题	(120)
--	----	-------

第七章	子程序	(123)
------------	------------	-------

7.1	过程的定义和调用	(123)
-----	----------	-------

7.1.1	函数过程的定义和调用	(124)
-------	------------	-------

7.1.2	SUB 过程的定义和调用	(128)
-------	--------------	-------

7.2	参数的传递	(130)
-----	-------	-------

7.3	过程的说明	(135)
-----	-------	-------

7.4	变量作用域	(136)
-----	-------	-------

7.4.1	全程变量	(136)
-------	------	-------

7.4.2	局部变量	(137)
-------	------	-------

7.4.3	共享变量	(141)
-------	------	-------

7.5	过程的编辑及显示	(142)
-----	----------	-------

7.5.1	过程的编辑	(142)
7.5.2	过程的显示	(143)
7.6	递归	(144)
7.7	GOSUB 子程序和自定义函数	(146)
7.7.1	GOSUB 子程序	(146)
7.7.2	自定义函数	(147)
7.7.3	SUB 过程、FUNCTION 过程与 GOSUB 子程序和 DEF FN 函数的比较	(149)
7.8	程序举例	(150)
	习题	(161)
第八章	多模块结构程序设计	(163)
8.1	多模块中共享变量的说明及其传递	(163)
8.1.1	模块间共享变量	(163)
8.1.2	模块间共享变量的传递	(164)
8.2	蕴含文件的使用	(167)
8.3	多模块程序的运行	(171)
8.3.1	控制始终在主模块的程序运行	(171)
8.3.2	在各模块间转换控制的程序的运行	(171)
8.4	程序举例	(173)
第九章	数据文件	(176)
9.1	文件的打开和关闭	(176)
9.1.1	文件的打开	(176)
9.1.2	文件的关闭	(180)
9.1.3	设备文件的打开和关闭	(181)
9.2	文件的操作语句和函数	(181)
9.2.1	文件操作的相关语句	(181)
9.2.2	文件操作的相关函数	(183)
9.3	顺序文件	(185)
9.3.1	顺序文件的写操作	(186)

9.3.2	顺序文件的读操作	(188)
9.3.3	顺序文件的使用和维护	(189)
9.4	随机文件	(195)
9.4.1	随机文件的读写语句	(196)
9.4.2	随机文件中记录变量的构成及记录长度的 计算	(196)
9.4.3	随机文件的建立与读取	(197)
9.5	二进制文件	(199)
9.5.1	二进制文件特点	(199)
9.5.2	二进制文件的读写语句	(200)
	习题	(202)

第十章 图形..... (203)

✓ 10.1	参数设置语句.....	(203)
10.1.1	SCREEN 语句	(203)
10.1.2	SCREEN 函数	(205)
10.1.3	视见区语句.....	(206)
10.1.4	窗口语句.....	(207)
10.1.5	文本视区语句.....	(209)
10.1.6	清屏语句.....	(210)
10.1.7	屏幕颜色设置语句.....	(210)
10.1.8	屏幕颜色点测试函数和屏幕坐标测试 函数.....	(212)
10.2	绘图语句.....	(212)
10.2.1	画点语句.....	(213)
10.2.2	画线语句.....	(214)
10.2.3	连续画线语句.....	(216)
10.2.4	CIRCLE 语句	(218)
10.2.5	填充颜色语句.....	(220)
10.3	动画功能.....	(222)

10.3.1 图形动画	(222)
10.3.2 快速动画	(224)
第十一章 Quick BASIC 编程环境	(231)
11.1 QB 的启动与退出	(231)
11.1.1 Quick BASIC 的启动	(231)
11.1.2 Quick BASIC 屏幕	(232)
11.1.3 Quick BASIC 的退出	(234)
11.2 菜单及其命令选择	(235)
11.3 对话框的使用	(237)
11.4 窗口管理	(237)
11.4.1 观察窗口	(238)
11.4.2 直接窗口	(239)
11.5 文本编辑	(240)
11.5.1 文本输入	(240)
11.5.2 编辑操作及命令	(241)
11.6 源文件管理	(244)
11.6.1 程序管理	(244)
11.6.2 模块管理	(247)
11.7 Quick BASIC 程序的运行	(248)
11.7.1 在 Quick BASIC 环境下运行 Quick BASIC 程序	(248)
11.7.2 在 DOS 环境下运行 Quick BASIC 程序	(249)
附录 A: ASCII 字符代码	(256)
附录 B: QB 保留字	(258)
附录 C: QB 数据类型	(260)
附录 D: QB 运算符及优先级	(260)
附录 E: QB 语句一览表	(261)
附录 F: Quick BASIC 键盘命令一览表	(267)
附录 G: Quick BASIC 内部函数一览表	(269)

第一章 计算机系统及程序概念

1.1 计算机系统

计算机系统包括系统硬件和系统软件。

系统硬件是计算机的电子和机械设备,而系统软件则指计算机系统程序和应用程序。硬件与软件的适当配合,则产生出计算机的生命,二者缺一计算机都不能工作。

计算机的主要系统硬件包括:中央处理机、主存储器,外存储器及输入输出设备,如图 1.1 所示。

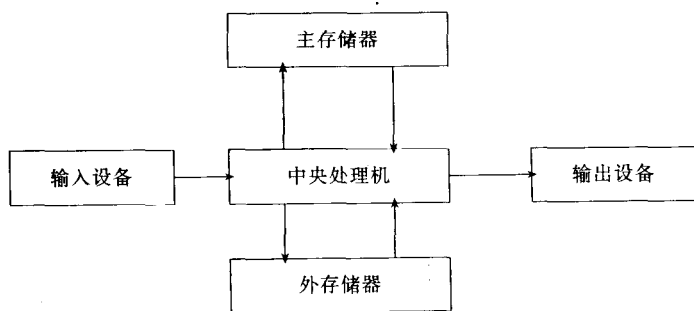


图1.1 计算机硬件组成

中央处理机由运算器和控制器组成,控制器从存储器中取出指令,经解释后,指挥计算机相应的部分按一定顺序执行这些指令。运算器就是算术逻辑单元,作为控制对象。控制器提供给运算器适当的数据,令运算器完成指定的运算,其中包括算术运算和逻辑运算。

主存储器有时也称为内存储器。主存储器用来存储常驻内存

的信息和当前正在被处理的信息和数据,它的存取速度快,但是,相对于外存储器来讲存储容量小。存储容量以字节(Byte)为单位,1个字节即为8个二进制位,即 $1 \text{ Byte} = 8 \text{ bit}$ 。主存储器的容量一般从几百 KBytes 到几十 MBytes,即由几百千字节到几兆甚至几十兆字节。

外存储器通常是由磁带、磁盘、光盘等存储介质加上读写装置组成。外存储器的存取速度慢,但存储容量非常大,其容量可以从几十兆字节到几千兆字节,所以外存储器主要用来存储暂时不使用的数据和文件。

另外,计算机中还有一个术语,即字长,字长的单位不是字节,它是依不同的机器所表示的数的大小而定。如果某种微机中能表示的数为 $0 \sim 255$,则该机的字长为8位,即8位机,如机器表示的数大,则它有更大的字长,如16位机、32位机,甚至64位机等。

输入设备向计算机输入程序、数据或命令,主要有键盘、磁带机、读卡机等。

输出设备是由计算机向外输出信息,包括计算结果和出错信息等。常用的输出设备有计算机终端显示器(有时叫监视器),各种打印机等。

系统的软件包括操作系统及其所支持的一些实用程序和各种语言编译程序。其作用是统一管理计算机系统的硬件和软件资源,包括输入输出设备的管理、存储器管理、文件管理、作业调度及编译源程序等。它是用户与计算机硬件之间的介面,是整个计算机系统指挥部分。所以,只有硬件而无软件的计算机称作裸机是不能有效工作的。

1.2 源程序的编译和程序的运行

程序实际上就是一系列的指令,每个指令告诉计算机去执行什么操作,如加、减、乘、除等操作。由人所编写的程序称作源程序或称源代码,它是依某种计算机语言规则写成的。对于源程序,计

计算机是不认识也不能执行的。计算机能执行的是可执行程序,它是由源程序经过编译和连接而成。源程序经编译得到目标程序,目标程序和可执行程序都是机器代码,即由(0、1)两个数码构成的代码。源程序也称源代码或源码。把源代码翻译成机器代码的过程称作编译,编译的结果是得到目标代码。编译过程在计算机上称作(Compilation),这个工作由一个称作编译器(Compiler)的程序来完成。不同的计算机语言配有不同的编译器。在编译的过程中,同时还要对源程序作语法检查,对于不符合语法的句子给出的错误信息。

解释器:把源代码翻译成机器代码的另一种程序叫作解释器。它的功能与编译器一样,所不同的是它仅对源程序作一行一行的翻译,而不是像编译器那样把源程序从头到尾一次翻译完。解释器在翻译源程序时,边翻译边执行,边解释边执行,解释完一行,接着就执行这一行,执行完了再解释下一行,再执行下一行,直至终了。早期的 BASIC 只有解释器。Quick BASIC 配有两种翻译程序,编译器和解释器。

编译器与解释器各有优点,编译器对源程序编译完成后,形成目标文件,再经连接便得到可执行程序,可执行程序可以在 QB 环境下运行,也可以在 DOS 环境下在任何时候运行。而解释器的优点在于可以使用户快速嵌入文件,并帮助用户找出程序中的错误。比较来说,编译器要比解释器先进得多。

1.3 计算机中的数制

1.3.1 数制及数值的表示

数制:即记数方法。

十进制是逢十进一,二进制是逢二进一,而十六进制是逢十六进一,分与秒是六十进一。计算机中常用的数制为十进制、二进制、十六进制、八进制等。

数值的表示:由各位数字的加权和表示。如十进制数

3	0	5	2
10^3	10^2	10^1	10^0

$$3 \times 10^3 + 0 \times 10^2 + 5 \times 10^1 + 2 \times 10^0 = 3052$$

其中 10 为十进制的基数。而 $10^0, 10^1, 10^2, 10^3$ 分别表示从右往左各位的权值。把每位数字与相应的权值相乘, 再把各位乘积相加即为十进制数的值。

对于二进制数, 计算数值与十进制相同, 但各位权值不同。如二进制数

1	0	1	1
2^3	2^2	2^1	2^0

从右往左各位二进制数的权值为 1, 2, 4, 8, 16 等。上边的二进制数 $(1011)_2$ 的值为 $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (11)_{10}$

其数字括号外的下标表示数制。

对于十六进制, 基数为 16, 每位的数字符号有 16 个, 分别是 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。其值仍是各位数字的加权和。如 16 进制

3	A	F
16^2	16^1	16^0

其 16 进制的各位权值, 从右往左分别是 $16^0, 16^1, 16^2 \dots$ 。上面的 16 进制数的值为:

$$(3AF)_{16} = 3 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = (943)_{10}$$

八进制的基数为 8, 每位的字符为 0~7。

1.3.2 数值在不同数制中的转换

一个数值可以用不同的数制表示, 如数 $(79)_{10} = (1001111)_2 = (257)_8 = (4F)_{16}$

一个数可以从一种记数制中的表示转换为另外一种记数制中表示。常用的做法是将 10 进制数转换为其他进制的数。具体做法是连除取余法。即将十进制数连除要转换数制的基数, 每次记下余数, 直到商为 0 止, 把每次的余数以逆序排列, 即为要转换的数值。

如上例中的十进制值 79, 转换为二进制是下面的过程

2	79	余(1)
2	39	余(1)
2	19	余(1)
2	9	余(1)
2	4	余(0)
2	2	余(0)
2	1	余(1)
0		

把余数逆序排列为 $(1001111)_2 = (79)_{10}$ 。

$(79)_{10}$ 转换成 8 进制

8	79	... (7)	$(117)_8 = (79)_{10}$
8	9	... (1)	
8	1	... (1)	
0			

$(79)_{10}$ 转换成 16 进制

16	79	... 15	$(79)_{10} = (4F)_{16}$
16	4	... 4	
0			

二进制数转换成 16 进制数较容易, 把二进制数从右边开始每四位作为一个 16 进制位, 再把每个四位二进制值换成一位 16 进