

新编 FORTRAN 程序设计教程

李惠然 刘辰生 主编
范贻明 主审



海洋出版社

新编 FORTRAN 程序设计教程

李惠然 刘辰生 主编

范贻明 主审

海洋出版社

1999 年·北京

内容提要

本书详细地介绍了 FORTRAN 语言结构化程序设计方法和风格。介绍了近年来发布的 MS-FORTRAN5.1 的新的扩展功能,弥补了 FORTRAN 不能进行图形图像处理的缺陷。全书程序可在 Windows 及 DOS 两种平台下调试运行。本书可作为大学本、专科程序设计课的教材,也可作为参加全国计算机考试(二级)的辅导教材。

图书在版编目(CIP)

35350/22

新编 FORTRAN 程序设计教程/李惠然,刘辰生主编. —北京:海洋出版社,1999. 3
ISBN 7-5027-4427-4

I. 新… I. ①李… ②刘… III. FORTRAN 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1997)第 25033 号

责任编辑:阎世尊

责任校对:俞丽华

海洋出版社 出版发行

(100081 北京市海淀区大慧寺路 8 号)

北京兰空印刷厂印刷 新华书店发行所经销

1999 年 3 月第 1 版 1999 年 3 月北京第 1 次印刷

开本:787×1092 1/16 印张:20.18

字数:500 千字 印数:1—5000

定价:23.00 元

海洋版图书印、装错误可随时退换

编委会成员

主 编：李 惠 然 刘 辰 生
副主编：柴 欣 李 阳 明 刘 建 臣
编 委：付 灵 丽 陈 冀 川 张 红 梅
侯 立 坤 陈 德 生
主 审：范 贻 明

前 言

FORTRAN 语言是世界上最早出现的高级语言,从 1954 年第一个 FORTRAN 语言版本问世到现在已有 45 年以上的历史,它在上百种高级语言中保持经久不衰,是目前世界上流行最广泛、最适用于数值计算的高级语言。在科学研究、工程计算、实验数据处理等许多领域具有众多用户。特别是近年来,自 FORTRAN77 之后出现了 Fortran90、Microsoft FORTRAN4.0、5.0、5.1 等版本。MS-FORTRAN5.1 语言与过去普遍应用的 FORTRAN77 相比,除继承了 FORTRAN77 的全部优点外,在图形和图像的处理方面实现了突破。本书以 MS-FORTRAN5.1 为背景,由浅入深,介绍了基本 FORTRAN 语言的内容外,又介绍了 MS-FORTRAN5.1 的集成化的编程环境,使用户除可以在 DOS 环境下工作外,也可以在 Windows 界面下工作。书中提供了屏幕图形输入的命令及扩展了的数据类型等。MS-FORTRAN5.1 的图形处理功能完全能与 Turbo C、Pascal 等相媲美。

本书共十二章,它们之间具有相对的独立性和内在的联系,第一章到第十章详细地介绍了 FORTRAN 结构化程序设计的方法和程序设计的风格,同时突出介绍了常用算法。第十一章介绍 MS-FORTRAN5.1 的安装和集成环境的使用,第十二章介绍 MS-FORTRAN5.1 的高级功能。

本书适于做大学本、专科理工各类学生的程序设计课程的教材,也可做国家和各省市计算机二级考试的辅导教材。读者在学习完全书内容以后,配合上机实习,能独立的编写出解决各专业典型问题的程序,为以后课程设计,毕业设计打下牢固基础。

本书前十章讲授 34~38 学时,上机 34 学时。

参加本书编写的有李惠然、刘辰生、柴欣、李阳明、刘建臣、付灵丽、陈冀川、张红梅、侯立坤、陈德生等。

本书由范贻明教授做了细致地审阅。在编写及程序的调试过程中,得到了周建伟、薛美云、吴德宏等同志的大力帮助和支持,编者在此一并表示感谢。

由于编者水平有限,时间仓促,书中不妥之处在所难免,恳请读者及有关专家批评指正。

编 者
1999 年

目 录

第一章 程序设计及算法的基本概念	1
1.1 算法及算法的表示方法	1
1.1.1 算法概念	1
1.1.2 计算机算法应具备的特点	1
1.1.3 算法的表示方法	2
1.2 算法的三种基本结构	3
1.2.1 顺序结构	3
1.2.2 选择结构	3
1.2.3 循环结构	3
1.2.4 用 N—S 结构化流程图表示算法	5
1.2.5 用 PAD 图及其它图示方法表示算法	8
1.3 结构化程序设计与程序设计风格	8
1.3.1 结构化程序设计的由来	8
1.3.2 结构化程序设计的方法	9
1.3.3 程序设计风格	9
1.4 FORTRAN 的发展概况	10
习题一	11
第二章 FORTRAN 程序设计基础知识	12
2.1 FORTRAN 语言简单程序分析	12
2.2 FORTRAN77 源程序的书写格式	14
2.3 FORTRAN77 字符集	15
2.4 FORTRAN77 数据类型	16
2.4.1 在 FORTRAN77 中所使用的数据及数据类型	16
2.4.2 FORTRAN77 数据类型的特点	16
2.5 常量	16
2.5.1 整型常量	17
2.5.2 实型常量	17
2.6 变量	18
2.6.1 变量名	18
2.6.2 变量的类型	19
2.7 FORTRAN77 的内部函数	20
2.8 数组	22
2.8.1 数组概念的引出	22

2.8.2 数组的定义	23
2.8.3 数组元素及下标	25
2.8.4 数组在内存中的存储顺序	26
2.9 FORTRAN77 算术表达式	27
2.9.1 算术运算符和运算规则	28
2.9.2 算术表达式及其运算顺序	28
2.9.3 算术表达式运算中的类型问题	29
习题二	30
第三章 顺序结构及基本语句	31
3.1 顺序结构	31
3.2 赋值语句	32
3.2.1 赋值语句的性质和作用	32
3.2.2 算术赋值语句	32
3.3 输入和输出	33
3.3.1 输入/输出的基本概念	33
3.3.2 输入语句	33
3.3.3 输出语句	34
3.4 表控输入/输出	34
3.4.1 表控输入语句	34
3.4.2 表控输出语句	35
3.5 格式输入/输出	37
3.5.1 微机常用格式输入语句	37
3.5.2 微机常用格式输出语句	37
3.5.3 格式语句	37
3.5.4 可重复格式说明符	38
3.5.5 非重复格式说明符	42
3.6 参数语句	44
3.7 DATA(初值)语句	45
3.8 END、STOP 及 PAUSE 语句	45
3.8.1 结束语句	45
3.8.2 停止语句	45
3.8.3 PAUSE 语句	46
3.9 程序举例	46
习题三	47
第四章 逻辑运算及选择结构	49
4.1 关系运算符和关系表达式	49
4.1.1 关系运算符	49
4.1.2 关系表达式	49
4.2 逻辑运算符和逻辑表达式	50
4.2.1 逻辑常量	51

4.2.2 逻辑变量	51
4.2.3 逻辑运算符	51
4.2.4 逻辑表达式	52
4.2.5 逻辑表达式的运算规则	52
4.3 逻辑赋值语句与逻辑数据的输入输出	53
4.3.1 逻辑赋值语句	53
4.3.2 逻辑数据的输入输出	54
4.4 用块 IF 实现选择结构	55
4.4.1 构成块 IF 的基本语句	55
4.4.2 块 IF 的基本结构	55
4.5 块 IF 的嵌套	59
4.6 GOTO 语句和逻辑 IF 语句	61
4.6.1 无条件转移语句——GOTO 语句	61
4.6.2 逻辑 IF 语句	62
4.7 程序举例	63
习题四	70
第五章 循环结构及数组操作	74
5.1 循环的定义和类型	74
5.2 用 GOTO 语句实现循环	74
5.3 DO 循环和 CONTINUE 语句	75
5.3.1 DO 语句的一般形式及循环次数的计算	75
5.3.2 DO 循环的结构及执行过程	76
5.3.3 对 DO 循环的有关规定	77
5.3.4 CONTINUE 语句	80
5.3.5 DO 循环语句举例	81
5.3.6 DO 循环的嵌套(多重循环)	82
5.3.7 DO 循环程序举例	84
5.4 用 DO WHILE 语句实现当型循环	89
5.4.1 用 DO WHILE 语句实现当型循环	89
5.4.2 用 GOTO 语句、块 IF、READ 语句实现当型循环举例	90
5.5 用 DO UNTIL 语句实现直到型循环	91
5.5.1 用 DO UNTIL 语句实现直到型循环	91
5.5.2 用逻辑 IF 语句内嵌 GOTO 语句实现直到型循环举例	92
5.6 几种循环形式的关系与比较	93
5.7 数组常见操作(数组的输入/输出)	95
5.7.1 用 DATA 语句给数组赋初值	95
5.7.2 用数组名输入/输出	96
5.7.3 按数组元素名输入/输出	97
5.7.4 用 DO 语句控制输入/输出	98
5.7.5 隐含输入/输出表	99

5.8 DO 循环与数组应用举例	100
习题五	106
第六章 FORTRAN 数据结构	109
6.1 程序中的数据结构	109
6.2 双精度型数据	110
6.2.1 双精度型常数	110
6.2.2 双精度型变量和数组	110
6.2.3 双精度型表达式和赋值语句	111
6.2.4 双精度型数据的输入/输出	111
6.3 复型数据	113
6.3.1 复型常数	113
6.3.2 复型变量和数组	113
6.3.3 复型表达式及复型运算	114
6.3.4 复型数据的输入/输出	115
6.3.5 举例	115
6.3.6 数值型数据小结	118
6.4 字符型数据	119
6.4.1 字符型常数	119
6.4.2 字符型变量和数组	120
6.4.3 子字符串	121
6.4.4 字符表达式及字符赋值语句	122
6.4.5 字符型关系表达式	123
6.4.6 字符型函数	124
6.4.7 字符型数据的输入/输出	125
6.4.8 举例	126
习题六	133
第七章 常用基本算法及程序举例	135
7.1 穷举法	135
7.2 递推和迭代算法	137
7.2.1 递推算法	137
7.2.2 迭代算法	138
7.2.3 用迭代法解一元方程	139
7.3 一元方程求根的其它算法	143
7.3.1 用二分法求方程的根	143
7.3.2 用弦截法求方程的根	146
7.4 排序算法	147
7.4.1 比较互换法	147
7.4.2 选择法排序	149
7.4.3 冒泡法排序算法	151
7.4.4 希尔法排序	153

7.5 检索	155
7.5.1 线性检索	155
7.5.2 对分检索	156
7.6 数值积分	158
7.6.1 矩形法	159
7.6.2 梯形法	160
7.6.3 辛普生法	161
7.6.4 变步长辛普生求定积分	163
7.7 高精度运算的程序设计	164
7.8 各进位制之间数据转换的程序设计	169
7.8.1 十进制与二、八进制间的转换	169
7.8.2 十进制与十六进制间的转换	170
习题七	171
第八章 过程	173
8.1 过程的分类	173
8.2 语句函数	174
8.2.1 语句函数的定义	174
8.2.2 语句函数的调用与执行	175
8.2.3 调用形式举例	175
8.3 函数子程序	177
8.3.1 函数子程序的引入	177
8.3.2 函数子程序的定义	178
8.3.3 函数子程序的调用	180
8.3.4 函数子程序调用时的哑实结合	181
8.3.5 函数子程序应用举例	185
8.4 外部语句(EXTERNAL)和内部语句(INTRINSIC)	187
8.4.1 EXTERNAL 语句(外部语句)	187
8.4.2 INTRINSIC 语句(内部语句)	190
8.5 子例行程序	192
8.5.1 子例行程序的引入	192
8.5.2 子例行程序的定义	193
8.5.3 子例行程序的调用	194
8.5.4 子例行程序应用举例	194
8.6 可调数组及其应用	196
8.6.1 可调数组的概念	196
8.6.2 使用可调数组时的注意事项	197
8.6.3 可调数组应用举例	197
8.7 综合应用举例	198
习题八	204
第九章 数据联系语句和数据块子程序	205

9.1 公用语句(COMMON 语句)	205
9.1.1 公用语句概述	205
9.1.2 公用(COMMON)语句的一般形式	205
9.1.3 无名公用区及其使用规则	206
9.1.4 有名公用区及其使用规则	209
9.1.5 举例	210
9.2 数据块子程序	211
9.3 等价语句(EQUIVALENCE 语句)	214
习题九	216
第十章 文件及其使用	218
10.1 引言	218
10.2 文件的基本概念	218
10.2.1 记录的格式与长度	219
10.2.2 文件的存取方式	219
10.2.3 外部文件	220
10.2.4 文件读写(输入/输出)过程概述	220
10.3 文件的打开与关闭	221
10.3.1 文件的打开	221
10.3.2 文件的关闭	223
10.4 文件读写语句	225
10.4.1 文件读写语句的一般形式及功能	225
10.4.2 有格式顺序文件的读写	226
10.4.3 有格式直接文件的读写	227
10.4.4 无格式文件	228
10.5 文件的定位与查询	230
10.5.1 文件定位	230
10.5.2 文件查询	232
10.6 文件使用举例	233
习题十	236
第十一章 MS—FORTRAN5.1 的安装和集成环境的使用	238
11.1 MS—FORTRAN5.1 系统的安装	238
11.2 集成开发环境 PWB 的使用	240
11.2.1 集成开发环境 PWB 的启动和退出	240
11.2.2 窗口菜单与命令	241
11.2.2.1 文件菜单命令(File)	241
11.2.2.2 编辑命令(Edit)	244
11.2.2.3 查看窗口命令(View)	246
11.2.2.4 查找命令(Search)	246
11.2.2.5 生成菜单命令(MAKE)	250
11.2.2.6 运行菜单命令(RUN)	252

11.2.2.7 选择菜单(OPTION)	254
11.2.2.8 浏览菜单(BROWSE)	256
11.2.2.9 帮助菜单命令(HELP)	259
11.3 集成环境下的程序开发	260
11.3.1 编辑源程序	260
11.3.2 源程序的编译、连接和运行	261
11.3.3 多模块程序的调试	262
第十二章 MS—FORTRAN5.1 简介	263
12.1 FORTRAN 标识符的作用范围	263
12.2 数据类型	263
12.3 FORTRAN 语句	264
12.4 子程序和函数	270
12.5 MS—FORTRAN5.1 的元命令语言	272
12.6 结构、联合和内存映像	279
12.6.1. 结构	279
12.6.2 内存映像和联合	280
12.7 MS—FORTRAN5.1 的图形功能	281
12.7.1 MS—FORTRAN5.1 的图形环境	281
12.7.1.1 MS—FORTRAN5.1 的图形功能	281
12.7.1.2 图形坐标系统及颜色设置	281
12.7.1.3 图形模式的设置和取消	282
12.7.2 基本绘图过程	283
12.7.2.1 图像要素的设置	283
12.7.2.2 屏幕绘图命令	288
12.7.2.3 图形中的文本输出	296
12.7.3 绘图程序举例	301
附录	313
参考文献	315

第一章 程序设计及算法的基本概念

计算机的应用虽已深入到各行各业,但是要用计算机处理问题,就必须写出使计算机按人们意愿进行工作的计算机程序。所以每一个学习计算机知识的人都应当学习如何进行程序设计。

为了有效地进行程序设计,至少要具有两方面的知识,即:①掌握解决问题的方法和步骤——算法。②掌握一门计算机语言的语法规则,编写出计算机程序。一般来说程序设计过程中的基本工作是:定义问题、算法设计和数据结构设计、流程图(或N-S图等)设计、用选定的语言编制程序、调试程序、测试程序、编制文档资料、维护和再设计等。

计算机中目前常用的高级语言可划分为两大类:面向过程的和面向对象的。面向过程的常用语言(Procedure-Oriented Language)有:各种版本的BASIC、FORTRAN、C、PASCAL、LIST、COBOL等。面向对象(Object-Oriented Programming)的高级语言有:Visual BASIC、Visual C++、FoxPro、PROLOG(人工智能语言)、Java(网络软件开发语言)等。80年代中期C++语言问世后,面向对象程序设计开始广泛流行,现正在走向普及。

1.1 算法及算法的表示方法

1.1.1 算法概念

日常生活中做任何事情都有一定的步骤。所谓算法,就是为解决某一个问题而采取的方法和步骤。当然对同一个问题,可以有不同的解题方法和步骤,这样算法就有优劣之分。

一般应采用方法简单、运算步骤少的算法,除保证算法正确外,还要考虑算法的质量。

计算机算法分为数值运算算法和非数值运算算法两类。数值运算算法其目的是求数值解。例如,代数方程的数值解,常微分、偏微分方程的数值解算法等。对数值算法的研究比较深入和成熟。非数值算法多用于信息处理和人工智能等方面。例如,情报检索,排序算法等。

1.1.2 计算机算法应具备的特点

- 1) 有限性(有穷性)。一个算法应包括有限的操作步骤,应在有限次执行后完成。
- 2) 确定性。一个算法中的每一个步骤,必须有明确的定义。而不应含糊不清,模棱两可,算法的含义应当是唯一的。
- 3) 有零个或多个输入。所谓输入是指在执行算法时需要从外界取得必要信息。算法总是要施加到运算对象上,输入描述了运算对象的初始情况。
- 4) 可行性。算法中的每一步都是能够准确地执行,并能得到确定的结果。例如,在执行 $1/N$ 除法中,当 $N=0$ 时,计算机运算 $1/N$ 溢出,这表明算法无法执行。
- 5) 有一个或多个输出。算法的最终归宿就是要得到输出结果。没有输出结果的算法是没有意义的。

1.1.3 算法的表示方法

对一个算法的描述可以采用多种方式,主要有以下几种表示方法:

- ① 用自然语言;
- ② 用框图或流程图;
- ③ 用计算机语言或伪代码。

用计算机语言描述的算法就是计算机程序。

(1) 用类似于数学语言的自然语言

自然语言就是人们日常使用的语言,可以是汉语、英语或其它文字语言。用自然语言描述一个算法通俗易懂,但文字冗长,易出现“多义性”,含意不太严格,除一些很简单的问题外,一般不用自然语言来描述算法。

(2) 用框图或流程图

在用图示的方法来描述算法的方法中,有传统流程图, N-S 结构化流程图和 PAD 图等。

1) 用传统流程图表示算法。

一个算法可以用流程图(框图)来表示。传统的流程图是用一些图框来表示一些特定含意的操作。用图形表示算法,一目了然,易于理解。

美国国家标准化协会(ANSI)规定了一些常用的流程图符号,如图 1-1 所示。现已被世界各国所普遍采用。

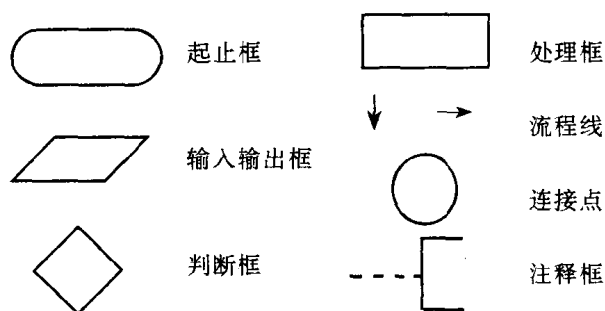


图 1-1 流程图使用符号

图 1-1 中的处理框,表示一般的叙述事项;菱形框的作用是对一个给定的条件进行判断,它有一个入口两个出口,根据判断给定条件是否成立,而选定两出口之一向下运行;连接点(圆圈)是将两幅流程图连起来的接点。采用连接点,可以避免流程线过长或交叉,避免流程图的图幅过大。

2) 用 ISO 流程图表示算法。

随着结构化程序设计方法的出现,使流程图也暴露出一些不足之处,比如它没有表示循环结构的图框,使用带箭头的流程线,允许流程的随意 GOTO 转移等。针对上述情况,国际标准化组织(ISO)在 ANSI 提出的流程图符号的基础上,提出了 ISO 流程图符号。除包括 ANSI 的流程图符号外又增了表示循环的符号;表示多出口的判断符号;规定当流程由上到下和由左到右时,流程线可以不带箭头,反之,若流程从下向上或从右向左时,流程线要加箭头。ISO 补充 ANSI 的流程图符号见图 1-2。

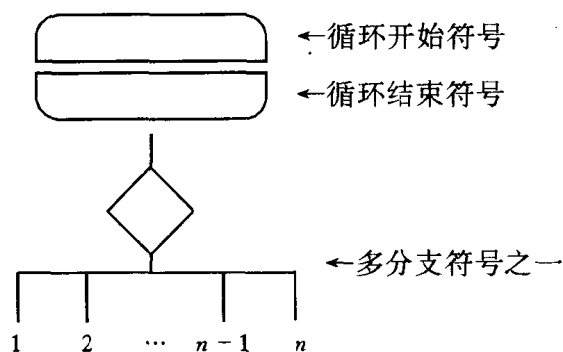


图 1-2 ISO 部分流程图符号

我国已接受 ISO 标准提出的程序流程图符号,并在此基础上制订了国家标准。国标中除保留了 ISO 的图框符号外又增加了少量框图符号。

为了提高算法的质量,人们设想,如果规定出几种基本结构,由这些基本结构由上而下按一定规律组成一个算法结构。这样的算法结构,结构清晰,易被人理解,从而能提高算法的质量。

1.2 算法的三种基本结构

1966 年,Bohra 和 Jacopini 提出了三种基本结构,即顺序结构、选择结构和循环结构。人们可以用这三种基本结构作为表示一个良好算法的算法结构。

1.2.1 顺序结构

顺序结构如图 1-3 所示。虚线框的内容是一个顺序结构,其中图框 A 和图框 B 是顺序执行的处理框。顺序结构是最简单的一种结构,其入口为 a 点,出口为 b 点。

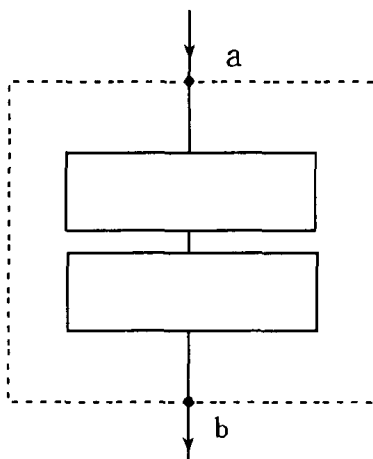


图 1-3 顺序结构框图

1.2.2 选择结构

选择结构如图 1-4 所示。此种结构中必定包含一个判断框。根据给定的条件 P 是否成

立而选择执行 A 框或 B 框。当条件成立时, 执行 A, 否则执行 B。根据需要, A 框或 B 框可以是空框, 即不执行任何操作。但判断框中的两个分支, 执行完 A 或 B 之后都是经过唯一的出口 b 点。

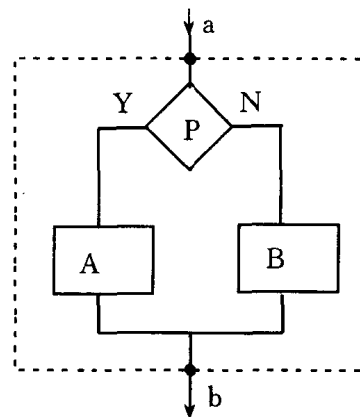


图 1-4 选择结构框图

1.2.3 循环结构

循环结构又称重复结构。分为当型循环结构和直到型循环结构两类。二者都是重复执行某些操作。

1.2.3.1 当型(WHILE 型)循环结构

图 1-5(a), (b) 分别是 ANSI、ISO 规定的当型循环结构框图。其功能是, 当给定条件 P_1 成立时, 执行 A 框操作, 执行完 A 框后, 再判断 P_1 条件是否成立; 若成立再次执行 A 框, 如此重复执行 A 框, 直到判断 P_1 条件不成立为止。此时不再执行 A 框, 结束循环, 从出口点 b 脱离循环结构。

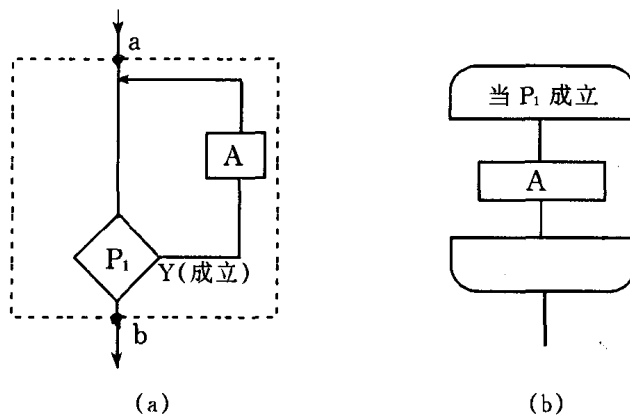


图 1-5 当型循环结构框图

1.2.3.2 直到型(UNTIL 型)循环结构

图 1-6(a), (b) 分别是 ANSI、ISO 规定的直到型循环结构框图。其功能是, 先执行 A 框操作, 然后判断给定条件 P_2 是否成立; 如果不成立, 则再执行 A 框, 然后再对 P_2 条件判断, 如此反复直到给定的条件 P_2 成立为止。此时不再执行 A 框, 结束循环从出口点 b 脱离本循环结构。

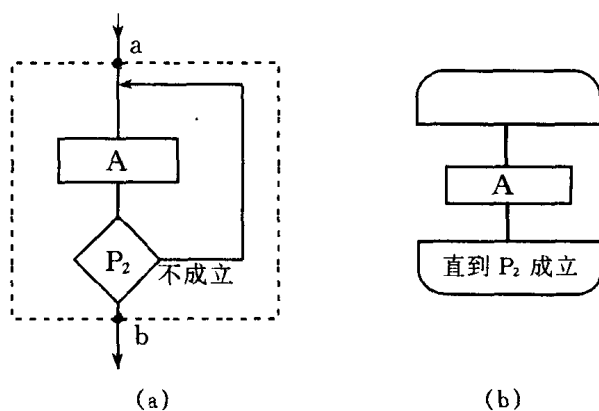


图 1-6 直到型循环结构框图

上面介绍的当型循环和直到型循环是可以互相转换的，也就是说对同一个问题既可以用当型循环处理，也可以用直到型循环处理。当型循环是先判断条件，条件成立执行 A 框；直到型循环是先执行 A 框，后判断条件，条件不成立时执行 A 框。对同一个问题判断框中的条件 P_1 与 P_2 是为互逆条件，例如 P_1 条件是“ $X < 5$ ”时， P_2 条件应是“ $X \geq 5$ ”。对于一个具体问题，究竟用当型循环还是用直到型循环呢？用户可以自主决定。但由于当型循环执行 A 框（循环体）可以是 0 次到 N 次，而直到型循环则至少执行一次循环体（A 框）。所以，当事先不能确定是否至少执行一次循环体的情况下，一般用当型循环。在第五章中配合举例将更详细的介绍实现两种循环结构的各种方法。

以上表示算法的三种基本结构有以下共同特点：

- 1) 只有一个入口和一个出口。在选择结构菱形框中虽有两个分支，但选择结构只有一个出口，二者不要混淆。
- 2) 结构内的每一个框都有机会被执行。也就是说对每一个框都有一条从入口到出口的畅通路程。
- 3) 基本结构内不存在“死循环”（无终止的重复执行）。

实际上具有上述三个特点的算法结构都可以作为基本结构。也就是说，用基本结构的顺序组合可以表示任何复杂的算法结构。比如，一个顺序结构中，也可以包含有选择结构或者循环结构，用户可以自己定义基本结构。由基本结构所构成的算法属于结构化的算法。根据结构化的算法才能编写出结构化的程序。

1.2.4 用 N-S 结构化流程图表示算法

1973 年美国学者 I. Nassi 和 B. Shneiderman 提出在流程图中去掉流程线，全部算法写在一个矩形框内，在该框内还可以包含其它的基本框。这种流程图被人们称为 N-S 结构化流程图。

(1) N-S 流程图中使用的流程图符号

- 1) 顺序结构。图 1-7 所示为由 A、B 两个框组成的基本顺序结构。
- 2) 选择结构。图 1-8 所示为基本选择结构。与图 1-4 相对应。当条件 P 成立时执行 A，否则执行 B。同样 A 块或者 B 块可能是一个空框，无任何操作。
- 3) 循环结构。
 - a) 当型循环结构。图 1-9 为当型循环结构，当条件 P_1 成立时反复执行 A 框操作，直