高等院校计算机教育系列教材

C/C++ 程序设计教程

（第2版）

张　莉　主编
陈　雷　雷宏洲　孙龙清　编著
孟超英　审校

- 通过实例应用解析语义规范
- 图表规范直观，便于快速理解掌握现代编程技能
- 内容简明扼要，突出知识要点
- 追求现代教育理念

赠送电子课件

清华大学出版社

高等院校计算机教育系列教材

C/C++程序设计教程

(第2版)

张　莉　主　编
陈　雷　雷宏洲　孙龙清　编　著
孟超英　审　校

清华大学出版社
北　京

内 容 简 介

C 程序设计编程灵活、风格优美，编程自由流畅，是现代程序设计的基础。C 语言是一种通用性语言，具有简洁、实用、代码质量高、可移植性强，既可用于应用软件编制，也可用于系统软件的开发。

随着计算机信息技术飞速发展与普及，C 语言自身也在不断地进化与发展，如面向对象技术的 C++、可视化编程的 Visual C++、网络编程的 Java 语言等，但都保持着基本的语法规则与编程风格。可以说掌握了 C 语言程序设计，也就掌握了现代编程的钥匙。

本教材第 1 版作为重点课程建设教学改革使用教材，经过两年多的教学实践，此次推出第 2 版，在内容细节上作了比较多的修改与补充，结构紧凑、编排规范合理、举例详实，符合现代教育的改革思路，可作为高等学校教学用书，也可作为编程爱好者的自学用书。

图书在版编目(CIP)数据

C/C++程序设计教程(第 2 版)/张莉主编；陈雷，雷宏洲，孙龙清编著；孟超英审校. —北京：清华大学出版社，2007.2
(高等院校计算机教育系列教材)
ISBN 978-7-302-14514-1

Ⅰ.C…　Ⅱ.①张…②陈…③雷…④孙…⑤孟…　Ⅲ.C 语言—程序设计—高等学校—教材　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2007)第 003527 号

责任编辑：彭　欣　宣　颖
封面设计：山鹰工作室
版式设计：杨玉兰
责任校对：马素伟
责任印制：何　芊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　　**邮购热线**：010-62786544
投稿咨询：010-62772015　　**客户服务**：010-62776969
印 刷 者：北京市清华园胶印厂
装 订 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：25　**字　数**：600 千字
版　　次：2007 年 2 月第 2 版　　**印　　次**：2007 年 2 月第1 次印刷
印　　数：1～4000
定　　价：33.00 元

前　言

随着现代编程技术的广泛应用与发展，C/C++程序设计更为普及。学习和掌握 C/C++语言成为现代软件开发人员的必备知识之一。

C 程序设计自 20 世纪 90 年代后期用于开发了 UNIX 操作系统，使 UNIX 成为世界上第一个易于移植的计算机操作系统以来，在软件开发领域得到广泛的应用与发展，为专业系统开发人员和专业程序设计人员所青睐，并随着现代计算机技术的发展，被不断完善，经久不衰。

使用 C 语言进行程序设计，编程结构简洁流畅、编译运行效率高。C 语言因其具有不同于其他语言的优点，具有强劲的生命力，因此在快速发展的计算机技术领域，C 程序设计才能够持久地存在并得以丰富和发展。至今学习 C 程序设计仍是掌握各种 C 语言集成开发环境的重要基础。

C 程序设计不仅有高级语言的语义特点，可以编写操作系统、编译程序、数据库软件系统等，还具有低级语言底层操作的能力，可以方便地进行字节运算、字位运算、寄存器操作、取地址、设计中断、访问设备端口等。C 语言是现代技术人员不可或缺的软件系统开发与制作工具。

随着计算机技术的飞速发展，C 语言本身也在不断地改进与发展着，这种发展仍在继续，如现在广为使用的面向对象技术的 C++、可视化编程的 Visual C++、网络编程语言等，但其基本语法规范与编程风格没变。可以说掌握了 C 语言，就掌握了深入系统学习和应用计算机的钥匙，用户再学习理解其他现代编程技术也就容易了许多。如面向对象程序设计的 C++语言，功能很强大。C++语言对 C 语言做了很多改进，如增加了运算符；增加了类型的安全性；允许函数重载等。而 C++语言与 C 语言的本质区别是增加了面向对象的技术，比如类、对象的封装性，基类、派生类的继承性，重载、动态联编多态性等。又比如像 Sun Microsystems 公司于 1995 年正式推出的新一代编程语言 Java，是一种面向对象、跨平台，适合于网络计算模式的分布式语言，而 Java 程序设计就保留了许多广为人知的与 C、C++同样的语法规范与结构风格。

在计算机应用领域，C 语言不仅仅是实现算法的程序设计语言环境，熟练掌握 C 语言程序设计与掌握许多其他高级程序设计相比，还有一大优点就是 C 语言所具有的强大功能，充分体现在 C 程序设计可以很好地发挥人的创新与创造思维，达到“能想到的就能实现”的境地。

当然，学习和掌握 C 语言并不像其他高级语言那样容易。特别是在掌握 C 语言程序设计精华或在深入系统应用等方面，还需要涉及一些计算机专业基础知识，如操作系统、数据结构、数据库系统、软件工程等。有些内容还需要大量的实践积累，才能提高综合应用能力，充分发挥自己的创新思维，让创造付诸于实际应用与开发。

本教程针对现代教学改革理念，在提高教学效率的同时，指导学生有效提高综合实践能力。参加编写、使用和修订的教学梯队均是多年来在软件开发与 C/C++程序设计教学方

面富有实践经验的一线教师，他们在使用第 1 版两年多教学实践的基础上，作了比较多的补充、修改与完善，使教材的教学适用性更强。

全书分为 14 章，其中第 1 章、第 2 章由张莉教授负责编写；第 3 章由孟超英教授和雷宏洲副教授负责编写；第 4 章、第 13 章由雷宏洲副教授负责编写；第 5 章、第 8 章由郑立华副教授负责编写；第 6 章、第 7 章由冀荣华博士负责编写；第 9 章、第 10 章由段清玲副教授负责编写；第 11 章由陈雷副教授负责编写；第 12 章由李琼飞副教授负责编写；第 14 章由孙龙清教授负责编写；全书由张莉教授统一修改定稿。

本教材在编写、使用和修订过程中，得到了北京大学谢柏青教授、中国人民大学杨小平教授等著名专家学者的具体指导，另外参加本书编写审校的教师还有张筠、杨丽丽、吕春利、马钦、李林、张兰、姜八月、侯凤玲、戴伟、李娟、李国成等年轻教师梯队，在此致以衷心谢意。

本教程作为教学改革和精品课程建设教学用书，配有网络辅助教学资源，在内容取材与编写方面力求要点突出、结构紧凑，易读易懂、实例详实，适合当前教改新要求，可以帮助读者尽快掌握 C/C++程序设计的实际应用技巧，系统全面地了解 C 程序设计的重点和要点，掌握实际编程及应用。

鉴于本教材源于多年教学改革实践，有些尝试在教学改革不断地摸索中还有许多探讨的空间，还可以进一步结合现代教育技术，在实际教学中提炼与完善。由于新时期、新技术、新的教学理论与要求，本教程第 2 版也难免有不足之处，诚望广大读者提出宝贵的意见，共同探索 IT 教育的新理念和新方法，以适应现代人材培养目标对信息技术教育的需要。

编　者

目　录

第 1 章　计算机程序设计与算法

学习计算机信息技术并不仅仅是学习操作和使用一般应用程序，计算机作为一种工具，熟练掌握计算机程序设计技术与算法，是开发利用信息技术最基本的技能之一。当前，以计算机技术、网络技术与通信技术为基础的信息技术得到普及与应用，推动了知识经济的发展，现代社会要跟上信息时代的步伐，使整个经济持续稳定地发展，就必须发展知识经济。知识经济也是信息经济，是以信息产业为标志、以知识为基础的经济，而发展知识经济的关键是提高现代人才的信息技术水平。熟练掌握一到两种程序设计语言以及现代编程设计方法则可以把我们带到一个全新的知识领域，使我们真正接触到更为广阔的信息科技世界。

本章主要内容：

- 程序设计语言
- 程序设计过程
- 程序设计算法
- 自然语言描述计算机算法
- 程序流程图描述计算机算法
- N-S 图描述计算机算法
- 用程序设计语言描述计算机算法

1.1　程 序 设 计

学习程序设计并不是简单地学习计算机语法规范或程序设计语言的本身，而是要学会怎么用计算机程序设计语言解决实际问题、提高工作效率和工作质量。计算机技术应用领域随学科交叉融合而日益广泛，计算机技术应用能力要求也日渐提高，学习掌握程序设计基本原理与技能是应用计算机解决行业领域实际问题的基本要求。

1.1.1　程序设计语言

多年以来，国内外教育界专家一直认为计算机教育必须学习程序设计语言及设计方法，特别是高等教育，要加强信息技术的教育，更应加强力度培养学生掌握程序设计语言及现代编程方法。不仅可以培养学生运用程序设计算法来解决实际问题的应用能力，提高专业技术拓展能力，也可以提高信息技术综合应用技能。如今我国许多中小学在具备了相应的师资与教学实验条件后，也相继开设了不同层次的计算机教育与编程语言课，主要以 BASIC

语言或 Visual Basic 语言为主，但由于地区差异很大，有些中学的计算机设备和实验条件相当好，还接入了互联网，实现了远程教学，而更多的边远地区的学生却很少触及现代信息设施，更不要说掌握应用技能了。我国高校普遍开设的计算机基础教育课程系列，即在相对保障的教学计划和教学实验条件下，使非计算机专业的学生系统有效地掌握现代信息技术的基础理论和应用技能，成为培养现代人才的重要环节，甚至造就出一大批现代 IT 精英，不仅提高了个人的就业实力，也提高了国际经济活动的行业竞争能力。目前，国家教育部仍在进一步加强高等教育中现代信息技术环节的教育。

实际上能实现各种算法、称之为语言的集成环境很多，可以说高校公共基础教育教什么样的语言就其基本教学目标而言并无多大差异，语言实现算法的本质结构都是相似的，如果有区别就是应用功能上的不同、应用平台上的不同和函数库的不同。若对语言进行分类，可以从许多方面来分：可以按对机器的依赖程度分类、按程序设计方法分类、按计算方法分类以及按应用领域分类等。例如，从应用领域的角度可以分为以下几类。

(1) 科学计算语言。用于科学计算，基础是数学模型，过程描述的是数值计算，如 FORTRAN 语言。

(2) 系统开发语言。用于编写编译程序、操作系统、数据库管理系统 DBMS 等，如 C 语言。

(3) 实时处理语言。及时响应环境信息，可以根据外部信号对不同的程序段进行并发控制执行。

(4) 商用语言。主要用于商业处理、经济管理，基础为自然语言模型。

(5) 人工智能描述语言。模拟人的思维推理过程，实现智能化控制等。

(6) 模拟建模语言。用于模拟实现客观事物的发展与变化过程，以提前预测未来发展的结果。

(7) 网络编程语言。在网络技术基础上进入深层次的应用研究与开发的语言，如 Java 是一种新型跨平台分布式程序设计语言，Dephi 适于网络化环境的编程等，不再列举。

就技术性、基础性、实用性而言，选用哪种程序设计语言系统学习最为合适？可以说在各种现代编程语言尽显魅力的今天，仍当属 C 程序设计语言最具有代表性，这也是国内外高校计算机教育中热选不衰的原因之一。C 语言编程流畅、编译效率高、应用广泛，也是现代网络编程如 Java、面向对象可视化编程 Visual C++等程序设计的基础。我国高校于 1993 年前后逐步引入了 C 语言程序设计。由于 C 语言最早产生于 UNIX 操作系统平台后普及到微机上，由于其编码效率较高，能进行底层操作，比如机器的中断地址、设备端口、寄存器、字位操作等，还可以处理字符、图形和编写界面，并且使用方便，是开发系统软件的良好选择。对于学生来说，若能掌握一些硬件的基本知识，并熟练掌握 C 语言，将会成为现代就业的强项之一。

如今，学好计算机技术已经成为人们完善自我、提高工作技能的重要需求，成为人们现代生活中的一部分，因为相当多的职业都离不开计算机，而计算机的发展与运用正是现代科学技术发展的重要标志。

计算机可以代替人从事重复性的劳动，并能完成人所不能做到的事情。有人误以为计算机就是操作使用的工具，计算机工作的每一步都要靠人手工操作完成；而另一些人对计算机的作用估计过高，认为只要有了计算机，一切事情都可以自动办到。其实，计算机是

按照事先输入的人为设计的步骤，编制的程序，执行后才能自动连续并重复运行，从而精确、快速、规范地代替人进行复杂繁琐的工作的。计算机是集人类智慧设计和制造并为人类服务的工具，因此计算机不可能完全代替人的智慧和劳动，但需要利用计算机进行开创性的劳动，利用程序设计算法与程序设计应用实现人们要做的许多复杂的、繁琐的工作。

1.1.2　程序设计过程

任何程序设计过程都是为解决实际问题而进行的，用计算机解决问题的过程就是算法设计与程序实现的过程，最终以程序设计语言具体实现。

1. 算法设计与程序实现过程

程序设计是指我们使用一种计算机语言为实现解决实际问题的算法去设计编写计算机程序的过程。计算机语言是人与计算机进行交流的媒介，通过语言编写的程序，计算机就会准确地按程序步骤执行操作，计算机解决实际问题的一般过程如图 1.1 所示。

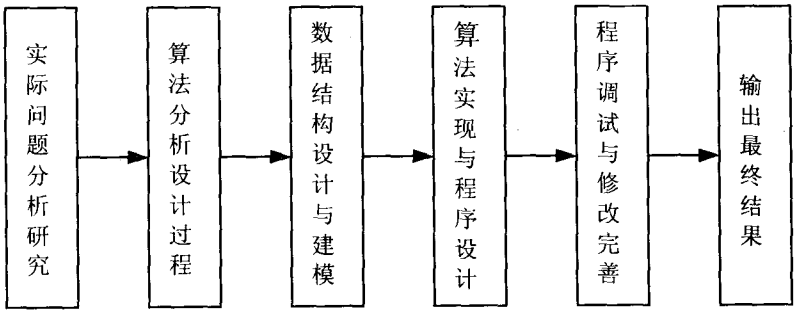


图 1.1　计算机解决实际问题的一般过程

其中，实际问题的分析与研究是算法设计的基础，根据待解决问题的难易程度，需要做大量的实地调研和分析工作，确定实现算法的运行环境，设计合理的数据结构，建立实现算法的模型，这些都是解决实际问题的具体步骤，不仅要有必要的 IT 专业基础理论与专业背景，还应具备实际设计调试经验，以及程序应用业务本身的要求，因此常常由具有专业技术背景的人员与程序员共同协作完成，然后才是程序员的程序设计与调试工作，整个工作过程中的关键是算法设计，最终是程序实现。

2. 程序设计的步骤

程序设计大致包含以下几个步骤。

(1)　分析问题。

程序设计首先要对问题进行分析，明白我们要做什么，确定要使用的数学模型。

(2)　确定算法。

确定算法即确定解决问题时要执行的一系列步骤。

(3)　算法描述。

算法描述就是使用计算机语言对算法予以描述，其中包括数据结构的设计与算法流程

的实现。

(4) 确定程序设计语言。

由于不同的计算机程序设计语言有不同的特点，根据实际情况与需要选好程序设计语言后，就可以用该语言编程实现算法。

(5) 调试和运行程序。

1.2 程序设计算法

简单地说，计算机算法就是用计算机解决问题的方法和步骤。例如，计算任意圆柱体的体积。分析这个问题，我们了解这是与数学计算有关的问题，用计算机语言实现时，有操作命令，还有操作对象，即操作数。程序运行时首先应该允许用户输入圆柱体的圆截面半径 r 和柱体高度 h 的数值数据；然后根据求圆柱体体积公式

$$V=\pi\times r^2\times h$$

计算其体积值数据，其中 π 在计算机中一般没有现成的、可任取精度的数据值可取，也没有 π 这个语言命令可以识别符号，因此或是定义 pi=3.14159 数据类型，或建立其他数学模型实现其达到的精度，计算后结果还应能正确地输出；分析正确后，选择合适的计算机程序设计语言，编写程序以实现整个算法过程，最后程序经过录入调试，正确运行后，便能实现“计算任意圆柱体的体积”算法的输入、计算、输出整个过程。这样就完成了计算机的算法设计与程序实现的过程。

语法计算机语言程序设计的文法规则，不能有误，否则计算机语言编译或解释系统就会检测出程序有语法错误，用户计算机程序就不会被“翻译”成机器语言正确执行。因此，语法规定应掌握准确。而解决问题的算法可以自由灵活地优化设计，因此同一个问题的算法可以有多种，可编的程序也不是惟一的。所以学习程序设计，熟练掌握语法规范和语言结构是必要的，而算法实现是核心，综上所述，计算机程序可以这样表示：

程序=算法+数据结构

其中，数据结构是对数据的描述，包括对数据类型的描述和对数据组织形式的描述定义。算法是对操作过程的描述，即操作步骤的描述。

如果考虑现代编程的工程化与多样性，可以这样表示：

程序=数据结构+算法+(程序设计方法+编程工具+语言环境)

其中算法是关键，是实现程序设计的依据和基础，算法分析做的完整，做的精细，才能有完整的程序设计，才可能对程序进行优化，所以掌握算法至关重要。

1.3 计算机算法的表示

人们在实际生活和工作中，无论做什么事情，都需要有计划、按步骤去完成，如果你计划去看一场电影或听一场音乐会，就需要先看海报，完成这项活动需要按指定的方式到

指定的地点付款、购票，再按规定的时间到规定的地点，验票进门找到自己的座位，接下来看电影或听音乐会，完成计划和步骤。实施执行这件事情的整个过程是按时间、按步骤顺序进行的，有条不紊直至完成，这仅仅是人们日常生活所熟悉的许许多多事情中的一项活动，这个过程实施的步骤就是“算法”，只是人们习以为常，从没有去细究其思维支配的过程，也就不必细细描述。

要让计算机去做一件事情，就必须把如何做这件事情分拆成许许多多小步骤，每一步正好由计算机语言的命令来完成，计算机按着顺序执行这一系列小步骤，最后执行完一系列命令，把所要做的事情做完。

描述这些步骤的方法有自然语言、流程图、N-S 图等，描述每一步骤的计算机命令选用的就是计算机语言。把计算机每一条命令对应的一系列步骤组成一个规则的整体，就是计算机程序。计算机程序设计“算法”就是利用计算机语言，让计算机完成解决问题的步骤。那么，假设解决问题的一项活动还穿插有其他活动，则需要周密计划、严谨安排，先做什么后做什么，再做什么，才可能有效完成，对应的计算机语言编程则是流程控制问题，所有的计算机程序流程控制结构都可以分为顺序结构、条件分支结构、循环结构三种方式。

顺序结构是程序设计最基本的结构，由顺序命令组成，程序执行流程是按顺序执行方式运行的。条件分支结构根据不同的条件执行不同的命令。循环结构根据给定的条件反复执行某一段程序，并不断修改给定的条件，最终结束循环。

1.3.1 自然语言描述

用自然语言描述算法是非常直接的一种表示方法，易于表达、易于理解，所以自然语言描述算法的最大优点是便于人们之间直接交流，可是最大的缺点也产生于人们之间的交流，表现在自然语言表述不够严谨和理解不一致时，容易产生歧义性。

例如，“学期年末获优秀学生奖的条件是：考核达标、平均成绩 90 分以上或从未迟到早退者，即可参加评奖。”

听起来很容易理解，但理解方式可见不是惟一的。

理解 1：考核达标、平均成绩 90 分以上，即可参加评奖。

理解 2：考核达标、从未迟到早退者，即可参加评奖。

理解 3：从未迟到早退者，即可参加评奖。

所以，自然语言表述不能有效合理地表达算法，显然也不能利用计算机有效地进行程序设计和实现算法。

1.3.2 程序流程图描述

用自然语言描述程序执行步骤，如果是顺序结构还比较容易，但是对于稍微复杂一点的程序流程，比如说条件判断分支结构或循环结构等，就不是很方便。因此，美国国家标准化协会 ANSI(American National Standard Institute)规定了一些常用的流程图符号，表示程序的执行步骤与控制流向，这就是程序流程图。如图 1.2 所示。

使用程序流程图表示程序算法，流程清晰、易看易懂，用这样一些符号描述程序的执行步骤为国际通用表示方法，在程序设计中普遍使用。

例如，用程序流程图表示这样一个算法：输入两个数，如果这两数之和大于 0，则输出结果；如果这两个数之和小于 0，则重新输入两个数，等于 0 结束。程序流程图如图 1.3 所示。

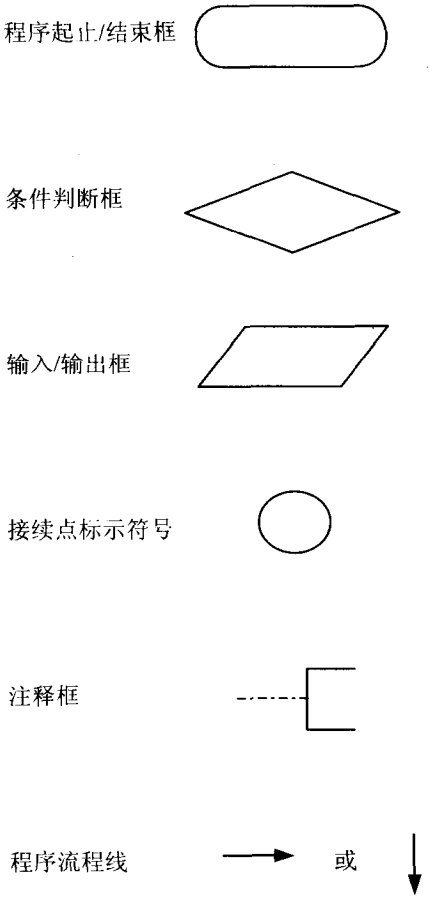


图 1.2　程序流程图基本符号

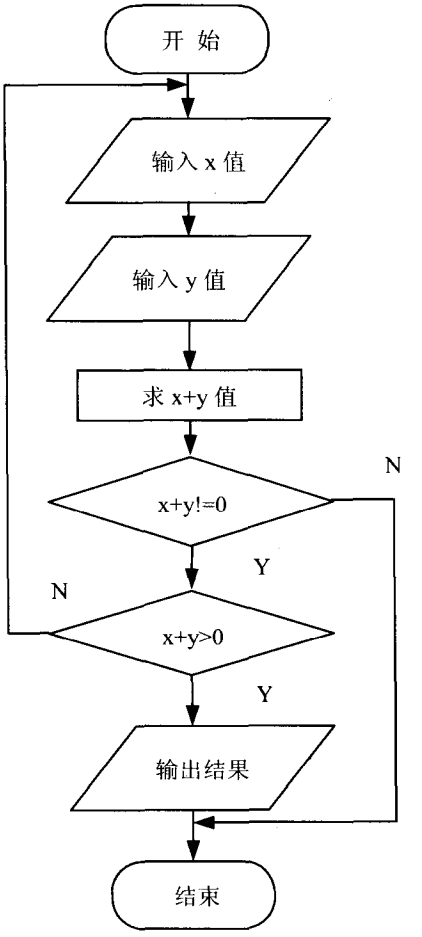


图 1.3　算法程序流程图

虽然用程序流程图表示算法，程序步骤和过程可以表达清楚，使用方便，简明易看，容易理解。但是，要表示复杂一些的算法就会显得凌乱繁杂，如图 1.4 所示。

由于这个原因，人们想到应该规定几种基本的算法设计结构，然后按照一定的规范组合成整体的算法结构，这些基本结构可以组成各种新的基本结构，然后按顺序搭接构成整个程序的算法结构，从而使程序设计质量大为提高。

1966 年，Bohra 和 Jacopini 提出了程序设计的三种基本结构，至今仍是各种计算机程序设计语言支持的程序流程控制基本结构。三种程序控制结构表示如图 1.5 所示。

实际上，使用上述三种基本结构很容易组成任何一种复杂的程序算法。很明显，以上

三种基本结构都是一个入口一个出口，用这三种基本结构组成的程序算法都属于结构化程序设计(Structured Programming)的算法，每种基本结构向外应该没有无规则转向流程，即每种基本结构只有一个入口和一个出口，否则就不能称为结构化程序设计。

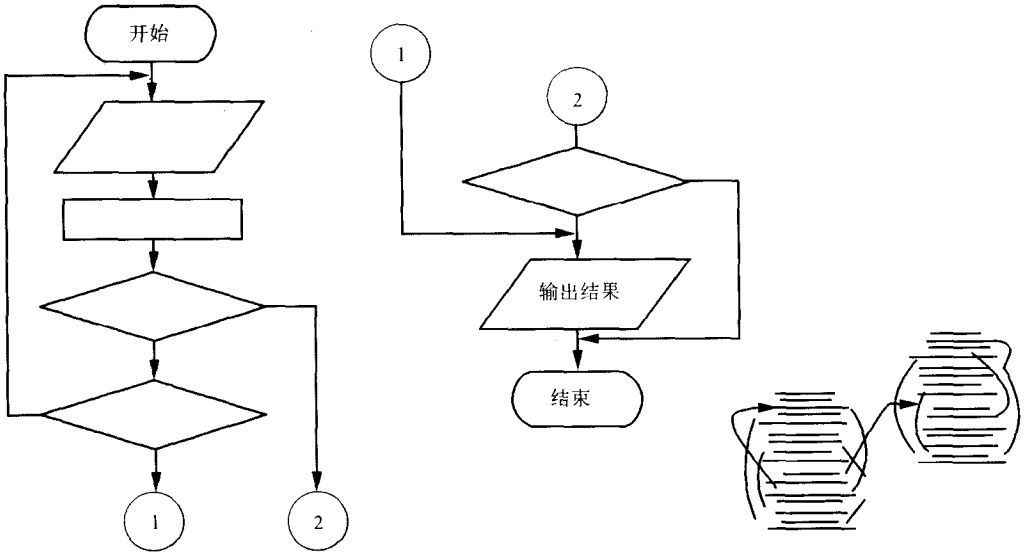


图 1.4　表示复杂的算法看似凌乱繁杂

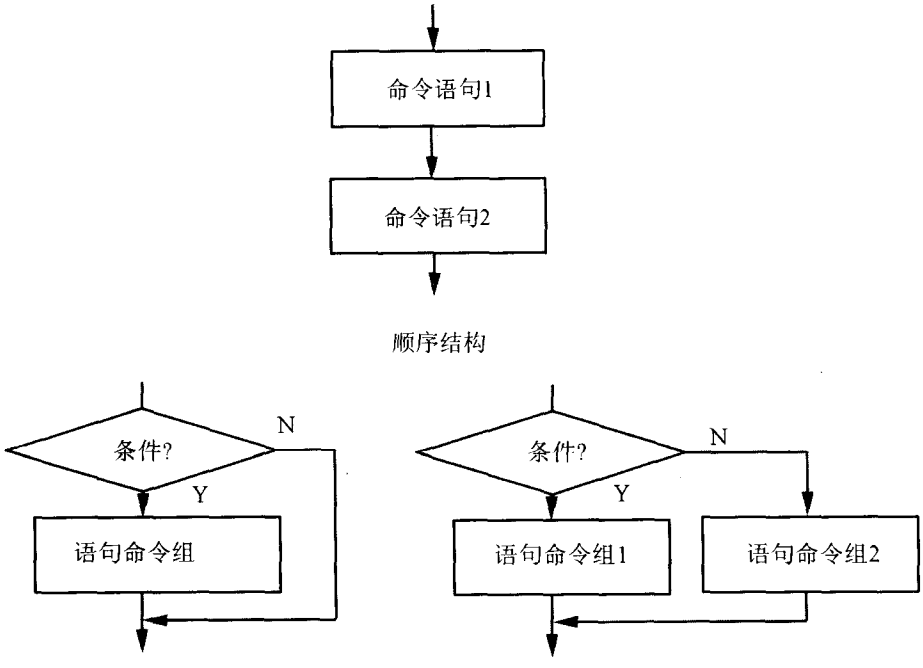
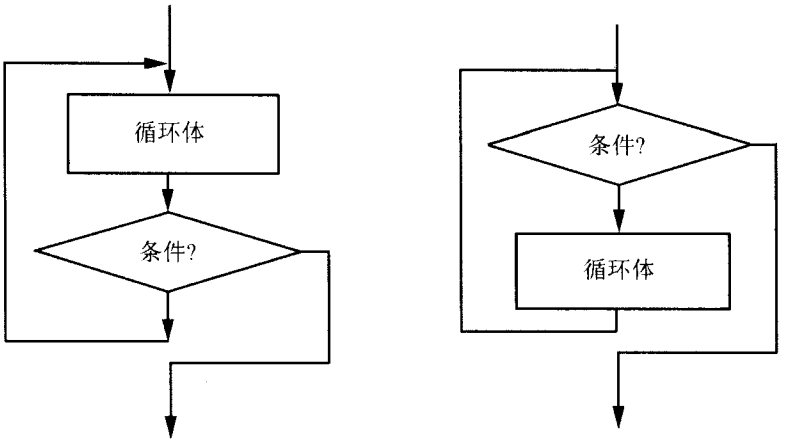


图 1.5　程序流程图表示的三种程序控制结构



循环控制结构

图 1.5(续)

1.3.3 N-S 图描述

结构化程序设计可以用程序设计基本结构组成的构件，顺序组合成各种复杂的算法结构，这样使用程序流程图的程序流程线就显得多余，于是 1973 年美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的程序控制流程图的表示方法，即可以使用矩形框表示三种基本结构，而使用三种基本结构的矩形框嵌套，就可以组成各种复杂的程序算法了，这就是 N-S 图，如图 1.6 所示。

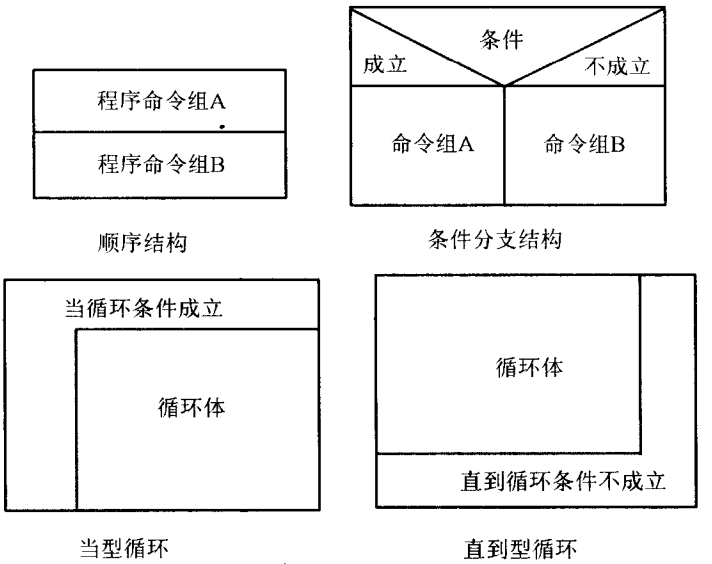


图 1.6 用 N-S 图表示三种基本结构