

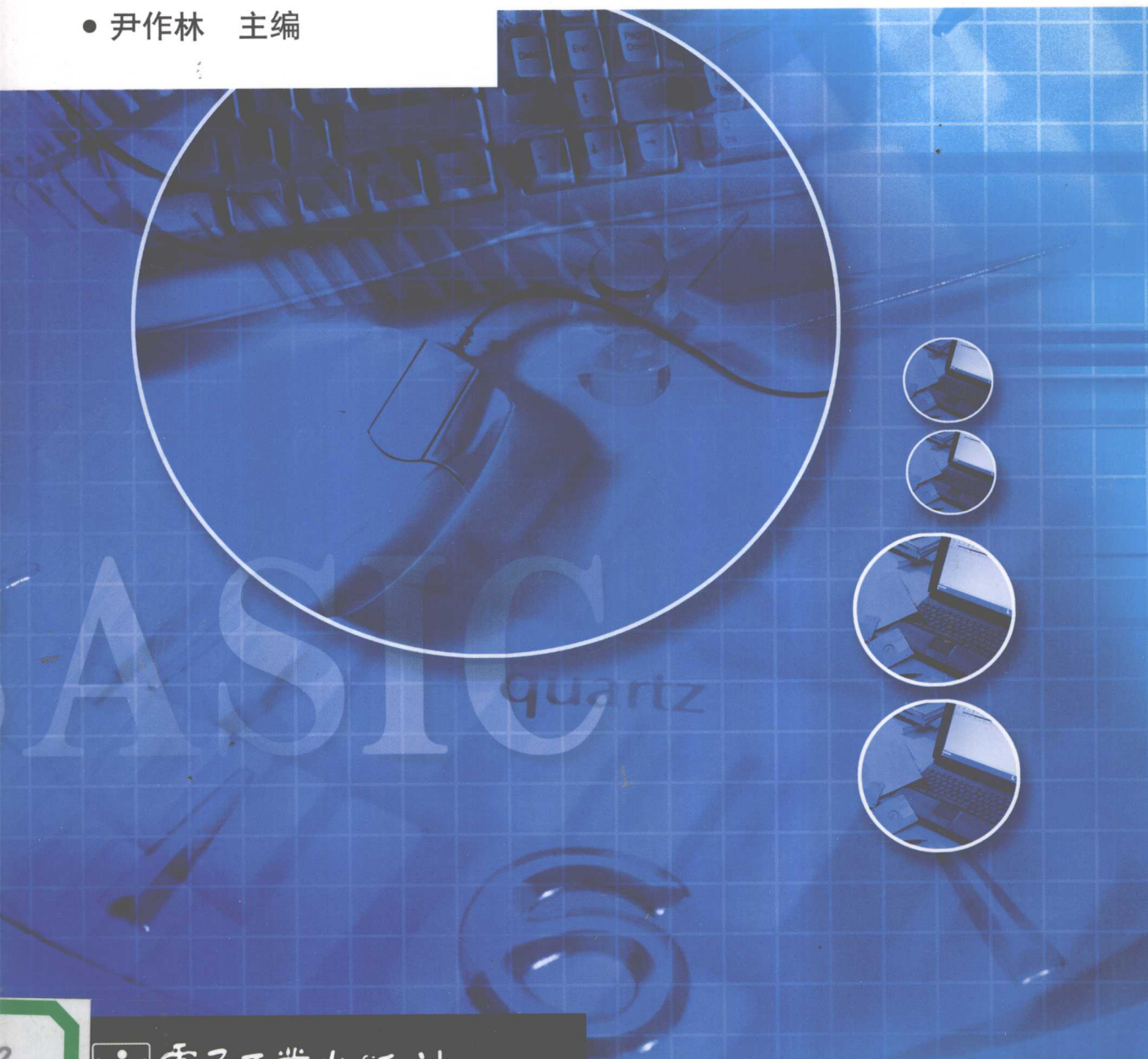
高等职业教育电子信息类贯通制教材

· 计算机技术专业



QBASIC语言程序设计

• 尹作林 主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

TP312
Y606:1

教育电子信息类贯通制教材（计算机技术专业）

QBASIC 语言程序设计

尹作林 主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书共 10 章, 前 8 章主要介绍了 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 语言的屏幕控制语句和作图语句, 以增加教材使用的弹性; 第 10 章介绍了 QBASIC 语言的程序文件和数据文件的建立及读写语句。

本书可作为高等职业教育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。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

QBASIC 语言程序设计/尹作林主编. —北京: 电子工业出版社, 2004.2

高等职业教育电子信息类贯通制教材·计算机技术专业

ISBN 7-5053-9605-6

I. Q… II. 尹… III. BASIC 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 002778 号

责任编辑: 朱怀永

印 刷: 北京四季青印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 11.25 字数: 288 千字

印 次: 2004 年 2 月第 1 次印刷

印 数: 6 000 册 定价: 15.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。
联系电话: (010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。



近年来,信息产业已快速发展成为中国第一大产业,每年平均以 30%左右的增长速度飞速发展。作为高新技术产业的信息产业,从产业岗位群的需求来看,需要大量的技能型、应用型人才。而技能型、应用型人才最适合高等职业技术教育来培养。

高等职业技术教育的培养目标是:“培养高素质技术应用型人才”。由于培养目标的定位,高等职业教育的教学模式就是要注重学生职业技能的训练和职业岗位能力的培养。

QBASIC 语言程序设计是高等职业教育计算机技术类专业学生必修的一门基础专业课程,也是其他各类专业学生学习使用计算机解决实际问题的一门重要实用技术课程。通过本课程的学习使学生掌握计算机编程语言——QBASIC 语言的基本知识,掌握利用 QBASIC 语言进行程序设计的基本思想和方法,锻炼学生应用 QBASIC 语言进行程序设计的能力,为学生学习计算机技术类专业知识和职业技能以及学习其他计算机技术课程打下坚实基础,进一步锻炼学生运用计算机解决一些实际问题的能力,达到培养高素质计算机应用型人才的基本要求。

本书的每章开始部分有本章要点,结束部分有本章小结、习题以及上机操作等内容,这样形成完整的教材结构体系,同时也更好地帮助学生掌握和巩固所学知识。

教学安排参考学时为 72 学时(见下表)。

序号	课 程 内 容	学时数			
		合计	讲授	上机操作	机动
1	概述	4	2		
2	QBASIC 语言程序设计的基础知识	6	4	2	
3	顺序结构语句	10	4	2	
4	分支结构语句	14	6	2	
5	循环结构语句	8	6	4	2
6	数组	12	4	2	
7	模块化结构程序	6	6	4	
8	字符串操作	6	4	2	2
9	*屏幕控制与作图	6	4	2	
10	文件	8	4	2	2
11	总计	72	44	22	6

本书由内蒙古电子信息职业技术学院尹作林老师主编,安徽电子信息职业技术学院苏传芳老师、河北商贸学校尚艳钦老师、内蒙古经贸学校马艳秋老师参加了编写。

本书还配有数学指南、电子教案及习题答案(电子版),有需要的教师请与电子工业出版社

目 录



第 1 章 概述	(1)
1.1 程序及程序设计语言简介	(1)
1.2 QBASIC 语言的发展及特点	(2)
1.2.1 QBASIC 语言的发展	(2)
1.2.2 QBASIC 语言的特点	(4)
1.3 QBASIC 语言的简单程序结构	(5)
1.4 QBASIC 程序的上机调试和运行	(6)
1.4.1 QBASIC 语言的进入	(6)
1.4.2 QBASIC 语言的工作窗口	(7)
1.4.3 QBASIC 语言的菜单功能	(8)
1.4.4 QBASIC 程序的输入、编辑和运行	(8)
本章小结	(11)
习题 1	(11)
第 2 章 QBASIC 语言程序设计的基础知识	(12)
2.1 QBASIC 语言使用的各种数据类型	(12)
2.2 常量和变量	(13)
2.2.1 常量	(13)
2.2.2 变量	(16)
2.3 标准函数	(19)
2.4 运算符和表达式	(20)
2.4.1 算术运算符和算术表达式	(20)
2.4.2 关系运算符和关系表达式	(21)
2.4.3 逻辑运算符和逻辑表达式	(22)
2.4.4 不同类型运算符的优先次序	(22)
本章小结	(22)
上机操作 1 QBASIC 语言的基本操作练习	(23)
习题 2	(23)
第 3 章 顺序结构语句	(25)
3.1 赋值语句 (LET 语句)	(25)
3.1.1 赋值语句的一般格式	(25)
3.1.2 使用赋值语句注意的地方	(25)

3.2 输出语句 (PRINT 语句)	(26)
3.2.1 PRINT 语句	(26)
3.2.2 LPRINT 语句	(30)
3.3 暂停语句 (STOP 语句) 和结束语句 (END 语句)	(31)
3.3.1 STOP 语句	(31)
3.3.2 END 语句	(31)
3.3.3 注释语句 (REM 语句)	(31)
3.4 输入语句	(32)
3.4.1 键盘输入数据语句 (INPUT 语句)	(32)
3.4.2 读数据语句 (READ 语句) 和置数据语句 (DATA 语句)	(33)
3.4.3 恢复数据区指针语句 (RESTORE 语句)	(35)
3.5 交换变量数据语句 (SWAP 语句)	(35)
3.6 顺序结构程序举例	(36)
本章小结	(38)
上机操作 2 顺序结构语句的使用	(38)
习题 3	(39)
第 4 章 分支结构语句	(42)
4.1 分支语句 (IF 语句)	(42)
4.1.1 行 IF 语句	(42)
4.1.2 行 IF 语句的嵌套	(43)
4.2 块 IF 语句	(44)
4.2.1 块 IF 语句	(44)
4.2.2 块 IF 语句的嵌套	(45)
4.3 多分支结构语句 (SELECT CASE 语句)	(46)
4.3.1 多分支结构语句	(46)
4.3.2 多分支转移语句 (ON GOTO 语句)	(48)
4.4 无条件转移语句 (GOTO 语句)	(48)
4.5 分支结构语句的应用举例	(49)
本章小结	(51)
上机操作 3 分支结构语句的使用	(52)
习题 4	(52)
第 5 章 循环结构语句	(56)
5.1 FOR—NEXT 循环语句	(56)
5.2 WHILE 循环语句	(63)
5.3 DO 循环语句	(65)
5.4 循环语句的嵌套结构	(69)
5.5 循环结构的应用举例	(71)
本章小结	(73)
上机操作 4 FOR—NEXT 循环语句的使用	(74)
上机操作 5 WHILE 循环语句的使用	(75)

习题 5	(76)
第 6 章 数组	(80)
6.1 数组及其建立	(80)
6.1.1 数组简介	(81)
6.1.2 数组的建立	(82)
6.1.3 数组的使用说明	(84)
6.2 一维数组和二维数组	(85)
6.2.1 一维数组	(85)
6.2.2 二维数组	(86)
6.3 静态数组和动态数组	(88)
6.3.1 释放数组的内存空间	(88)
6.3.2 重新定义数组	(89)
6.3.3 求数组下标界限值的函数	(90)
6.4 数组的应用举例	(91)
本章小结	(97)
上机操作 6 数组的定义和使用	(97)
习题 6	(98)
第 7 章 模块化结构程序	(103)
7.1 模块化结构概述	(103)
7.2 自定义函数的定义和使用	(104)
7.2.1 单行自定义函数	(104)
7.2.2 多行自定义函数	(106)
7.3 子程序的定义和使用	(107)
7.3.1 块内子程序 GOSUB 语句	(108)
7.3.2 独立模块的子程序 SUB 语句	(111)
7.4 模块化函数的定义和调用	(116)
7.4.1 模块化函数的定义和调用	(116)
7.4.2 模块化函数的应用举例	(117)
7.5 全局变量与局部变量	(118)
7.5.1 局部变量	(118)
7.5.2 全局变量	(119)
7.6 函数和子程序的嵌套和递归调用	(120)
7.6.1 过程的嵌套调用	(120)
7.6.2 过程的递归调用	(121)
本章小结	(123)
上机操作 7 自定义函数和子程序的定义使用	(124)
上机操作 8 模块化函数的应用	(124)
习题 7	(125)
第 8 章 字符串操作	(129)

8.1 字符串常量和变量	(129)
8.1.1 字符串常量	(129)
8.1.2 字符串变量	(130)
8.2 字符串的基本操作	(131)
8.2.1 字符串变量的赋值	(131)
8.2.2 求字符串长度	(132)
8.2.3 字符串的合并	(132)
8.2.4 字符串的比较	(132)
8.2.5 求子字符串	(133)
8.3 字符串其他函数	(134)
8.3.1 键盘读字符函数	(134)
8.3.2 字符与 ASCII 码的转换函数	(134)
8.3.3 字符串与数值的转换函数	(135)
8.3.4 大小写字母的转换函数	(135)
8.3.5 求子字符串序号的函数	(135)
8.3.6 日期和时间函数	(135)
8.4 字符串操作的应用举例	(135)
本章小结	(137)
上机操作 9 字符串的基本操作	(138)
习题 8	(139)
* 第 9 章 屏幕控制与作图	(141)
9.1 屏幕控制语句	(141)
9.1.1 LOCATE 语句	(141)
9.1.2 WIDTH 语句	(141)
9.1.3 CLS 语句	(141)
9.1.4 CSRLIN 和 POS 函数	(141)
9.2 显示模式的控制语句 (SCREEN 语句)	(142)
9.3 颜色的设置语句	(142)
9.3.1 文本模式下的 COLOR 语句	(142)
9.3.2 图形模式下的 COLOR 语句	(143)
9.4 作图语句	(143)
9.4.1 画点语句 (PSET 和 PRESET 语句)	(143)
9.4.2 画线段、矩形语句 (LINE)	(144)
9.4.3 画圆语句 (CIRCLE)	(144)
9.4.4 连续画线语句 (DRAW 语句)	(147)
9.5 图形的着色语句 (PAINT 语句)	(148)
9.6 应用举例	(149)
本章小结	(151)
上机操作 10 作图语句的使用	(151)
习题 9	(152)

第 10 章 文件	(153)
10.1 文件及文件的基本操作	(153)
10.1.1 文件的类型	(153)
10.1.2 文件的说明	(154)
10.1.3 文件的打开与关闭	(154)
10.2 顺序文件的操作	(155)
10.2.1 顺序文件的建立和写操作	(155)
10.2.2 顺序文件的读操作	(156)
10.3 随机文件的操作	(157)
10.3.1 记录型变量	(158)
10.3.2 随机文件的建立和写操作	(158)
10.3.3 随机文件的读操作	(159)
本章小结	(161)
上机操作 11 文件的基本操作	(161)
习题 10	(161)
附录 A QBASIC 的保留字总表	(163)
附录 B 常用字符及其 ASC II 码对应表	(165)
附录 C QBASIC 标准函数一览表	(167)
参考文献	(168)

第1章 概述

本章要点：本章首先向同学们简要介绍程序及程序设计语言，目的是要求学生了解计算机语言的发展历史，明白掌握一门计算机程序设计语言基础的重要性。

本章还向同学们介绍了 QBASIC 语言的发展简史，向大家清楚地说明了掌握 QBASIC 语言是进行计算机程序设计的基础。熟练掌握 QBASIC 语言后，同学们可以轻松地学习和掌握其他高级语言。

通过对本章的学习，要求重点掌握 QBASIC 语言的基本概念、QBASIC 语句的基本书写规则等。特别是要求同学们尽快熟悉 QBASIC 语言工作环境，掌握 QBASIC 语言窗口中各菜单的功能。

对 QBASIC 语言的简单程序结构做一般了解，以便为后面各章的学习打下一个良好的基础。

1.1 程序及程序设计语言简介

计算机从诞生之日起就有自己的专用“语言”。不过，计算机的“语言”是人们为了控制计算机正确工作而专门为它设计的人为语言——“形式语言”。计算机语言，实际上是“电脑”与“人脑”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，是当今人类社会新文化——“计算机文化”的精神文明基础。

向计算机提供有关的控制命令信息，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）。而指令系统是要用某种语言来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。用以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换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成程序，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，编程语言的产生和发展也经历了由简单到复杂、从低级到高级的发展过程。程序设计语言经历了机器语言、汇编语言和高级语言三个发展阶段。

1. 机器语言

计算机刚出现时，使用的是机器语言，通过用二进制代码和计算机的指令符号来编写程序。这种程序虽然计算机能够理解并直接执行，但对初学者来说不够直观，很难接受，出现错误也难检查和修改，而且不便于交流。所以机器语言的应用有很大的局限性。



2. 汇编语言

人们用一些符号来代替每一个具体的指令，即符号语言，也称“汇编语言”。其实，计算机并不能识别汇编语言，要求将汇编语言翻译成计算机可识别的语言，即由“汇编程序”将汇编语言翻译成机器语言，然后让计算机执行。机器语言和汇编语言都与具体的计算机有关，因而称为面向机器的语言。

3. 高级语言

人们希望有一种与具体的机器无关的语言，最好是接近于数学公式的描述。这就出现了各种编程语言，也称为“高级语言”。高级语言编写的程序易读、易修改，但要让计算机执行，必须通过“编译程序”翻译成机器语言。如 PASCAL 语言、FORTRAN 语言和 C 语言都必须由“编译程序”翻译后才能被计算机执行，而 BASIC 语言通常靠“解释程序”边解释边由计算机执行。

计算机通过程序指令来自动执行，而程序要靠程序设计人员利用编程语言来编写和调试。程序设计人员根据实际任务提出问题，然后确定计算方法，构造数学模型，画出流程图，选择合适的编程语言，编写程序并上机反复调试，直到调试结果正确为止。这就是利用编程语言进行程序设计的过程。

本书所介绍的 QBASIC 语言是一种计算机的高级语言。该语言是用英文单词、数学表达式等易于理解的符号，并按严格的语法规则和一定的逻辑关系组合的计算机语言。QBASIC 语言编写的程序独立于机型、可读性好、易于维护，程序设计的效率较高。

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 Thomos·E·Kurtz 开发的 BASIC 语言是计算机语言发展史上的一件大事，它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”的特点，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，乃至从美国传播到全世界，成为初学者学习计算机的首选语言。三十多年来，BASIC 语言经过了以下几个发展阶段。

1. 初期的 BASIC

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，也能用于解决小型的问题。美国标准化协会 ANSI (American National Standard Institute) 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。1975 年美国的比尔·盖茨与他的伙伴在第一台微型机上将编译型 BASIC 语言压缩成只有 4KB 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, 能处理文件, 具有绘图和音乐功能。BASIC 语言不仅以它的优美的语法和容易学习的特点得以广泛流行, 而且它的实用性使它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 BASICA)。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 年代的计算机普及就是以 BASIC 语言为主要内容的。

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。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, 它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、编程和调试简单等优点, 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。它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 BASICA 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了,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它的环境下运行,即所谓“for Windows”的软件。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于 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 的组成部分提供的,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。QBASIC 语言是一种完全结构化和模块化的语言。

QBASIC 语言主要具有以下优点:

- 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 160KB(代码和数据总长),而 GW BASIC 和 BASICA 程序最大长度只有 64KB。
- 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化的调试工具。
- 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

没有行号或标号,行号或标号只是一个地点标记,但与程序的执行顺序无关。行号或标号的作用只是便于程序流程可以转移到该地点继续执行。用 GW BASIC 或 BASICA 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专业 BASIC 语言,如 Quick BASIC, Visual BASIC。

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现代语言相比并不逊色。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 语言的简单程序结构

为了说明 QBASIC 语言的程序结构,我们先看两个简单的 QBASIC 程序的例子。

【例 1.1】 假如用 a, p, t 分别表示银行存款的本金、利率和年限,求本息之和(按单利计算)。

程序如下:

```
REM sum = ?
INPUT "a, p, t=" a, p, t
LET sum = a + a * p * t
PRINT sum
END
```

这是一个简单的 QBASIC 程序,只有一个称为“主程序”的程序段。在输入源程序时, QBASIC 系统自动将程序格式做规范处理,包括:

- ① 将语句的关键字(语句定义符、函数名等)全部改为大写。
- ② 在赋值语句中赋值号(=)及运算代号(+, -, *, / 等)两侧各留出一个空格。
- ③ 在每一个语句行的行号后留一个空格,如一个语句行有多个语句,则在分隔语句的冒号后面留一个空格。

以上操作是由 QBASIC 系统自动进行的,也就是说,用户在输入某一行源程序时不用顾及这些格式是否规范,系统会自动地将其按上述规范格式处理。

【例 1.2】 编程计算 $\text{sum}=4!+5!+6!+7!$ 。

程序如下:

```
DECLARE SUB f(s!, n!)
CLS
FOR i=4 TO 7
    CALL f(s, i)
    sum = sum + s
NEXT i
PRINT "sum ="; sum
```



```
END
```

```
SUB f (s, n)
```

```
Fact = 1
```

```
FOR j = 1 to n
```

```
fact = fact * j
```

```
NEXT j
```

```
s = fact
```

```
END SUB
```

这个程序由两个部分组成，END 语句以上为主程序模块，以下为子程序模块。

对 QBASIC 的源程序的结构作以下几点说明：

① 一个 QBASIC 源程序由一个或多个模块组成。

② 在一个源程序中，必须有且只能有一个主程序模块，可以有一个或多个子程序模块，也可以没有子程序模块。

③ 一个程序模块由一系列语句行（程序行）组成。

④ 一个语句行可以由一个或多个语句构成。当在同一个语句行中有一个以上的语句时，语句之间必须用冒号（:）隔开。同一程序行的语句数量不限，但是语句的总字符数量不得超过 255 个。

⑤ 语句行可以有行号，也可以没有行号。行号为纯数字，由 1~40 个数字组成，行号后面不加冒号。标号可以是数字、英文字母或二者的组合，但是标号后面必须加一个冒号（本书后面用到行号或标号时会有详细的介绍）。行号或标号的位置必须写在每一行的最左端，而不能出现在一行中两个语句的中间。

⑥ 解释语句也是可选的，它的功能是对语句行（程序行）或程序段的功能、意义进行解释，以增加程序的可读性。

⑦ 一个语句一般包括两部分：语句定义符和语句体。语句定义符是用来标识该语句的性质，而语句体是用来表示该语句执行的对象。个别语句没有语句体，例如 END 和 CLS 语句。

⑧ 一个程序可以有多个 END 语句，但每一个模块必须有一个结束语句。如果程序中只有一个 END 语句，应该放在整个程序的最后。主程序模块在 END 语句处结束，子程序模块在 END SUB 或 END FUNCTION 语句处结束。

1.4 QBASIC 程序的上机调试和运行

本节介绍的是简单的 QBASIC 语言程序的上机调试和运行方法。

1.4.1 QBASIC 语言的进入

QBASIC 语言是随 MS-DOS 5.0 或 MS-DOS 6.x 一起发行的，如果用户的微机上已经安装了以上 DOS 版本，在 DOS 子目录下，就会有下面两个文件。

qbasic.exe

qbasic.hlp

qbasic.exe 是 QBASIC 语言的解释程序，qbasic.hlp 是与解释程序联合使用的在线帮助文



件,这两个文件必须同时存在。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。

另外,如果用户使用的 DOS 版本不是 MS-DOS 5.0 或 MS-DOS 6.x,只需将 qbasic.exe 和 qbasic.hlp 两个文件复制到用户的系统中, QBASIC 语言就可以在软盘或硬盘上正常执行了。

QBASIC 语言可以运行在所有直接写屏的汉字系统下,如 UC DOS。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 批文件的环境变量 PATH 中设置了 DOS 文件所在的路径,那么在任何路径下,只要在 DOS 的提示符后输入:

qbasic

回车后则进入 QBASIC 语言解释环境中。

屏幕上有“Welcome to MS-DOS QBasic”及“Copyright (c) Microsoft Corporation,1987-1992.All rights reserved.”等字样。同时提示用户有两个选择:

① Press Enter to see the Survival Guide, 提示用户按回车键可调用联机帮助文件。

② <Press ESC to clear this dialog box>, 即按 ESC 键,得到一个工作窗口。

1.4.2 QBASIC 语言的工作窗口

进入 QBASIC 语言后,此时如按回车键,就进入联机帮助屏幕;按 ESC 键,就进入一个空白的工作窗口,显示的工作窗口如图 1.1 所示。

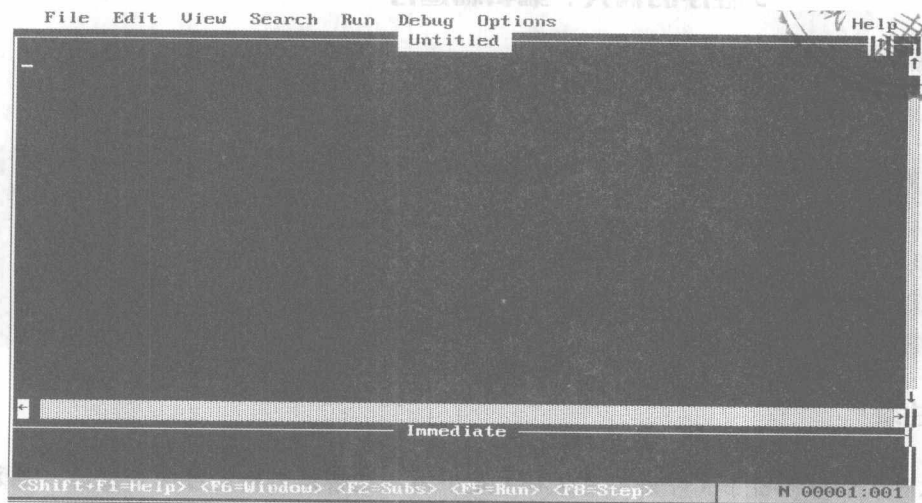


图 1.1 QBASIC 语言的工作窗口

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,此语句被立即执行并输出结果。