

全国计算机等级考试

最新大纲

FORTRAN

语言程序设计

(二级)

● 郑莉 邵毓华 编著

FORTRAN 语言程序设计(二级)

等考
直通车



机械工业出版社
China Machine Press

全国计算机等级考试

FORTRAN 语言程序设计(二级)

郑莉 邵毓华 编著



机械工业出版社

本书针对 1998 年教育部考试中心修订的“全国计算机等级考试（二级）大纲”而编写的，内容包括 FORTRAN 语言程序设计的基础知识和结构化程序设计，完全覆盖了大纲中“FORTRAN 语言程序设计”部分的知识点。着重帮助读者学习程序设计方法，提高编程能力。全书共分 11 章，每章包括：正文、应用举例、小结和习题、习题答案。所有例题都在 Microsoft FORTRAN 5.1 编译环境中进行调试，本书主要适用对象是准备参加全国计算机等级考试（二级）FORTRAN 语言考试的读者，同时也是初学编程者很好的入门教材，还可用作大专院校或成人培训教材。

图书在版编目 (CIP) 数据

全国计算机等级考试 FORTRAN 语言程序设计（二级）/郑莉，邵毓华编著.

—北京：机械工业出版社，2000.10

ISBN 7-111-01892-3

I. 全... II. ①郑... ②邵... III. FORTRAN 语言-程序设计-水平考试-自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 66557 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划：胡毓坚

责任编辑：刁明光

责任印制：郭景龙

北京京丰印刷厂印刷·新华书店北京发行所发行

2000 年 10 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{32}$ ·12.5 印张·307 千字

0 001—3000 册

定价：20.00 元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话：(010) 68993821、68326677 - 2527

出版说明

全国计算机等级考试自从推出以来,已有上百万人次参加了考试,从而有力地推动了计算机应用技术在中国的发展。

为了更好地普及计算机基础知识,全方位地为广大应试者服务,机械工业出版社聘请了清华大学、北方交通大学、北京科技大学等院校长期从事全国计算机等级考试教育、具有丰富教学经验的老师,编写了本套《全国计算机等级考试教程》系列教材。

本套教材思路严谨、立意深刻,是在这些老师长期积累的教学经验的基础上编写而成的,因而紧扣考试大纲;此外,本套教材概念清晰、层次分明、深入浅出,是一套比较完整、系统的应试教材;所附习题完全模拟全国计算机等级考试的考试样题,每道习题均附有答案,实用性、参考性较强,因此对应试者在全国计算机等级考试的学习中起到指导作用。本套教材最大的特点是既有理论讲解,又有实践环节,应试者通过反复练习,使那些平时难以深入理解和灵活运用的理论得以理解和运用,通过自己动手动脑解答习题,达到举一反三的效果,从而为顺利通过全国计算机等级考试打下坚实的基础。

另外,为了使应试者能够尽快通过考试,机械工业出版社还配套出版了《全国计算机等级考试试题详解及模拟试卷》,欢迎广大读者提出宝贵意见。

前 言

全国计算机等级考试是由教育部考试中心主办,用于测试应试人员计算机应用知识与能力的等级水平考试。其目的在于推进计算机知识的普及,促进计算机技术的推广应用,以适应社会主义经济建设的需要,为用人单位录用和考核工作人员服务。考试分为四个等级,其中二级考核应试者软、硬件基础知识和使用一种高级计算机程序设计语言编制程序、上机调试的能力。

本书是根据教育部考试中心 1998 年重新修订的“全国计算机等级考试(二级)”大纲而编写的,内容包括程序设计基础知识和 FORTRAN 语言结构化程序设计,完全覆盖了大纲中“FORTRAN 语言程序设计”部分的知识点。着重帮助应试者学习程序设计方法,提高编程能力。作者多年来在清华大学从事计算机基础教学、考试工作,深感对于初学程序设计的读者来说,要提高编程能力有赖于大量的上机练习,通过练习掌握分析问题和解决问题的方法。因此本书概念和语法讲解简明清晰,主要通过丰富的例题介绍概念和语法的应用,解题思路和程序设计方法,引导读者通过实践来学习。这一点对于准备参加等级考试的读者尤为重要,只有经过平时大量的训练,考试时才能从容应对。另外,与一般教科书不同,本书中的多数例题和习题都是按照样卷和往届考题的题型设计的,并且直接包含了大量往届考题,这样有利于读者有针对性地准备考试。

全书共分 11 章,每一章包括:学习要求、正文、应用举例、小结和习题。全部例题都在 Microsoft FORTRAN 5.1 编译环境中进行调试并能正确运行。小结部分是对每章知识点和例题的简要总结,方便读者复习记忆。习题包含了新大纲要求的全部知识点和题型,并附有答案。

本书的主要适用对象是准备参加二级 FORTRAN 语言考试的读者,同时也是初学编程者很好的入门教材,还可用作大专院校或成人培训教材。

参加本书编写的还有董渊博士和孟鸿俐。黄维通博士为本书的编写提供了很多帮助,在此深表感谢。

· 编 者

目 录

出版者的话

前言

第 1 章 程序设计基础知识	1
1.1 信息表示的方法——二进制	1
1.1.1 二进制数与十进制数之间的转换	1
1.1.2 十进制数、八进制数与十六进制数之间的转换	3
1.1.3 计算机中信息的表示	3
1.1.4 溢出与误差	4
1.2 算法与结构化程序设计	5
1.2.1 算法的基本特征	5
1.2.2 算法的类型	6
1.2.3 算法的基本结构	6
1.2.4 算法的描述工具	7
1.2.5 结构化程序设计	8
1.2.6 编程、测试和调试	9
1.3 计算机语言、程序及上机过程	9
1.3.1 计算机语言	9
1.3.2 高级语言的翻译	10
1.3.3 操作系统介绍	11
1.3.4 FORTRAN 程序设计上机流程	11
1.4 本章小节	12
习题一	12
第 2 章 FORTRAN 程序的结构及书写格式	14
2.1 FORTRAN 程序的结构特点	14
2.2 FORTRAN 语言的基本成分	15
2.3 程序的书写格式	17
2.4 本章小结	17
习题二	18
第 3 章 数据类型及其运算	20
3.1 常量	20
3.2 变量	21
3.2.1 变量及变量的命名	21
3.2.2 变量的类型说明	22
3.3 符号常量	23

3.4	运算符	23
3.5	表达式及其运算规则	25
3.6	应用举例	27
3.7	本章小结	29
	习题三	30
第4章	FORTRAN 语言最基本的语句	32
4.1	赋值语句	32
4.1.1	算术赋值语句	32
4.1.2	逻辑赋值语句	33
4.2	表控格式的输入和输出	33
4.2.1	表控输出	33
4.2.2	表控输入	34
4.3	格式输出	35
4.3.1	格式输出语句的一般形式	35
4.3.2	格式编辑符	36
4.3.3	格式输出的纵向控制	38
4.4	格式输入	39
4.4.1	格式输入语句的一般形式	39
4.4.2	格式编辑符	39
4.5	格式输入输出时应注意的问题	41
4.5.1	格式的重复作用	41
4.5.2	输入/输出语句与格式语句的相互作用	42
4.6	暂停语句、停止语句和结束语句	42
4.7	赋初值语句	43
4.8	转移语句	43
4.9	应用举例	44
4.10	本章小结	45
	习题四	45
第5章	选择结构程序设计	48
5.1	逻辑 IF 语句	48
5.2	块 IF 结构	51
5.3	简化的块 IF 结构	53
5.4	选择结构的嵌套	55
5.5	ELSEIF 选择结构	58
5.6	应用举例	60
5.7	本章小结	63
	习题五	64
第6章	循环结构程序设计	68
6.1	当型循环	68

6.2	直到型循环	70
6.3	DO 循环	74
6.3.1	一般形式	74
6.3.2	逻辑流程	75
6.3.3	循环次数的计算	75
6.3.4	注意事项	76
6.4	三种循环的比较	77
6.5	循环的嵌套	79
6.5.1	多重 DO 循环	79
6.5.2	循环和选择结构的嵌套	80
6.5.3	注意事项	81
6.6	隐 DO 循环	81
6.7	本章小结	82
	习题六	83
第 7 章	数组的应用	86
7.1	数组的定义	86
7.1.1	类型说明语句	86
7.1.2	DIMENSION 语句	87
7.1.3	DIMENSION 语句和类型说明语句结合	87
7.1.4	可调数组的概念	88
7.2	数组元素的表示方式	88
7.3	数组的存储	88
7.4	用 DATA 语句对数组赋初值	90
7.5	数组的输入与输出	92
7.5.1	用数组名对整个数组进行输入/输出	92
7.5.2	用 DO 循环进行数组的输入/输出	93
7.5.3	用隐 DO 循环进行数组的输入/输出	94
7.6	应用举例	95
7.6.1	一维数组的应用	95
7.6.2	二维数组的应用	103
7.7	本章小结	106
	习题七	107
第 8 章	字符处理	111
8.1	字符型数据	111
8.1.1	字符串	111
8.1.2	字符型变量	111
8.1.3	字符型数组	112
8.1.4	子字符串	112
8.2	字符型数据的赋值	113

8.2.1	用赋值语句赋值	113
8.2.2	用 DATA 语句赋初值	114
8.2.3	用 PARAMETER 参数语句定义字符型符号常量	114
8.3	字符型数据的运算	114
8.3.1	字符表达式	114
8.3.2	字符型数据的关系运算	114
8.3.3	用于字符处理的内部函数	115
8.4	字符型数据的输入与输出	115
8.4.1	字符型数据用表控方式输入和输出	116
8.4.2	字符型数据的格式输入与输出	116
8.5	字符型常量的应用举例	117
8.6	本章小结	119
	习题八	120
第 9 章	函数和子程序	123
9.1	内部函数	123
9.2	语句函数	124
9.2.1	语句函数的定义	124
9.2.2	语句函数的调用	124
9.2.3	语句函数调用举例	125
9.3	函数子程序	126
9.3.1	函数子程序的定义	126
9.3.2	函数子程序的调用	128
9.4	子例程子程序	129
9.4.1	子例程子程序的定义	129
9.4.2	子例程子程序的调用	130
9.5	模块间的数据传递——形实结合	133
9.5.1	变量作为形参	133
9.5.2	数组作为形参	134
9.5.3	子程序名作为形参	136
9.6	应用举例	137
9.7	本章小结	140
	习题九	140
第 10 章	数据联系	145
10.1	公用语句	145
10.1.1	无名公用区	145
10.1.2	有名公用区	146
10.1.3	公用区的使用方法	146
10.2	数据块子程序	147
10.3	应用举例	148

10.4 本章小结·····	152
习题十·····	152
第 11 章 文件 ·····	155
11.1 文件与记录·····	155
11.1.1 文件 ·····	155
11.1.2 记录 ·····	155
11.2 文件的分类及操作·····	156
11.2.1 文件的分类 ·····	156
11.2.2 文件操作 ·····	156
11.3 文件的打开与关闭·····	156
11.3.1 文件打开 ·····	156
11.3.2 文件关闭。 ·····	158
11.4 文件的读写·····	158
11.5 文件定位·····	158
11.6 顺序文件的存取·····	159
11.7 直接文件的存取·····	164
11.8 本章小结·····	166
习题十一·····	166
附录 ·····	169
附录 A FORTRAN77 内部函数·····	169
附录 B 习题答案 ·····	171
附录 C 全国计算机等级考试 FORTRAN 语言程序设计(二级)笔试样卷 ·····	173
附录 D 全国计算机等级考试 FORTRAN 语言程序设计(二级)笔试样卷答案 ·····	185
附录 E 全国计算机等级考试考试大纲 FORTRAN 语言程序设计(二级) ·····	186
附录 F ASCII 字符编码表 ·····	189

第 1 章 程序设计基础知识

1.1 信息表示的方法——二进制

计算机采用的是二进制数系,所有的外界信息在被转化为二进制数后,计算机才能进行传送、存储和加工处理。

计算机信息的计量单位有以下几种:

(1)位(bit):表示一个二进制位,是计算机中信息存储的最小单位。

(2)字节(Byte):八个二进制位就是一个字节(1Byte = 8 bit)。是计算机信息存储中最常用的基本单位。存储器的容量通常是用多少字节来表示的,比如我们常用的标有 1.44M 的 3.5in 软盘,就是指其容量为 1.44M 字节。字节常用的单位有:

K 字节 1KB = 1024 B

M (兆) 字节 1MB = 1024 KB

G 字节 1GB = 1024 MB

(3)字(word):又称计算机字。是由一定数量的字节组成,是计算机信息表示的独立单位。常用的固定字长有 8 位、16 位、32 位等。

我们最熟悉的是十进制数系,但在计算机中使用的却是二进制,虽然便于计算机的存储和运算,但人们阅读、处理却不方便。由于 3 位二进制数刚好可以转换为 1 位八进制数,4 位二进制数刚好可以转换为 1 位十六进制数,因此在计算机程序设计中,也经常使用八进制和十六进制。各种进制的基数、进位原则、基本符号如表 1-1 所示。

表 1-1 进 制 表

进 制	基 数	进位原则	基本符号
二进制	2	逢 2 进 1	0, 1
八进制	8	逢 8 进 1	0, 1, 2, 3, 4, 5, 6, 7
十进制	10	逢 10 进 1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
十六进制	16	逢 16 进 1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1.1.1 二进制数与十进制数之间的转换

二进制数采用的是“逢二进一”的进位记数方式,而十进制数是“逢十进一”的方式,一个十进制数 N,把它写成以 2 为底的幂级数的形式:

$$N = \underbrace{\sum_{i=0}^{n-1} a_i 2^i}_{\text{整数部分}} + \underbrace{\sum_{i=-1}^{-m} a_i 2^i}_{\text{小数部分}}$$

下面我们详细说明转换方法:

(1) 幂级数展开法

(2)除 2 取余法

【例 1-1】将十进制数 68 转化为二进制数。

$$(68)_{10} = 64 + 4$$

方法二:除 2 取余法。

所以: $(68)_{10} = (1000100)_2$

【例 1-2】将二进制 $(11111111.11)_2$ 转化为十进制数

• 2 •

1.1.2 十进制数、八进制数与十六进制数之间的转换

1. 二进制数与八进制数、十六进制数之间的转换

二、八、十六三种进制的权值有内在的联系,每位八进制数相当于三位二进制数($2^3 = 8$),每位十六进制数相当于四位二进制数($2^4 = 16$)。下面我们结合实际例题来学习它们之间的转换。

将二进制数从小数点开始,向左右分别按三(四)位为一个单元划分,每个单元单独转换成为一个八进制(十六进制)数,就完成了二进制到八、十六进制数的转换。八(十六)进制数的每一位,分别独立转换成三(四)位二进制数,除了左边最高位,其他位如果不足三(四)位的要用0来补足,按照由高位到低位的顺序写在一起,就是相应的二进制数。

【例 1-3】将二进制数 $(1000100)_2$ 转换为八、十六进制数

解: $(1000100)_2 = (\underline{1\ 000\ 100})_2 = (104)_8$

$(1000100)_2 = (\underline{100\ 0100})_2 = (44)_{16}$

【例 1-4】将十六进制数 $(F7)_{16}$ 转换为二进制数

解: $(F)_{16} = (\underline{1111})_2$

$(7)_{16} = (\underline{0111})_2$

$(F7)_{16} = (\underline{1111\ 0111})_2 = (11110111)_2$

注意:在八、十六进制数到二进制数的转换中,将每一位分别独立转换成二进制数即可,但是每一位都必须写满三或四位(分别对应于八、十六进制数)。在这个例题中,低位十六进制数7,转换为二进制是 $(7)_{16} = (111)_2$,在完成最后的拼接时,一定要在高位补一个0,否则, $(F7)_{16}$ 成为了 $(1111111)_2 = (7F)_{16}$,真可谓“失之毫厘,谬以千里”。

2. 十进制数、八进制数、十六进制数之间的转换

在进行这三种数制转换时,我们可以采取以二进制数为过渡的方法:把要转换的数先全部转换为二进制数,再由二进制数转换成我们需要的最终结果。

【例 1-5】将十六进制数 $(1000)_{16}$ 转换为十进制数结果是()。

A) 4096 B) 1024 C) 2048 D) 8192

解: $(1000)_{16} = (1\ 0000\ 0000\ 0000)_2 = 1 \times 2^{12} = (4096)_{10}$

答案: A。

1.1.3 计算机中信息的表示

1. 数的表示

数不仅有整数、小数之分,还有正数、负数之分。计算机用二进制编码表示数,也要能把它们区别开来才行,一般情况下,计算机内部用二进制数最高位的“0”和“1”来分别表示正数与负数。计算机中,小数点固定在某一位置上的数,称为定点数。当小数点放在符号位之后,数值最高位之前,表示的是定点小数;当小数点放在右边最低位之后,表示的是定点整数。小数点位置不固定的数称为浮点数。一个实数数 N 可以写成:

$$N = \pm S \times 2^E$$

其中 E 是一个整数,确定 N 的小数点的位置,称为阶码;S 是一个定点小数,表示 N 的有效位数,称为尾数。这样形式的数称为浮点数。

2. 字符的表示

在本节开始时,我们讲过,所有的信息在输入计算机后,都是用二进制数来表示的,这就要靠编码技术。西文字符的最流行编码方案是“美国信息交换标准代码”,简称 ASCII 码。包括了十个数字,大小写英文字母和专用字符共 95 种可打印字符和 33 个控制字符。ASCII 码用一个字节中的七位二进制数来表示一个字符,最多可以表示 $2^7 = 128$ 个字符。

3. 中文的表示

西文只有 26 个的字母,中文却有几千个常用字,每一个都要有唯一的编码对应,怎么表示呢?现在,已经有很多种的编码方式,其中在我国的“国家标准信息交换用汉字编码”(GB2312 - 80 标准)中,用两个字节中的两个七位二进制来表示一个汉字。

4. 其他信息

大家知道,现在的计算机处理的信息远不止数字和字符,多媒体计算机处理的声音、图象等又是怎样表示的呢?无一例外,全是二进制数,但大多数要经过特殊的输入设备的转换,如果有兴趣,可以参考这方面的有关书籍。

1.1.4 溢出与误差

一个字节的二进制数也只能表示 28 个数,可我们在计算中完全有可能会用到更大的数,怎么办?还有,在本节前面讲数制转换时,我们举的例子中,转换都可以很精确地完成,是不是每个十进制数都可以精确用二进制数表示?下面我们来看看。

1. 溢出

用二进制表示数,一定长度的二进制数所能表示的范围总是有限的。在运算中,比如两数相加的结果,超出了机器所能表示的这个范围,就会产生溢出。这时,计算机一般就会给出错误的信息。

2. 误差

【例 1-6】用二进制表示 0.625 和 0.626。

解: $(0.625)_{10} = 1/16 = 2^{-4} = (0.0001)_2$

$(0.626)_{10} = 1/16 + 1/1000 = 1/16 + 1/1024 + \dots \approx (0.0001000001)_2$

在上面的例子中,如果同样用 8 个二进制位来表示,那么,计算机就不能区别出 0.625 和 0.626 的差别,都是 0.00010000。事实上,我们并不能准确的将十进制换算为等值的二进制,再加上表示的位数有限,因此,判断两个差别很小的实数时,就会有问题,这种情况就是出现了误差。不能准确换算而产生的误差叫换算误差,由于位数限制进行四舍五入而产生的误差叫舍入误差。在科学计算中误差是一个很重要的问题,因为在计算过程中误差的多次累积就可

能导致结果的不准确。

溢出和误差产生的很重要的原因之一是二进制位数限制而造成的。解决的一个方法就是增加位数,用更多的二进制位,就可以表示更大的数;用更多的二进制来表示同样的数,精度就可以提高。

1.2 算法与结构化程序设计

这本书介绍的是 FORTRAN 程序设计,但是,对于什么是程序设计,怎样设计程序,在程序设计时应当注意哪些问题,也许每个人都会有自己的想法。“程序设计就是用某种语言编写一个程序来完成某些特定工作,就是写程序”,很多人都会这么想,真是这么简单吗?其实不然,著名的计算机科学家沃思先生曾给出这样一个公式:

$$\text{程序} = \text{算法} + \text{数据结构}$$

也就是说,程序设计至少要从算法设计和数据结构设计两个方面来考虑。那软件又是什么呢?程序设计在软件开发过程中的地位又是什么呢。最初,软件和程序是一个概念,后来,程序越做规模越大,软件的概念发生了变化,软件是由程序以及开发、使用、维护过程中的全部文档所组成,软件开发成为了一个重要的工程学科——软件工程。软件工程项目开发过程如表 2-1 所示。

表 2-1 软件的生命期

阶 段	内 容	任 务	成 果
定义阶段	计划	软件功能、可行性分析、经费预算、进度等	软件计划说明书
	需求分析	理解用户的要求并表达	需求分析说明书
开发阶段	设计	模块化设计,自顶向下、逐步细化	设计说明书
	编程	编写结构化的源程序	源程序
	调试 测试	发现并排除错误	可执行系统 测试报告
运行阶段	运行	维护	改进的系统

用软件工程的眼光来看,程序设计只是软件工程中的一部分工作。

不管是沃思先生给出的公式,还是软件工程的观点,算法都是程序设计的一个主要部分。算法就是解决问题的方法和步骤。在计算机科学意义之下,算法主要用来描述计算机解决特定问题的操作步骤。

1.2.1 算法的基本特征

1. 有穷性

一个算法必须在每一种可能情况下,都能在有限步执行后结束。

回想一下关于误差的讨论,一个小数,可以展开成一个以 2 为底的幂级数,理论上讲,只要取足够多的项,就能满足任意精度的要求,当我们取无穷多项,就是完全精确的。事实上,由于

数字位数的限制,我们只能取有限项,于是有了误差。从算法的角度讲,累加的计算必须在有限步执行后完成,也就是说,只能对有限项进行计算,而不可能对无穷项进行,这也就是误差不可避免的原因。

2. 确定性

算法的每一个步骤,执行顺序和内容必须有确切的含义,对应于一个确定的操作,不能有第二种理解方式。

3. 有效性

又称为可操作性,算法的每一个步骤,必须是可以实现的而且能够得到预期的结果。分母为零的分式运算就是一个典型的无效算法。

1.2.2 算法的类型

算法是为解决特定问题而设计的,而要解决的问题门类繁多,因此,算法也会呈现出多样性与复杂性,但这并不妨碍我们对算法作一个大致的分类。对于计算机处理而言,算法根据其应用方向可以大概分为数值算法和非数值算法两种。

数值算法可以定义为“数学问题构造性解法的一个完备而确切的描述。并规定方法中仅允许加、减、乘、除等基本算术运算”。数值算法常用于科学计算领域中。

非数值算法则广泛应用在信息(数据)处理的场合。这类问题常常要对大量的数据进行加工处理(搜集、转换、分类、组织、检索、存储、维护等),有时还要绘制数据分布曲线或印出报表,还可以根据加工后得到的信息寻找规律,进行预测。这些处理工作一般不涉及复杂的数学问题,但数据量大,数据的类型和结构也较复杂。

1.2.3 算法的基本结构

一个算法是由“结构”和“原操作”两个部分构成的。最基本的结构有三种,它们是:顺序结构、分支结构和循环结构。原操作包括:输入、输出、求一个表达式的值并赋给一个变量、比较二个变量等。

下面我们用工来分别描述一下三个基本结构。

1. 顺序结构

这个结构是由若干个依次执行的处理块组成。图 1-1 是包含二个处理块的序列,其中 S1、S2 分别代表不同的处理块。

顺序结构是任何一个算法都离不开的基本主体结构。

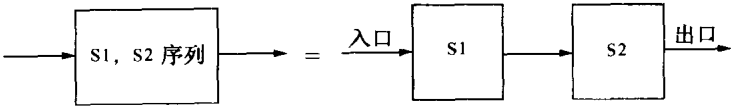


图 1-1 顺序结构示意图

2. 分支结构

最基本的分支结构是二分支结构。它是根据某一逻辑条件是否成立而决定选择哪一个分支上的处理块去执行,所以分支结构又称选择结构。图 1-2 是分支结构示意图。

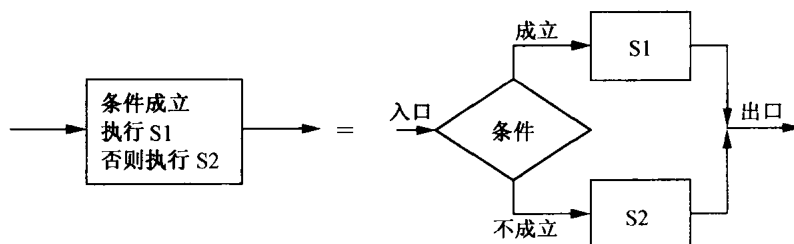


图 1-2 分支结构示意图

分支结构总是以条件或情况的判断为起始点的,它是人脑思维判断活动的抽象。

3. 循环结构

循环结构是指在算法设计中,从某处开始有规律地反复执行某一处理块,该处理块称为循环体。循环体执行多少次是由一个控制循环的条件决定的。当控制条件成立时,重复执行特定的处理块。典型循环结构示意图如图 1-3 所示。

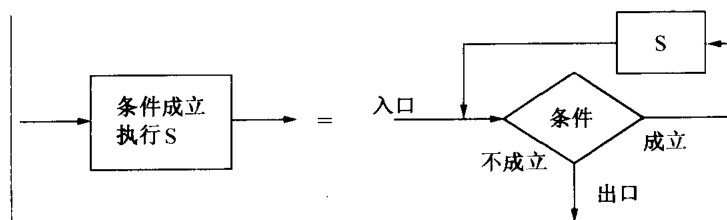


图 1-3 循环结构示意图

循环结构反映了人们在处理某一事件时,对不同数据执行同一操作的工作方式。

这三个结构块中的每一个都具有一个入口和一个出口,而结构块中的每一处理块,如上面图中的 S、S1、S2,也都具有一个入口和一个出口。由这三种基本结构可以繁衍出无限多的结构来,可以表示任意一个复杂的算法。

1.2.4 算法的描述工具

一个设计好的算法需要用一种语言来描述。用自然语言来描述,是最容易想到的。但是用自然语言描述算法时,往往一个人认为明确的操作,另一个人却觉得不明确,或者两个人都认为明确了,但实际上有着不同的理解。算法的基本特征,特别是确定性,要求有一种精确的,无歧义的描述语言对算法进行描述。

计算机语言既能描述算法,又能实际执行,但算法思想容易被语言细节所掩盖。从软件开发人员方面看,在使用程序设计语言进行编写之前,需要对算法的逻辑关系进行详细分析,并给出清晰的表达,使它成为编码的依据。有了算法描述工具,程序员可以不受具体计算机语言的语法约束,集中精力去研究所给问题的逻辑。从用户和维护人员来看,如果需要