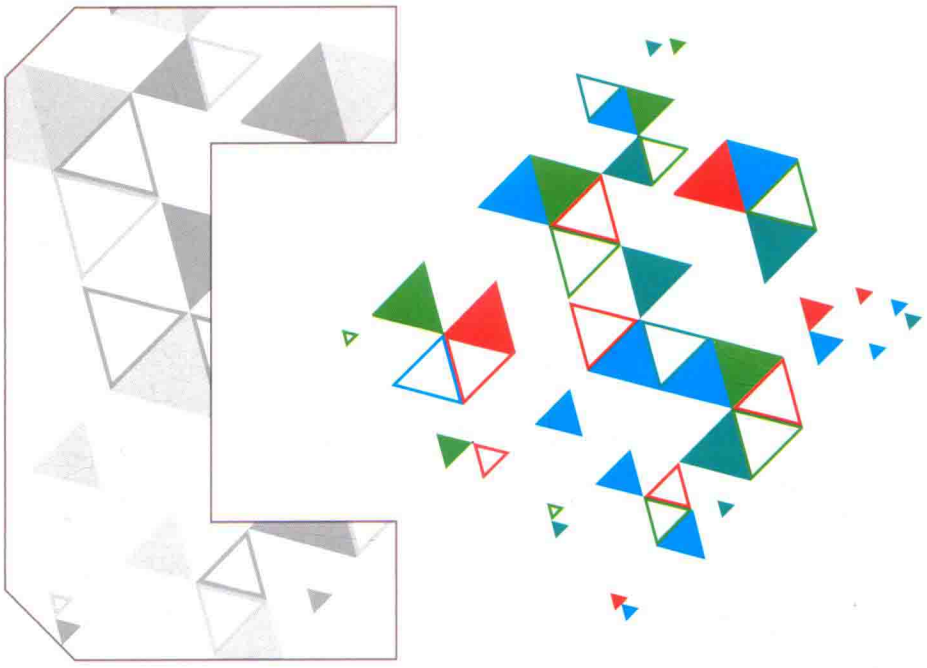
普通高等教育“十三五”规划教材



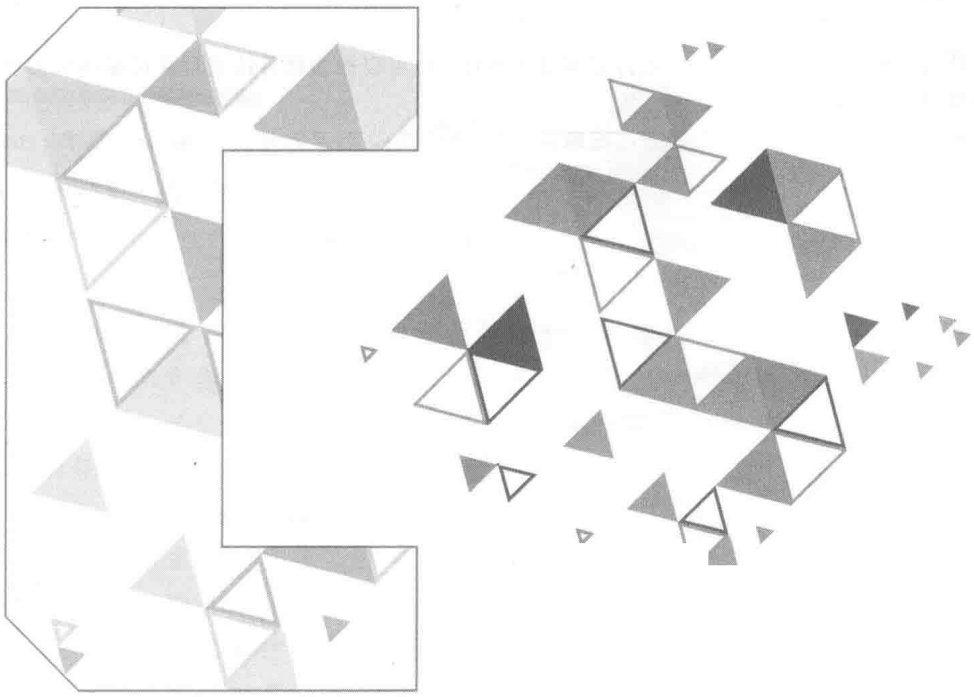
程序设计基础教程

—C语言

常东超　刘培胜　郭来德　等编著

化学工业出版社

普通高等教育 十二五 规划教材



程序设计基础教程
——C语言

◆◆◇◆ 常东超 刘培胜 郭来德 等编著

化学工业出版社
·北京·

本书是参照全新计算机等级考试（二级C语言）教学大纲及C99的新特性并根据高校全新C语言程序设计教学大纲要求编写而成；全书分为10章，主要内容有程序设计基础理论和C程序的基本组成以及程序开发过程；C语言的基本数据类型、运算符、表达式、数据类型转换及标准的输入输出函数；C语言的基本语句和流程控制语句；数组、函数、指针的概念及用法；C语言的编译预处理功能；C语言结构体与共用体、C语言中文件的相关概念以及文件的各种操作方法；最后附录部分介绍了C程序设计的常用库函数。

本书既可以作为高等学校本科计算机C语言程序设计教材，也可以作为培养读者计算机编程能力和参加全国计算机等级考试（C语言）的自学参考书。

本书另配有电子教案（PPT格式）与课后习题解答（Word格式），联系邮箱：changdc885@126.com。

图书在版编目(CIP)数据

程序设计基础教程：C语言/常东超等编著. —北京：化学工业出版社，2019.2
普通高等教育“十三五”规划教材
ISBN 978-7-122-33343-8

Ⅰ.①程… Ⅱ.①常… Ⅲ.①C语言-程序设计-高等学校-教材 Ⅳ.①TP312.8

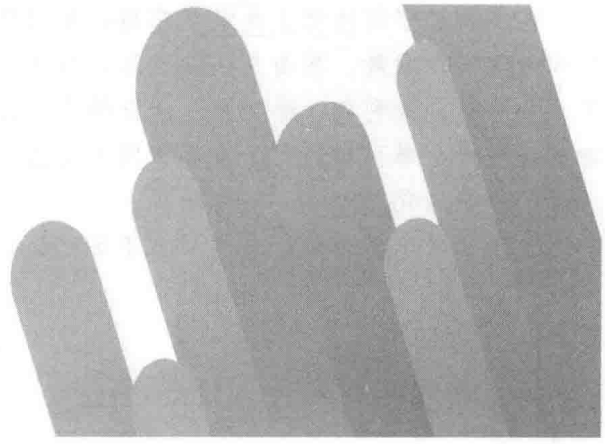
中国版本图书馆CIP数据核字（2018）第270243号

责任编辑：满悦芝 石 磊　　文字编辑：吴开亮
责任校对：王鹏飞　　装帧设计：张 辉

出版发行：化学工业出版社（北京市东城区青年湖南街13号 邮政编码100011）
印　　装：三河市延风印装有限公司
787mm×1092mm 1/16 印张17 字数525千字 2019年3月北京第1版第1次印刷

购书咨询：010-64518888　售后服务：010-64518899
网　　址：http://www.cip.com.cn
凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：48.00元



前言

2017年2月以来，教育部积极推进新工科建设，先后形成了“复旦共识”“天大行动”和“北京指南”等多项共识，并发布了《关于开展新工科研究与实践的通知》《关于推荐新工科研究与实践项目的通知》，全力探索形成领跑全球工程教育的中国模式、中国经验，助力高等教育强国建设。

为了适应时代发展和人才培养的需求以及计算机技术的发展，在C语言标准及编译技术、集成开发环境不断变化的背景下，本书在作者多年《C语言程序设计》讲义的基础上，结合数位一线教师多年教学实践与研发经验，并考虑到学生的反馈信息，对各个章节的内容、结构等进行了修订、调整、完善和补充。特别在复杂结构的叙述方法上，作者根据多年的心得体会对教学内容进行了重新组织和编排，力求深入浅出、通俗易懂，使广大读者尽早、尽快掌握C语言程序设计方法，具备一定的程序设计能力。

全书分为10章，主要内容有程序设计基础理论和C语言程序的基本组成以及程序开发过程；C语言的基本数据类型、运算符、表达式、数据类型转换及标准的输入输出函数；C语言的基本语句和流程控制语句；数组、函数、指针的概念及用法；C语言的编译预处理功能；C语言结构体与共用体、C语言中文件的相关概念以及文件的各种操作方法；附录部分介绍了C程序设计的常用库函数。尤其值得一提的是，作者在C语言语法结构等基础知识的介绍方面作了进一步的总结和归纳，增加了一些和专业相关的案例内容，从而使读者阅读此书和学习C语言时更能感觉到有章可循。

本书紧紧围绕《全国计算机二级C语言程序设计考试大纲》，采用“案例驱动”的编写方式，以基础语法、语义训练为中心，语法介绍精炼，内容叙述深入浅出、循序渐进，程序案例与实际相结合、生动易懂，具有很好的启发性。每章均配备教学课件和精心设计的习题。本书配套的《C语言程序设计实验指导与习题精选》与教材结合紧密，对教学质量的提高可起到积极的促进作用。

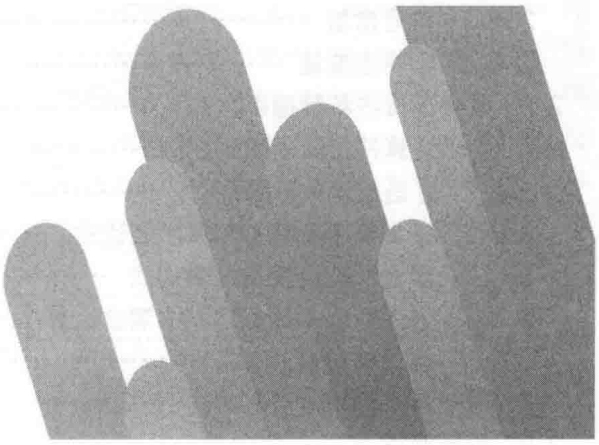
本书与传统教材相比，在下面三个方面进行了改进和强化：(1) 结合教学心得对部分知识点的叙述方法做了仔细修改，以使读者更容易理解；(2) 为了扩大读者视野和更深入掌握C语言程序设计的方法，本书增加了有关编程的部分新内容并删改了不适应当前编程需要的陈旧内容，创新了部分习题；(3) 结合全国计算机等级考试全新编程环境，采用新的标准对全书和例题内容进行了调试。

全书由辽宁石油化工大学常东超、刘培胜、郭来德等编著，参加编写和书稿校对工作的教师有卢紫微、苏金芝、吕宝志、吉书朋、杨妮妮、张国玉、韩云萍、王杨、张凌宇、李会举、张利群、徐晓军、胡玉娥等。辽宁石油化工大学计算机与通信工程学院李艳杰教授对全书内容进行了全面审阅并提出了很多宝贵建议，在此致以诚挚的谢意。全书由辽宁石油化工大学常东超统稿。

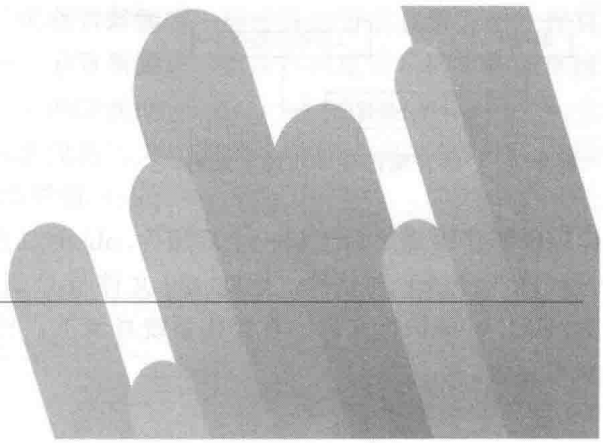
限于作者水平有限，书中如有不足之处，敬请读者批评指正，以利作者改进。

编著者

2019 年 1 月



目录



第1章

C语言程序设计概述

1.1 程序和程序设计的基本概念

目前计算机的应用已经深入到社会的各个领域，成为人们生活、学习和工作的必备工具。早期的计算机是用来计算的机器，但是目前的计算机尤其是微型计算机已不再是简单的计算机器了，而是具有强大的存储和计算能力、由程序自动控制的智能化电子设备。人与人之间交流需要语言，那么人与机器要进行交流也需要“语言”，这种“语言”就是我们要说的“程序”。程序来自生活，通常指完成某一事物的既定方式和过程。比如，我们常说做什么事情需要什么样的程序，在日常生活中可以看成是对一系列执行过程的描述。以学生去食堂打饭为例，为了完成这件事情需要几个步骤呢？首先，带上饭卡去食堂；其次，到相应的窗口去排队；然后，挑选饭菜并刷卡；接着，食堂职工负责盛取饭菜相应事宜；最后，离开食堂窗口，这样就完成了去食堂打饭的程序。

那么计算机中的程序是怎样定义的呢？它是指为计算机执行某些操作，或解决某个问题而编写的一系列有序指令的集合。这里涉及“指令”的概念，以日常生活中的指令来解释，比如，办公室里的老板对秘书发出指令，需要秘书做口述笔记，键入信函内容、发传真等等，那秘书就需要按照指示逐步完成老板发出的每一条指令，最终完成老板交代的事情。在计算机中，就是由程序员对计算机发出指令，想让计算机解决某个问题，就可将解决问题的过程用计算机能够接受的方式或者选择某一种计算机语言将它一步一步地描述出来，计算机就会按照预先存储在它里面的代码逐步去执行，这就是计算机中的指令，而指令的集合就构成了计算机中的程序，编写程序的过程就称为“程序设计”。

计算机程序就是人与计算机交流的“语言”，也就是程序设计语言。正如人与人交流有不同的语言一样，程序设计语言也有很多种，基本上分为高级语言和低级语言两大类。目前常见的高级语言有 Visual Basic、C++、Java、C 等，这些语言都是用接近人们习惯的自然语言和数学语言作为表达形式，使人们学习和操作起来感到十分方便。但是，对于计算机本身来说，它并不能直接识别由高级语言编写的程序，只能接受和处理由 0 和 1 的代码构成的二进制指令或数据。由于这种形式的指令是面向机器的，因此也称为“机器语言”。

我们把由高级语言编写的程序称为“源程序”，把由二进制代码表示的程序称为“目标程序”。为了把源程序转换成机器能接受的目标程序，软件工作者编制了一系列软件，通过这些软件可以把用户按规定语法写出的语句一一翻译成二进制机器指令。这种具有翻译功能

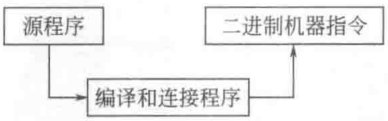


图 1.1 C 语言编译程序功能示意图

的软件称为“编译程序”，每种高级语言都有与它对应的编译程序。例如，C 语言编译程序就是这样的一种软件，功能如图 1.1 所示。

我们无论采用哪种语言编写程序，经过编译（compile）最终都将转换成二进制的机器指令。C 源程序经过 C 编译程序编译之后生成一个后缀为.obj 的二进制文件（称为目标文件），然后由称为“连接程序”（link）的软件，把此.obj 文件与 C 语言提供的各种库函数连接起来生成一个后缀为.EXE 的可执行文件。在操作系统环境下，只需点击或输入此文件的名字，该可执行文件就可运行。

1.2 算法与程序设计

所谓算法，是指为解决某个特定问题而采取的确定且有限的步骤。

对于一个问题，如果可以通过一个计算机程序，在有限的存储空间内运行有限的时间而得到正确的结果，则称这个问题是算法可解的。但是算法不等于程序，也不等于计算方法。当然，程序也可以作为一种描述，但通常还需考虑很多与方法和分析无关的细节问题，这是因为在编写程序时要受到计算机系统环境的限制。通常程序的编制不可能优于算法的设计。

1.2.1 算法的基本特征

作为一个算法，一般具有以下几个特征。

(1) 可行性

针对实际问题设计的算法，人们总是希望得到满意的结果。但一个算法又总是在某个特定的计算工具上执行的，因此，算法在执行过程中往往受到计算工具的限制，使执行结果产生偏差。例如，在进行数值计算时，如果某计算工具具有 7 位有效数字，则在计算

$$X=10^{12},Y=1,Z=-10^{12}$$

3 个量的和时，如果采用不同的运算顺序，就会得到不同的结果，即：

$$X+Y+Z=10^{12}+1+(-10^{12})=0$$

$$X+Z+Y=10^{12}+(-10^{12})+1=1$$

而在数学上，$X+Y+Z$ 与 $X+Z+Y$ 是完全等价的。因此，算法与计算公式是有差别的。在设计一个算法时，必须要考虑它的可行性，否则是不会得到满意结果的。

(2) 确定性

算法的确实性，是指算法的每一个步骤都必须是有明确定义的，不允许有模棱两可的解释，也不允许有多义性。这一性质也反映了算法与数学公式的明显差别。在解决实际问题时，可能会出现这样的情况：针对某种特殊问题，数学公式是正确的，但按此数学公式设计的计算过程可能会使计算机系统无所适从。这是因为，根据数学公式设计的计算过程只考虑了正常使用的情况，而当出现异常情况时计算机就不能适应了。

(3) 有穷性

算法的有穷性，是指算法必须能在有限的时间内完成，即算法必须能在执行有限个步骤

之后终止。数学中的无穷级数，在实际计算时只能取有限项，即计算无穷级数数值的过程只能是有穷的。因此一个数的无穷级数表示只是一个计算公式，而根据精度要求确定的计算过程才是有穷的算法。

算法的有穷性还应包括合理的执行时间的含义。因为，如果一个算法需要执行千万年，显然就失去了实用价值。

(4) 输入

一个算法可以有零个或多个输入。由于在计算机上实现的算法是用来处理数据对象的，因此在大多数情况下这些数据对象需要通过输入来得到。

(5) 输出

一个算法有一个或多个输出，这些输出是同输入有着某些特定关系的量。

一个算法是否有效，还取决于为算法所提供的情报是否足够。通常，算法中的各种运算总是施加到各个运算对象上，而这些运算对象又可能具有某种初始状态，这是算法执行的起点或依据。因此，一个算法的执行结果总是与输入的初始数据有关，不同的输入将会有不同的结果输出。当输入不够或是输入错误时，算法本身也就无法执行或导致执行出错。一般来讲，当算法拥有足够的情报时，此算法才是有效的，而当提供的情报不够时，算法可能失效。

综上所述，所谓算法，是一组严谨的定义运算顺序的规则，并且每一个规则都是有效的，且是确定的，此顺序将在有限的次数下终止。

1.2.2 算法的基本要素

一个算法通常由两种基本要素组成：一是对数据对象的运算和操作，二是算法的控制结构。

(1) 算法中对数据对象的运算和操作

每个算法实际上是按解题要求从环境能够进行的所有操作中选择合适的操作所组成的一组指令序列。因此计算机算法就是计算机能处理的操作所组成的指令序列。

通常，计算机可以执行的基本操作是以指令的形式描述的。一个计算机系统所能执行的所有指令集合称为该计算机系统的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令所组成的指令序列。在一般的计算机系统中，基本的运算和操作有以下4种。

① 算术运算：主要包括加、减、乘、除等运算。

② 逻辑运算：主要包括“与”“或”“非”等运算。

③ 关系运算：主要包括“大于”“小于”“等于”“不等于”等运算。

④ 数据传输：主要包括赋值、输入、输出等操作。

(2) 算法的控制结构

一个算法的功能不仅仅取决于所选用的操作，还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

算法的控制结构给出了算法的基本框架，它不仅决定了算法中各操作的执行顺序，还直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有传统流程图、N-S结构化

流程图、算法描述语言等。一个算法一般都可以用顺序、选择、循环 3 种基本控制结构组合而成。

1.2.3 算法描述的方法

算法是描述某一问题求解的有限步骤，而且必须有结果输出。设计一个算法，或者描述一个算法，最终是由程序设计语言来实现的。但算法与程序设计又是有区别的，主要是一个由粗到细的过程。算法是考虑实现某一个问题求解的方法和步骤，是解决问题的框架流程；而程序设计则是根据这一求解的框架流程进行语言细化，实现这一问题求解的具体过程。

一般可以使用下面几种类型的工具描述算法：

(1) 自然语言

自然语言即人们日常进行交流的语言，如英语、汉语等。自然语言用来描述算法，分析算法，作为用户相互之间进行交流，是一种较好的工具。但是将自然语言描述的算法直接在计算机上进行处理，目前还存在许多困难，包括有诸如语音语义识别等方面的问题。

(2) 专用工具

要对某一个算法进行描述，可以借助于有关图形工具或代码符号。20 世纪 50～60 年代兴起的流程图几乎成了程序设计及算法描述的必用工具，是描述算法很好的形式。

人们已经提出了多种描述算法的流程图。这种方法的特点是用一些图框表示各种类型的操作：圆角矩形表示起止框，在算法开始和结束的时候使用；菱形表示判断框；矩形表示处理框；平行四边形表示输入/输出框；带点不封闭矩形表示注释框；基本图形用流程线连接起来，通过流程线反映出它们之间的关系；小圆圈表示连接点，大型系统中流程线和基本图形连接起来时，通常用小圆圈表示。图 1.2 为一般流程图所用的基本符号图形。

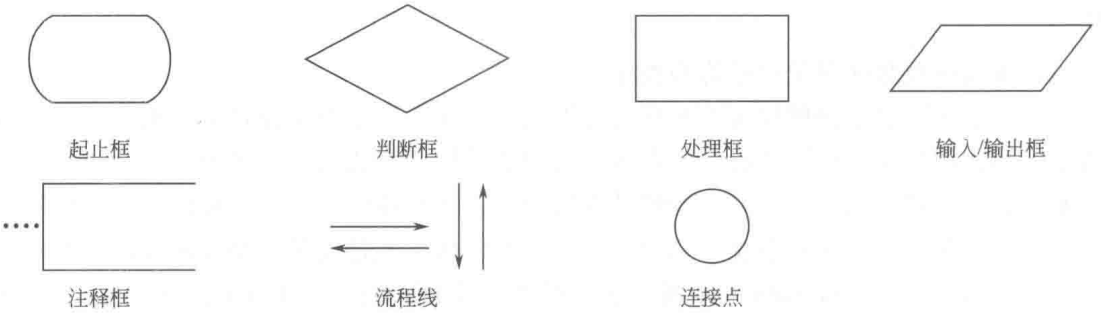


图 1.2 流程图所用的基本符号图形

流程图这里主要介绍传统流程图和 N-S 流程图。

传统流程图的优点是形象直观、简单方便，根据流程线能很直观地看出程序的走向；缺点是它对流程线的走向没有任何的限制，导致谁都可以修改流程的走向，使流程随意转向，而且它在描述复杂算法时，所占篇幅较多。

N-S 流程图是在 1973 年，由美国学者 I. Nassi 和 B. Shneiderman 提出的一种新的流程图形式，这种流程图完全去掉了流程线，算法的每一步都用一个矩形框来描述，把一个个矩形框按执行的次序连接起来就是一个完整的算法描述。这种流程图用两位学者的第一个英文字母命名，因此称为 N-S 流程图。在下一小节中将结合结构化程序设计中的三种基本结构来介绍这种流程图的基本结构。

(3) 部分常用的算法

在程序设计时，常常要通过一些特定的算法来求解。现在比较常用的算法有下列几种：

① 迭代法　一般的一元五次或更高次的方程，以及几乎所有的超越方程、微分方程问题都无法用解析方法通过求根公式来求解，人们只能用数值方法求其近似解。用事先估计的一个根的初始值 X_0，通过迭代算式 $X_K+1=G(X_K)$ 求出另一个近似值 X_1，再由 X_1 求出 X_2，从而获得一个求解的序列 $\{X_0, X_1, X_2, \cdots, X_n, \cdots\}$ 来逼近方程 $f(x)=0$ 的根。这种求解的过程称为迭代。

② 枚举法　枚举法的基本思想是首先依据题目部分条件确定答案的大致范围，然后逐一验证所有可能的情况，这种算法也称为穷举法。

③ 递归法　递归是指一个过程直接或间接地调用它自身，递归过程必须有一个递归终止条件。

如：

$$\begin{cases} n!=1 & n=0 \\ n!=n(n-1)! & n>0 \end{cases}$$

④ 递推法　算法从递推初始条件出发，应用递推公式对问题进行求解。如 Fibonacci 数列存在递推关系：

$$F(1)=1, F(2)=1, F(n)=F(n-1)+F(n-2), (n>2)$$

若需求第 30 项的值，则依据公式，从初始条件 $F(1)=1$、$F(2)=1$ 出发，可逐步求出 $F(3)$、$F(4)$……，直到求出 $F(30)$。

除此之外，还有回溯法（一种基本策略）、分治法（问题分解成较小部分，求解再组合）等其他的常见算法，在此不一一阐述。

1.2.4　程序设计

(1) 程序设计

程序设计的过程一般包含以下几个部分：

① 确定数据结构：是指根据任务书提出的要求、指定的输入数据和输出结果，确定存放数据的数据结构。

② 确定算法：针对存放数据的数据结构来确定解决问题、完成任务的步骤。

③ 编码：根据确定的数据结构和算法，使用选定的计算机语言编写程序代码，输入到计算机并保存在磁盘上，简称编程。

④ 调试：目的是验证代码的正确性，用各种可能的输入数据对程序进行测试，消除由于疏忽而引起的语法错误或逻辑错误；使之对各种合理的数据都能够得到正确的结果，对不合理的数据能进行适当的处理。

⑤ 整理并写出文档资料。

这就是编写程序的五个步骤。

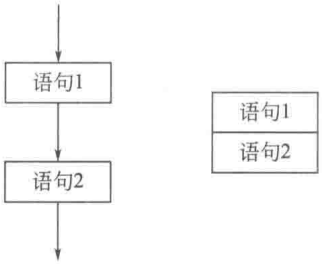
(2) 结构化程序设计

对于任何一个程序来说，都有自身的结构。就像我们读文章一样，我们说文章的结构是顺叙的、倒叙的或是插叙的等，结构化的程序设计由三种基本结构组成。

① 顺序结构　在本书第 3 章中将要介绍的如赋值语句，输入、输出语句都可以构成顺

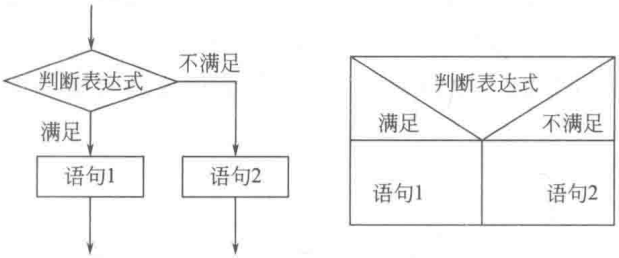
序结构。当执行由这些语句构成的程序时，将按这些语句在程序中的先后顺序逐条执行，没有分支，没有转移。顺序结构可用图 1.3 所示的流程图表示，其中图（a）是一般的流程图，图（b）是 N-S 流程图。

② 选择结构　在本书第 4 章中将要介绍的 if 语句、switch 语句都可以构成选择结构。当执行到这些语句时，将根据不同的条件去执行不同分支中的语句。当判断表达式满足条件时，执行语句 1，否则执行语句 2。选择结构用图 1.4 所示的流程图表示，其中图（a）是一般的流程图，图（b）是 N-S 流程图。



(a) 一般的流程图　(b) N-S流程图

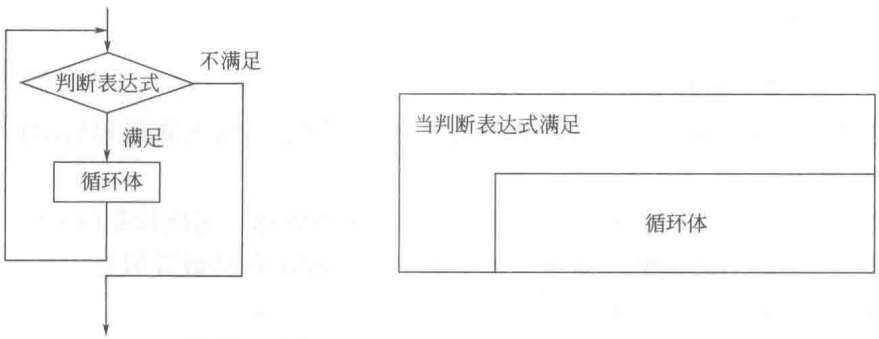
图 1.3　顺序结构流程图



(a) 一般的流程图　(b) N-S流程图

图 1.4　选择结构流程图

③ 循环结构　在本书第 5 章中将要介绍不同形式的循环结构，它们将根据各自的条件，使同一组语句重复执行多次或一次也不执行。循环结构的流程图如图 1.5 和图 1.6 所示，每个图中图（a）是一般的流程图，图（b）是 N-S 流程图。图 1.5 是当型循环流程图。当型循环的特点是：当指定的条件满足（成立）时，就执行循环体，否则就不执行。图 1.6 是直到型循环流程图。直到型循环的特点是：执行循环体直到指定的条件满足（成立）时就不再执行循环体。



(a) 一般的流程图　(b) N-S流程图

图 1.5　当型循环流程图

已经证明，由三种基本结构组成的算法可以解决任何复杂的问题。由三种基本结构所构成的算法称为结构化算法；由三种基本结构所构成的程序称为结构化程序。

(3) 模块化程序设计

当计算机在处理较复杂的任务时，程序代码量非常大，通常会有上万条语句，需要由许多人来共同完成。这时常常把这个复杂的任务分解为若干个子任务，每个子任务又可分为很多小的子任务，每个小的子任务只完成一项简单的功能。在程序设计时，用一个个小模块来