

# FORTRAN77 程序设计

高培志  
傅德胜  
编著

江苏科学技术出版社

**FORTRAN77**

**程 序 设 计**

高培志 傅德胜 编著

江苏科学技术出版社

## 内 容 简 介

本书以标准 FORTRAN77 全集语言文本为依据,详细介绍了 FORTRAN77 语言的基本概念、特性、程序设计的基本方法和技巧,辅以大量实例示范各种语句的应用和编程技巧。全书各章均有丰富的练习题供读者自测学习效果 and 编程实践,所有一百多个程序例题均在 MS-FORTRAN V5.0 编译系统下调试运行正确。

本书特别介绍了一种运行于 Windows 3. X、Win95、Windows NT 和 DOS 平台的 FORTRAN 语言集成调试环境 FORTRAN PowerStation。

本书可作为 FORTRAN 语言等级考试(二级)培训教材、普通高校和各类成人教育 FORTRAN 语言程序设计教程,也可用作各类 FORTRAN 语言程序设计人员的自学读本。

# 前 言

FORTRAN 语言是世界上出现最早、应用最广的程序设计语言。大量工程及科学上的程序、专业性软件都是用 FORTRAN 语言编制的。为使 FORTRAN 语言能利用计算机科学与技术最新研究成果,吸收其他语言的某些长处,不断满足新的应用要求,发展成为功能更强大、效率更高、使用更方便的现代语言,国际标准化组织(ISO)每隔 10 年左右就为 FORTRAN 制定一个新的国际标准文本,供各国遵循。这是 FORTRAN 语言几十年来长盛不衰的重要原因。

为适应计算机应用能力等级考试对 FORTRAN77 语言的内容范围和深度,以及考试程序调试环境的变化要求,作者以 FORTRAN77 全集语言标准文本为依据,结合多年 FORTRAN 语言教学实践,突出教学重点,注重理论与实践相结合,编写了这本教材。

计算机语言是运用计算机解决问题的一种工具,重在应用。在学习中既不应忽略其语法规范,也不应满足于对语法条文的了解,而应在弄懂语句的语法、语义基础上,大量做程序阅读和编程上机调试练习,巩固、深化对语言的理解,提高运用计算机的能力。

随着计算机的发展,人们希望能在 Windows 3.X、Win95、Windows NT 平台上开发 FORTRAN 程序,提高程序调试的效率。集成调试环境软件应运而生。本书介绍的 FORTRAN PowerStation 是一个可运行于 Windows 和 DOS 平台的功能强大的 FORTRAN 语言集成调试环境,是一种调试 FORTRAN 程序的有效工具。

本书由南京理工大学高培志主编。南京气象学院傅德胜编写了第一章至第三章;南京理工大学高培志编写了第四章至第十二章并对全书进行了修改和定稿。在编写过程中,南京航空航天大学吕兵曾审阅了本书,并提供了宝贵的修改意见,在此表示深切的谢意。

由于编者学识有限,书中缺点、错误难免,恳请同行和读者批评指正。

编者

1998 年 6 月于南京

## 文档约定

本书在语句的一般形式中,使用:

< >	代表必选项
[ ]	代表可选项
...	代表重复项

在执行程序人机对话显示信息中,使用:

带下划线部分	代表输入信息
不带下划线部分	代表输出信息
□	代表一个空格字符
↵	代表回车操作

# 目 录

<b>第一章 编程基础知识</b> .....	1
§ 1.1 计算机、语言和程序设计简介 .....	1
1.1.1 计算机系统简介 .....	1
1.1.2 程序设计语言概述 .....	3
1.1.3 计算机程序设计概述 .....	4
§ 1.2 FORTRAN 语言简介 .....	11
1.2.1 FORTRAN77 的程序结构 .....	12
1.2.2 FORTRAN77 源程序举例 .....	13
1.2.3 FORTRAN 字符集 .....	14
1.2.4 FORTRAN 源程序的书写格式 .....	14
习题一 .....	15
<b>第二章 数据类型、常数和变量</b> .....	17
§ 2.1 FORTRAN77 的数据和数据类型 .....	17
§ 2.2 常数 .....	17
§ 2.3 变量及其类型说明 .....	19
习题二 .....	21
<b>第三章 基本 FORTRAN 语句</b> .....	23
§ 3.1 算术表达式 .....	23
§ 3.2 算术赋值语句 .....	27
§ 3.3 DATA 语句 .....	28
§ 3.4 内部函数 .....	30
§ 3.5 基本输入输出语句 .....	37
3.5.1 几个基本概念和术语 .....	37
3.5.2 表控格式输入/输出语句 .....	40
§ 3.6 STOP, PAUSE, END 语句 .....	44
§ 3.7 PARAMETER 语句 .....	45
§ 3.8 应用举例 .....	46
习题三 .....	48
<b>第四章 逻辑运算和选择结构</b> .....	52
§ 4.1 关系表达式 .....	52
§ 4.2 逻辑表达式和逻辑赋值语句 .....	54

§ 4.3	无条件 GOTO 语句 .....	56
§ 4.4	算术 IF 语句 .....	57
§ 4.5	逻辑 IF 语句 .....	58
§ 4.6	块 IF .....	61
§ 4.7	应用举例 .....	68
习题四 .....		71
<b>第五章 循环结构 .....</b>		<b>76</b>
§ 5.1	循环结构引例 .....	76
§ 5.2	“当型”和“直到型”循环 .....	77
§ 5.3	DO 语句和 DO 循环 .....	80
§ 5.4	有关 DO 循环的一些限制 .....	83
§ 5.5	嵌套 DO 循环 .....	85
§ 5.6	应用举例 .....	87
习题五 .....		90
<b>第六章 编辑控制格式输入/输出 .....</b>		<b>94</b>
§ 6.1	格式(FORMAT)语句和格式说明 .....	94
§ 6.2	编辑控制格式输出 .....	95
§ 6.3	编辑控制格式输入 .....	106
§ 6.4	应用举例 .....	110
习题六 .....		115
<b>第七章 数组 .....</b>		<b>119</b>
§ 7.1	数组和数组元素 .....	119
§ 7.2	数组说明和 DIMENSION 语句 .....	120
§ 7.3	数组的存储和输入输出 .....	123
§ 7.4	应用举例 .....	127
习题七 .....		134
<b>第八章 过程 .....</b>		<b>139</b>
§ 8.1	语句函数 .....	139
§ 8.2	函数辅程序 .....	143
§ 8.3	子程序辅程序 .....	148
§ 8.4	哑实结合数据通讯 .....	151
§ 8.5	公用区数据通讯和 COMMON 语句 .....	160
§ 8.6	SAVE 语句 .....	163
§ 8.7	BLOCK DATA 辅程序 .....	164
§ 8.8	EQUIVALENCE 语句 .....	166

§ 8.9 应用举例 .....	169
习题八 .....	174
<b>第九章 文件 .....</b>	<b>178</b>
§ 9.1 文件的基本概念 .....	178
§ 9.2 外部文件的打开与关闭 .....	180
§ 9.3 输入/输出语句的一般形式 .....	183
§ 9.4 文件的查询 .....	187
§ 9.5 文件的定位 .....	189
§ 9.6 应用举例 .....	192
习题九 .....	201
<b>第十章 字符数据处理 .....</b>	<b>203</b>
§ 10.1 字符型数据及其类型说明 .....	203
§ 10.2 字符表达式和字符赋值语句 .....	206
§ 10.3 字符型数据的输入/输出 .....	207
§ 10.4 应用举例 .....	210
习题十 .....	214
<b>第十一章 上机操作指导 .....</b>	<b>217</b>
§ 11.1 MS-FORTRAN77 V5.0 编译系统简介 .....	217
§ 11.2 全屏幕编辑程序 ED.COM 的使用 .....	217
§ 11.3 FORTRAN77 程序调试过程 .....	220
§ 11.4 上机练习 .....	227
<b>第十二章 FORTRAN 语言集成调试环境 .....</b>	<b>234</b>
§ 12.1 概述 FORTRAN PowerStation Development System .....	234
§ 12.2 可视化工作台编辑器的使用 .....	236
12.2.1 可视化工作台编辑器的特点 .....	236
12.2.2 文件编辑与文件管理 .....	237
12.2.3 项目管理 .....	251
12.2.4 编译文件或构造项目 .....	255
12.2.5 应用程序的调试 .....	262
<b>附 录 .....</b>	<b>274</b>
附录 A 程序单位中语句和注释行的次序 .....	274
附录 B ASCII 表 .....	275
附录 C FORTRAN77 语句分类及索引 .....	277
附录 D 常用算法索引 .....	278

# 第一章 编程基础知识

## § 1.1 计算机、语言和程序设计简介

电子计算机是本世纪最重大的科学技术发明之一,它的出现和发展迅速地改变着人类的生产方式和生活方式。为了在各个领域应用计算机,有必要了解计算机基础知识,掌握与计算机交流信息的工具——计算机语言。

用计算机进行科学计算和辅助业务管理,最常用的一类语言是过程型命令语言,使用这类语言控制计算机工作时,不仅要告诉计算机“做什么”,还要告诉计算机“怎样做”。控制计算机完成特定任务的一系列步骤构成了解决问题的算法。用某种计算机语言实现算法的过程叫编程。实现解题算法的一系列语句(或指令)称为程序。包括分析问题、确定算法、编写程序(编码)和上机调试在内的整个过程叫做程序设计。要使用计算机,就要学习用计算机语言进行程序设计。

### 1.1.1 计算机系统简介

世界上第一台电子计算机 ENIAC 是 1946 年问世的。五十多年来计算机发展迅速,经历了四个阶段:电子管计算机(第一代);半导体计算机(第二代);集成电路计算机(第三代);大规模/超大规模集成电路计算机(第四代)。与此同时,计算机软件也有了迅速的发展。

这四代计算机都属于冯·诺依曼型计算机,即在计算机内数据以二进制表示,以程序存储控制为基本工作原理,由程序控制对数据进行加工处理和输入输出。

#### 一、电子计算机的特点

##### ● 运算速度快

计算机的运算速度是指它一秒钟能存取指令的数量。例如,每秒能存取一万条指令的计算机,其运算速度为一万次/秒。第四代计算机运算速度最快可达万亿次/秒。

##### ● 精确度高

计算机能表示的数值大小和精度,一般与存放数据的存储单元字长有关。字长越长,表示数据的有效位数越多,数值精度也越高。目前微机字长以 16 位和 32 位居多;微小型机一般为 16~32 位;中型机多为 64 位;大型机或巨型机一般为 64~128 位。计算机为了提高运算精度,通常还采取双倍或多倍字长运算,精度可达十几位有效位数。

● 具有记忆能力,能存储大量程序和数据。

● 由程序控制计算机自动工作。

#### 二、计算机系统的组成

计算机系统由硬设备(或称硬件)和软设备(即程序系统,或称软件)组成。

##### 1. 硬件

计算机系统硬件主要包括控制器、运算器、存储器、输入/输出设备。

控制器是整个系统的指挥中枢。它根据存储在存储器中程序的要求,依次从存储器中取

出一条条指令解释执行,向各有关部件发出控制信号,协调各部件完成程序指定的任务。

运算器是计算机的主要运算装置。在控制器控制下完成各种算术和逻辑运算。

控制器和运算器总称为中央处理器(简称 CPU),是计算机系统硬件的核心部件。

存储器是具有记忆功能的部件,用于存储程序指令和数据。

存储器由一系列存储单元构成,这些用于存放信息的存储单元,通常称为字。为了区分不同的单元,给每个存储单元顺序编号,存储单元的编号称为该单元的地址。一个存储单元由一系列二进制位(bit)组成。作为一个整体处理的相邻 8 个二进制位组成字节。一个存储单元能包含的二进制位数叫做字长。存储器中能存储信息的总数称为存储容量。存储器的容量通常以 KB 和 MB 为单位,1KB 代表 1024 个字节,1MB 代表 1024KB。例如,内存为 512KB 的微型计算机,实际存储容量为  $1024 \times 512$  个字节。

存储器分内存储器和外存储器两大类:内存储器包括随机读/写存储器(RAM)和只读存储器(ROM),是计算机的一个重要组成部分。运行程序、读/写原始数据、中间结果和最终结果都需使用 RAM。

外存储器作为计算机的外部设备,一般采用磁介质存储数据,例如,软磁盘、硬盘和磁带等。与内存储器相比,外存储器虽然存取速度相对较慢,但具有容量大的特点。13.34 厘米(5¼英寸)的高密软盘容量为 1.2MB;8.89 厘米(3½英寸)高密度软盘容量可达 1.44MB;硬盘容量以 MB(兆字节)和 GB(吉字节)为单位( $1\text{GB}=1024\text{MB}$ ),容量几十兆~几吉字节不等。

输入设备是把数据和程序输入计算机存储器的装置。有多种输入设备,键盘是最常用的一种。

输出设备是用来输出信息、计算的中间结果和最终结果的装置。有多种输出装置,显示器、打印机和磁盘驱动器是最常用的几种。

计算机系统硬件及各部分关系可用图 1-1 表示。

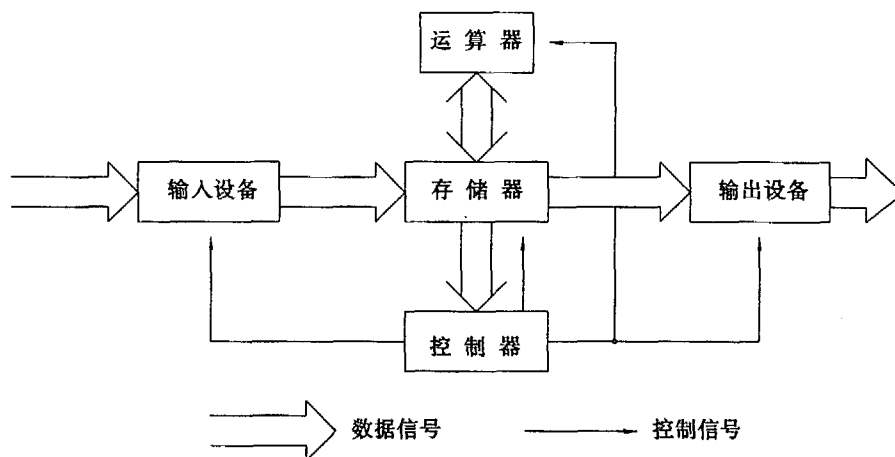


图 1-1 计算机硬件及各部分之间关系示意图

## 2. 软件

计算机系统的软件是程序系统和文档资料的总称。软件分为系统软件和应用软件。

系统软件是为充分发挥硬件效能、方便用户使用而编制的基本软件，一般由计算机厂家或软件公司开发。系统软件包括操作系统，各种程序设计语言的编译、汇编或解释程序，数据库管理系统，各种诊断程序，实用程序和程序开发工具等。

应用软件是为解决某个应用领域的具体任务专门编制的程序。应用程序一般由用户自行开发，有的计算机厂家也提供应用软件。应用软件经标准化、模块化而形成解决实际问题的应用程序组合，统称为软件包。

### 1.1.2 程序设计语言概述

要使用计算机，就必须以某种方式告诉计算机解题需要的特定操作步骤和数据，这就需要某种语言。计算机语言是我们与计算机交流信息的工具。计算机语言经历了机器语言、汇编语言和高级语言几个发展阶段。

#### 一、机器语言

计算机能直接识别和执行的是为计算机硬件设计的某些基本操作(如：加、减、取数、存数等)命令，称为机器指令。机器指令是用按特定规则组成的一系列二进制代码(0和1)表示的。一台计算机能执行的全部指令的集合称为指令系统。由于不同机型的计算机硬件设计不同，指令系统也因机而异。指令系统是计算机唯一能直接执行的语言，称为机器语言。用机器指令编写的程序叫机器语言程序，也叫手编程序。

【例 1-1】 用 Z-80 指令编写的计算  $32+17$  的机器语言程序：

地 址	指 令	说 明
0040	00111110	
0041	00100000	取 32 送入累加器 A
0042	00000110	
0043	00010001	取 17 送入累加器 B
0044	10000000	把 B 的内容加到 A 上
0045	00110010	把 A 的内容送到 0049 单元
0046	01001001	
0047	00000000	
0048	01110110	停机
0049	00110001	计算结果

程序的左半部分是存放指令的存储单元地址，以 16 进制表示；右半部分是指令，以二进制表示。

由上例可见，用机器语言编写程序，必须具备一定的计算机硬件知识，熟悉其指令系统。编写程序繁琐枯燥，易错难改，程序的可读性、可维护性和可移植性差。

#### 二、汇编语言

汇编语言是用特定的助记符号代替机器语言中以二进制表示的指令操作码和操作地址，帮助人们记忆指令含义，所以汇编语言又叫符号语言。用汇编语言符号编写的程序叫汇编语言程序。

【例 1-2】 用汇编语言编写的上例程序如下：

汇编指令	说 明
LD A,20H	将 20H(是 32 的十六进制表示)送累加器 A

LD B,11H	将 11H(是 17 的十六进制表示)送寄存器 B
ADD A,B	A,B(分别是累加器和寄存器)内容相加,结果保存在 A 中
LD (0049H),A	把 A 的内容存入存储单元 0049H
HALT	停语句

### 三、高级语言

高级语言是一种面向问题的语言,使人们摆脱了具体计算机硬件对编程的约束,把精力集中在用类似英语和数学公式的语句表达解题算法上。用高级语言编写的程序叫高级语言源程序。高级语言源程序在很大程度上脱离了对具体机型的依赖,有较好的可读性、可维护性和可移植性。程序设计语言 FORTRAN、C、BASIC、COBOL、PASCAL、Ada、LISP、PROLOG 等是目前最流行的高级语言。

【例 1-3】 用 FORTRAN 语言编写的上例程序:

```
PROGRAM ADD
  INTEGER A,B
  A=32
  B=17
  WRITE(*,*)A+B
END
```

计算机不能直接识别和执行高级语言源程序,源程序必须先经语言处理程序(编译程序)“翻译”成机器语言程序(目标程序)后执行,或用解释程序对源程序解释执行。有些系统还要用连接程序把目标程序与运行必须的库文件连接起来,生成可执行程序,才能运行。一般高级语言源程序在计算机上运行步骤如图 1-2 所示。

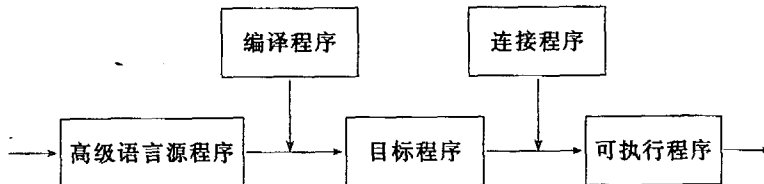


图 1-2 源程序在计算机上运行步骤

## 1.1.3 计算机程序设计概述

### 一、计算机程序设计

计算机程序设计是为计算机规划、安排解题步骤的过程,一般由四个步骤(如图 1-3)组成:

#### 1. 分析问题

在着手解决给定问题之前,要通过分析,充分理解问题,明确原始数据、解题要求、需要输出的数据及形式等。

#### 2. 设计算法

所谓算法是一步一步解题过程。算法设计和描述过程是一个结构化分析过程,即首先集中精力于算法的总体规划,然后逐层降低问题的抽象性,逐步充实其细节,直到最终把抽象的问题具体化成可用程序语句表达的算法。这是一个自顶向下、由粗到精、逐步细化的过程。

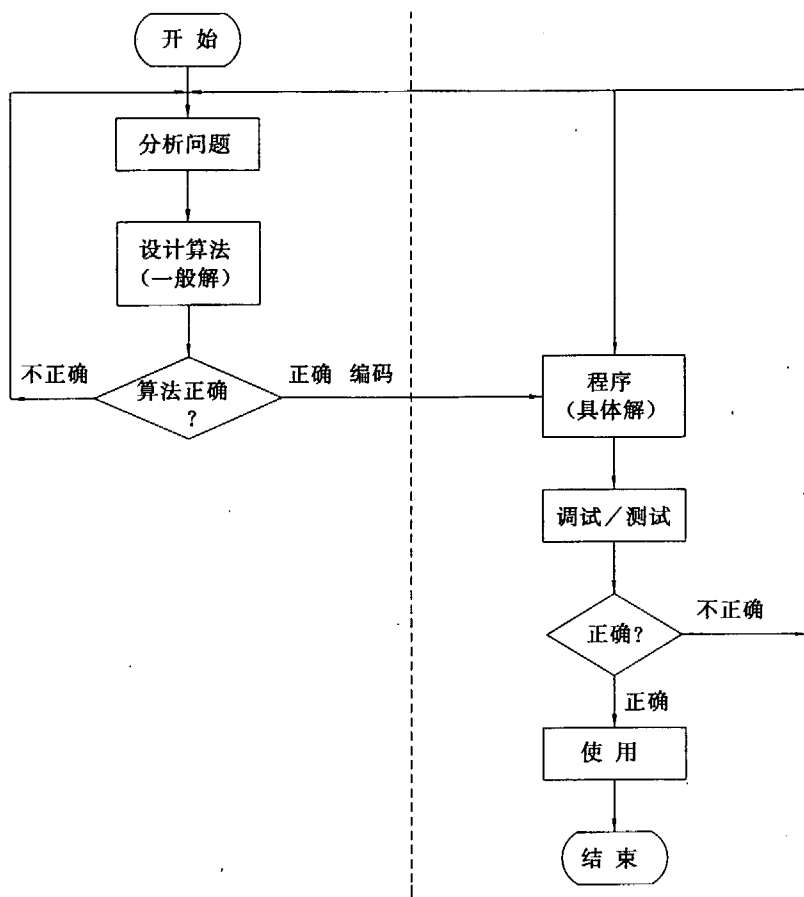


图 1-3 程序设计过程

### 3. 编码

在结构化分析的基础上,进行结构化程序设计,即把结构化分析成果的有关部分分别归纳到三种基本程序结构形态(顺序结构、选择结构、循环结构)中,并使每一结构只有一个入口与上一结构衔接,只有一个出口与下一结构衔接,使整个程序呈一串由这三种模块衔接起来的链条状。用某种计算机语言表示算法的过程称为编码(即通常所说的编程)。程序是用计算机语言编码的解题算法。

### 4. 调试程序

编好的程序还要输入计算机,进行编译。编译程序对源程序进行语法检查。发现源程序中有不符合语法规则的语句,则给出“出错信息”,程序员根据编译错误信息的提示,查找并改正错误后再行编译,直到没有语法错误为止,编译将其转换成目标程序。大多数语言还要用连接程序把目标程序与库文件连接成可执行程序。在连接过程中若程序中使用了错误的内部函数名,则会引起连接错误,可执行程序在运行过程中有可能因使用数据不当,出现“表达式除以零”而引起的“溢出”等运行错误,此时要修改程序,甚至要重新考虑算法。对运行顺利结束并得到结果的程序,对所得结果要分析校核,只有能得到正确结果的程序才是正确的程序。上述过程叫调试程序。一些重要的软件还用精心设计的数据进行测试,考验其运行的

可靠性、稳定性。

归纳起来,包括分析问题、设计算法、编码和调试与测试在内的整个过程就是程序设计。前两个步骤解决“问题是什么”,“如何解决它”,构成程序设计的解题阶段;后两个步骤把问题的一般解(算法)具体化为一个可用的程序,构成了程序设计的实现阶段。

## 二、描述算法的工具和方法

常用的描述算法的工具具有:

### 1. 用自然语言描述算法

【例 1-4】 给定两个正整数  $m$  和  $n$ ,求它们的最大公因子。

算法:

- (1) [求余数]以  $n$  除以  $m$ ,令  $r$  是所得余数, $0 \leq r < n$ ,执行步骤(2)。
- (2) [判余数是否等于零]如果  $r=0$ ,输出  $n$  的当前值,算法结束;否则执行第(3)步。
- (3) [更新被除数和除数] $m \leftarrow n, n \leftarrow r$ ,转步骤(1)。

### 2. 用算法描述语言描述算法

用某种高级语言或类高级语言来描述解题算法。

例如,用类 PASCAL 程序描述例 1-4 的解题算法如下:

```
procedure EVCLID
  Var M,N,R: integer
begin
  read(M,N);
  while N≠0 do
    begin
      R←M mod N,
      M←N,
      N←R
    end;
  write(N)
end
```

### 3. 用伪码描述算法

所谓“伪码”是用符号名、语言保留字(关键字)和自然语言混合描述算法。例如:

```
      READ m,n
标号  r=m/n
      IF(r≠0)THEN
          m←n
          n←r
      ELSE
          WRITE n 的当前值
          STOP
      ENDIF
      GOTO 标号
```

END

#### 4. 用流程图描述算法

所谓“流程图”是将人为约定的一组图形符号,用有向线连接起来形象化地表达解题算法步骤(或称流程)。

本书使用的流程图符号如图 1-4 所示。这些符号与 1963 年美国标准化协会电子计算机和信息处理委员会发布的一套流程图符号基本一致。

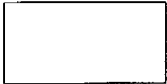
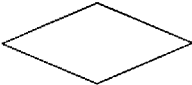


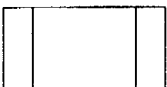



流程图符号	名称	功能
	处理框	表示一般处理功能
	判断框	根据条件判断 选择几个路径中的一个
	循环框	表示循环
	输入/输出框	表示输入/输出
	过程调用框	表示子程序或函数调用
	端框	表示流程的起、止点
	连接点	表示与其它部分流程图之 间的连接点
	有向线	表示流程的路径和走向

图 1-4 流程图符号

例 1-4 的解题算法可用图 1-5 所示的流程图描述。

### 三、程序设计举例

**【例 1-5】** 已知一组学生某课程考试成绩及其性别代码(“0”代表男生,“1”代表女生),成绩和性别代码由键盘输入,当输入的性别代码为负值时结束输入。编程计算这组学生的平均分数,同时分别计算男、女生的平均分数,并把结果显示出来。

#### 1. 分析问题

已知数据:学生成绩及学生性别代码,要求由键盘输入。因为学生总数事先不能确定,应每次输入一名学生的数据,并对其成绩和人数计数。由题意知,性别代码为负值结束输入,它是控制循环的特征数据,所以每输入一名学生数据都要对其检测,看是否取负值。

要求:计算并输出全组学生平均成绩,男、女生平均成绩。

为计算平均成绩,必须计算全组学生总分和统计总人数。计算男、女生平均成绩,可以先计男生人数,累加男生总分,求出男生平均成绩;女生总分和人数可由总成绩和总人数中分别减去男生总分和男生人数求得。

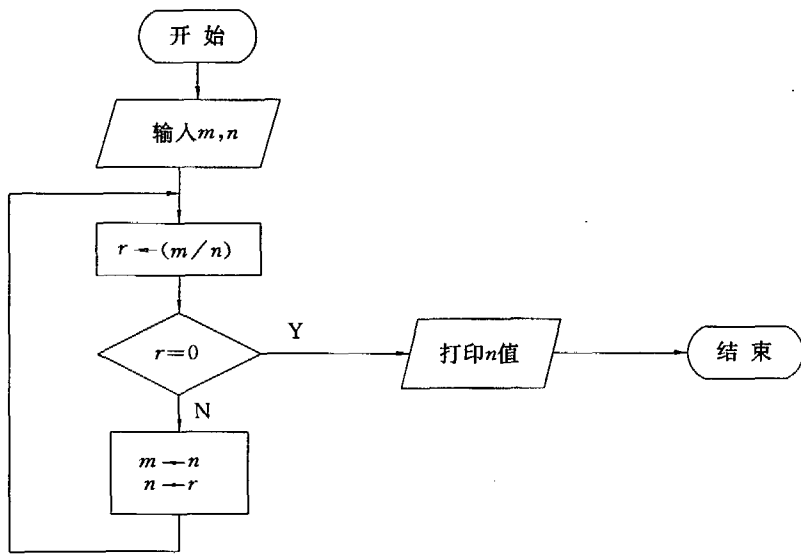


图 1-5

## 2. 设计算法

基于上述分析,可拟定解题算法步骤。算法描述可能是一个由粗到精的渐进过程,即构造总体结构,然后逐步细化。如可用文字描述如下:

(1) 初始化。

(2) 读入一组已知数据。若性别代码是非负值,则处理全组学生及男生有关数据,然后重复做第(2)步,否则做第(3)步。

(3) 若总人数不为零,处理女生有关数据,然后做第(4)步,否则转到第(5)步。

(4) 计算并打印显示平均成绩。

(5) 算法结束。

该算法可进一步细化:

(1) 设总分数、总人数、男生总分、男生人数变量,并初始化其值为零。

(2) 当输入学生的性别代码为非负值时,重复下列步骤:

① 读入一个学生的成绩、性别代码;

② 若性别代码为负值,则到第(3)步;否则

③ 计算学生总数和总成绩;

④ 若性别代码是零(男生),则

a 累加男生总分;

b 男生人数加 1。

(3) 计算女生人数及其总成绩。

(4) 计算并显示平均成绩:

① 计算显示全组平均成绩;

② 若组中包含男生,则计算显示男生平均成绩;否则显示'班上没有男生';

③ 若有女生存在,计算显示女生平均成绩;否则显示'班上没有女生'。

(5) 算法结束。

描述该算法的流程图如图 1-6 所示。

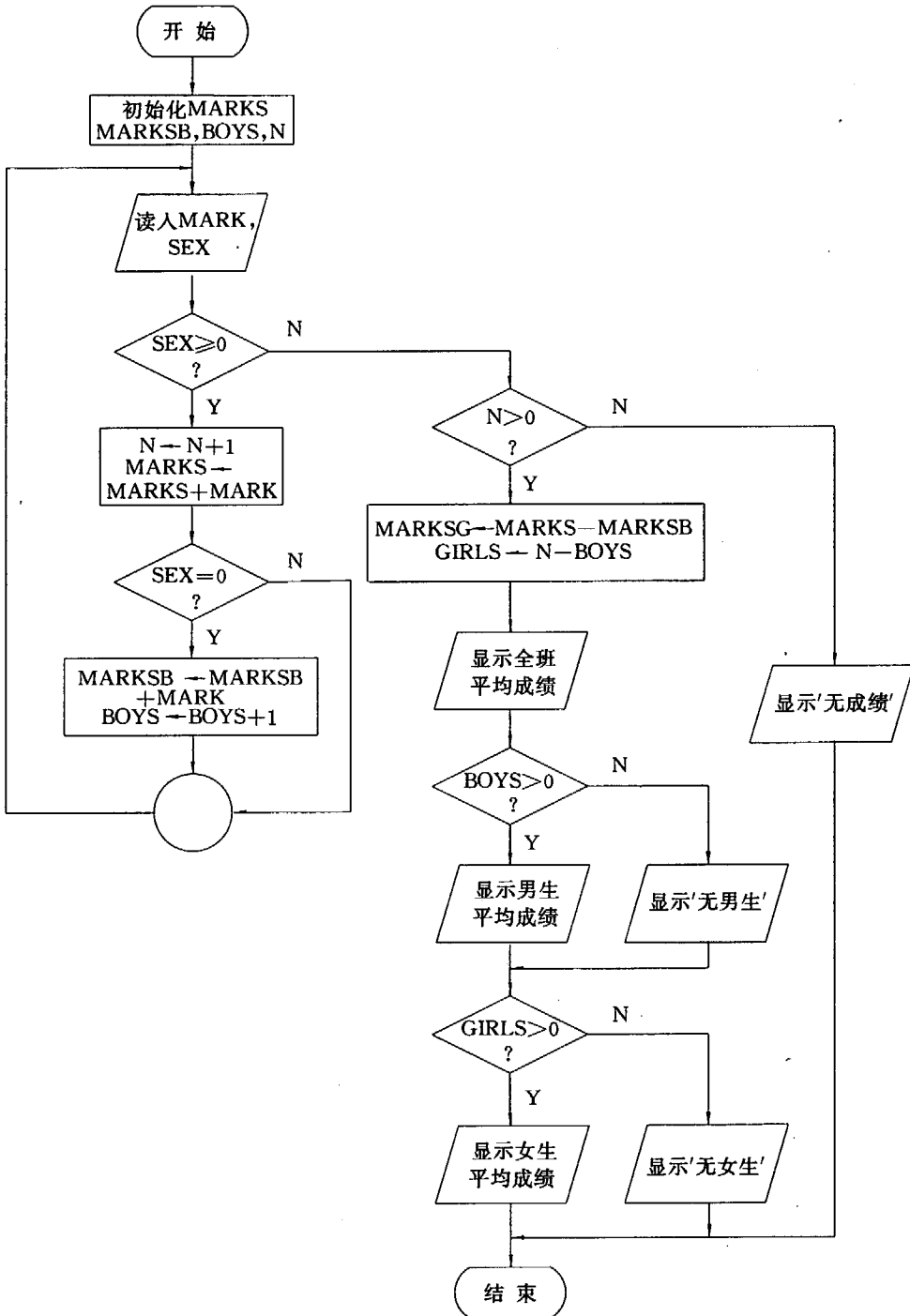


图 1-6