

计·算·机·等·级·考·试·丛·书

98 大纲

谭浩强 主编

谭浩强 马素霞 殷光复 编著

QBASIC 程序设计(二级)教程



特

邀计算机教育专家编写



有

配套辅导用书和样题汇编



用

于考试前系统的学习与培训

计算机等级考试
从书



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



(京)新登字 158 号

内 容 简 介

本书是根据教育部考试中心 1998 年公布的《全国计算机等级考试 考试大纲》中关于 QBASIC 语言程序设计的考试要求而编写的。内容包括:QBASIC 语言的基本概念、QBASIC 语言的语法规则以及利用 QBASIC 语言进行程序设计的有关知识。本书覆盖了考试大纲所要求的范围,系统而简明地介绍了考生应该掌握的内容。每章之后均附有精选的练习题,以帮助读者学习和掌握。

本书可以作为参加计算机等级考试人员的自学或培训教材,也可以作为大专院校 QBASIC 语言程序设计课程的教学用书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: QBASIC 程序设计(二级)教程

作 者: 谭浩强 马素霞 殷光复 编著

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市清华园胶印厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 21.5 **字数:** 497 千字

版 次: 2000 年 9 月第 1 版 2001 年 7 月第 3 次印刷

书 号: ISBN 7-302-00807-8/TP·291

印 数: 10001~14000

定 价: 26.00 元

计算机等级考试丛书('98 大纲)

序

进入 20 世纪 90 年代以来,我国掀起了第二次计算机普及高潮。人们已经认识到,要建设一个现代化国家,离开计算机将寸步难行;无论从事什么工作,都必须学习计算机知识,掌握计算机应用。不少单位已经把通过计算机考试作为任职条件。

为了适应社会的需要,原国家教委考试中心于 1994 年推出了“全国计算机等级考试”,受到社会各界的热烈欢迎。至 1998 年底,已有 160 多万人报名考试,其中 70 多万人获得了等级证书。全国许多地区和部门也组织了本地区或本系统的计算机统一测试。许多同志认为,“学历”是从整体上反映了一个人的知识水平,而“证书”则反映了一个人在某一方面的能力。证书制度是学历制度的必要补充,是人才市场的需要,因而受到各方面的欢迎。

当然,计算机统一考试无论从内容上还是形式上都有一些问题需要进一步探索,例如,如何更好地反映计算机新技术的发展,怎样才能测试出应试者的实际应用能力等。相信随着时间的推移和经验的累积,这项制度会日趋完善。

全国和地区性的计算机等级考试在内容上、分级上、考试形式上大体相似。全国计算机等级考试目前分为四个等级。一级:具有计算机的初步知识和使用微机系统的初步能力。一级(B):面向公务员和在职干部,要求大致与一级相当,侧重应用能力。二级:具有计算机软件、硬件的基础知识和使用一种高级语言编制程序、上机调试的能力,可以从 QBASIC, FORTRAN, PASCAL, C, FoxBASE 五种语言中任选一种应试。三级分为两类:三级(A):具有计算机应用基础知识和计算机硬件系统开发的初步能力;三级(B):具有计算机应用基础知识和软件系统开发的初步能力。四级:具备深入而系统的计算机知识和较高的计算机应用能力。

为了帮助广大读者准备参加等级考试,1995 年我们组织编写了“计算机等级考试辅导”丛书,由清华大学出版社出版。该丛书包括一级、一级 B、二级(五种)的考试辅导共 8 本书,在出版后三年多时间内已发行 120 多万册。

根据计算机应用的发展,1998 年教育部考试中心修改了全国计算机等级考试大纲。和原大纲相比,新大纲有以下改动:一级除了可以选择 DOS 环境应试外,还可以选择 Windows 环境应试;二级的 BASIC 语言改为 QBASIC 语言;在二级的计算机基础知识部分中增加了对网络和多媒体的要求。

根据广大读者的要求,我们在原来的“计算机等级考试辅导”丛书的基础上,组织出版这套“计算机等级考试丛书('98 大纲)”。该丛书由以下三个系列构成:(1) 计算机等级考试教程:全面而系统地介绍考试大纲所规定的内容,是便于自学的教材。(2) 计算机等级

考试辅导:用来帮助已学过该课程的读者复习和准备考试,每本书的内容均包括各章要点、各章难点、例题分析、思考题,并附有模拟试题。(3)计算机等级考试样题汇编:按照计算机等级考试的内容和试题形式,提供了500~600道样题,供应试者参考。

本丛书中各书的作者都是高等学校或计算机应用部门中具有丰富教学经验并对计算机等级考试有较深入研究的教授、专家。相信本丛书的出版一定会受到广大准备参加计算机等级考试的读者的欢迎。

欢迎读者对本丛书提出宝贵意见。

“计算机等级考试丛书('98大纲)”主编

全国计算机等级考试委员会副主任

谭浩强

1999年3月

前 言

BASIC 语言是为广大初学者开发的计算机语言,由于它易学易用,自 1964 年问世以来,深受初学者欢迎,几十年来长盛不衰。QBASIC 语言是由美国 Microsoft 公司于 90 年代初推出的第 3 代 BASIC 语言,它是完全结构化和模块化的语言,适合于结构化程序设计。它是没有程序设计经验的人学习程序设计的较理想的一种语言。

近年来,国内学习 QBASIC 语言的人愈来愈多,许多大学、中学、中专都开设了 QBASIC 课程,许多学习计算机程序设计的人也把 QBASIC 作为入门语言。教育部考试中心在 1998 年的全国计算机等级考试的考试大纲中,把 QBASIC 作为二级(程序设计)考试的可任选的语种之一。

为了帮助准备参加全国计算机等级考试的读者掌握 QBASIC 语言程序设计,我们根据全国计算机等级考试'98 大纲的要求组织编写了本书。本书的叙述深入浅出,即使没有任何计算机基础的人,也能通过学习本书掌握 QBASIC 程序设计的基本知识。

本书由谭浩强教授担任主编,在参考谭浩强教授主编的《QBASIC 程序设计》(清华大学出版社出版)和其他 QBASIC 教材的基础上编写,其中马素霞副教授负责第 1 章~第 5 章,殷光复教授负责第 6 章~第 9 章,全书由谭浩强教授修改定稿。苏秀丽老师也参加了部分收集资料和编写工作。

本书如有不足,敬请批评指正。

编著者

2000 年 2 月

• II •

目 录

第 1 章 QBASIC 的基本概念	1
1.1 概述	1
1.1.1 BASIC 语言的发展简况	1
1.1.2 QBASIC 语言的特点	5
1.1.3 QBASIC 程序的组成	7
1.2 数据类型	9
1.2.1 概述	9
1.2.2 整型	10
1.2.3 实型	10
1.2.4 字符型	11
1.3 常量	11
1.3.1 整型常量	11
1.3.2 实型常量	12
1.3.3 字符串常量	12
1.3.4 符号常量	12
1.4 变量	13
1.4.1 变量名和变量的值	14
1.4.2 变量的类型	14
1.5 标准函数	17
1.6 QBASIC 表达式	18
1.6.1 算术运算符	18
1.6.2 算术表达式	19
1.6.3 不同类型数据的混合运算	20
习题	20
第 2 章 顺序结构程序设计	22
2.1 输出语句(PRINT 语句)	22
2.1.1 PRINT 语句的一般格式	22
2.1.2 PRINT 语句的作用	23
2.1.3 PRINT 语句的输出格式	26
2.1.4 LPRINT 语句	29
2.2 赋值语句(LET 语句)	30
2.2.1 赋值语句的一般格式	30

2.2.2	赋值语句的作用	30
2.2.3	对变量的赋值操作	32
2.3	键盘输入语句(INPUT 语句)	34
2.4	读数语句(READ 语句)和置数语句(DATA 语句)	37
2.5	恢复数据指针语句(RESTORE 语句)	39
2.6	结束语句(END 语句)和暂停语句(STOP 语句)	40
2.6.1	结束语句(END 语句)	40
2.6.2	暂停语句(STOP 语句)	41
2.7	注释语句(REM 语句)	41
	习题	42
第 3 章	选择结构程序设计	47
3.1	关系表达式和逻辑表达式	47
3.1.1	关系运算符和关系表达式	47
3.1.2	逻辑运算符和逻辑表达式	48
3.2	二路分支选择结构	49
3.2.1	行 IF 语句	49
3.2.2	块 IF 语句	53
3.2.3	ELSEIF 结构	57
3.3	多路分支选择结构	58
3.3.1	最基本的 SELECT CASE 语句	58
3.3.2	在 CASE 子句中使用“TO”指定值的范围	60
3.3.3	在 CASE 子句中使用“IS”指定条件	61
3.3.4	在 CASE 子句中使用多个条件	62
3.3.5	SELECT CASE 语句的一般格式	62
3.3.6	多分支转移语句(ON GOTO 语句)	69
3.3.7	用不同语句实现多分支选择结构的比较	70
	习题	71
第 4 章	循环结构程序设计	77
4.1	WHILE 循环结构	77
4.2	DO 循环结构	78
4.2.1	最简单的 DO 循环格式	78
4.2.2	用 EXIT DO 语句终止循环	79
4.2.3	带条件的 DO 循环	80
4.3	FOR 循环结构	85
4.3.1	FOR 循环的一般格式	85
4.3.2	FOR 循环的执行过程	87

4.3.3	EXIT FOR 语句	91
4.4	循环的嵌套	92
4.5	程序设计举例	96
	习题	103
第 5 章	数组与记录	111
5.1	数组和数组元素的概念	111
5.2	数组的定义与引用	112
5.2.1	数组的定义	112
5.2.2	数组的引用	114
5.3	一维数组	115
5.4	二维数组	128
5.5	多维数组	133
5.6	静态数组和动态数组	134
5.7	记录	137
5.7.1	记录的定义与使用	137
5.7.2	记录数组	139
	习题	141
第 6 章	函数与子程序	150
6.1	模块化程序设计概述	150
6.2	函数	151
6.2.1	自定义函数(DEF)	151
6.2.2	独立模块的函数(FUNCTION)	157
6.2.3	函数的嵌套与递归	163
6.3	块内子程序——子例程	167
6.3.1	GOSUB-RETURN 语句	168
6.3.2	ON GOSUB-RETURN 语句	169
6.3.3	ON KEY(n) GOSUB-RETURN 语句	170
6.4	独立模块的子程序	171
6.4.1	子程序的定义和调用	171
6.4.2	在 QBASIC 环境中有关子程序的操作	173
6.4.3	带参子程序——虚实结合	174
6.4.4	递归子程序与递归程序设计	179
6.5	全局变量与局部变量	187
6.5.1	局部变量	187
6.5.2	全局变量	189
	习题	192

第 7 章 字符串	198
7.1 概述	198
7.2 字符串常量	198
7.3 字符串变量	199
7.3.1 变长字符串变量	199
7.3.2 定长字符串变量	200
7.4 字符串传送	201
7.4.1 用 LET 语句传送字符串数据	201
7.4.2 用 READ/DATA 或 INPUT 语句传送字符串常量	202
7.4.3 用 LINE INPUT 语句传送字符串常量	203
7.5 字符串的比较	203
7.5.1 字符比较的规则	203
7.5.2 字符串比较的规则	204
7.6 字符串数组	205
7.7 字符串的连接	206
7.8 子字符串	207
7.9 有关字符串的函数	211
7.9.1 测字符串长度的函数	211
7.9.2 字符串转换函数	213
7.9.3 字符串重复函数	218
7.9.4 日期和时间函数	218
7.9.5 INKEY \$ 函数	220
习题.....	220
 第 8 章 屏幕控制与图形	 225
8.1 显示模式	225
8.1.1 显示模式	225
8.1.2 屏幕坐标系	226
8.2 屏幕控制语句	226
8.2.1 LOCATE 语句.....	226
8.2.2 WIDTH 语句	227
8.2.3 CLS 语句	227
8.2.4 CSRLIN 和 POS 函数	227
8.2.5 SCREEN 语句	228
8.2.6 COLOR 语句	228
8.3 绘图语句	229
8.3.1 画点语句	229
8.3.2 画线语句	230

8.3.3 画圆、椭圆和圆弧语句	232
8.4 图形着色语句	235
8.5 综合练习	236
习题	238
第9章 数据文件	241
9.1 文件的概念	241
9.1.1 文件与记录	241
9.1.2 文件的分类	242
9.1.3 文件标识	243
9.1.4 文件缓冲区	244
9.1.5 文件位置指针	245
9.1.6 文件结束标志	245
9.2 顺序文件	246
9.2.1 顺序文件的打开和关闭	246
9.2.2 对顺序文件的输出——写文件	248
9.2.3 从顺序文件输入——读文件	253
9.2.4 对顺序文件的修改	255
9.2.5 与顺序文件有关的函数	258
9.3 随机文件	261
9.3.1 随机文件的打开与关闭	261
9.3.2 对随机文件的读写	262
习题	267
附录	270
附录1 字符与ASCII码对照表	270
附录2 QBASIC保留字	273
附录3 QBASIC语句一览表	274
附录4 QBASIC函数一览表	277
附录5 PRINT USING语句的格式字符	280
附录6 1999年4月全国计算机等级考试二级笔试试卷 基础部分和QBASIC语言程序设计	281
附录7 本书各章习题的参考答案	292
参考文献	334

第 1 章 QBASIC 的基本概念

1.1 概 述

1.1.1 BASIC 语言的发展简况

BASIC 是英文 beginner's all-purpose symbolic instruction code(初学者通用的符号指令代码)的缩写。它是由美国 Dartmouth 学院数学系的 John. G. Kemeny 和 Thomas E. Kurtz 两位教授于 1963—1964 年间为初学者开发的一种小型程序设计语言。

BASIC 语言的最大特点是容易学习,它从诞生之日起,就一直是初学者学习计算机的入门语言。到现在,它不仅仅是初学者的良师益友,而且已经广泛应用于数值计算、数据处理、实时控制、绘图、游戏等多种领域。

随着计算机的不断发展,BASIC 语言也在发展。可将 BASIC 语言划分为以下几个发展阶段:

(1) 基本 BASIC

最初的“基本 BASIC”只有 17 条语句(1964 年 5 月 1 日问世时只有 14 条语句),在一些小型机上使用,采用的是编译方式。虽然它的功能还较弱,但由于使用简单,使得计算机的使用第一次从少数专家的手里解脱出来,被更多的人所接受。在最初的几年里,BASIC 语言先后在多种计算机上实现。各个厂家在实现时都设法对 BASIC 的功能做了扩充和改进,语句数量增加,运行速度提高,使其不仅能用于学习,也能用于解决一些小型的问题。

(2) 微机 BASIC

1971 年是 BASIC 语言发展过程中的一个重要时期,这一年,BASIC 语言已经成为一种相当稳定的通用语言,正当它向着多功能、标准化方向发展的时候,计算机的硬件取得了划时代的成就——美国英特尔(Intel)公司研制出世界上第一个微处理器,并由此开辟了微型计算机的时代。由于当时计算机硬件配置较低(尤其是内存很小),其它计算机高级语言难以在微机上实现。而 BASIC 由于对硬件要求比较低,容易在微机上实现。

第一个为微机配备 BASIC 语言的是微软(Microsoft)公司的总裁比尔·盖茨(Bill Gates)。1975 年比尔·盖茨与他的伙伴在第一台微型机上将编译型的 BASIC 压缩成 4KB 的解释型 BASIC 语言。在盖茨的第一个微机 BASIC 的带动下,以后出现的微机都毫无例外地配备了 BASIC 语言,有的还把它固化在内存中。微型计算机因体积小、功耗低、工作可靠、价格便宜等优点,得到了迅猛发展,并载着 BASIC 语言奔向人类社会的各个角落。微型计算机产品的激烈角逐,使得 BASIC 语言的功能不断强化,但也带来了各种问题。

当初设计 BASIC 语言时,曾规定 BASIC 语言应不要求用户了解机器的硬件结构。可是在微型计算机的角逐战中,各计算机厂商都急于卖弄自己机器的优势,给 BASIC 强加

了许多依赖于机器硬件的特性。于是出现了众多的与计算机硬件相联系版本,人们称之为计算机的“方言性”。BASIC 的方言性破坏了它的通用性和可移植性,给标准化造成很大困难。

(3) 结构化 BASIC

随着计算机硬件及计算机应用的发展,计算机要解决的问题越来越复杂,根据问题直接写出程序已不可能,往往需要借助于图形工具将解题步骤(也称为算法)描述出来。人们最早使用的图形工具为流程图,它用一些几何图框表示各种类型的操作,在框内写上简明的文字或符号表示具体的操作,用箭头的流程表示操作的先后顺序,对箭头的使用没有限制。图 1.1 为 ANSI(美国国家标准化协会)规定的一些常用的流程图符号。图 1.2 表示的是求三个数中的最大数的算法,图 1.3 表示的是求 $\sum_{i=1}^{100} i$ 的算法。

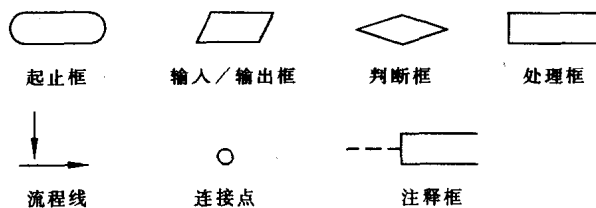


图 1.1 常用流程图符号

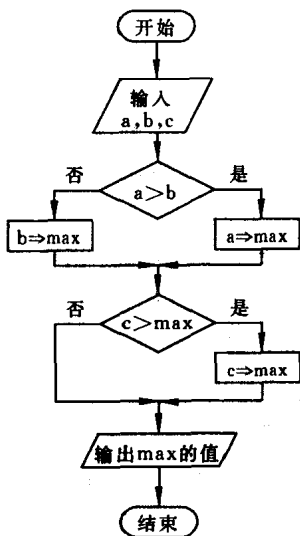


图 1.2 求三个数中的最大数算法

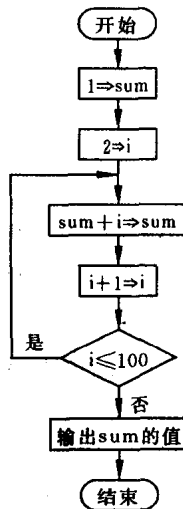


图 1.3 求 $\sum_{i=1}^{100} i$ 的算法

在图 1.2 中有两个判断框,判断框中的内容称为“条件”,根据条件成立与否决定执行两个分支中的一个分支,这种结构称为分支结构。在图 1.3 中有一个判断框,当判断框中的条件成立时,第 3、4 个处理框重复执行,这种结构称为循环结构。

在早期的 BASIC 语言中,没有直接实现分支及循环结构的专门语句,需要借助 GOTO 语句及行号来实现。

例 1.1 求 $\sum_{i=1}^{100} i$ 的 BASIC 程序。

```
10 REM compute sum=1+2+3+...+100
20 sum = 1
30 i = 2
40 sum = sum + i
50 i = i + 1
60 IF i <= 100 GOTO 40
70 PRINT "sum="; sum
80 END
```

OK (OK 是提示符,说明机器处在命令状态)

RUN (RUN 是通知系统进行运行的命令)

sum= 5050 (程序执行结果)

OK

在上面的程序中,10~80 都是行号。在没有转移语句(GOTO)的情况下,按行号由小到大的顺序执行各语句,执行到行号 60 时,若满足条件“ $i \leq 100$ ”,则转到行号 40 执行,40~60 行是一个循环。

由于对 GOTO 语句的使用不加任何限制,程序员可以信手使用 GOTO 语句把流程从程序的一行转向程序内的任何一行,这就容易导致程序出现图 1.4 所示的 BS 型算法结构(BS 是 bowl of spaghetti 的缩写,意为“一碗面条”,即线索不清,搅成一团),使程序难以阅读、难以修改,软件的质量难以保证。在 20 世纪 70 年代爆发了“软件危机”。人们将软件危机爆发的原因归结为对 GOTO 语句的无限制使用。为了克服使用 GOTO 语句带来的弊端,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中有直接实现顺序、选择及循环三种基本结构的语句,并且要求程序模块化,即主程序和子程序分别属于不同的模块。

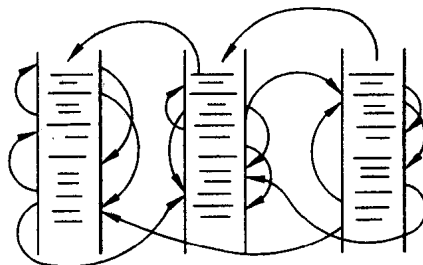


图 1.4 BS 型算法结构

用流程图表示的三种基本结构如图 1.5 所示。已经证明,由以上三种基本结构按线性顺序组成的算法,可以解决任何复杂的问题。由基本结构构成的算法属于“结构化”的算法,它不存在无规律的转移,只在本结构内才允许存在分支或向前或向后的跳转。请读者思考一下,图 1.3 所示的算法是否是结构化算法?

既然用基本结构的顺序组合可以表示任何复杂的算法结构,那么基本结构之间的流程线就属多余的了。1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。在这种流程图中,完全去掉了带箭头的流程线。全部算法写在一个大矩形框中,这种流程图称为 N-S 流程图,又称为盒图(box diagram)。用 N-S 图表示的三种基本结构如图 1.6

既然用基本结构的顺序组合可以表示任何复杂的算法结构,那么基本结构之间的流程线就属多余的了。1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。在这种流程图中,完全去掉了带箭头的流程线。全部算法写在一个大矩形框中,这种流程图称为 N-S 流程图,又称为盒图(box diagram)。用 N-S 图表示的三种基本结构如图 1.6

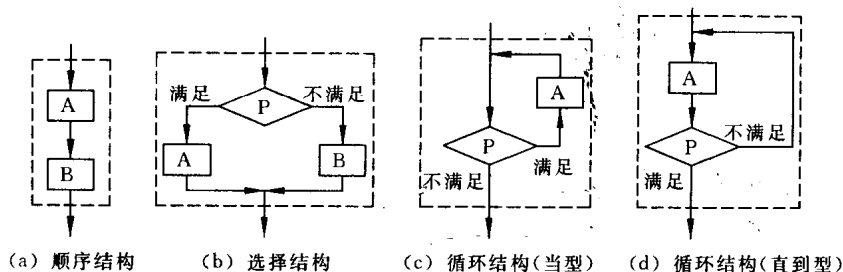


图 1.5 用流程图表示的三种基本结构

所示。图 1.7 表示的是求三个数中的最大数的算法,图 1.8 表示的是求 $\sum_{i=1}^{100} i$ 的算法。

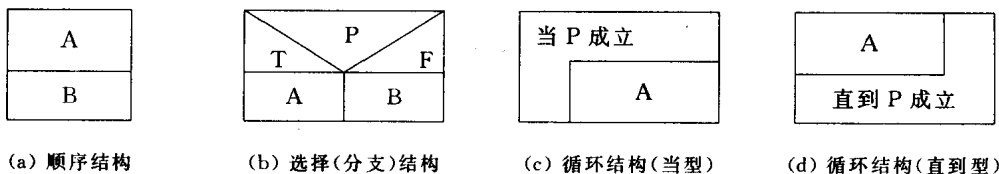


图 1.6 N-S 图表示的三种基本结构

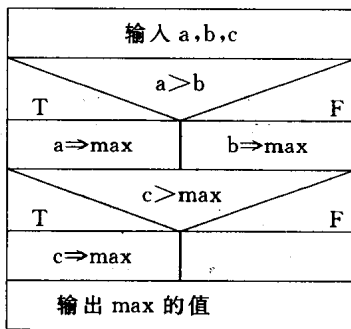


图 1.7 求三个数中的最大数算法

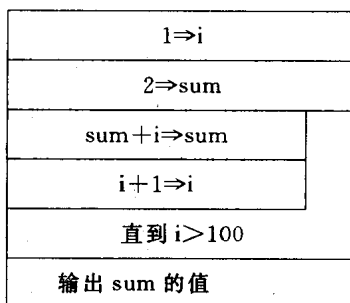


图 1.8 求 $\sum_{i=1}^{100} i$ 的算法

显然,BASIC 已有的控制结构难以满足结构化程序设计的要求,往往需要借助 GOTO 语句及行号来实现分支及循环结构。面对 BASIC 的这些缺点,人们已从多方面对它进行了改进,如制定 BASIC 标准,推出编译 BASIC 版本,改良并增加结构化的控制结构等,但这些都没有从根本上消除 BASIC 的缺点。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,这一标准于 1987 年正式通过。在此前后,出现了一些结构化的程序设计语言,主要有 True BASIC, Quick BASIC, Turbo BASIC 等。

True BASIC 是 BASIC 语言的两位创始人 Kemeny 和 Kurtz 于 1985 年推出的。True BASIC 意为“真正的 BASIC”,以区别于各个厂家自行开发的“街头 BASIC (street BASIC)”。True BASIC 对 BASIC 语言做了重大改进和发展,它严格遵循 ANSI 标准,不仅完全适应结构化和模块化程序设计的要求,而且保留了 BASIC 语言易学易懂、程序易

编易调试等优点。

Quick BASIC 是 Microsoft 公司 1987 年推出的。Quick BASIC 提供了一个开发程序的集成环境,用户在编辑、编译、调试、运行程序时均可通过菜单进行操作,使用十分方便;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;采取编译执行方式,能建立可执行文件(.EXE),提高了执行效率;允许用户建立常用的过程库;提供更强的调试工具,使用户对程序设计环境有更强的控制能力。

在 Quick BASIC 的基础上,微软公司又在几年前推出了 QBASIC 语言,并将 QBASIC 作为 MS-DOS 5.0 版的一个组成部分免费提供给用户。也就是说,如果你使用的计算机上已经装了 MS-DOS 5.0 或更高的版本,你就可以自由地使用 QBASIC 软件(不需要另外购买此软件)。

(4) 图形界面的 BASIC

1991 年微软公司推出了第一个直接运行在 Windows 环境下的 BASIC 语言——Visual BASIC 语言,意为“可视化 BASIC”语言,用它可以开发出具有良好用户界面的 Windows 应用软件。

在 Windows 环境下运行的 Visual BASIC 与 QBASIC 完全兼容,也就是说,用 QBASIC 所编写的程序能在 Visual BASIC 环境下顺利执行。因此掌握了 QBASIC 再过渡到 Visual BASIC 是很方便的。

除了可以向 Visual BASIC 过渡之外,Microsoft 公司的字处理程序 Word for Windows 以及 Word for OS/2 都附加了和 QBASIC 很相像的宏语言,称为 Word BASIC。Microsoft 还计划在它的多个操作环境中将 QBASIC 的子集作为统一的语言。因此,学习 QBASIC 能够使人们较容易地进入程序设计的大门,并为以后应用计算机打下良好的基础。

1.1.2 QBASIC 语言的特点

QBASIC 是模块化和结构化的 BASIC 语言,它扩充了以下几个方面的功能:

(1) 扩充了变量和常量的类型

QBASIC 允许变量名的长度达 40 个字符。变量类型除了整型、实型、双精度型和字符串类型外,还增加了长整型、定长字符串类型。此外还增加了符号常量。

(2) 提供了新的选择结构

过去的 BASIC 需要借助 GOTO 语句和行号实现选择结构,IF 语句是写在一行内的。这种语句的功能受到限制,使用不便,结构不清晰。QBASIC 允许使用多行的 IF 结构,即块 IF 结构。这种选择结构以 IF 开始,以 END IF 结束。例如:

```
IF <条件> THEN
```

```
    <语句组 1>
```

```
ELSE
```

```
    <语句组 2>
```

```
END IF
```

这种结构使程序清晰,功能较强,使用方便。

QBASIC 还提供了多分支选择结构的 SELECT CASE 语句。

(3) 增加了新的循环结构

除了保留 FOR-NEXT 循环结构外,增加了 WHILE 型循环和 DO 循环。WHILE 型循环的形式为:

```
WHILE <条件>
    <循环体>
WEND
```

它是实现当型循环的语句。

DO 循环包括两类:DO WHILE 型循环和 DO UNTIL 型循环。

DO WHILE 型循环的结构为:

```
DO WHILE <条件>          或          DO
    <循环体>                <循环体>
LOOP                      LOOP WHILE <条件>
```

当“条件”满足时继续执行循环体。

DO UNTIL 型循环的结构为:

```
DO UNTIL <条件>          或          DO
    <循环体>                <循环体>
LOOP                      LOOP UNTIL <条件>
```

“条件”满足时终止执行循环体。

(4) 子程序和函数作为单独的模块

BASICA 和 GW-BASIC 的子程序并不是单独的模块,它和调用程序属于同一个模块,因而变量具有全局性,即主程序中的变量和子程序中的同名变量对应相同的内存单元。这样难以实现模块的独立性。QBASIC 的子程序是一个单独的模块,子程序中的变量只在本子程序中有效,是局部变量。

BASICA 和 GW-BASIC 只允许定义单行的自定义函数。QBASIC 允许用户定义多行的、单独的函数模块。函数中的变量是局部变量。

这就为模块化设计提供了有力工具。

(5) 不需要行号

考虑到与旧的基本语言的兼容性,QBASIC 中保留了行号。但在使用 QBASIC 语言编程序时,不希望程序员使用行号。不使用行号,就迫使程序员不能使用以行号为依托的 GOTO 语句,从而严格按结构化原则构造程序。

(6) 具有友好的用户界面

QBASIC 提供了集成的编辑、调试及运行环境,将一些重要的功能以菜单的形式提供给用户,用户可以同时使用键盘和鼠标器来控制操作。系统提供程序窗口和命令窗口,程序运行结果和源程序不显示在同一屏上,以便于观察运行结果。

(7) 提供联机“帮助”

QBASIC 提供联机“帮助”系统,在编程过程中可以要求计算机随时提供“帮助”信息。这使初学者尤感方便。

(8) 提供“分步执行”、“跟踪”等调试工具

QBASIC 提供的多种调试工具可以帮助程序员更有效地查找程序中存在的错误。

(9) 采用先进的解释执行方式

QBASIC 采用先进的解释程序,使程序运行比一般的 BASIC 快得多。

在输入一行 QBASIC 语句时,系统会立即将该行上所有的关键字改为大写,自动使循环向右缩进,并且立即进行语法检查,如有错,会即时给出出错信息。

以上这些功能使 QBASIC 与其它一些现代语言相比并不逊色。也就是说,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一贯传统。QBASIC 是学习用 BASIC 编程和开发软件的有力工具。

在学习使用 QBASIC 的基础上,如果想更好地开发应用程序,可以使用 Quick BASIC。QBASIC 是 Quick BASIC 的子集,它具有 Quick BASIC 的大部分功能,而且两者的语言内容和语法规律基本一样。学会了 QBASIC 很容易通过自学掌握 Quick BASIC 的使用。作为初学者,从学习 QBASIC 入手是较好的办法。

1.1.3 QBASIC 程序的组成

先举一个 QBASIC 源程序的例子。

例 1.2 已知圆的半径 $r = 2.5$,求圆的周长 l 和面积 s 。

求周长和面积的公式为:

$$l = 2\pi r$$

$$s = \pi r^2$$

源程序如下:

```
REM the perimeter and area of circle
LET r = 2.5
LET l = 2 * 3.14159 * r
LET s = 3.14159 * r * r
PRINT "l="; l
PRINT "s="; s
END
```

运行结果如下:

$l = 15.70795$

$s = 19.63494$

在输入源程序的过程中,当输入完一行,键入回车键时,QBASIC 系统自动将程序格式作规范化处理,如将语句关键字、函数名等改为大写,在赋值号(“=”号)及运算符(+、-、*、/、\等)两侧各留出一个空格等。除了对格式作规范化处理外,还进行语法检查,如果刚刚输入的一行存在语法错误,则在用户键入回车键时自动给出语法错误提示。

例 1.3 已知三角形的三条边 $a=3$ 、 $b=4$ 、 $c=6$,求三角形的面积 s 。