



21世纪高等学校计算机科学与技术规划教材

程序设计基础

CHENGXU SHEJI
JICHU

陈一明 主 编



北京邮电大学出版社
www.buptpress.com



21 世纪高等学校计算机科学与技术规划教材

程序设计基础

主 编 陈一明

副主编 吴良海 赖锦辉 左利云
苏海藏 陈晓龙

北京邮电大学出版社

• 北京 •

内 容 简 介

本教材以《计算机基础课程教学基本要求(2009版)》中“程序设计基础”课程教学要求为依据,加强以应用能力培养为核心的实践教学,重在编程的思想与方法。本教材重点介绍了算法、程序等基本概念,算法与程序设计的基本方法,程序控制结构等编程基础知识,以问题解决为案例,详细介绍 Visual Basic、Visual FoxPro、Visual C++ 三种程序设计语言在实际中的应用。本教材有《程序设计基础实验指导》作为配套教材。

本教材可以作为普通高校非计算机专业计算机基础程序设计类课程的教材,也可作为学习程序设计的参考材料。

图书在版编目(CIP)数据

程序设计基础/陈一明主编. -- 北京:北京邮电大学出版社,2010.8

ISBN 978-7-5635-2400-6

I. ①程… II. ①陈… III. ①程序设计—高等学校—教材 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2010)第 162150 号

书 名	程序设计基础
主 编	陈一明
责任编辑	李学锋
出版发行	北京邮电大学出版社
社 址	北京市海淀区西土城路 10 号(100876)
电话传真	010-82333010 62282185(发行部) 010-82333009 62283578(传真)
电子信箱	ctrd@buptpress.com
经 销	各地新华书店
印 刷	北京市梦宇印务有限公司
开 本	787 mm×1 092 mm 1/16
印 张	19
字 数	436 千字
版 次	2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷

ISBN 978-7-5635-2400-6

定价: 35.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

前 言

“程序设计基础”是高等学校重要的计算机基础课程,它以编程语言为平台,介绍程序设计的思想和方法。通过该课程的学习,学生不仅要掌握高等程序设计语言的知识,更重要的是在实践中逐步掌握程序设计的思想和方法,培养解决问题的思维习惯与能力。

教材以《计算机基础课程教学基本要求(2009 版)》中“程序设计基础”课程教学要求为依据,加强以计算思维及应用能力培养为核心的实践教学,重在“寻找”解决问题的思想与方法;强调“不是让学生学到什么,而是让其能做什么,能解决什么问题”。教材结构围绕“边讲、边练、边做”教学模式来进行设计,理论的内容通过案例展开。摒弃以程序设计语言为中心展开教学的传统做法,转而以掌握程序设计思想理念、解决问题方法为根本的素质培养。结合高校各专业需要,整合各种程序设计语言知识,融会贯通。突出以“任务、问题”为核心,以“提出问题→解决问题→总结提高→拓展创新”的递进模式组织学习内容。

教材分主教材——《程序设计基础》与配套辅助教材——《程序设计基础实验指导》。主教材以介绍理论知识与程序设计方法为主,不同专业可按需要选择学习某种语言模块。实验辅助教材则着重以案例的解决方法为主,指导学生实验操作。两教材内容相辅相成。

主教材内容分为三大部分:基础篇、案例篇、应用篇。

第一部分从问题解决案例入手,首先介绍算法与程序两个最基础概念,然后介绍算法与程序设计的基本方法与过程,并且把程序设计中相通的基础知识,如数据类型、常量变量、表达式、数组与函数、程序的组织结构、程序的功能结构等抽取出来,进行统一的介绍,有助于对知识的全面了解,培养计算机思维能力。同时,在这部分提供大量各种类型的习题,以加强对基础知识的细化处理,弥补教材中没有过多理论知识细节描述的情况。

第二部分则以解决问题的案例形式分别介绍 Visual Basic、Visual Forpro、Visual C++ 三种最常用的高级程序设计语言的基本应用。

第三部分则介绍这三种语言解决不同类型问题的实际应用。我们在第二、三部分力求以问题的解决为主线,重在介绍解决实际问题的思想与方法,不拘泥于某种语言的使用细节。在内容的选择上能结合各种语言特点与不同专业的知识特点与要求进行,但也注意教材体系的统一要求。在内容处理上突显“面向学生、面向应用、突出实践、重在思想”的特色。当学生在具备解决问题的初步能力时,引导学生逐步形成严谨、科学、周密的思维方式,养成良好的编程习惯,从而进一步增加知识的深度与广度。

教材由“构建高校计算机公共课程多元教学平台”项目组编写,其中陈一明任主编,吴良海、赖锦辉、左利云、苏海英、陈晓龙任主教材副主编,吴宪君任实验教材副主编。陈一明

负责全书编写思想、模式框架、章节大纲及各部分的初稿、修改、定稿等各阶段工作；吴良海负责主教材第一部分，赖锦辉、左利云、苏海英分别负责主教材第二、三部分的 VB、VC++、VFP 内容，陈晓龙负责主教材内容校对及示例程序的验证等，吴宪君负责实验教材内容的撰写。同时，包括何海燕、李金瑞在内的项目组成员都参加了教材的交叉修改工作，肖健老师设计与提供了一些案例。

教材的编写希望能达到内容整合、淡化语言、侧重思想与能力培养。由于时间与能力有限，还存在很多不理想的地方，敬请大家多提意见。

编 者

目 录

第一篇 基础篇

第 1 章 算法与算法设计	2
1.1 算法的概念	2
1.2 算法的组成要素与控制结构	3
1.3 算法的表示	5
1.4 算法的特征与算法的评价标准	8
1.5 算法的设计方法	9
1.6 常用算法简介	10
1.6.1 常用算法	10
1.6.2 算法举例	11
总结与提高	13
思考与练习	13
第 2 章 程序与程序设计	15
2.1 程序的概念	15
2.2 程序设计与程序设计基本流程	16
2.3 程序的功能结构	17
2.4 程序设计的演变	17
2.4.1 早期程序设计	17
2.4.2 结构化程序与结构化程序设计	18
2.4.3 面向对象程序设计	19
2.4.4 可视化编程	20
2.5 源程序与目标程序	21
总结与提高	23
思考与练习	23
第 3 章 程序设计语言	25
3.1 程序设计语言简介	25
3.1.1 程序设计语言的发展	25
3.1.2 高级程序设计语言的程序示例	27
3.2 程序设计语言特点与使用简介	27
3.2.1 常用的程序设计语言特点	27

3.2.2 常用的程序设计语言使用简介·····	29
总结与提高·····	35
思考与练习·····	36
第4章 编程基础 ·····	37
4.1 高级语言编程基础·····	37
4.1.1 数据与数据类型·····	37
4.1.2 常量与变量·····	39
4.1.3 表达式·····	42
4.1.4 函数与数组·····	44
4.2 数据输入与输出·····	45
4.2.1 数据输入·····	45
4.2.2 数据输出·····	46
4.3 程序基本控制结构·····	47
4.3.1 选择结构·····	48
4.3.2 循环结构·····	49
4.4 简单编程示例·····	51
总结与提高·····	56
思考与练习·····	56

第二篇 案例篇

第5章 Visual Basic 程序设计 ·····	68
5.1 Visual Basic 编程概述·····	68
5.1.1 引例·····	68
5.1.2 OOP 思想·····	69
5.1.3 VB 程序设计思想·····	72
总结与提高·····	77
思考与练习·····	77
5.2 VB 的常用标准控件·····	82
5.2.1 控件编程实例·····	82
5.2.2 键盘和鼠标事件·····	88
5.2.3 其他控件示例·····	90
总结与提高·····	92
思考与练习·····	92
5.3 过程与过程应用·····	96
5.3.1 过程·····	96
5.3.2 参数传递·····	98
5.3.3 过程应用·····	100
总结与提高·····	104

思考与练习	104
5.4 菜单、对话框和多重窗体设计	109
5.4.1 菜单和对话框程序设计	109
5.4.2 建立多重窗体应用程序	114
总结与提高	116
思考与练习	116
5.5 数据文件	120
总结与提高	124
思考与练习	124
第 6 章 Visual C++ 程序设计	127
6.1 Visual C++ 编程概述	127
6.1.1 简单的 VC++ 程序介绍	127
6.1.2 VC++ 编译预处理	129
总结与提高	129
思考与练习	130
6.2 函数与数组	134
6.2.1 函数的定义及参数传递	134
6.2.2 递归简介	138
6.2.3 数组的基本应用	138
6.2.4 多维数组	140
6.2.5 数组与函数	141
总结与提高	141
思考与练习	142
6.3 指针	146
6.3.1 指针与地址操作符	146
6.3.2 指针变量的声明	147
6.3.3 指针初始化	149
6.3.4 指针运算	149
6.3.5 指针与数组	150
6.3.6 指针的指针	151
6.3.7 void 指针	152
6.3.8 函数指针	152
总结与提高	153
思考与练习	154
6.4 类	158
6.4.1 构造函数和析构函数	161
6.4.2 构造函数重载	163
6.4.3 类的指针	165
6.4.4 struct 和 union 定义的类	166

总结与提高	166
思考与练习	166
6.5 文件的输入输出	170
6.5.1 打开文件	170
6.5.2 关闭文件	171
6.5.3 文本文件	171
6.5.4 状态标志符的验证	173
6.5.5 获得和设置流指针	173
6.5.6 二进制文件	174
6.5.7 缓存和同步	175
总结与提高	175
思考与练习	176
第7章 Visual FoxPro 程序设计	178
7.1 Visual FoxPro 数据库基础	178
7.1.1 Visual FoxPro 工作目录的设置	179
7.1.2 项目管理器	180
7.1.3 数据库管理	181
7.1.4 表的基本操作与维护	182
7.1.5 数据排序、索引、统计和多表操作	189
7.1.6 SQL 查询语言	196
总结与提高	198
思考与练习	198
7.2 Visual FoxPro 编程概述	201
7.2.1 Visual FoxPro 程序文件的建立	202
7.2.2 结构化程序设计	202
总结与提高	206
思考与练习	207
7.3 表单设计与常用控件使用	210
7.3.1 表单设计	210
7.3.2 常用控件使用	215
总结与提高	222
思考与练习	222
7.4 报表设计	224
7.4.1 使用报表向导创建报表	224
7.4.2 用“快速报表”创建报表	227
7.4.3 “报表设计器”创建报表	228
思考与练习	231
7.5 菜单设计	231
7.5.1 Visual FoxPro 菜单结构	232

7.5.2 下拉式菜单设计	232
总结与提高	235
思考与练习	235

第三篇 应用篇

第 8 章 Visual Basic 程序设计实践	238
8.1 简易计算器	238
8.2 打字练习程序	240
8.3 登录界面	243
8.4 学生信息管理系统	247
总结与提高	250
思考与练习	250
第 9 章 Visual C++ 程序设计实践	251
9.1 制作计算器	251
9.2 屏幕保护程序	252
9.3 利用 MSComm 控件串口传输数据	256
9.4 ADO 的数据库编程	257
总结与提高	259
第 10 章 Visual Foxpro 系统开发实例	260
10.1 系统开发的一般过程	260
10.2 超市销售管理系统	260
10.2.1 系统规划方案	260
10.2.2 系统的逻辑结构	263
10.2.3 系统详细设计	263
10.3 学生信息管理系统	273
10.3.1 数据库设计	273
10.3.2 系统的逻辑结构	273
10.3.3 系统详细设计	274
总结与提高	279
附录 A 关键字	280
附录 B 常用函数	282
附录 C 程序测试方法	290
参考文献	294

第一篇 基础篇

余英时文集 卷一

第 1 章 算法与算法设计

教学目标

1. 理解算法的相关概念;
2. 掌握算法的表示方法;
3. 掌握分析问题、设计解决问题算法的方法;
4. 了解一些常用算法。

1.1 算法的概念

〔例 1-1〕 有 A、B 两个杯子,分别装有不同的液体,现要求把这两个杯子中的液体交换放置,如何操作?

■ 步骤如下

- 第一步,准备第三个杯子 C;
第二步,把杯子 A 的液体倒到杯子 C 中, $A \rightarrow C$;
第三步,把杯子 B 中液体倒到杯子 A 中, $B \rightarrow A$;
第四步,把杯子 C 中液体倒到杯子 B 中, $C \rightarrow B$;
第五步,操作完成。

从上例可以看到,解决问题的过程都是由一定的规则、步骤组成的,这种规则实际上就是算法。

算法(Algorithm)是一系列解决问题的清晰指令,也就是说,能够对一定规范的输入,在有限时间内获得所要求的输出。如果一个算法有缺陷,或不适合于某个问题,将不能正确解决这个问题。

算法通常是指按照一定规则解决某一类问题的明确和有限的步骤描述。现在,使用计算机程序设计语言,可以把算法编写成计算机程序,让计算机执行程序来解决问题。

〔例 1-2〕 计算函数 $M(x)$ 的值。函数 $M(x)$ 为:

$$M(x) = \begin{cases} bx + a^2 & x \leq a \\ a(c - x) + c^2 & x > a \end{cases}$$

其中, a, b, c 为常数。

■ 问题分析

这是一个数值运算问题,根据 x 的取值决定采用哪一个表达式计算 M 的值。

■ 算法如下

- ① 将 x 的值输入到计算机;
- ② 判断 $x \leq a$, 如果条件成立, 执行第③步, 否则执行第④步;
- ③ 按表达式 $bx + a^2$ 计算出结果存放到 M 中, 然后执行第⑤步;
- ④ 按表达式 $a(c - x) + c^2$ 计算出结果存放到 M 中, 然后执行第⑤步;
- ⑤ 输出 M 的值;
- ⑥ 算法结束。

[例 1-3] 给定两个正整数 m 和 n ($m \geq n$), 求它们的最大公约数。

■ 问题分析

这也是一个数值运算问题(我国数学家秦九韶在《数书九章》一书中曾记载了这个算法), 一般用辗转相除法(也称欧几里德算法)求解。

例如: 设 $m = 35, n = 15$, 余数用 r 表示。它们的最大公约数的求法如下:

① 35 除以 15 商为 2, 余数 r 为 5, 然后将分母辗转为分子 $n \rightarrow m$, 余数辗转为分母 $r \rightarrow n$, 继续相除;

② 15 除以 5 商为 3, 余数为 0, 当余数为零时, 所得 n 即为两数的最大公约数;

③ 所以 35 和 15 两数的最大公约数为 5。

■ 算法如下

- ① 将两个正整数存放到变量 m 和 n 中;
- ② 求余数: 计算 m 除以 n , 将所得余数存放到变量 r 中;
- ③ 判断余数是否为 0: 若余数为 0 则执行第⑤步, 否则执行第④步;
- ④ 反复更新被除数和除数: 将 n 的值存放到 m 中, 将 r 的值存放到 n 中, 并转向第②步继续循环执行;
- ⑤ 输出 n 的当前值;
- ⑥ 算法结束。

由上述例子可以看出, 一个算法由若干操作步骤构成, 并且这些操作是按一定的控制结构所规定的次序执行。如[例 1-1]中的 4 个操作步骤是按先后顺序逐个执行的, 称之为顺序结构。[例 1-2]则是按设计好的操作步骤有选择性地执行, 而不是所有步骤都执行, 如第 3 步和第 4 步的两个操作就不能同时被执行, 它们需要根据条件决定执行哪个操作, 这种结构称之为选择结构。在[例 1-3]中不仅包含了判断, 而且需要重复执行。如第二步到第五步之间的步骤就需要根据条件判断是否重复执行, 并且一直延续到条件“余数为 0”为止, 这种具有重复执行功能的结构称之为循环结构。

1.2 算法的组成要素与控制结构

一个算法由一些操作组成, 而这些操作则按控制结构所规定的次序执行, 也就是说算法由操作与控制结构两个要素组成。计算机的基本操作包括以下四个方面:

- (1) 算术运算。加、减、乘、除;
- (2) 数据比较。大于、小于、等于、不等于、大于等于、小于等于;

(3) 逻辑运算。与、或、非；

(4) 数据传送。输入、输出、赋值。

算法的实现过程是由一系列操作组成的,这些操作之间的执行次序就是程序的控制结构。算法的基本控制结构通常包括顺序结构、选择结构和循环结构。

1996 年,计算机科学家 Bohm 和 Jacopini 证明了这样的事实:任何简单或复杂的算法都可以由顺序结构、选择结构和循环结构这三种基本结构组合而成。所以,这三种结构就被称为程序设计的三种基本结构。

1. 顺序结构

顺序结构表示程序中的每一个操作都是按从上到下的先后次序执行的。整个顺序结构只有一个入口点和一个出口点。这种结构的特点是:程序从入口点开始,按顺序执行所有操作,直到出口点处,所以称为顺序结构。事实上,不论程序中包含了什么样的结构,而程序的总流程都是顺序结构。

2. 选择结构

选择结构表示程序的处理步骤出现了分支,它需要根据某一特定的条件选择其中的一个分支执行。选择结构有单选择、双选择和多选择三种形式。

双选择是典型的选择结构形式,在结构的入口点处是一个判断操作,表示程序流程出现了两个可供选择的分支,如果条件满足执行其中一个分支,否则执行另一个分支。值得注意的是,在这两个分支中只能选择一个且必须选择一个执行,但不论选择了哪一个分支执行,最后流程都一定到达结构的出口点处。

当两个分支中的任意一个处理为空时,说明结构中只有一个可供选择的分支,如果条件满足执行分支处理,否则顺序向下到流程出口处。也就是说,当条件不满足时,什么也没执行,所以称为单选择。

多选择是指程序流程中遇到多个分支,程序执行方向将根据条件确定。如果满足条件 1 则执行分支 1 处理,如果满足条件 n 则执行分支 n 处理。总之要根据判断条件选择多个分支的其中之一执行。不论选择了哪一个分支,最后流程要到达同一个出口处。如果所有分支的条件都不满足,则直接到达出口处。

3. 循环结构

循环结构表示程序重复执行某一组操作,直到某条件为假(或为真)时才可终止。

应用循环最主要的是:什么情况下执行循环?哪些操作需要循环执行?

循环结构也只有一个入口点和一个出口点,循环终止是指流程执行到了循环的出口点。

循环结构重复执行的操作称为循环体,是指从循环入口点到循环出口点之间的处理步骤,这就是需要循环执行的部分。而什么情况下执行循环则要根据条件判断。循环体可以是一个或多个操作,也可以是一个完整的结构或一个过程。

循环结构的基本形式有两种:当型循环和直到型循环。

(1) 当型循环。表示先判断条件,当给定的条件满足时执行循环体,并且在循环终端处流程自动返回到循环入口;如果条件不满足,则退出循环体直接到达流程出口处。因为是“当条件满足时执行循环”,即先判断后执行,所以称为当型循环。

(2) 直到型循环。表示从结构入口处直接执行循环体,在循环终端处判断条件,如果条件不满足,返回入口处继续执行循环体,直到条件为真时再退出循环到达流程出口处。因为是

“直到条件为真时终止”，即先执行后判断，所以称为直到型循环。

从三种基本控制结构可以看到，程序中的任意基本结构都有唯一入口和唯一出口，并且程序不会出现死循环。

1.3 算法的表示

描述算法通常有自然语言、流程图、N-S图、伪代码、程序设计语言等。如前面几个例子就是用自然语言来表示算法。

1. 自然语言和伪代码表示法

自然语言：就是指人们日常使用的语言符号，可以是汉语、英语或其他语言。

伪代码：是用介于自然语言和计算机语言之间的文字和符号来描述算法。

如[例 1-1]可用如下的伪代码表示：

Begin(算法开始)

输入 A,B

A→C

B→A

C→B

Print A,B

End (算法结束)

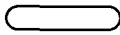
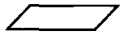
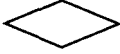
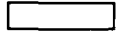
2. 流程图表示算法

流程图是一种用规定的图框、流程线及文字说明来准确、直观地表示算法的图形。一个流程图包括以下几部分：

- ① 表示相应操作的图框；
- ② 带箭头的流程线；
- ③ 图框中必要的文字和符号说明。

传统流程图中的基本符号如表 1-1 所示。

表 1-1 传统流程图中的基本符号

流程图符号	名 称	含 义
	起止框	表示一个算法的起始和结束，是任何流程图不可少的
	输入、输出框	表示一个算法输入和输出的信息，可用在算法中任何需要输入、输出的位置
	判断框	判断某一条件是否成立，成立时在出口处标明“是”或“Y”；不成立时标明“否”或“N”
	处理框	赋值、计算，算法中处理数据需要的算式、公式等分别写在不同的用以处理数据的处理框内
	流程线	表示流程进行的方向

画算法流程图(亦称框图)的规则如下:

- 使用标准的图形符号;
- 框图一般按从上到下、从左到右的方向画;
- 除判断框外,大多数流程图符号只有一个进入点和一个退出点。判断框可具有超过一个退出点;
- 判断框分两大类,一类判断框“是”与“否”两分支的判断,而且有且仅有两个结果;另一类是多分支判断,有几种不同的结果;
- 在图形符号内描述的语言要非常简练清楚。

用流程图表示的三种结构如图 1-1、图 1-2、图 1-3 所示。

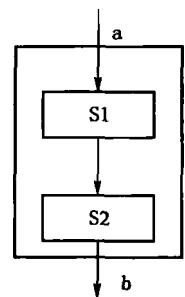


图 1-1 顺序结构

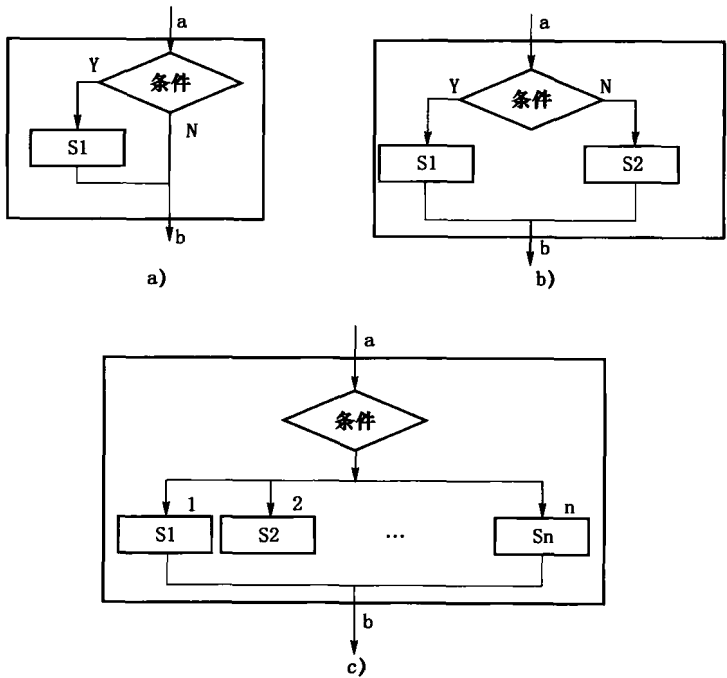


图 1-2 选择结构

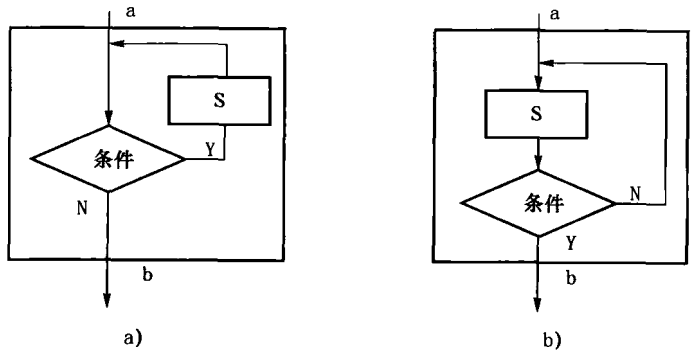


图 1-3 循环结构

3. N-S 流程图表示算法

N-S 流程图是结构化程序设计方法中用于表示算法的图形工具之一(对于非结构化的程序,用 N-S 流程图是无法表示的)。两位美国学者 Nassi 和 Shneiderman 于 1973 年提出了一种新的流程图形式,这就是 N-S 流程图,也称为 Nassi Shneiderman 图。

三种基本结构的 N-S 图表示如图 1-4 所示。

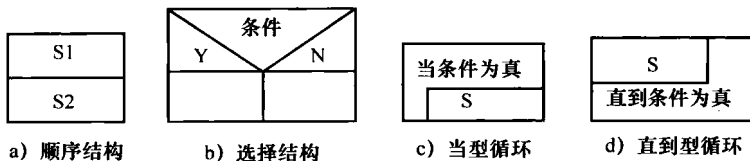


图 1-4 N-S 流程图

N-S 图的基本单元是矩形框,它只有一个入口和一个出口。长方形框内用不同形状的线来分割,可表示顺序结构、选择结构和循环结构。在 N-S 图中,完全去掉了带有方向的流程线,三种基本结构分别用三种矩形框表示,将这种矩形框进行组装就可表示全部算法。这种流程图从表达形式上就排除了随意使用控制转移对程序流程的影响,限制了不良程序结构的产生。

图 1-5 是用 N-S 流程图表示[例 1-2]中求函数值 m 的算法。

将[例 1-3]中的算法稍作调整,使流程图采用单选择结构形式,其中的条件改为 $r \neq 0$,这样就可以用直到型循环的 N-S 图表示这个算法。图 1-6 是表示[例 1-3]的 N-S 流程图。

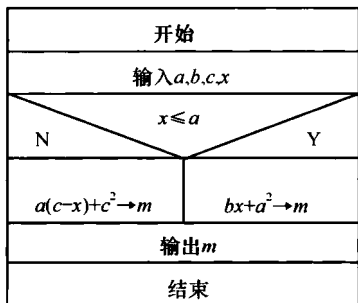


图 1-5 计算函数值的 N-S 图

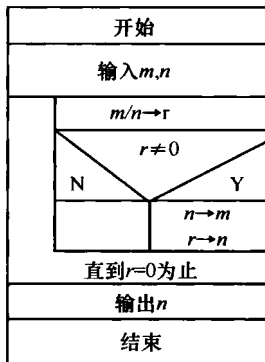


图 1-6 求最大公约数的 N-S 图

流程图、N-S 图是表示算法的图形工具。其中,流程图具有直观性强、便于阅读等特点,是本书主要介绍和使用的工具;N-S 图符合结构化程序设计要求,是软件工程中强调使用的图形工具,有兴趣的读者可参阅有关软件工程或软件开发技术等方面的著作。

[例 1-4] 请分别用流程图和 N-S 图描述“输入 10 个数,打印输出其中的最大数”的算法。

算法设计如图 1-7 所示。

4. 程序设计语言表示算法

前面介绍的算法描述方法是帮助程序开发人员阅读、编写程序而设计的一种辅助工具,