

高等院校计算机基础
教育系列课程教材

FORTRAN77语言 程序设计

余冬梅 主编 韩卫 副主编

王肇荣 主审



中国科学技术出版社

高等院校计算机基础教育系列课程教材

FORTRAN 77 语言程序设计

余冬梅 主 编

韩 卫 副主编

王肇荣 主 审

中国科学技术出版社

• 北京 •

图书在版编目(CIP)数据

FORTRAN77 语言程序设计/余冬梅主编·—北京:中国科学技术出版社,1996.8

ISBN 7-5046-2201-X

I. F… II. 余… III. FORTRAN 语言-程序设计 IV. TP312F0

中国版本图书馆 CIP 数据核字 (96) 第 04268 号

中国科学技术出版社出版

北京海淀区白石桥路 32 号 邮政编码:100081

新华书店北京发行所发行 各地新华书店经售

顺义康华印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:13.5 字数:320 千字

1996 年 8 月第 1 版 1996 年 10 月第 2 次印刷

印数:5001—10000 册 定价:16.00 元

内 容 提 要

在计算机应用技术飞速发展的今天,FORTRAN 语言仍然是当今应用最广并深受用户欢迎的一种程序设计语言,尤其是在科学计算方面更具特色。

全书共分八章,分别介绍了 FORTRAN 语言的数据类型、程序设计的三种基本结构(顺序、分支、循环),输入输出格式的设计方式,FORTRAN 的函数和子程序以及数据文件的存取方式。每一章都进行了小结并列举了大量的程序实例,每章还附有相应的习题。

本书可作为高等院校、大中专学校非计算机专业 FORTRAN 语言程序设计课程的教学用书、等级考试或自学参考书,也可作为各行各业初学计算机程序设计人员的入门读物。

机械工业部部属高校计算机基础教育 系列课程教材编审委员会

主任委员：张奠成

副主任委员：袁鹤龄 王杰臣 张明毫

委 员：郝忠孝 梁文林 朱逸芬 贝嘉祥

王肇荣 田瑞庭 陈金华

责任编辑：张秀智

封面设计：炎 尘

机械工业部部属高校计算机基础教育系列课程教材出版说明

《中国教育改革和发展纲要》指出：“要按照现代科学技术文化发展的新成果和社会主义现代化建设实际需要，更新教学内容，调整课程结构。”人类即将进入的 21 世纪，将是高度信息化的社会，作为这种社会的重要基础的计算机，将渗透到社会各个角落。计算机不仅作为一种工具来使用，而且作为一种文化来普及；计算机科学技术不仅是一门独立的学科，而且是所有学科知识结构中的重要组成部分。

在我国，多年来数、理、化及外语等基础教育在高等技术人才的素质培养中发挥了十分重要的作用。面对新的形势，加强高校非计算机专业计算机基础教育，将是促进高校教育质量提高的必要措施。在这方面，除了设备配置、师资培养、计划安排和课程设置等工作外，教材是加强计算机基础教育的基本建设。为此，机械工业部部属高校在总结多年来实施非计算机专业计算机基础教育的教学经验基础上，发挥知识群体优势，组织编写本系列教材，供有关专业选择使用。

根据目前各专业课程设置，本系列教材包括三个层次：即计算机基础及高级程序设计语言、微型计算机原理及应用、软件技术基础及计算机网络概论。考虑到不同专业对计算机知识要求的差异，部分教材将按不同学时编写。由于计算机技术发展迅速，教材内容将不断更新，因此第一轮教材安排在 1996 年秋季全部出齐，以便在使用过程中修改完善，使第二轮教材以更新和更好的面目问世。

本系列教材除适合于高校非计算机专业本科生、大专生使用外，也可用于成人教育及自学教育教学，同时也可作为工程技术人员自修计算机技术的指导参考。

机械工业部部属高校计算机基础教育系列
课程教材编审委员会
1995 年 6 月 5 日

前 言

计算机知识和计算机应用能力是当代大学生知识结构的重要组成部分,也是当今各行各业科技人员知识结构中不可缺少的内容。因此在非计算机专业中开展计算机基础教学是十分必要的。非计算机专业计算机教学的特点是以应用知识和应用能力的培养和训练为主要目标的,而程序设计则是计算机基础教育的基础和重点。

本书是部属院校非计算机专业计算机基础系列课程教材之一。作者在总结多年来的教学经验的基础上,按照国家教委颁布的《高等学校理工科类非计算机专业〈高级语言程序设计〉课程教学基本要求》,以及近年来计算机等级考试大纲的内容和要求精心编著而成的。本书力求语言精练通俗,选材举例简明适当,概念清晰。每章均有要点小结和程序举例。全书内容丰富,重在实用,便于自学。本书是以标准“FORTRAN 77”为背景,书中所列举的程序均在计算机上运行并通过。

全书共分八章,第一章介绍了 FORTRAN 语言的概况和 FORTRAN 程序的上机操作过程;第二章介绍了 FORTRAN 语言的基本元素,包括数据类型、运算符和表达式;第三章介绍了顺序结构程序设计的基本方法;第四章介绍了分支结构的程序设计;第五章介绍了循环结构的程序设计方法,其中包括三种循环结构(当型循环,直到型循环,DO 循环);第六章介绍了 FORTRAN 语言数据的输入、输出格式的设计方法;第七章介绍了 FORTRAN 的过程(函数和子程序);第八章介绍的是 FORTRAN 语言中数据文件的建立(将数据存入磁盘)和数据文件的访问(把数据从磁盘中取出)的操作过程和方式。另外还有附录,其中附录 I 介绍了 EDITOR(全屏幕编辑软件)的用法;附录 II 给出了常用字符与 ASCII 代码的对照表;附录 III 给出了 FORTRAN 77 的内部函数;附录 IV 给出了 FORTRAN 77 语句总览。

本书第一章及附录 I 由湖南大学黄国均编写,第二章由湖南大学张利平编写,第三、四章及附录 IV 由燕山大学韩卫编写,第五章及附录 II、III 由湖南大学余冬梅编写,第六章由哈尔滨科技大学吕励编写,第七章由湖南大学袁立香编写,第八章由湖南大学王荣湘编写。余冬梅负责全书的统稿。本书主编为余冬梅,副主编为韩卫,主审由西安理工大学王肇荣教授担任,王教授对本书全稿进行了认真的审阅,并提出了许多宝贵的修改意见。此外,本书还得到了编审委员会和合肥工业大学孙家启教授、湖南大学教务处及计算机系有关领导的大力支持和指导,在此向他们表示诚挚的感谢!

由于作者水平有限,若有错误或不妥之处,恳请读者批评指正。

编 者

1996 年 2 月

目 录

第一章 FORTRAN 语言的初步认识	(1)
1.1 FORTRAN 语言简介	(1)
1.2 简单的 FORTRAN 程序介绍	(2)
1.2.1 FORTRAN 程序实例	(2)
1.2.2 FORTRAN 程序的组成	(3)
1.3 FORTRAN 源程序的书写格式	(5)
1.4 算法与流程图	(5)
1.5 FORTRAN 源程序的上机操作过程	(7)
1.5.1 概述	(7)
1.5.2 源程序的编辑方法	(8)
1.5.3 程序的编译、连接及运行过程	(9)
小结	(10)
习题一	(11)
第二章 FORTRAN 语言的基本元素	(12)
2.1 FORTRAN 语言的数据类型	(12)
2.1.1 常数	(12)
2.1.2 变量	(15)
2.2 FORTRAN 语言的运算符	(19)
2.2.1 算术运算符与优先级	(19)
2.2.2 关系运算符	(19)
2.2.3 逻辑运算符	(20)
2.2.4 字符运算符	(20)
2.3 FORTRAN 语言的表达式	(20)
2.3.1 算术表达式	(20)
2.3.2 字符表达式	(22)
2.3.3 关系表达式	(22)
2.3.4 逻辑表达式	(23)
小结	(24)
习题二	(24)
第三章 顺序结构程序设计	(26)
3.1 顺序结构程序的算法表示	(26)
3.2 顺序执行语句的格式和用法	(27)

3.2.1	赋值语句	(27)
3.2.2	简单输入语句	(32)
3.2.3	简单输出语句	(35)
3.2.4	应用举例	(37)
3.3	参数说明语句(PARAMETER)	(40)
3.4	变量赋初值语句(DATA)	(41)
3.5	结束语句(END)	(45)
3.6	停语句(STOP)	(45)
3.7	暂停语句(PAUSE)	(46)
3.8	程序说明语句(PROGRAM)	(47)
3.9	无条件转移语句(GOTO)	(47)
	小结	(48)
	习题三	(48)
第四章	分支结构的程序设计	(51)
4.1	分支结构程序的算法表示	(51)
4.2	分支结构程序引例	(51)
4.3	关系表达式及关系运算符	(52)
4.3.1	关系运算符	(53)
4.3.2	关系表达式	(53)
4.4	逻辑表达式及逻辑运算符	(54)
4.4.1	逻辑表达式	(55)
4.4.2	逻辑表达式的运算规则	(55)
4.5	产生分支结构的语句及用法	(57)
4.5.1	逻辑 IF 语句	(57)
4.5.2	块 IF 语句	(58)
4.5.3	算术 IF 语句	(63)
4.6	程序举例	(63)
	小结	(65)
	习题四	(65)
第五章	循环结构的程序设计	(67)
5.1	概述	(67)
5.2	循环结构程序的算法表示	(67)
5.3	用逻辑 IF 语句实现直到型循环结构	(67)
5.4	用块 IF 语句实现当型循环结构	(70)
5.5	用 DO 语句实现循环	(71)
5.5.1	DO 语句的形式	(71)
5.5.2	DO 循环结构及执行过程	(72)
5.5.3	CONTINUE 语句在循环结构中的用法	(74)
5.5.4	DO 循环的嵌套结构	(76)

5.6	数组元素的输入输出方式	(80)
5.6.1	用 DO 循环方式	(80)
5.6.2	用隐 DO 循环方式	(82)
5.6.3	用数组名方式	(83)
5.6.4	用 DATA 语句给数组赋初值	(84)
5.7	程序举例	(85)
小结		(96)
习题五		(97)
第六章	FORTRAN 数据的输入和输出	(99)
6.1	输入、输出语句的一般形式	(99)
6.1.1	READ 语句的格式	(99)
6.1.2	WRITE 语句的格式	(100)
6.2	FORMAT 语句的格式	(100)
6.3	格式编辑符	(100)
6.3.1	I 编辑符	(100)
6.3.2	F 编辑符	(101)
6.3.3	E 编辑符	(102)
6.3.4	H 编辑符	(102)
6.3.5	X 编辑符	(103)
6.3.6	A 编辑符	(103)
6.3.7	撇号编辑符	(104)
6.3.8	纵向走纸控制	(104)
6.3.9	斜杠编辑符	(105)
6.3.10	G 编辑符	(105)
6.3.11	D 编辑符	(106)
6.3.12	L 编辑符	(106)
6.4	重复系数的用法	(107)
6.5	采用格式方式时数据的输入方式	(107)
6.6	数据的输出格式设计	(108)
6.7	省写 FORMAT 语句时的 READ, WRITE 语句的表示方法	(109)
6.8	程序举例	(110)
小结		(112)
习题六		(113)
第七章	FORTRAN 的过程	(115)
7.1	FORTRAN 的过程概述	(115)
7.2	FORTRAN 的函数	(115)
7.2.1	FORTRAN 的内部函数	(115)
7.2.2	FORTRAN 的语句函数	(118)
7.3	FORTRAN 的子程序	(122)

7.3.1	子程序概述	(122)
7.3.2	函数子程序	(123)
7.3.3	子例行子程序	(127)
7.3.4	程序单位间数据的传递	(130)
7.3.5	EQUIVALENCE 语句和 SAVE 语句	(149)
7.3.6	数据块子程序	(151)
7.4	程序举例	(152)
	小结	(157)
	习题七	(158)
第八章	FORTRAN 的数据文件	(161)
8.1	文件的概念	(161)
8.1.1	文件和记录	(161)
8.1.2	文件指针	(162)
8.1.3	文件的存取方式	(162)
8.2	文件操作的语句	(162)
8.2.1	文件的打开语句	(162)
8.2.2	文件的关闭语句	(164)
8.2.3	输入语句	(164)
8.2.4	输出语句	(165)
8.2.5	重绕语句	(165)
8.2.6	回退记录语句	(166)
8.2.7	写结束文件记录语句	(166)
8.3	文件的建立与访问	(166)
8.3.1	建立文件	(167)
8.3.2	访问文件	(169)
8.4	程序举例	(171)
	小结	(183)
	习题八	(183)
附录		(183)
附录 I	EDITOR——全屏幕编辑软件及用法	(185)
附录 II	常用字符与 ASCII 代码对照表	(190)
附录 III	FORTRAN 77 内部函数	(192)
附录 IV	FORTRAN 77 语句总览	(196)

第一章 FORTRAN 语言的初步认识

1.1 FORTRAN 语言简介

FORTRAN 是英文 FORMULA TRANSLATION 的缩写,可直译为“公式翻译”。FORTRAN 语言是国际上广泛流行的高级语言,在数值计算方面应用极广。它的发展历史大致如下:

1954 年发表了第一个 FORTRAN 文本。1957 年在 IBM 704 机上实现。

1958 年 IBM 公司提出了 FORTRAN I。1962 年 IBM 公司又提出了 FORTRAN IV。

1966 年美国标准化协会(ANSI)公布了美国标准 FORTRAN (X3.9—1966)和美国标准基本 FORTRAN (X3.10—1966)。这两个文本的前者相当于 FORTRAN IV;后者相当于 FORTRAN I。

1976 年美国标准化协会(ANSI)对 FORTRAN(X3.9—1966)进行了修订,到 1978 年正式公布了新的美国国家标准 FORTRAN(X3.9—1978)。为了区别于旧标准(FORTRAN 66),将它称为 FORTRAN 77。

FORTRAN 66(相当于 FORTRAN IV)流行了十几年,在数值计算领域内影响很大,在我国也得到推广应用。

FORTRAN 77 与 FORTRAN 66 是兼容的,除个别例外,用 FORTRAN 66 写成的程序可在 FORTRAN 77 编译系统支持下正常运行。

但是,FORTRAN 77 对于 FORTRAN 66 有许多重大的修改与扩充。例如,增加了“块 IF”语句,提供了 IF—THEN—ELSE 等判断选择结构,大大有利于程序设计结构化的实现。又如,允许不同类型数据的混合运算,解除了旧标准的限制,极大地方便了用户,等等。因此,FORTRAN 77 很快取代了 FORTRAN 66 并得到推广应用。它在我国已流行了十几年,目前仍在广泛应用之中,所以学习它仍有极大的实用价值。

FORTRAN 77 设有两个级别:全集 FORTRAN,子集 FORTRAN。

另外,IBM—FORTRAN 或 MS—FORTRAN 都是以子集作为蓝本扩充、修改而成的,并未实现全集的全部功能,用户应当注意这点。IBM 与 MS 这两个公司推出的 FORTRAN 77 版本,目前在微型机上应用极广,性能良好。

FORTRAN 语言还得到了国际标准化组织(ISO)的认可。1972 年 ISO 接受美国标准,公布了 ISO FORTRAN 标准。1980 年,FORTRAN 77 又被接受为国际标准(称为 FORTRAN ISO 1539—1980)。

在三十几年的时间内,FORTRAN 语言一直广泛流传,未被淘汰,这是难能可贵的。FORTRAN 语言有一个重要的特点,就是它能博采众长,推陈出新,不断推出新版本。

1.2 简单的 FORTRAN 程序介绍

1.2.1 FORTRAN 程序实例

首先介绍几个简单的 FORTRAN 程序,让读者初步认识一下什么是 FORTRAN 程序。

例 1—1 已知并联电路的电阻公式为:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$$

给定 R_1, R_2 , 求 R 。

设 $R_1=100, R_2=100$, 单位为欧姆, 求 R 的程序可写为:

```
C      EXAM-1
      R1=100
      R2=100
      Y=1/R1+1/R2
      R=1/Y
      WRITE(*,*)'R:',R
      END
```

几点说明:

程序的第一行是注解行。这样的行内,第一个字符必须写 C 或 *, 不能写其他字符,也不能不写字符。本行只有注解作用(或提示作用),对运行(计算)不起任何作用。本行也可以不写。

以下四行(2~5 行),每行内写有一个赋值语句。它的作用是将赋值号(=)右边的值传送给左边的变量(存储起来)。例如,将 100 传送给 R_1 ;将 100 传送给 R_2 ;将 $1/R_1+1/R_2$ 传送给 Y ;将 $1/Y$ 传送给 R ;等等。

第六行是一个输出语句(WRITE 语句即写语句)。它的基本作用是输出,在一行内将输出字符 R : 和 R 的值(50.0000000)。

第七行称为结束行(END 行)或结束语句(END 语句),它表示本程序到此结束。FORTRAN 程序要求有一个结束行,不可不写。

注意:从第二行起,每行的第一个字符都从第 7 列开始;前面六列都是空白,不能写其他字符。

读者试算本例时,可让 R_1 或 R_2 换上不同的值,以观察运行结果的变化。

例 1—2 求 S 的值,已知:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{100}$$

程序可写为:

```
C      EXAM-2
      S=0
      N=1
10     S=S+1.0/N
      N=N+1
      IF (N.LE.100) GOTO 10
```

```

WRITE(*,*) 'S=',S
END

```

几点说明：

本程序的第一、二、三行及第七、八行与例 1—1 类似，不重复解释。第四行也是一个赋值语句，但它带有一个标号 10，标号 10 必须写在第一至第五列上。 $S=S+1.0/N$ 是一个不同一般的赋值语句，它实现将 S 的老值加 $1.0/N$ 后又传送给 S (变成 S 的新值)。当这个语句被重复执行时，S 就将变成一个累加的值。第五行 $N=N+1$ 实现了一个自动修改功能，N 由 1 变成 2，由 2 变成 3，由 3 变成 4，等等。它总是把 N 的老值加 1，又传送给 N，使 N 获得一个新值。第六行是一个单行 IF 语句 (逻辑 IF 语句)，在这里起控制、转移作用。当 N 小于或等于 100 时，转去执行 10 指示的那一行语句。控制条件是 $N.LE.100$ ，这里 $.LE.$ 是小于或等于的意思 (如同数学符号 $\leq$)。整个语句可以理解为：如果 N 小于等于 100，就去执行 10 行语句；如果条件 $N.LE.100$ 不成立，则依次执行下一行语句。本程序运行结果为 $S=5.1873780$ 。

例 1—3 计算输出 $8!, 10!, 20!$ 的值。程序全文可以写为如下形式：

```

C  MAIN
  Y8=FAC(8)
  Y10=FAC(10)
  Y20=FAC(20)
  WRITE(*,*)Y8,Y10,Y20
  END
C  SUB
  FUNCTION FAC(N)
    Y=1.0
    K=1
10  Y=Y*K
    K=K+1
    IF (K.LE.N) GOTO 10
    FAC=Y
  END

```

几点说明：

本程序分为两个单元 (UNIT)，一个主程序单元，一个函数子程序单元，分别加上了注解行：C MAIN 与 C SUB。每一个单元都有一个结束行 (END 行)。函数子程序单元定义了一个函数 FAC(N)，它的自变量就是 N，它的功能就是求出 $N!$ (N 的阶乘)。这个子程序的算法后文再讲。主程序单元内，使用了 $Y8=FAC(8)$ 等语句，是对 FAC 函数进行调用，此处调用函数 FAC(N) 三次，分别求出 $8!, 10!, 20!$ ，然后输出它们的值。运行结果为：

40320 3.6288 $\times 10^8$ 2.432902 $\times 10^{18}$ (显示在屏幕上的形式与此有别，后文将讲到)。

1.2.2 FORTRAN 程序的组成

通过上节的例题，可以了解以下几点：

(1) FORTRAN 程序可以由一个或几个单元组成。只有一个单元的程序必须是一个主程序单元。有几个单元的程序中，也必须有一个主程序单元，不能多也不能少。每一个单元都必

须有一个自己的 END 行(END 语句)。它的作用是表示本单元结束,也可以表示程序运行到此应当停下来。主程序单元与子程序单元如何区别以后将专门讲到。

(2)每个单元都由若干行组成。行,可以分为语句行或非语句行。例如赋值语句行、结束语句行、写语句行等,这些都是典型的语句行。又如注解行,它只起注解作用,对于程序的编译、运行不起作用,是非语句行。

(3)语句的概念。一个语句至少占一行,也可以一连占几行。当一个语句一连占几行时,第一行叫开头行,以后各行叫继续行。当一个语句只占一行时,这行本身就是开头行。每一个语句,都有一定的语法规则,都有确定的语义,都能完成确定的功能。FORTRAN 语言又规定,在一行之内不允许写两个或多个语句。例如以下几行:

```
READ(*,*)X,Y
WRITE(*,*)A,B,C
END
```

这都是一个语句占一行。语句又分为可执行语句与非执行语句。可执行语句的基本特点是,命令计算机系统产生确定的动作,完成指定的任务,因而也对应一定的计算机指令(或指令系列)。非执行语句则不要求计算机产生具体的动作,只起为程序提供说明信息的作用。例如:

```
FUNCTION    FAC(N)
100  FORMAT(1X,I4,F5.2)
```

以上都是非执行语句。又如:

```
READ(*,*)X,Y
S=0.5*A*B*SIN(C)
WRITE(*,*)X1,X2
```

以上都是执行语句。

(4)标号的概念。FORTRAN 语言允许使用标号,标号总是放在语句的前面。但是语句并不要求一定带上标号,因此有的语句可能带标号,另一些语句则可能不带标号。标号是由无符号整数组成,允许最多五个数字,由 1~99999 之间的数组成。例如:

```
10      IF (N.LT.100)  THEN
100     FORMAT(1X,I4,F5.2)
```

这里的 10,100 都是标号。

(5)FORTRAN 语言对于各类语言的位置(次序)有明确的规定,不允许写错。例如,一般地说非执行语句应放在可执行语句的前面;PROGRAM 语句、FUNCTION 语句、SUBROUTINE 语句应放在一个单元的第一行;END 语句应放在一个单元的最后一行;等等。但也有若干例外,此处从略。

(6)FORTRAN 字符集的概念。FORTRAN 源程序由一系列字符组成,它们是:

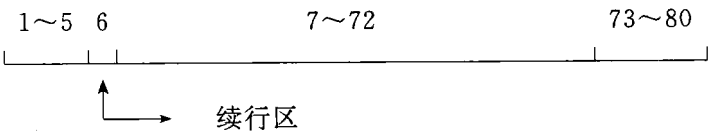
- 字母。由 A 到 Z,26 个英文字母,大写、小写均可,一共 52 个字符。
- 数字。0,1,2,3,4,5,6,7,8,9 一共 10 个字符。
- 特殊字符。其他 ASCII 字符集中的可打印字符。

以上三类字符的集合就是 FORTRAN 字符集。写入源程序的字符,必须都是属于 FORTRAN 字符集的字符。

1.3 FORTRAN 源程序的书写格式

标准的 FORTRAN 程序纸,应当每一行分为 80 列(即 80 个空格),每一列上写一个字符(即每一个格子里写一个符号)。这 80 列分为四个区:标号区、续行区、语句区、注解区。具体安排如下:

- 标号区 1~5 列
 - 续行区 第 6 列
 - 语句区 7~72 列
 - 注解区 73~80 列
- 示意如下:



注意:

- (1)注解行的书写不受上述限制,它只要求该行第 1 列(第一格)写 C 或写 *。
- (2)对于开头行,第 6 列应当为空白或者写数字 0,不允许写其他字符。
- (3)对于继续行,第 6 列上必须写续行标志。FORTRAN 规定,续行标志可以使用除空格、数字 0 之外的其他字符。例如,可以用+号或\$号等。建议用\$为宜。并且第 1 至第 5 列都只能为空格。
- (4)如果使用编译系统提供的元命令,则必须从某行的第 1 列写起,第 1 列写\$符号,后面紧跟元命令字符,并且不许随意留空格。例如:
\$DO66

这就是元命令行。建议初学者暂不使用元命令。

- (5)写在注解区(73~80 列)内的字符,只起注解作用,可为用户注解提供方便(例如写各行行号),对编译系统不起作用。建议初学者暂不使用注解区,都置为空白,以免引出不必要的麻烦。

1.4 算法与流程图

算法,应当广义地理解为使用计算机解决问题的方法与步骤。这里讲的问题,可以是通常说的“计算题目”,要求算出一个或一批数;也可以是所谓“绘图题目”,由计算机绘制出一个或一批图形、图案;也可以是一切计算机系统可以完成的工作任务。

大家都知道,在使用计算机解题之前,必须编制控制计算机工作的程序(即程序设计);而在编制程序之前,又应当选定合适的工作方法与工作步骤(即确定算法)。因此,算法是必不可少的,算法是程序设计的前提和基础。

选定了某个算法,就必须完整准确地表达出来,记录下来,以利于自己的工作和与他人的交流。特别是当某个算法比较复杂、比较庞大时,表达与记录尤其重要。

表达、记录算法,可以有多种方法。当然,人们首先使用的总是某种自然语言,例如中文、英文等。

现在举一个很简单的例子。

问题:求某个铅球的体积。

算法:

第一步,尽量准确地确定这个铅球的直径 d ;

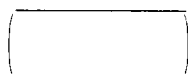
第二步,选定圆周率 π 的近似值为 3.14159;

第三步,完成计算 $\frac{3}{4}\pi(\frac{d}{2})^3$,得到一个数 V ;

第四步,认定 V 值就是该铅球的体积值。

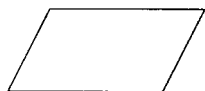
另外一种表达、记录算法的方法就是画程序框图,或称为画流程图。所谓画流程图,就是用一些约定的几何图形来描述算法。在编写复杂的程序之前,先画出流程图,在流程图上反复检查、核对算法,是很有好处的,也是很有必要的,因为流程图总是比自然语言更加直观、简明、清楚,使人能有一目了然之感。

但是目前对于流程图符号尚无统一的标准,只有一些大体一致的约定。现将常用符号及其意义说明如下:



圆边框

表示工作的开始或停止



平行四边形框

表示输入或输出



判断框

表示逻辑判断或检查



矩形框

表示各种处理功能



箭头线

表示流程的路线



圆圈

表示连接点

下面举一个例子,说明流程图的应用。

现将例 1—2 的流程图画在下面。注意:它能实现求 S 值,

$$S=1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{100}$$

它与第 2 页所写程序是互相对应的。