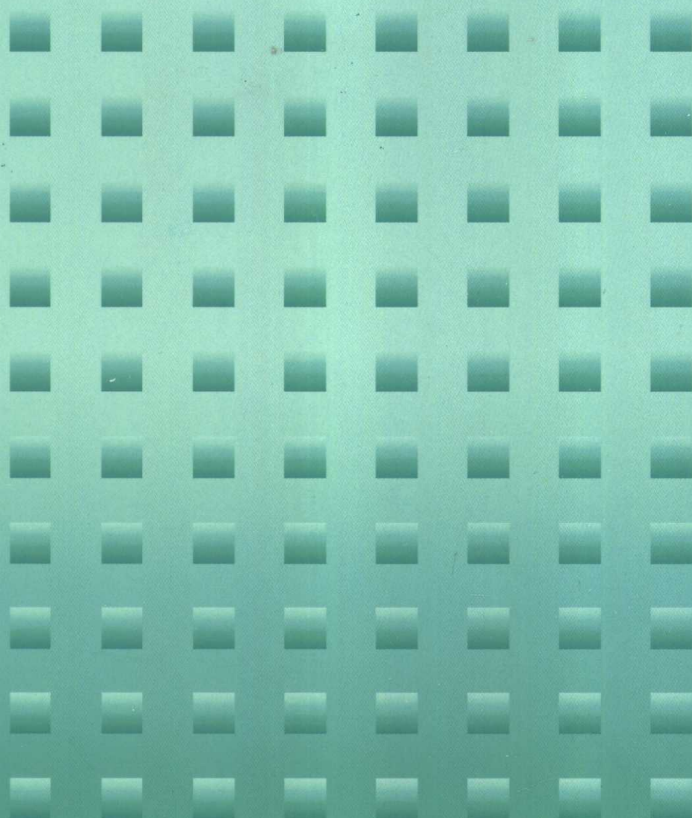


QUICK BASIC 与 VISUAL BASIC 语言

常用化学计算方法

徐洪文 编著



化学工业出版社
教材出版中心

QUICK BASIC 与 VISUAL BASIC 语言

常用化学计算方法

徐洪文 编著

化学工业出版社
教材出版中心
·北京·

(京)新登字 039 号

图书在版编目(CIP)数据

QUICK BASIC 与 VISUAL BASIC 语言 常用化学
计算方法/徐洪文编著. —北京: 化学工业出版社,
2003.4

ISBN 7-5025-4365-1

I. Q… II. 徐… III. BASIC 语言-应用-化学-计
算-高等学校-教材 IV. 06-04

中国版本图书馆 CIP 数据核字(2003)第 019778 号

QUICK BASIC 与 VISUAL BASIC 语言
常用化学计算方法

徐洪文 编著
责任编辑: 何曙霓
文字编辑: 周 红
责任校对: 蒋 宇
封面设计: 于 兵

*

化学工业出版社 出版发行
教材出版中心

(北京市朝阳区惠新里 3 号 邮政编码 100029)

发行电话: (010) 64982530

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

*

新华书店北京发行所经销

北京市管庄永胜印刷厂印刷

三河市东柳装订厂装订

开本 787 毫米×1092 毫米 1/16 印张 14½ 字数 352 千字

2003 年 5 月第 1 版 2003 年 5 月北京第 1 次印刷

ISBN 7-5025-4365-1/G · 1184

定 价: 23.00 元

版权所有 违者必究

该书如有缺页、倒页、脱页者, 本社发行部负责退换

前 言

计算机现已成为高等学校学生非常喜欢的工具。作为一名化学类专业的大学生,除了会用计算机进行文字编辑外,能用一种计算机语言进行程序设计,解决化学专业中较为复杂的计算是应该掌握的基本功。目前在化学计算中经常碰到的数学问题主要有方程求根、函数插值、线性方程组的求解、一元或多元回归、积分方程和微分方程的数值解等问题。学习过高等数学的大学生对于这些数学问题的理解并不十分困难,但对这些问题的求解过程进行程序设计是需要学习和训练的,同时为比较同一算法不同解法之间优劣差异,本书对某些问题进行求解推照,希望读者比较分析。

Quick Basic 和 Visual Basic 语言都是比较容易学习的计算机语言,Quick Basic 在 DOS 操作系统下使用,Visual Basic 在 Windows 操作系统下使用,用 Quick Basic 和 Visual Basic 语言编程其编辑环境都非常友好,随时给你提示编程中出现的语法问题,只要你正确地输入本书中每一个程序,一定会得到满意的结果。

计算机求解问题是由程序来实现的,而对实际问题的求解进行程序设计是要由人来完成的,程序设计需要用逻辑思维来考虑计算机解决问题的可行性,即你设计的解题步骤计算机必须能一步一步实现。同时还要考虑你设计的每一步解题程序会不会产生分母为零的情况,会不会产生负数开平方或负数求对数情况,会不会产生数据类型运算超界溢出情况,会不会产生逻辑判断前后矛盾情况、会不会产生参数传递匹配失误情况。在输入程序时也要非常认真仔细,应配套的语句一定要配套,例如条件语句、循环语句等;注意区别字母“o”与数字“0”,字母“i”和字母“l”与数字“1”。刚开始学习编程时要多上机模仿练习,练习多了自然就掌握了其中的一些共性的编程思想。

一个合格的程序除了能获得正确的计算结果外,其格式设计应该是结构化的,并且其程序的可读性要好。结构化的程序容易移植,程序的可读性好便于交流,这是信息化社会人们的一个共同要求。

要想学好程序设计,必须耐心看懂程序的每一行,并且想一想:

- (1) 为什么那样设计?有什么数学计算要求?
- (2) 能不能用另外一种方式取代?

当你确实看懂了后就设法用另外一种方式改变它,比如改变输入输出格式、改变逻辑判断次序、改用另一种循环方式等。只要你坚持做下去,一定会在较短的时间真正掌握计算机语言。

学习程序设计能培养严格的逻辑思维方式和耐心细致的性格,学好常用化学计算方法并看懂程序,需要高等数学、英语、计算机语言和化学专业综合知识,因此学好常用化学计算方法需要其他相关学科知识的支持。当今信息社会知识更新速度非常快,我们今天的努力学习,就是为今后的再(或继续学习)学习打个基础,构成大学基础的每一门课,或许是你喜欢的或许是你不喜欢的,但都是不可缺的,都是需要认真学习的。

学习和实践是我们一生的事业。

内 容 简 介

本书主要介绍了 QUICK BASIC 与 VISUAL BASIC 语言在常用化学计算方法上的具体应用。全书分四章,分别讲述了 QB 语言的使用、常用化学计算方法及 QB 程序、VB 语言的使用和常用化学计算方法及 VB 程序。书中重点和难点分析透彻清晰,化学计算中涉及的方程求根、线性方程组求解、插值、回归分析、微分和积分方程的数值解等数学方法都有实例并结合程序讨论,简明易懂。

本书适合各高等院校化学与化工类学生、教师作选用教材,也可供有关科技人员及计算机爱好者阅读参考及使用。

目 录

第 1 章 QUICK BASIC 语言的使用	1
1.1 QUICK BASIC 环境	1
1.1.1 QUICK BASIC 的启动和退出	1
1.1.2 QUICK BASIC 工作窗口介绍	2
1.1.3 QUICK BASIC 程序的编辑与运行	2
1.2 QUICK BASIC 程序设计初步	4
1.2.1 QUICK BASIC 程序结构	4
1.2.2 数据类型	4
1.2.3 符号常量	5
1.2.4 变量	6
1.2.5 变量的作用域	7
1.2.6 算术运算符、关系运算符和逻辑运算符	9
1.2.7 内部函数和表达式的运算顺序	10
1.3 输入输出语句	12
1.3.1 数据输入语句	12
1.3.2 结果输出语句	14
1.4 分支结构	17
1.4.1 行 IF 语句	17
1.4.2 块 IF 语句	17
1.4.3 SELECT CASE 语句	19
1.4.4 GOTO 语句	20
1.5 循环语句	20
1.5.1 FOR-NEXT 循环	20
1.5.2 DO-LOOP 循环	22
1.5.3 WHIL-EWEND 循环	24
1.6 SUB 过程和 FUNCTION 过程	27
1.6.1 SUB 过程	27
1.6.2 FUNCTION 过程	32
1.6.3 FUNCTION 过程递归	36
1.6.4 STATIC 语句和 STATIC 属性	38
1.7 数据文件	39
1.7.1 顺序文件的打开和关闭	40
1.7.2 顺序文件的建立与使用	40
第 2 章 常用化学计算方法及 QB 程序	43
2.1 方程求根	43

2.1.1	二分法	43
2.1.2	迭代法	44
2.1.3	牛顿法	47
2.1.4	割线法	52
2.2	函数插值	54
2.2.1	线性插值	54
2.2.2	拉格朗日三点插值	56
2.2.3	拉格朗日全节点插值	59
2.2.4	埃特金插值	61
2.2.5	拉格朗日二元三点插值	64
2.3	线性方程组的求解	68
2.3.1	列主元高斯消去法	68
2.3.2	高斯-约当消去法	74
2.3.3	逆矩阵法	77
2.3.4	解线性方程组的一般迭代法	81
2.3.5	解线性方程组的高斯-赛德尔迭代法	83
2.4	回归分析	85
2.4.1	一元线性回归	85
2.4.2	多元线性回归	88
2.4.3	多项式回归	93
2.5	数值积分	96
2.5.1	定步长梯形积分法	96
2.5.2	变步长梯形积分法	98
2.5.3	辛普森积分法	101
2.5.4	龙贝格积分法	106
2.5.5	离散点下的定积分	109
2.6	常微分方程的数值解	112
2.6.1	欧拉(Euler)法及改进欧拉法	112
2.6.2	龙格-库塔法	117
2.6.3	一阶常微分方程组的数值解	121
第3章	VISUAL BASIC 语言的使用	125
3.1	VISUAL BASIC 环境	125
3.1.1	VISUAL BASIC 的启动和退出	125
3.1.2	VISUAL BASIC 窗口介绍	126
3.1.3	可视化编程方法	128
3.2	Visual Basic 程序设计基础(一)	130
3.2.1	数据类型	130
3.2.2	变量	130
3.2.3	运算符和表达式	131
3.2.4	内部函数	131

3.2.5	输入数据方式	132
3.2.6	输出数据方式	133
3.3	Visual Basic 程序设计基础 (二)	138
3.3.1	条件选择结构	138
3.3.2	各种循环结构	141
3.3.3	数组的使用	145
3.4	Visual Basic 程序设计基础 (三)	150
3.4.1	过程的概念	150
3.4.2	通用过程	152
3.4.3	递归	157
3.4.4	变量及过程的作用域	158
3.4.5	多窗体应用实例	165
3.4.6	数据文件的生成、读取和调用	168
第4章	常用化学计算方法及 VB 程序	171
4.1	方程求根	171
4.1.1	二分法	171
4.1.2	迭代法	172
4.1.3	牛顿法	174
4.1.4	割线法	176
4.2	函数插值	178
4.2.1	线性插值	178
4.2.2	拉格朗日三点插值	180
4.2.3	拉格朗日全节点插值	182
4.2.4	埃特金插值	184
4.2.5	拉格朗日二元三点插值	185
4.3	线性方程组的求解	188
4.3.1	列主元高斯消去法	188
4.3.2	高斯-约当消去法	190
4.3.3	逆矩阵法	192
4.3.4	解线性方程组的一般迭代法	194
4.3.5	解线性方程组的高斯-赛德尔迭代法	195
4.4	回归分析	197
4.4.1	一元线性回归	197
4.4.2	多元线性回归	198
4.4.3	多项式回归	201
4.5	数值积分	203
4.5.1	定步长梯形积分法	203
4.5.2	变步长梯形积分法	204
4.5.3	辛普森积分法	206
4.5.4	龙贝格积分法	207

4.5.5 离散点下的定积分	208
4.6 常微分方程的数值解	211
4.6.1 欧拉(Euler)法及改进欧拉法	211
4.6.2 龙格-库塔法.....	214
4.6.3 一阶常微分方程组的数值解	216
参考文献.....	220

第 1 章 QUICK BASIC 语言的使用

1.1 QUICK BASIC 环境

1.1.1 QUICK BASIC 的启动和退出

1.1.1.1 在 DOS 环境下的启动

如果用户使用的计算机硬盘上安装了 MS-DOS 5.0 或更高的 DOS 版本，在子目录 DOS 中就已包含了 QBASIC。开机后，使当前盘为 C 盘，然后按以下步骤启动 QBASIC。

C:\>CD\DOS✓（进入 DOS 子目录）

C:\DOS>QBASIC✓（将 QBASIC 调入内存）

此时进入 QUICK BASIC 屏幕，而后就可在此屏幕下编辑运行程序。

如果用户使用的计算机硬盘上未安装 MS-DOS 5.0 或更高的 DOS 版本，用户可自己在 C 盘上建立一个子目录，然后将 QBASIC 文件（或将 QUICK BASIC 高版本系统）拷贝到该子目录下，在该子目录下键入命令“QBASIC”（或“QB”）就可启动。

1.1.1.2 在 WINDOWS 环境下的启动

从“开始”菜单中选择“程序”，然后选择“MS-DOS 方式”，屏幕上出现 DOS 提示符，输入 C:\>CD\DOS✓

C:\>DOS>QBASIC✓（在 DOS 子目录中包含 QBASIC 系统）

即可进入 QUICK BASIC 屏幕。见图 1-1。

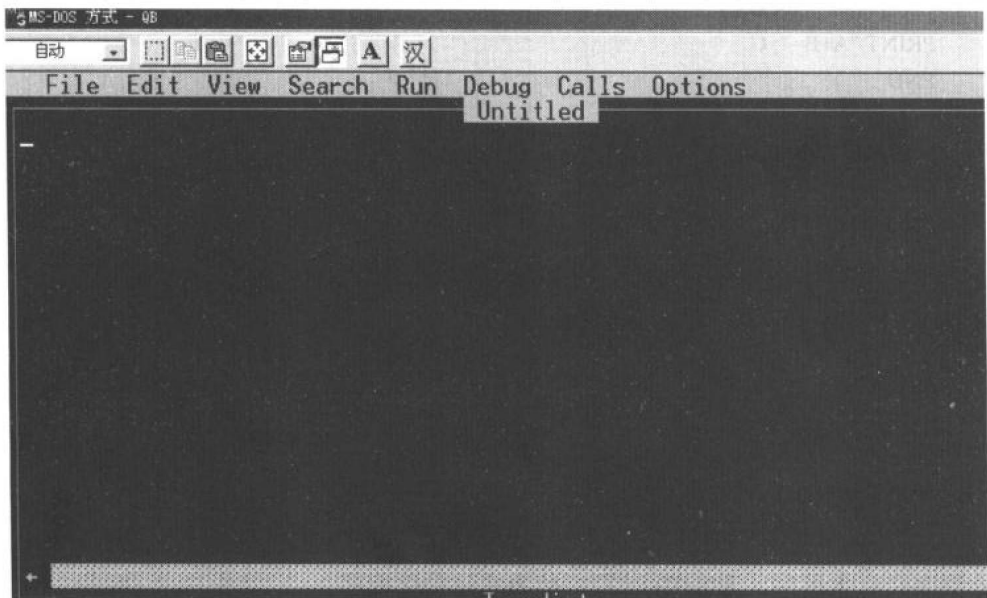


图 1-1

当编辑计算结束后可以从菜单 File 中选择 Exit 项退出，返回 DOS 系统子目录下。

1.1.2 QUICK BASIC 工作窗口介绍

当在存有 QUICK BASIC 系统的子目录下输入“QB”(或“QBASIC”)后回车,即可进入 QUICK BASIC 屏幕。

屏幕各部分说明如下:

顶部为菜单栏,共有 File, Edit, View, Search, Run, Debug, Calls, Option 等八项;

屏幕上上部为观察窗口,此窗口用于输入、编辑程序,同时程序名显示于该窗口上界线的中部,如果程序没有命名,则显示为 Untitled;

在窗口的下部和右部各有一个滚动条,显示光标所在的位置;

屏幕中下部为立即窗口(按 F6 键使光标在观察窗口与立即窗口切换),在此窗口中输入“QB”命令表达式或函数后回车,结果将立即显示出来,供人们调试程序时验证某些命令表达式用。

QB 编程操作可以通过选择菜单来实现,按 Alt 键进入菜单条(或用鼠标选择),按 Esc 键可以从菜单内退到原工作状态。Alt 键是一个切换开关,按一次 Alt 键进入菜单条,再按一次 Alt 键退出菜单条,进入菜单条后有两种方式选择所需操作:

- ① 用方向键“←”,“→”移动高亮条到被选择项上后回车;
 - ② 按下菜单项中的第一个字母(该字母与其他字母颜色不一样),就选中该菜单。
- 然后用方向键“↑”,“↓”移动选择子菜单。

1.1.3 QUICK BASIC 程序的编辑与运行

【例 1-1】(Q111) 求 A+B 的和。

```
DECLARE SUB JIAFA(A,B,C)    此行不用输入,存盘时自动生成
A=123
B=234
CALL JIAFA (A,B,C)
PRINT "A+B="; C
END

SUB JIAFA (A,B,C)
C=A+B
END SUB
```

此程序有主程序和 SUB 过程,故为结构化程序。

【例 1-2】(Q112) 求 A 和 B 的和。

```
A=123
B=234
C=A+B
PRINT "A+B="; C
END
```

此程序只有主程序,无 SUB 过程,故为非结构化程序。

这两个程序运行结果相同,但例 2 程序没有 SUB(或 FUNCTION)过程。对于简单程序可以没有 SUB(或 FUNCTION)过程,但对于复杂程序,利用 SUB 或 FUNCTION 过程会使程序清晰易读,便于交流,所以本书程序不论简单与否,都设计为结构化程序。

QB 程序的输入（应首先输入主程序）：

以例 1-1 程序为例，首先选择 File 菜单中的 New Program 子菜单，回车，即可在编辑窗口输入主程序，如图 1-2。

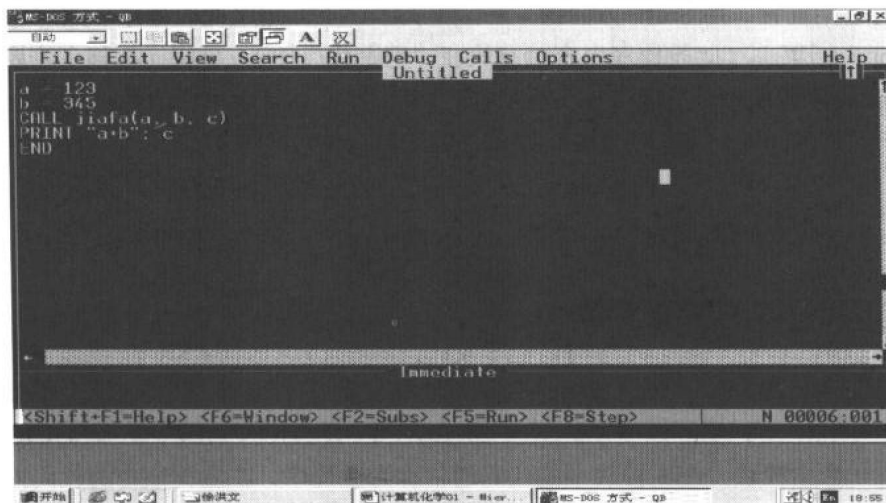


图 1-2

而后选择 Edit 菜单中的 New Sub 子菜单，弹出一个对话框，在 Name 框中添上过程名 JIAFA，单击 OK，进入 SUB JIAFA — END SUB 编辑窗口，在 SUB JIAFA 后面添上 (A, B, C)，在其随后一行上输入 C = A+B，如图 1-3。

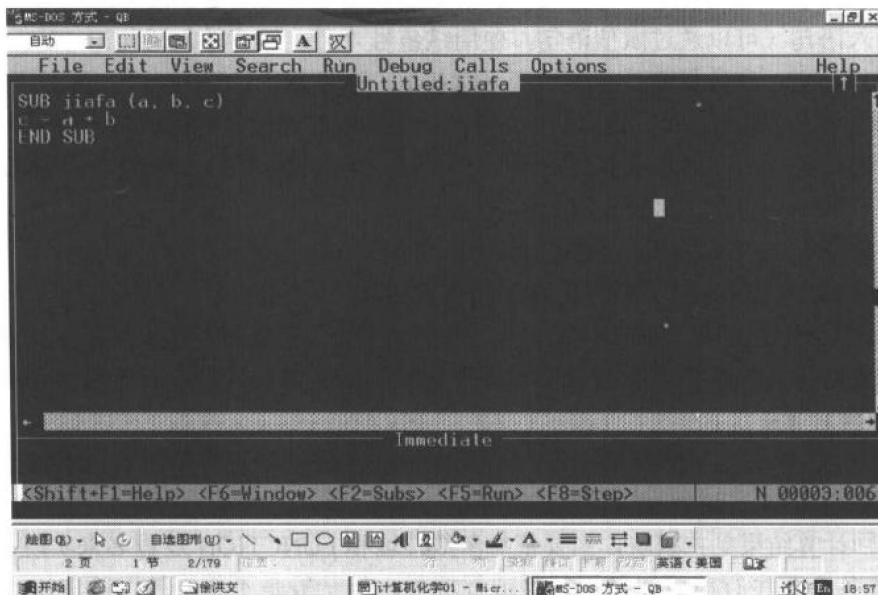


图 1-3

程序输入完后一定要先保存起来，选择 File 菜单 Save As 子菜单，弹出对话框，在 File Name 对话框中输入盘符 \ 子目录 \ 文件名.BAS，单击 OK，文件就保存好，要注意“过程”不是以独立磁盘文件方式存在，它需要与包含它的主程序一起作为文件存入磁盘。

而后选择 Run 菜单中 Start 子菜单（或按 F5 键），如程序无错，即可看到运行结果，如程序有错，系统马上提示出错信息。改正错误后，再按 F5 键，可看到改正后的运行结果。

在编辑程序时，有时需要将某一段程序拷贝到程序的另一个位置，这时首先把光标移到要定义的起始位置，用 Shift+方向键（←，↓，↑，→）定义此段程序，然后选择 Edit 菜单中的 Copy 子菜单（此段程序内容就被放在“剪贴板”中），再将光标移到需要粘贴的位置，选择 Edit 菜单 Paste 子菜单，此段程序内容就被粘贴到光标所在位置。如选择 Edit 菜单中的 Cut 子菜单，此段程序内容也被放在“剪贴板”中，但在原处定义的此段程序就被剪掉了。

1.2 QUICK BASIC 程序设计初步

1.2.1 QUICK BASIC 程序结构

QUICK BASIC 程序应该是结构化程序设计，即在 QUICK BASIC 中应该包括一个主模块（又称之为主程序）和一个（或多个）子模块（又称之为过程）。在过程里的代码通常以 SUB 或 FUNCTION 开头，以 END SUB 或 END FUNCTION 结尾；一个 QUICK BASIC 程序只能有一个主模块，在主模块中的代码通常称为主程序。在主程序中的代码内容一般应包括：

- ① DECLARE 语句（说明引用的 SUB 或 FUNCTION 过程传递的参数，用户不用专门输入，此语句在存盘时自动生成）；
- ② 注释语句（REM 或用撇号“'”）用于注释程序的某些功能；
- ③ 变量类型说明语句（当一个变量没有类型说明符时，将被隐含地说明为单精度变量），本书中化学计算主要用到整型变量、单精度型变量和数组定义语句；
- ④ 输入语句（可以通过赋值语句，使用赋值号“=”、键盘输入语句 INPUT、读数置数语句 READ / DATA 和数据文件给各种变量赋初值，以便用于计算）；
- ⑤ 调用 SUB 或 FUNCTION 过程语句（以便进入过程做各种运算）；
- ⑥ 输出语句（主要通过屏幕显示语句 PRINT 和格式显示语句 PRINT USING 输出计算结果）；
- ⑦ 结束语句（程序结束时必须加上 END 语句）。

SUB 或 FUNCTION 过程代码内容一般应包括：

- ① 变量类型说明语句和数组定义语句（仅说明和定义在本过程使用的局部变量和数组）；
- ② 功能运算，主要通过顺序语句、条件语句和循环语句的组合表达数学计算方程式，完成各种数学计算；
- ③ 调用 SUB 或 FUNCTION 子过程语句，在子过程中也可以再调用本程序中的其他 SUB 或 FUNCTION 子过程，以便完成复杂的运算；
- ④ 返回计算结果到主模块，主要依靠参数传址和 FUNCTION 过程名来实现；

在上述各个程序行前可以使用行号或行标号，但一般是不使用的，除非用到无条件转移语句 GOTO，才不得不使用行号或行标号。

行号是由 0 到 65529 的整数组成（如 5，10，300 等），行标号是由字母和数字组成（如 A1，B34 等）。

1.2.2 数据类型

基本数据类型分为字符串型数据（变长字符串和定长字符串）和数值型数据，本书仅

讨论数值型数据，数值型数据又分为整型、长整型和实型，实型数从精度上看又有单精度和双精度之分。

① 整型数（2 字节）存储为一个有符号的 16 位二进制数，范围是从-32768 ~ +32768。

② 长整型数（4 字节）存储为一个有符号的 32 位二进制数，范围是从-2147483648 ~ +2147483647。

③ 单精度浮点数（4 字节）其精度准确到 7 位十进制数，范围是：负数从-3.402823E+38 ~ -1.40129E-45；正数是从 1.40129E-45 ~ 3.402823E+38。

④ 双精度浮点数（8 字节）其精度准确到 15 或 16 位十进制数，范围是：负数从-1.79763134862316D+308 ~ -4.94065D-324；正数是从 4.94065D-324 ~ 1.79763134862316D+308。

例如：

23D+23 属于双精度型，3456.21E+11 属于单精度型

23471 属于整型，-387648 属于长整型

输入 a=123.456789!，按回车键后立刻变为 a=123.4568

输入 a=1231.45678e6，按回车键后立刻变为 a=1231456780#

输入 a=1234.435675，按回车键后立刻变为 a=1234.435675#

输入 a=1453.6689e30，按回车键后立刻变为 a=1.4536689d+33

1.2.3 符号常量

在主程序中用 CONST 语句定义的常数变量是全局量，在本程序的各个过程都可以直接使用；而在过程中使用 CONST 语句定义的常数变量是局部量，在主程序和其他过程是不能直接使用的。

如在主程序中用 CONST 语句定义变量 PI 和 Kw，其变量是全局量：

```
CONST PI = 3.1415926, Kw = 1.4E-14
```

如在主程序中使用赋值语句给变量 PI 和 Kw 赋值，其变量是局部量：

```
PI = 3.1415926, Kw = 1.4E-14
```

因此，修改 CONST 语句定义的变量的值，将影响整个程序。

【例 1-3】(Q121)

```
DECLARE SUB HOT(Y)
REM 求和                      REM 为注释语句
CONST A=3                      A 是全局量
Y=2
CALL HOT(Y)
PRINT "Y=" ; Y
END
SUB HOT(Y)
Y=Y+A
END SUB
```

运行程序（按 F5 键），显示：

Y=5

【例 1-4】 (Q122)

```

DECLARE SUB HOT(Y)
CALL HOT(Y)
PRINT "Y="; Y, "A="; A
END
SUB HOT(Y)
CONST A=3           A 是局部量
Y=A+7
END SUB

```

运行程序（按 F5 键），显示：

```
Y=10      A=0
```

1.2.4 变量**1.2.4.1 变量的构成**

QUICK BASIC 的变量名由字母，数字和小数点组成，变量名必须以字母开头，最后一个字符也可以是类型说明符（如变量名字后加 %，表示该变量为整型，加 &，表示该变量为长整型，加 ! 表示该变量为单精度型，加 # 表示该变量为双精度型）。

例如：A, CH2, AMOT, W%。

变量名不能是 QUICK BASIC 中的保留字（例如 SIN, DATA, CONST 等就是非法的），但可以是嵌入保留字的字符串（例如 PSIN, DATAVOLUM, ACONST 就是合法的）。同时，变量名也不能是末尾带有类型说明符的保留字（DATA\$）。

变量名还应该与过程名，符号常量名和 TYPE 语句中用户定义的类型名、元素名有区别。

1.2.4.2 变量的类型**(1) 用类型说明符来标识**

把类型说明符放在变量的末尾，可以表示不同的变量类型，其中 % 表示整型，! 表示单精度型，# 表示双精度型，& 表示长整型，\$ 表示字符串型。

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：AMAX%, AMAX&, AMAX# 分别是三个不同的变量。

(2) 用 DIM 变量名 AS 类型来定义

```
DIM A AS INTEGER, B AS LONG, C2 AS SINGLE
```

```
DIM W3 AS DOUBLE, AREA AS STRING
```

INTEGER 表示整型，LONG 表示长整型，SINGLE 表示单精度，DOUBLE 表示双精度，STRING 表示字符串。

(3) 用 DEFtype 来定义

DEFINT A~D 表示以 A, B, C, D 字母开头的变量名为整型(INT);

DEFSNG E~F 表示以 E, F 字母开头的变量名为单精度型(SNG);

DEFDBL G~J 表示以 G, H, I, J 字母开头的变量名为双精度型(DBL);

DEFLNG K~L 表示以 K, L 字母开头的变量名为长整型(LNG);

DEFSTR M~O 表示以 M, N, O 字母开头的变量名为字符串型(STR)。

注意：类型说明符（%, &, !, #, \$）比 DEFtype 语句优先起作用，实际应用中，应根

据实际需要来设置变量的类型。在一般化学计算中使用单精度变量就够了，如设置为双精度变量即浪费内存空间，又影响处理速度。

1.2.4.3 数组变量

数组是一批具有相同类型且彼此间存在先后顺序关系的集合变量，其中各个元素间是用下标来区分的。数组的命名规则与简单变量命名规则完全相同。

一切未定义就使用的数组都是隐含定维数组，QB 系统对这种数组一律分配给 11 个元素（即 0~10 各分量）。大于 11 个元素的数组在使用前要先定义说明。数组只有经过定义后，编译程序才能为其分配存储空间。

数组的维数就是数组下标的个数，例如：一维数组只有一个下标，二维数组有两个下标，以此类推。如果小于 10 的数组，可以直接使用而不用定义（但最好定义，以便养成对数组变量先定义后使用的习惯）。

```
DIM A(2 TO 20) AS INTEGER
```

```
DIM SHARED B(1 TO 13, 30 TO 90)
```

```
DIM C(5, 3 TO 6, 4 TO 15)
```

其中数组 A() 为整型一维数组，下界为 2，上界为 20，共可安排 $(20 - 2 + 1) = 19$ 个元素；数组 B() 为具有共享属性（SHARED 为共享属性选择项）二维数组，表示模块中的所有过程都可以共用该数组，说明该数组变量是一个全局量，共可安排 $(13 - 1 + 1) \times (90 - 30 + 1) = 793$ 个元素；数组 C() 为三维数组，共可安排 $(5 - 0 + 1) \times (6 - 3 + 1) \times (15 - 4 + 1) = 288$ 个元素。同样下面的定义也是正确的：

```
DIM A(20) AS INTEGER
```

```
DIM SHARED B( 13, 60)
```

```
DIM C(5, 8, 15)
```

此时数组 A() 为整型一维数组，共可安排 $(20 - 0 + 1) = 21$ 个元素；数组 B() 为具有共享属性二维数组，共可安排 $(13 - 0 + 1) \times (60 - 0 + 1) = 854$ 个元素；数组 C() 为三维数组，共可安排 $(5 - 0 + 1) \times (8 - 0 + 1) \times (15 - 0 + 1) = 864$ 个元素。

可以省略下界和关键字 TO，但此时下界是从 0 开始。

1.2.5 变量的作用域

在 QUICK BASIC 中使用的变量主要有三种类型：即全局变量、局部变量和共享变量。这三种变量的使用使得结构化程序设计成为可能，因此必须掌握这三种变量的使用。

1.2.5.1 全局变量

在主程序中使用 DIM SHARED 就可以定义全局变量，例如：

```
DIM SHARED A, B, AREA
```

此时变量 A, B, AREA 就被定义为全局变量，变量 A, B, AREA 可以在程序中的任何地方被引用。

【例 1-5】 (Q123)

```
DECLARE SUB SS(Y)
```

```
DIM SHARED A
```

A 是全局量

```
A=5
```

```
CALL SS(Y)
```

```
PRINT "Y2=" ;Y
```



```

END
SUB SS(Y)
PRINT "Y1=";Y
Y=Y+A
END SUB

```

运行程序（按 F5 键），显示：

```

Y1=0
Y2=5

```

1.2.5.2 局部变量

局部变量存在于主程序代码或过程级代码中，其作用域为所在的模块或过程。使用 DIM 语句可以定义局部变量。

【例 1-6】 (Q124)

```

DECLARE SUB SS(C)
A = 5                局部变量
C = 1                局部变量
CALL SS(C)
PRINT "SECOND " "A="; A; "B="; B; "C="; C
END
SUB SS(C)
B = 10              局部变量
C = C+A
PRINT "FIRST " "A ="; A; "B ="; B; "C ="; C
END SUB

```

运行程序（按 F5 键），显示：

```

FIRST  A = 0  B = 10  C = 1
SECOND A = 5  B = 0  C = 1

```

此程序中 C 也是局部变量，但它处于变量传递之中。

如果在主程序中已经将一个变量 A 定义为全局量(DIM SHARED A)，但又想在某个过程中把它作为同名的局部变量使用，则可以在该过程中通过 STATIC 语句来加以说明(STATIC A)。STATIC 语句说明变量为静态变量，它只能放在需要定义的过程中，说明本过程使用的简单变量或数组变量是局部变量，并在调用时保留该变量或数组变量的值。

【例 1-7】 (Q1241)

```

DECLARE SUB SS(Y)
DIM SHARED X
X=5
FOR k=1 TO 3
CALL SS(Y)
NEXT k
PRINT "Y="; Y
END

```

(Q1242)

```

DECLARE SUB SS(Y)
DIM SHARED X
X=5
FOR k=1 TO 3
CALL SS(Y)
NEXT k
PRINT "Y="; Y
END

```