

信息学(计算机) 国际奥林匹克 QUICK BASIC 4.5

甘宪甫 主编

中南工业大学出版社

信息学(计算机)国际奥林匹克

QUICK BASIC 4.5

中南

1981.9

社

381181

7/31-2
G 11

信息学(计算机) 国际奥林匹克

QUICK BASIC 4.5

甘宪甫 主编

中南工业大学出版社

内容简介

Quick BASIC 4.5 是当前国际上流行的 BASIC 最佳版本,1994 年定为信息学国际奥林匹克“IOI”(International Olympiad in Informatics)竞赛语言。Quick BASIC 4.5 结构化程序设计能力强,运算速度快,解释与编译并行,具有和 Pascal、C 语言并驾齐驱的完美程度,但难度却与基本 BASIC 差不多。

本书包括“语言”和“解题”上下两篇,文字通俗易懂,内容由浅入深,它既可以作为普及计算机教育用教材,也可以作为信息学奥林匹克竞赛、计算机课外活动培训用书。

JS196/60

【相】新登字 010 号

信息学(计算机)国际奥林匹克

Quick BASIC 4.5

甘宪甫 主编

责任编辑:文 刀

*

中南工业大学出版社出版发行
湖南大学印刷厂印装
新华书店总店北京发行所经销

*

开本: 787×1092 1/16 印张: 13.5 字数: 340 千字

1994 年 5 月第 1 版 1995 年 3 月第 2 次印刷

印数: 5201—11200

*

ISBN7-81020-650-8/TP·035

定价: 9.00 元

本书如有印装质量问题,请直接与承印厂家联系解决。

序

计算跟语言一样,是人类社会不可缺少的东西,是一种基础性的思维工具,离开了它,社会的生产、人们的生活就无法正常进行。在只有算盘的时代,学生要学打算盘;发明了算尺之后,理工科学生要学习拉计算尺;现在计算机步入学校,走进家庭,学生要学用计算机,这本是顺理成章之事。但我觉得这样理解还远远不够,因为计算机不同于一般的工具,它是“人类通用智力工具”。从一般工具到智力工具,这是科技史上划时代的进步。所以过去我们说,没有电我们将寸步难行;而我们现在要说,没有计算机就没有现代化。

坚持科学技术是第一生产力,把经济建设真正转移到依靠科技进步与提高劳动者素质的轨道上来,是一场广泛而深刻的变革,核心是重视科学,重视人才,重视教育,重视符合时代精神的现代化的教育。随着时代的发展,教育的内容、方法与手段也要不断更新。计算机与基础教育相结合是当今世界的一大趋势。计算机的普及从娃娃作起是时代的需要,跨世纪的人才所应具备的知识结构中,应用计算机的能力不容忽视。计算机作为一门课程纳入教学计划,教什么内容和怎么教,是大家普遍关心的一个问题。我以为,对中学生从学习一门简单的计算机语言入手来了解计算机可能是一条捷径。经验表明,青少年通过学习计算机语言,可以学会初步的设计程序的思路与方法,受到现代的科学思维方式的启迪,提高逻辑思维能力、抽象化形式化描述问题能力和科学地分析问题与解决问题的能力。

众所周知,BASIC 语言最初是为便于教学而设计的,它简单易学,使用方便,特别适合初学者的需要。这种为大众所喜闻乐见的语言经过 30 年的演化与改进,功能越来越强,象 True BASIC 和 Quick BASIC 已经成了真正结构化的程序设计语言。在选定语言版本,具备了师资和设备条件的情况下,教材就显得十分重要了。由甘宪甫老师编写的《信息学(计算机)国际奥林匹克 Quick BASIC 4.5》一书,全面介绍了这一优选的语言版本,内容深入浅出,文字通俗易懂,既可作为普及计算机的教材,又可作为信息学奥赛培训用书,与国内国际的信息学奥林匹克接轨。

在许多场合我都说过,各种级别的竞赛都不是目的,而是手段,是推动计算机普及的手段。国际信息学奥林匹克是在世界范围计算机普及与发展的产物。没有普及,中国的青少年也绝不可能冲向世界。五届世界大赛,中国队给人们留下了总体实力很强的印象,表明了中国的青少年有志于掌握高科技,有能力去攀登科技领域的高峰。

今天在为这本书作序时,恰逢小平同志提出“计算机的普及要从娃娃做起”十周年。看到改革开放给祖国大地带来的勃勃生机,想到普及计算机教育又将有一个大的发展,作为园丁,我们有责任自觉地将普及计算机的事业与中华民族光辉灿烂的未来联系起来,尽心尽力,让跨世纪的人才茁壮成长。

中国计算机学会普及委员会主任
国际信息学奥林匹克中国队总教练
清华大学计算机科学与技术系教授

吴 文 虎

1994.2.14 于清华园

前 言

信息学(计算机)已成为继人类语言和数学之后,人类必须掌握的第三种最主要的工具。作为未来 21 世纪的人才,当今的中学生、广大青少年同学们,很有必要学习好计算机,提高自身素质,以适应未来社会发展的需要。

信息学(计算机)国际奥林匹克是世界各国重视青少年中普及计算机教育的产物,它的竞赛是世界范围内对青少年智慧解题与操作能力的竞赛,定名为“International Olympiad in Informatics”,简称 IOI。到目前为止,信息学国际奥林匹克竞赛已历经了五届。我国在清华大学计算机系教授、享有盛誉的国际信息学奥林匹克中国队总教练吴文虎的带领下,历经艰辛,奋力拼搏,取得了举世瞩目的优异成绩,共获得 18 枚奖牌。

Quick BASIC 4.5 是当前国际上流行的 BASIC 最佳版本,1994 年定为信息学(计算机)国际奥林匹克竞赛语言。Quick BASIC 4.5 结构化程序设计能力强,使用下拉式菜单,全屏幕多功能显示,语言对话式操作,解释与编译并行。Quick BASIC 4.5 将程序的编辑、编译、连接、调试等融为一体,已达到和 Pascal、C 语言并驾齐驱的完美程度。由于 Quick BASIC 4.5 来源于 BASIC 语言,因此,它仍然保留了 BASIC 语言简单易学、通用性强、特别适合于初学者学习的特点。目前数以百万计的中国中、小学生正在接受计算机教育,在教学语言上,由于条件的限制,大多数仍为 BASIC 语言。Quick BASIC 4.5 为与国际奥林匹克接轨架起了理想的通向成功的桥梁。

1994 年是中国计算机发展史上值得纪念的一年。十年前的今天,邓小平同志以伟人的远见卓识,发出了“计算机的普及要从娃娃抓起”的伟大号召。在巨人之手的推动之下,中国的计算机普及教育工作蓬勃发展,开展计算机教育的中小学如雨后春笋,遍地开花,总数已逾万所。十年,在人类历史长河中只是短暂的一瞬,而中国的这一瞬间,是如此辉煌。作为耕耘在中小学计算机教育这遍沃土上的人民教师,不仅为过去的十年感到骄傲,也为今后的发展感到无比的欢欣鼓舞。

指导和参加本书编辑工作的有:吴文虎教授、陈松乔教授、吴耀斌、王重芳、范一苓、黄香君、刘尚、彭小路、赵宝鼎老师等。本书得到中国计算机学会普及委员会、国防科技大学计算机系、湖南省教委普教处、河南省教委中小学教研室、四川省教育科学研究所的大力支持与帮助,在此一并表示衷心的感谢。

全书中所有例题均上机验证,Quick BASIC 4.5 系统盘可得到提供。由于编者水平有限,加之时间仓促,难免存在不少缺点和错误,诚请读者批评和指正。

编 者

1994 年元月

目 录

上篇 语 言

第1章 Quick BASIC 4.5 语言基础	(1)
1.1 Quick BASIC 语言概述	(1)
习 题	(2)
1.2 Quick BASIC 4.5 基本字符和关键字	(2)
1.2.1 基本字符	(2)
1.2.2 关键字	(3)
习 题	(4)
1.3 Quick BASIC 4.5 程序行与语句结构	(1)
习 题	(5)
1.4 量的类型和作用域	(5)
1.4.1 常量和变量	(5)
1.4.2 变量的类型	(5)
1.4.3 定义类型语句	(6)
1.4.4 作用域及其语句	(8)
习 题	(10)
1.5 数据的表示及其运算	(11)
1.5.1 常数的表示	(11)
1.5.2 表达式和算术运算符	(12)
1.5.3 关系运算符和逻辑运算符	(12)
习 题	(14)
第2章 Quick BASIC 4.5 操作基础	(15)
2.1 标准键盘及其操作训练	(15)
2.1.1 标准键盘和键盘操作	(15)
2.1.2 指法训练	(16)
习 题	(17)
2.2 Quick BASIC 4.5 安装和启动	(17)
2.2.1 Quick BASIC 4.5 系统文件	(18)
2.2.2 Quick BASIC 4.5 安装	(19)
2.2.3 Quick BASIC 4.5 启动	(20)
习 题	(21)
2.3 Quick BASIC 4.5 屏幕窗口和菜单操作	(21)
2.3.1 屏幕窗口	(21)
2.3.2 文件(File)和编辑(Edit)	(22)

2.3.3 观察(View)和检索(Search)	(28)
2.3.4 运行(Run)和调试(Debug)	(31)
2.3.5 调用(Calls)、选择(Options)和帮助(Help)	(36)
习 题	(38)
第3章 输出和输入语句	(39)
3.1 输出显示语句	(39)
3.1.1 PRINT 语句	(39)
3.1.2 WRITE 语句	(41)
习 题	(41)
3.2 输出用光标控制	(42)
3.2.1 LOCATE 语句	(42)
3.2.2 CSRLIN 和 POS 函数	(42)
习 题	(43)
3.3 赋值和交换语句	(43)
3.3.1 LET 语句	(43)
3.3.2 SWAP 语句	(45)
习 题	(45)
3.4 键盘输入、结束和注释语句	(46)
3.4.1 INPUT 语句	(46)
3.4.2 END 语句	(47)
3.4.3 REM 语句	(47)
习 题	(47)
3.5 读数、置数和数据恢复语句	(48)
3.5.1 READ、DATA 语句	(48)
3.5.2 RESTORE 语句	(49)
习 题	(50)
第4章 程序流程图和基本结构语句	(51)
4.1 程序流程图	(51)
4.1.1 框图	(51)
4.1.2 N-S 图	(51)
习 题	(52)
4.2 条件和情况语句	(52)
4.2.1 IF—THEN...ELSE 行语句	(53)
4.2.2 IF—THEN...ELSE 块语句	(56)
4.2.3 SELECT CASE—END SELECT 语句	(57)
习 题	(58)
4.3 循环语句	(59)
4.3.1 FOR—NEXT 语句	(59)
4.3.2 WHILE—WEND 语句	(62)
4.3.3 DO—LOOP 语句	(64)
习 题	(65)

4.4	GOTO 型控制	(66)
4.4.1	GOTO 语句	(66)
4.4.2	ON—GOTO 语句	(66)
	习 题	(67)
4.5	子程序和自定义函数	(67)
4.5.1	GOSUB—RETURN 语句	(67)
4.5.2	DEF FN 语句	(70)
	习 题	(71)
第 5 章	数组、过程、函数和字符串	(72)
5.1	数组	(72)
5.1.1	DIM 定义数组语句	(72)
5.1.2	数组元素	(72)
	习 题	(71)
5.2	过程	(75)
5.2.1	过程的定义	(75)
5.2.2	过程的调用	(77)
5.2.3	过程和子程序、自定义函数间的关系	(79)
	习 题	(80)
5.3	函数	(80)
5.3.1	算术函数	(80)
5.3.2	三角函数	(81)
5.3.3	打印格式函数	(82)
5.3.4	随机函数	(82)
	习 题	(81)
5.4	字符串	(84)
5.4.1	字符串并置	(85)
5.4.2	字符串函数	(85)
5.4.3	字符串相关函数	(87)
	习 题	(88)
第 6 章	图形和音响语句	(90)
6.1	图形与绘图语句	(90)
6.1.1	显示模式及其语句	(90)
6.1.2	屏幕坐标及其语句	(92)
6.1.3	绘图语句和填色语句	(92)
6.1.4	应用举例	(97)
	习 题	(100)
6.2	音响和音乐语句	(101)
6.2.1	BEEP 嘟语句	(101)
6.2.2	SOUND 声语句	(101)
6.2.3	PLAY 音语句	(102)
6.2.4	应用举例	(105)

习 题.....	(107)
----------	-------

下篇 解 题

第 7 章 数据结构、程序设计与算法	(108)
7.1 数据结构	(108)
7.1.1 基本概念.....	(108)
7.1.2 线性表.....	(109)
7.1.3 树(tree)	(114)
7.1.4 图(graph)	(116)
7.2 程序设计	(119)
7.2.1 结构化程序设计.....	(120)
7.2.2 结构化程序设计的方法.....	(120)
7.3 算法	(122)
7.3.1 算法设计的基本思路.....	(122)
7.3.2 算法与解题.....	(123)
第 8 章 数字问题	(124)
(例 1~例 18)	
第 9 章 数值处理	(137)
(例 19~例 39)	
第 10 章 文本图形和字符串	(155)
(例 40~例 56)	
第 11 章 随机问题和逻辑判断	(172)
(例 57~例 75)	

附 录

附录一 Quick BASIC 4.5 关键字	(188)
附录二 ASCII 字符代码	(190)
附录三 出错信息	(192)
附录四 Quick BASIC 4.5 命令、语句和函数	(199)

上篇 语 言

和人类语言一样,计算机通过计算机语言进行信息交流。计算机语言可分为三类,这就是机器语言、汇编语言和高级语言。所谓机器语言,是计算机能够直接识读的语言,它由 0、1 代码组成,因此又称 0、1 代码语言,它不需要翻译便可直接为计算机所接受。所谓汇编语言,是用一种较直观、便于记忆的符号来表示的语言,因此又称符号语言,它要通过“汇编程序”翻译成机器语言后才被计算机所接受。所谓高级语言,是一种更加接近人类语言习惯的计算机语言,它接近人们的书写与语言习惯,便于记忆,容易掌握,通用性强,它通过“解释方式”或“编译方式”翻译后,被计算机所接受。Quick BASIC 4.5 是一种计算机高级语言,它通过解释与编译并行的方式,指挥计算机高效率地开展工作。

第 1 章 Quick BASIC 4.5 语言基础

1.1 Quick BASIC 语言概述

BASIC 语言是国际上广泛使用的一种计算机高级语言,它的全称是 Beginner's All-purpose Symbolic Instruction Code,即初学者通用符号指令代码语言。BASIC 语言可通过解释方式或编译方式翻译后被计算机所接受,因此,它可分为解释型和编译型。解释型 BASIC 在执行时需要不断地解释译码,执行的速度较慢,所编写的程序只能在相应的环境中执行。例如:Comxsoft BASIC、Applesoft BASIC、Microsoft BASIC (GW BASIC 是 Microsoft BASIC 高级版本)等均属解释型 BASIC。由于简单易学,调试、检测比较方便,尤其是 GW BASIC 功能也较强,在普及计算机教育中仍被广泛地采用。编译型 BASIC 一次性译码编译,执行的速度快,所编写的程序可以在不同的语言环境下执行,通用性好,功能更强,因此,编译型 BASIC 得到更广泛地应用。

Quick BASIC 是美国 Microsoft 公司 1987 年底推出的 BASIC 新版本,它吸收了 Pascal、C 等语言的优点,集解释型与编译型于一体,既具有解释型的简单易学与调试、检测的方便性,又具有编译的快速与通用性,同时还对 BASIC 语言的功能进行了较大的扩充,尤其是在结构化程序设计方面,在屏幕编辑、编译、连接、调试等方面,已具有和 Pascal、C 语言并驾齐驱的完美

程度。

Quick BASIC 语言具有以下特点：

- 1) 和一般 BASIC 语言一样,好懂易学,语言结构简单明了,易于初学者学习和掌握。
- 2) 采用下拉式菜单,对话方式操作,解释与编译并行。它把编译、运行和调试有机地合为一体,把解释型 BASIC 的方便性和编译 BASIC 快速性合为一体。
- 3) 交互式编辑。输入一程序后,就能马上对其进行程序语法检查,如果正确则被编译成可执行代码,否则给出错误信息。在编辑中,关键字将自动转换成大写字母。
- 4) 交互式调试。在调试程序时,可以将一个运行着的程序在任何一处暂停,对它进行编辑,然后从停止处继续执行。可以利用屏幕底部的直接窗口测试程序的任意语句。
- 5) 可快速生成 DOS 状态下直接运行的可执行文件(.EXE),有灵活的联机求助功能,随时可对语句、命令、ASCII 码等提供帮助信息。有很强的结构化程序设计能力,有库管理功能和丰富的数据处理、图形、音响语句等。

自 Quick BASIC 问世以来,已推出了好几个版本,主要有 2.0、3.0、4.0 和 4.5 等,还有 Quick BASIC 编译 6.0(其编程环境仍为 4.0)。Quick BASIC 4.5 是当前国际上流行的 BASIC 中最佳版本,1994 年定为信息学国际奥林匹克“IOI”(International Olympiad in Informatics)竞赛用语言。

为简便起见,在本书以后各章节中,BASIC 语言均指 Quick BASIC 4.5 语言。

习 题

- 1) BASIC 一词的全称及其含义是什么?
- 2) 计算机语言可分为哪几类,各有什么特点?
- 3) Quick BASIC 4.5 语言属哪一类计算机语言,其特点是什么?

1.2 Quick BASIC 4.5 基本字符和关键字

1.2.1 基本字符

Quick BASIC 基本字符,即字符集是由字母、数字和专用字符组成。其组成原则是:它们能被 Quick BASIC 语言所识读。专用字符包括有标点符、算术运算符、关系运算符和特殊说明符等。

1) 字母:

基本字母是英文中的 26 个字母,它包括大写字母 A~Z 和小写字母 a~z。

2) 数字:

数字由 0~9,共计 10 个基本阿拉伯数字组成。

3) 标点符:

- ! 感叹号或单精度说明符
- , 逗号或分区(标准)格式打印说明符
- . 句号或小数点

- ； 分号或紧凑格式打印说明符
- ： 冒号或语句分隔符
- ？ 问号或 PRINT 打印语句缩写符
- ' 单引号或注解定界符
- " 双引号或字符串定界符
- (开括号
-) 闭括号

4) 算术运算符：

- + 加号或并置运算符
- 减号或虚线符
- * 乘号或星号符
- / 除号或斜杠符
- ^ 乘方运算符
- \ 整除号或倒斜杠

5) 关系运算符：

- > 大于号
- >= 大于等于号
- = 等于号或赋值号
- < 小于号
- <= 小于等于号
- <> 不等于号

6) 特殊说明符：

- % 百分号或整型变量说明符
- # 数码标记(或英镑号),或双精度说明符
- \$ 美元号或字符串变量说明符
- & "和"符号或长整型变量说明符
- _ 下划线
- 空格符

空格符 主要用于拉开字符间的间隔,或指令性间隔说明,在屏幕显示和打印输出上什么也没有,仅是为说明空格多少而人为书写的虚符,在实际操作中可按长条键或位移键形成。

开、闭括号必须成对使用,而且可以重叠套用。

还有许多字符,尽管在 Quick BASIC 中没有特别的意义,但也能被打印或显示出来。详见附录二 ASCII 字符代码。

1.2.2 关键字

某些字在 Quick BASIC 中有特殊的意义,能完成对计算机某些特定的操作等,这些字称为关键字。关键字一般包括所有的 BASIC 语句、命令、函数和运算符等内容。

我们必须用空格或某些规定的字符,把关键字跟数据或 BASIC 语句的其它部分分隔开来,以便确保计算机识别 BASIC 语言中的这些关键字。

Quick BASIC 4.5 语言中的关键字, 详见附录一 Quick BASIC 4.5 关键字。

习 题

- 1) 基本字符即字符集是由什么组成的? 数学中的圆周率 π 是否为基本字符?
- 2) 什么是关键字, 它一般包括哪些内容?

1.3 Quick BASIC 4.5 程序行与语句结构

Quick BASIC 语言是由 Quick BASIC 程序及其命令组成的, 而程序是由一行一行的程序行构成的。一程序行可以由一条语句组成, 亦可以由多条语句组成, 但总的字符个数不能超过 255 个。当一程序行由多条语句组成时, 语句之间必须用冒号(:)分隔开。

程序行的一般格式是: (式中方括号中内容表示选择项可省略)

[行标识符] BASIC 语句 [:BASIC 语句] [注释]

说明如下:

1) [行标识符]

行标识符由行号或标号组成。由于 Quick BASIC 程序的执行顺序取决于程序行的输入顺序, 即行的先后顺序, 因此行标识符为可选择项, 这是和一般 BASIC 程序的重大区别 (一般 BASIC 程序必须具有行号, 并按行号顺序执行)。

行号由 1~65529 整数数字组成。标号由字母或数字组成, 但必须以字母开头、冒号结尾, 总字符不能超过 40 个字符, 不能是纯关键字。

2) BASIC 语句

BASIC 语句是程序行内容的基本结构, 它告诉计算机在这一程序行中要干什么或准备干什么。

BASIC 语句分为执行性语句和非执行性语句两类。执行性语句是一种程序指令, 这些指令告诉计算机下一步该作什么。例如: PRINT A 是一个执行性语句, 它告诉计算机下一步要在屏幕上显示出变量 A 的数值。而非执行性语句, 不产生任何程序动作, 仅表示对程序人员的提示, 或在其它执行性语句工作时, 提供一特定场所等。如: REM (注释语句), DATA (置数语句) 等。

在一条语句中, 一般包含指令名及指令内容两部分。如: PRINT A, 其中 A 就是指令 PRINT 的指令内容。如: RESTORE (恢复数据指令), 它无指令内容, 但 RESTORE 指令本身隐含恢复总体数据的含义。习惯上, 我们将语句中的指令名就称为该语句的语句名。

3) [注释]

说明程序行的作用, 进行注释以供程序阅读时参考。注释说明前, 必须用单引号隔开或采用注释语句。

值得注意的是, 当使用 Quick BASIC 编辑程序, 输入的内容超过一物理行时, 屏幕将自动右移, 不能把下划线()作为行的延续符; 而使用其它编辑程序, 即在非 Quick BASIC 状态下编辑 Quick BASIC 程序, 则可用下划线作为延续符, 将一个程序行放在几个物理行中。

但是, 下划线不能延续非执行性语句, 如: REM、DATA 等。

习 题

- 1) Quick BASIC 程序行的一般格式是什么?
- 2) Quick BASIC 程序的执行顺序取决于什么? 它和一般 BASIC 程序执行顺序有何区别?
- 3) BASIC 语句可分为哪两类? 各有什么作用?

1.4 量的类型和作用域

1.4.1 常量和变量

常量(Constant)和变量(Variable)是数据的两种量的形态。常量是在程序中始终保持不变的数据量,而变量是在程序中可以改变的数据量。计算机程序中所述的变量,和普通数学中使用的变量的含义是不同的。数学中的变量隐含着变化不定“未知”的概念,而计算机程序中的变量,在执行的每个细过程中是确定的,但在程序执行的整个过程中其值是可以变化的。在讨论问题时,我们一般讨论变量。

变量的命名:

Quick BASIC 变量的命名,名字的有效长度不能超过 40 个字符。变量名中允许出现字母、数字和小数点,但第一个字符必须是字母,这个字母既可大写也可小写。那些用以识别变量类型的专用符(如 \$, %, !, #)可以作为最后一个字符出现在变量名中。

变量名不能是纯关键字(详见附录一 Quik BASIC 4.5 关键字),也不能由关键字后面再加说明类型的专用字符组成。例如:

变量 ABS

是非法的,因为 ABS 是纯关键字。然而,

变量 ABSPROM

是允许的,因为这里 ABS 只是变量名 ABSPROM 中的一部分,不是纯关键字。

1.4.2 变量的类型

变量可分为两大类型:数字型和字符串型。数字型变量的值只能是数字,而字符串型变量的值只能是字符串。所谓字符串,就是用双引号括起来的、一串 BASIC 语言中允许存在的字符。

我们可以把常数赋给变量,如可以把程序中的计算结果或通过各种输入语句把值赋给变量。值得注意的是,变量的类型(字符串型和数字型)必须和赋给它的数据类型相匹配。一个尚未赋过值的数字型变量,它的值是 0;对于字符串型变量,其初值为空,也就是没有字符,其长度为 0。

字符串型变量中的字符串,可分为变长字符串和定长字符串。变长字符串的长度是不定的,它的长度取决于赋给它的字符串的长度;而定长字符串含有确定的字符串长度,它在变量后加上“* 数值”(字符个数)来表示。

变量的类型可包含在变量的名字中,也就是说变量的名字可决定它的类型。字符串型变量名的最后一个字符必须是“\$”符,而在数字型中,变量名还决定了它的精度。

例如: A\$="Computer" 意思是将字符串 Computer 赋给字符串变量 A\$, 这里 \$ 符是变量类型的说明符, 它说明了该变量是一个字符串型变量。

可以通过数字型变量的名字来说明它的值是整型、长整型、单精度还是双精度的。尽管用整型和单精度的变量进行运算得到的精度较低, 但是我们还是希望为变量规定严格的精度。精度高的变量占有较多的存储空间, 且需花费较多的运算时间, 这对空间内存容量是不利的。一个需要反复运算的程序若采用整型变量, 运行起来要快一些。

下面给出数字型变量的各种类型说明符及其所需要的存储空间字节数(8 位二进制数字组成一个字节):

% 整型变量(2 字节)

& 长整型变量(4 字节)

! 单精度变量(4 字节)

双精度变量(8 字节)

如果变量类型未被说明, 则这个变量就是单精度的。

例如: A% 整型变量 A

P# 双精度变量 P

B& 长整型变量 B

NA\$ 字符串变量 NA

MN! 单精度变量 MN

ACD 单精度 ACD

1.4.3 定义类型语句

1) TYPE 自定义记录类型语句

可以利用 TYPE 语句, 自定义新的记录数据类型。

格式:

TYPE 记录名

元素名 AS 类型

元素名 AS 类型

.....

END TYPE

说明:

(1) “记录名”和“元素名”的命名规则, 与变量的命名规则相同。

(2) “类型”可以是 INTEGER(整型)、LONG(长整型)、SINGLE(单精度)、DOUBLE(双精度)、STRING * n(定长字符串型)或自定义记录类型。

(3) 在 TYPE 自定义记录类型中, 不能含有数组。定义好后的记录类型, 可以用它说明另一记录(变量)类型。

例如:

TYPE Date

Year AS INTEGER

Month AS STRING * 3

Day AS INTEGER

END TYPE

以上 TYPE 定义了一个记录类型,记录名是 Data,记录中含有三个元素:Year、Month 和 Day。其中 Year 和 Day 是整型,Month 是定长字符串型,其长度为 3 个字符。

2) CONST 定义常量语句

格式:

CONST 常量名=表达式[,常量名=表达式](表)

说明:

(1) “常量名”是常量的名字,按变量的命名规则命名。“表达式”可由数字、运算符或字符串等组成。当对多个常量定义时,各式之间要用逗号分隔。

例如: CONST Max%=25,Min=3.1416

(2) 常量可以通过将常数赋给变量来实现,但定义后的常量不会因疏忽而改变,使程序设计更加可靠。

(3) 当常量定义后,同名量不能再用作变量。如: CONST n=60,则 n%、n& 等不能再用作变量名。

3) “declare”定义变量类型语句

可以通过变量名尾部的类型说明符说明变量的类型,也可通过定义语句来定义变量的类型。

格式:

“declare”变量名 AS 类型[,变量名 AS 类型](表)

说明:

(1) “declare”是一类说明型语句的统称,它可以是 DIM 语句、REDIM 语句、COMMON 语句、SHARED 语句或 STATIC 语句。

(2) “类型”可以是 INTEGER(整型)、LONG(长整型)、SINGLE(单精度)、DOUBLE(双精度)、STRING(字符串型)或自定义记录类型。

例如:

DIM a AS LONG 定义变量 a 为长整型变量。

DIM s AS STRING * 10 定义变量 s 为定长(10 个字符)字符串型变量。

DIM Ston AS date 定义变量 Ston 为记录 date 类型(如果记录 date 已经定义)。

4) DEF 定义变量类型语句

格式:

DEFtype 字母—字母[,字母—字母](表)

说明:

(1) “DEFtype”中的 DEF 是关键字, type 是变量类型标志符,可以是 INT(整型)、LNG(长整型)、SNG(单精度)、DBL(双精度)和 STR(字符串型)。

(2) “字母”是英文中的 26 个英文字母,大小写均可。字母可以作为该定义类型的变量名,而且以该字母开头的变量名,也是这种类型的变量。如:

DEFINT C—F 定义变量 C、D、E、F 为整型变量,而且以这些字母开头的变量也是整型变量,如 Com、Dele、Fil 等也是整型变量。

(3) 定义变量类型语句一般放在程序开头。

综上所述,我们将定义变量的符号总结如下(见表 1-1)。

表 1-1

类型符号

类型符号	类型说明符	类型标志符	类型名称
整 型	%	INT	INTEGER
长整型	&	LNG	LONG
单精度	! 或省缺	SNG	SINGLE
双精度	#	DBL	DOUBLE
字符串型	\$	STR	STRING
说 明	写在变量尾	DEF 语句中	TYPE, declare 中

注意:

(1) 变量名相同但类型说明符不同时,所表示的是不同的变量。例如 A、A %和 A&. 是不同的三个变量。

(2) 当一个变量没有类型符号说明时,就是一个单精度变量。

(3) 类型说明符总是比 DEFtype 定义类型语句优先起作用。

1.4.4 作用域及其语句

在结构化程序设计中,尤其是设计较复杂的程序时,通常把一个程序分成一个主程序(又称“模块”)部分和若干个相对独立的子程序(又称“过程”)部分。和一般 BASIC 语言不一样,在 Quick BASIC 中的变量和常量,存在着在不同的程序部分是否有效的问题,这就是变量和常量的作用域问题。变量和常量的作用域可以通过相应的语句给予定义。

1) 局部量

局部量包括有局部变量和局部常量。凡是在程序过程中定义的量称为局部量,局部量只有在相应的过程中才是有效的,即它的作用域只是在过程的内部。实质上,局部量只是在该过程被调用时才具有确定的存储单元,当程序返回模块后,局部量的存储单元被释放,从而变得无意义。因此,使用局部量不必担心所用的量名是否同模块或其它过程中的量名一样而造成混乱,从而大大提高程序设计的可靠性。由于局部量只在相应的过程中有效,说明和修改局部量是容易的,并且不影响程序的其它部分工作,因此保证了程序的易读性和增强了程序的可维护性。

局部量可用以下方式定义:

(1) 在模块中没有、而在过程中出现和使用的量,一般可作为局部量。

(2) STATIC 局部变量语句

格式:(过程中)

STATIC 变量[,变量](表)

说明:

STATIC 语句只能放在需定义的过程中。若该量已定义为全局量,STATIC 仍可将它作为同名的局部量定义使用。STATIC 语句定义的局部变量,在过程调用中保留变量的值,成为非自动变量(见后)。

例如:

STATIC a,b 将变量 a、b 定义为局部变量,并在过程调用中保留变量的值(非自动变量)。

2) 全局量