PASCAL
程序设计

黄顺珍 著

广东教育出版社

粤新登字 03 号

图书在版编目（CIP）数据

PASCAL 程序设计/黄顺珍

—广州：广东教育出版社，1994.6

ISBN 7—5406—2811—1

Ⅰ.P・・・

Ⅱ.黄・・・

Ⅲ.程序设计—计算机软件—计算技术

Ⅳ.TP311

PASCAL

程序设计

黄顺珍　著

*

广东教育出版社出版发行

广东省新华书店经销

南海彩印制本厂印刷

787×1092 毫米 16 开本　17.25 印张　390 000 字

1994 年 6 月第 1 版　1994 年 6 月第 1 次印刷

印数 1—3000 册

ISBN 7—5406—2811—1/TP・2

定价 11.15 元

说　明

根据国务院关于高等学校教材工作的规定，我部承担了全国高等学校和中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978～1990年，已编审、出版了三个轮次教材，及时供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻国家教委《高等教育“八五”期间教材建设规划纲要》的精神，“以全面提高教材质量水平为中心，保证重点教材，保持教材相对稳定，适当扩大教材品种，逐步完善教材配套”，作为“八五”期间工科电子类专业教材建设工作的指导思想，组织我部所属的九个高等学校教材编审委员会和四个中等专业学校专业教学指导委员会，在总结前三轮教材工作的基础上，根据教育形势的发展和教学改革的需要，制订了1991～1995年的“八五”（第四轮）教材编审出版规划。列入规划的，以主要专业主干课程教材及其辅助教材为主的教材约300多种。这批教材的评选推荐和编审工作，由各编委会或教学指导委员会组织进行。

这批教材的书稿，其一是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的，其二是在认真遴选主编人的条件下进行约编的，其三是经过质量调查在前几轮组织编写出版的教材中修编的。广大编审者、各编审委员会（小组）、教学指导委员会和有关出版社，为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还可能有缺点和不足之处，希望使用教材的单位，广大教师和同学积极提出批评和建议，共同为不断提高工科电子类专业教材的质量而努力。

机械电子工业部电子类专业教材办公室

前　言

本教材系按机械电子工业部的工科电子类专业教材 1991～1995 年编审出版规划，由应用电子技术专业（大专）教材评审委员会征稿并推荐出版，责任编委傅丰林。

本教材由深圳大学黄顺珍编写，清华大学方棣棠教授担任主审。

本课程的参考学时为 54 学时，内容包括三篇和一个独立章，第一篇介绍计算机的基础知识、PASCAL 入门知识、4 类标准数据类型；第二篇结合程序的 4 种基本结构（顺序、选择、循环、子程序调用）介绍 PASCAL 的各种语句及函数、过程、嵌套、递归，并通过许多例题介绍程序设计方法；第三篇围绕数据类型拓宽扩展，介绍枚举、子界、数组、集合、记录、文件、指针；最后的独立章介绍 TURBO-PASCAL5.0 及 MS-PASCAL 上机操作步骤，是一章完整的实验指导书。全书在整体组织上很大程度体现出程序结构思想，内容和例题历经筛选，力求由浅入深，循序渐进，各章内容后均配有较丰富的概念性及应用性习题，以加强练习。

本教材对理工科大专以上各专业普遍适用，教学过程中，可根据专业要求，对打“*”号的内容灵活选讲。

参加本教材审阅工作的还有马玉祥、张国全老师，他们都为本书提出许多宝贵意见，在编写过程中，鲍有立、徐中州、胡庆彬、鲍清平老师给予不少有益的帮助，在此一并表示诚挚的感谢，由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编　者

1994.1.20

目　录

第一篇　程序设计基础

第二篇　程序结构

第三篇 数据类型扩展

第一篇　程序设计基础

第一章　基础知识

1.1　计算机初步知识

1.1.1　计算机组成

电子计算机，简称计算机或电脑，以速度快、存贮量大为其突出标志，而且发展迅猛。1946年出现的世界上第一台计算机，运算速度是3000次/秒，而现在，巨型机的速度可望达到上亿次，微型机的速度也是几百万次。从存贮量来说，现在微型机上使用的一张5寸1.2M软盘可存60万个汉字，大型机的存贮量就可想而知了。从1946年至今，计算机的性能指标提高了好几个数量级，而价格却不断下降，其应用遍及工业、农业、国防、科学、教育、交通、公安，直至日常生活等各个领域，极大地推动着社会变革，从古至今，还没有出现过哪种技术的发展对社会的影响是如此之大。

计算机的内部结构是复杂的，搞清它不是朝夕之功，但可以给出一个粗略轮廓。所有的计算机，总体上由如下五大部件所组成：输入设备、输出设备、运算器、存贮器、控制器。五大部件之间的关系如图1—1所示。图中，“⇒”表示数据流向，“→”表示控制信号。

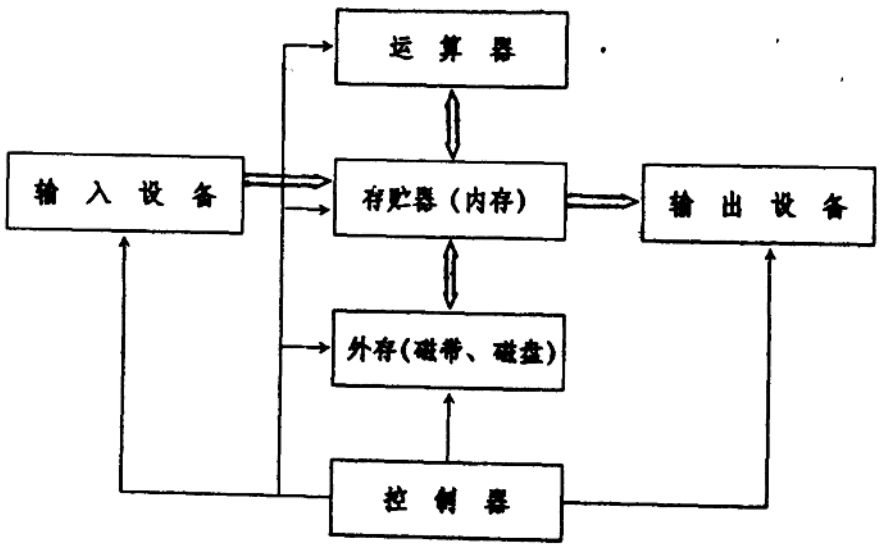


图1—1　计算机组成示意图

输入设备：用于将数据、程序等信息输入到计算机的设备，如键盘、卡片输入机等。

输出设备：用于输出中间结果或最后结果的设备，如打印机、显示器、绘图仪等。不同机器所配备的输入、输出设备通常不同。

存贮器：用于存贮数据、程序及其他各种信息（如声音数据、图象数据等）。存贮器由许多单元组成，每个单元有一个编号，称为存贮地址，简称地址，如图 1－2 所示。存贮器地址和每个单元的内容均以二进制表示，下节将介绍二进制的初步知识。

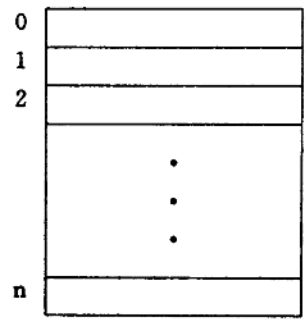


图 1－2　存贮器示意图

存贮器包括内存和外存，内存在主机内部，外存在主机外部（如磁带、磁盘等），做为辅助存贮器，存贮容量大，价格便宜，但存取速度比内存慢。

运算器：实现计算机系统的所有运算，包括两大类：算术运算和逻辑运算。算术运算是指加、减、乘、除，逻辑运算是指“逻辑与”、“逻辑或”、“逻辑非”，所有其他任何复杂的运算（如乘方、开方、指数、对数、微积分等），都由这些简单运算组合而成。

控制器：用以控制上述四大部件，使整台机器能自动地、连续地、有条不紊地工作，所以它相当于整台机器的司令部。

运算器和控制器总称中央处理器（Central Processing Unit），简称 CPU，或处理器。在大机器内，CPU 是由多个机柜组成，而在微型机内，CPU 是由大姆指大的一两片集成电路块组成。通常所说的 Z80，8088，80286 等，就是微型机上使用的 CPU 的型号。

输入设备和输出设备，通称为外部设备。

1.1.2　数制

数制，是指数的进位特征。

计算机的作用，从广义来说，是实现数据处理。任何数据，包括数值型（如 153，－36，17.8 等）和非数值型（如字符、声音、图象编码等），在机内都是以二进制形式表示。我们日常熟悉的是十进制，其进位特征是逢十进一，当然也有遇到其他数制，如 60 秒为 1 分，60 分为 1 小时，就是六十进制。二进制的特征是逢二进一，十进制的 3，写成二进制是 11，十进制的 4，写成二进制是 100，…，1～10 的十进制与二进制的对照关系如图 1－3（a）所示。

计算机中采用二进制是有其明显好处的。

其一，采用二进制，数据的产生、传送、识别都比较容易。

二进制只有两个状态：0 和 1，所以用以表示数据的元件只要有两个稳定状态即可，如电路的通和断、继电器的吸合与释放等，即可表示 0 和 1，显然，选择具有两个状态的元件比选择具有十个状态的元件容易得多，同理，信息在传送过程中，对方接收到信息

后，识别两种状态比识别十种状态也容易得多。

其二，二进制运算简单

加法：0+0=0

1+0=0+1=1

1+1=10

乘法：0×0=0

0×1=1×0=0

1×1=1

由于运算规律简单，制作一个运算器也就相对容易，成本也就随之下降。

至于其他好处，就不一一列举了。

十进制转化为二进制，可用除 2 取余方法，每次除 2 所得余数序列，即构成对应的二进制数，比如 38，转换过程如图 1－3 (b) 所示，最后得对应的二进制数为 100110。

十进制	二进制
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

(a) 对应关系

2 | 38
2 | 19 —余0(低位)
2 | 9 —余1
2 | 4 —余1
2 | 2 —余0
2 | 1 —余0
0 —余1(高位)

(b) 除 2 取余

图 1—3 十进制与二进制

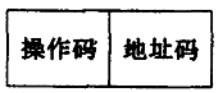
1.1.3 计算机语言

计算机语言是指计算机所能识别和执行的语言，是人机交换信息的工具。

随着计算机技术的发展，计算机语言经历了机器语言、汇编语言、高级语言三个阶段。

1. 机器语言

机器语言用于早期的计算机系统，那时软件缺乏，编写程序时，利用机器提供的指令系统，用二进制代码逐条编写。所谓程序，就是用某种语言所提供的指令或语句编写而成的指令或语句序列。例如，计算 f=b+cd，预先将 b、c、d 送内存预定单元，设 b 送 1000 单元，c 送 1001 单元，d 送 1010 单元，如图 1－4 所示，然后利用具体机器提供的指令系统编写程序，每条指令包括操作码和地址码两大部分：



操作码：指出这条指令执行什么操作，如取数，加法、减法、输出等，即指出指令的功能。

地址码：指出操作数的地址。

实现上述 f=b+cd 的解题步骤和对应的机器语言程序分别如图 1－5 和

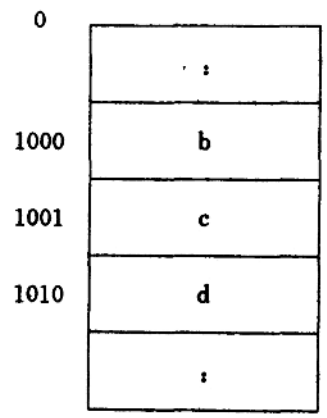


图 1—4 内存数据情形

图 1－6 所示：

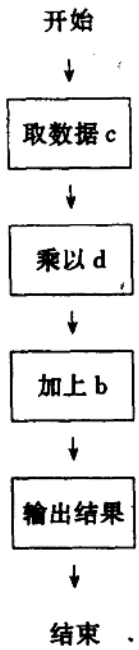


图 1－5　解题流程

```
10001001   ; c→A
01011010   ; c·d→A
00011000   ; c·d+b→A
11000001   ; A→1 号外设
00000000   ; 停机
```

图 1－6　机器语言程序

图 1－6 中，每条指令分号后为注释内容。

第 1 条指令 10001001，前 4 位 1000 是操作码，表明该指令是取数，后 4 位 1001 是地址码，表明从 1001 取数，完整意义是从 1001 取数，送 CPU 内部的一个累加器 A，由于 1001 事先存在的是 c，所以给出注释 c→A。A 称为累加器，因为它常用于寄存运算结果。

第 2 条指令 01011010，操作码 0101 表示做乘法，地址码 1010 表示乘数在 1010 单元，完整意义是累加器 A 中的内容乘以 1010 单元的内容（事先存的是 d），结果送回累加器 A。因做乘法之前，A 中已保存 c，所以给的注释是：c·d→A。

第 3 条指令 00011000，操作码 0001 指出该指令做加法，地址码 1000 指出加数在 1000 单元，完整意义是累加器 A 中内容加上 1000 单元内容（事先存的是 b），结果送回累加器 A。因执行此指令前 A 中内容为 cd，所以给的注释是：cd+b→A。

第 4 条指令 11000001，操作码 1100，指出该指令为输出指令，地址码 0001 指出在 1 号外部设备（设为打印机）输出。完整意义是将累加器 A 中内容在 1 号打印机上输出。

机器语言的特点是，整个程序都是二进制代码，编写工作量大，不直观，容易错，但机器能直接识别和执行，速度快。

读者对图 1－6 的机器语言程序是否完全明白关系不大，主要是为了与后面内容比较才列出。可以看出，对于编写一个稍大项目的程序，若用机器语言，那真是苦不堪言。

计算机是先进科学的产物，造价也很高，但如果编写程序停留在机器语言阶段，使用这么不方便，就会影响它性能的发挥，就不可能做到普遍使用，所以语言上必须有所发展，应该让用户很方便地使用，后来发展为汇编语言。

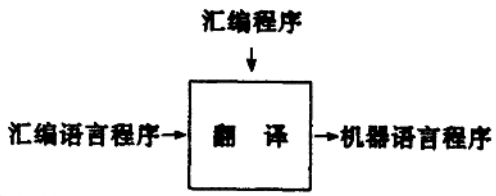
2. 汇编语言

汇编语言是把机器指令中的操作码用一个符号（通常是缩写的英语单词）来代替，使你对指令含义一目了然，所以汇编语言也叫符号语言。上述机器语言程序，改用汇编语言后如下所示，为方便对照，把机器语言重写在旁边。

机器语言程序	汇编语言程序	
10001001	mov A，1001	；c → A
01011010	mul 1010	；cd→A
00011000	add 1000	；cd+b→A
11000001	out 0001	；A→1 号打印机
00000000	halt	；停机

mov，取自英语单词 move，含义是移动。此处表示取数；mul 取自 multiplay，含义是相乘；add、out 的含义就很明白了。

显然，用汇编语言编写程序比用机器语言清楚得多，但汇编语言程序仍然依赖机器的指令系统，程序移植性差。另外，汇编语言程序不能直接执行，需要翻译成机器语言程序后才能执行，翻译工作由系统提供的专门程序（称为汇编程序）来完成，如下所示：



3. 高级语言

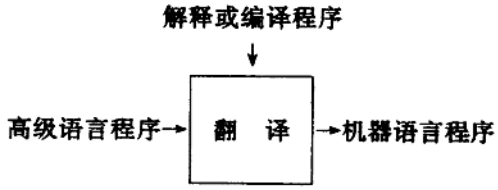
继汇编语言之后，发展为高级语言，亦称算法语言。其特点是，语句的形式接近口语化，运算表达式的形式接近数学式子，而且编写程序不必考虑具体机器的指令系统。常用的 BASIC、PASCAL、FORTRAN、COBOL 等语言，均属高级语言。上述 f=b+cd，若用 PASCAL 编写，程序如下：

```
PROGRAM sample(input,output);
VAR
  f,b,c,d:integer;
BEGIN
  write('Enter b,c,d:');
  readln(b,c,d);
  f:=b+c*d;
  writeln(f)
END.
```

由于高级语言脱离了具体的指令系统，人们编写程序时，就可把注意力集中于对具

体问题的分析和算法设计上，而且程序的移植也好，因此，高级语言为计算机的普遍使用带来极大方便。

高级语言也不能直接执行，也必须首先翻译成机器语言，翻译工作由系统提供的专门程序（解释或编译程序）来完成，如下所示：



翻译方式有两种：解释和编译。

解释：边翻译边执行，即翻译一条执行一条，速度较慢。早期的 BASIC 语言属于此类。

编译：整个程序翻译成机器语言后才能执行，速度比解释方式快。PASCAL、FORTAN、COBOL 均属编译方式。

1.2 程序设计初步知识

1.2.1 计算机解题过程

计算机靠执行程序发挥其作用，因此任何问题，首先须编成程序，才能交计算机执行，解题过程可大致归纳如下：

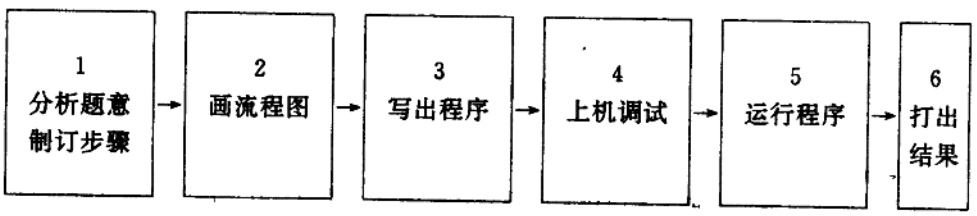


图 1－7 解题过程

下面请看具体实例。

例 1－1：设白炽灯每天用电 4 小时，电视机每天用电 3 小时，求每月用电多少度（每月按 30 天计，电压 220 伏）？

解：示意图如图 1－8 所示。

按电学基本知识，1 度＝100 瓦小时。

设白炽灯瓦数为 p1，电视机瓦数为 p2.

白炽灯每天用电量：

$w1=p1/1000\times4$（度）

电视机每天用电量：

$w2=p2/1000\times3$（度）

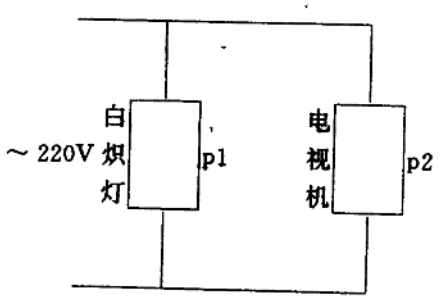


图 1－8 计算耗电量示意图

一个月用电总量：

$$w=(w1+w2)\times 30\text{（度）}$$

按以上分析，可得流程图如图 1－9，对应程序如图 1－10。

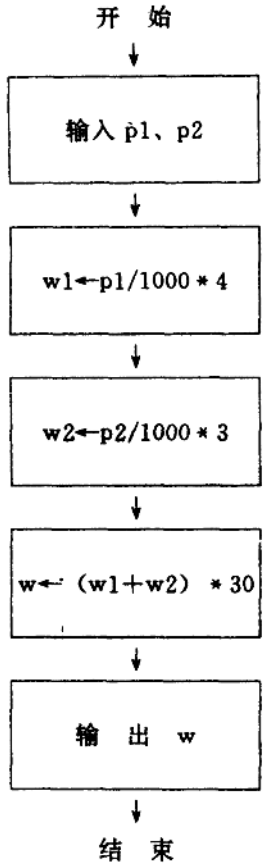


图 1－9 例 1－1 流程图

```
PROGRAM sample1 _ 1(input,output);
VAR
  p1,p2:integer;
  w1,w2,w:real;
BEGIN
  write('Enter p1,p2:');
  readln(p1,p2);
  w1:=p1/1000＊4;
  w2:=p2/1000＊4;
  w:=(w1+w2)＊30;
  writeln('w=',w)
END.
```

图 1－10 例 1－1 程序

对程序的解释见第二章。

如 1.1.3 节所述，PASCAL 属于编译性语言，图 1－10 的程序输入计算机后，必须先经编译系统翻成机器语言才能执行，但当你第一次把程序输入计算机，并开始编译时，计算机会指出你各种错误，这并不是说图 1－10 程序有什么错误，主要是你初次上机时缺乏经验，输入程序时可能这里少打了个分号，那里多打了个字母等等，编译程序都会认为是错误，然后经过修改，再编译，再修改，再编译，…，反复几次后，排除了所有错误，最后由编译程序生成可执行的机器语言程序，就可以在机上运行并输出结果了。

刚才以具体实例介绍了图 1－7 所示的解题过程。在图 1－7 中，1～2 号框的内容称为算法设计，算法，就是解题的思路或步骤。1 号框的内容主要是思考过程，2 号框的内容是把思考好的解题步骤（算法）以流程图的形式表达出来，通过分析流程图，可以检

查算法是否严密、准确，对于熟练的程序员，此步可省，或仅画出粗略框图。4～6号框属于上机实践的内容。有人说，学习程序设计主要是学算法设计，这不全面，试想，一个只会笔头写程序，不会上机调试，因而也无法判定所写程序是否正确的人，能说是学好了程序设计吗？一个合格的程序设计员，应当是既能编写程序，又能上机操作调试，同时还要能解剖别人的程序，所以应该说，学习程序设计，主要包括下面三方面内容：

程序设计能力；

程序分析能力；

上机调试能力；

这三方面能力统称为软件开发能力。

1.2.2 程序的四种基本结构

任何程序，无论多么复杂，都是由如下四种基本结构组成：顺序结构，选择结构，循环结构，子程序调用。下面分别介绍。

1. 顺序结构

图1－10所示的程序有这样特点：每执行完一条语句后，紧接着执行其后续的一条语句；即程序是按语句的先后顺序执行的，这种结构称为顺序结构。

2. 选择结构

有些程序，执行到某处时，按当时的条件，可能要跳过一部分程序，转到别处执行，产生分支操作。

例1－2：输入三个数a、b、c，求出最大数，解决这一问题的步骤，用文字叙述如下：

①输入三个数a、b、c；

②比较a与b，若a≥b，做第③步，否则做第④步；

③比较a与c，若a≥c，做第⑤步，否则做第⑥步；

④比较b与c，若b≥c，做第⑦步，否则做第⑥步；

⑤打出“a最大”，转⑧；

⑥打出“c最大”，转⑧；

⑦打出“b最大”，转⑧；

⑧结束。

这段文字叙述有点像走迷宫一样，难以把思路理直，若用流程图表示出来，则比较清楚，解决这一问题的流程如图1－11所示。

图1－11中，梭形框表示条件判断，比如②号框，用于判断a≥b否，判断结果只有两种可能：要么成立（yes），要么不成立（no），所以梭形框有两个出口，也就是两个分支，根据判断结果，条件成立转③号框，条件不成立转④号框，这就产生两种不同的选择，故称这种程序结构为选择结构。这一流程图所反映的算法，与上面的文字叙述相一致，但读者自然会感到，看这种流程图比看文字叙述清晰得多，后面介绍的结构图，比起流程图来，对于分析算法会更有利。

3. 循环结构

实际问题中，有些程序的部分语句需要重复执行多次，这种程序结构称为循环结构。

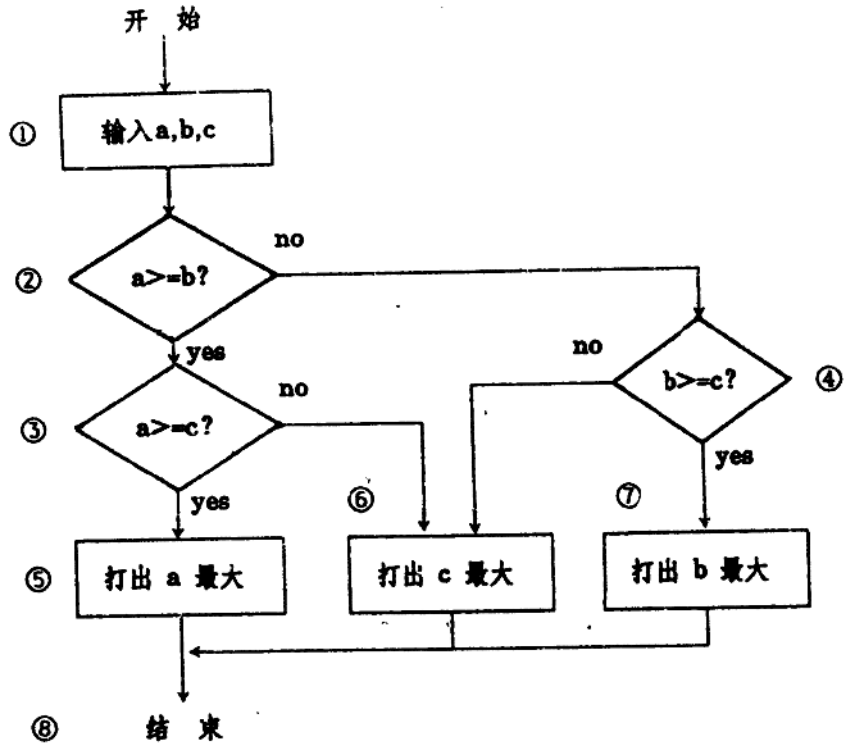


图 1－11　含选择结构的流程图

例 1－3：全班 30 个同学，输入每人三门课的成绩，求出每人平均成绩，并输出结果。

解决这一问题的步骤，用文字叙述如下：

①置学号 n＝1；

②输入第 n 号同学的三门课成绩：score1，score2，score3；

③求出三门课平均成绩：average＝(score1＋score2＋score3) /3；

④输出平均成绩 average；

⑤学号＋1，即 n←n＋1；

⑥判断 n，若 n≤30 转②，否则转⑦；

⑦结束。

与这段文字叙述对应的流程图如图 1－12 所示。

图 1－12 中，②～⑤号框的内容要重复执行 30 次，称这种程序结构为循环结构。

现在，请读者再比较图 1－9、图 1－11、图 1－12，进一步体会顺序结构、选择结构、循环结构的含义。

4. 子程序调用

一台复杂的机器，制造时通常是先分成多个部件单独制造，然后把这些部件装配起来，成为一台完整的机器。

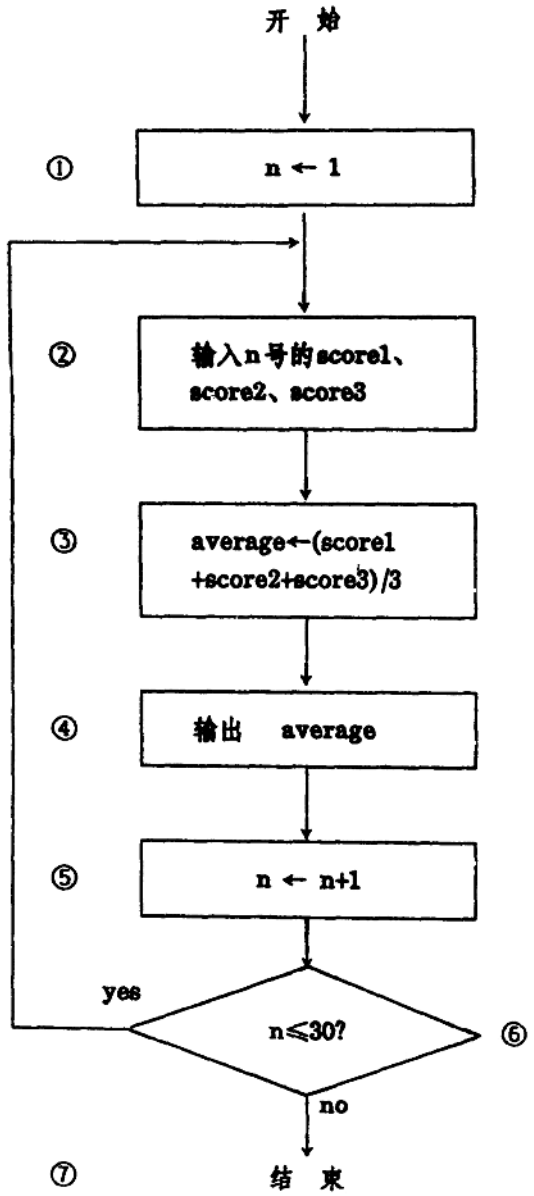


图 1－12 循环结构流程图

编写程序也是这样，一个实际课题，先从功能上分成几个模块，每个模块首先单独调试，然后供主程序调用，装配成一个大程序，从主程序角度看，每个功能模块就是一