

高等教育“十二五”规划教材

C语言程序设计

(第三版)

时景荣 李 鑫 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等教育“十二五”规划教材

C 语言程序设计

(第三版)

主 编 时景荣 李 鑫

副主编 罗时光 林 琳

参 编 季玉茹 吴雪莉 孙维福

内 容 简 介

本书通俗易懂、由浅入深、循序渐进地讲解了C语言程序设计的方法,突出基础知识、基本方法、基本技能的训练,以培养学生的程序设计能力。

全书共分12章,主要内容包括程序设计概述,数据、数据类型与数据运算,顺序结构程序设计,选择结构程序设计,循环结构程序设计,编译预处理,函数,数组,指针,结构体与共用体,数据文件,C语言综合应用等,并用实例阐述了应用程序的设计过程和可控菜单的设计方法。

为了满足教学和实验的要求,编者还编写了《C语言程序设计同步训练与上机指导(第三版)》与本书配套使用。

本书既可作为高等学校相关课程的教材,也可作为高职高专、培训机构的教学用书。

图书在版编目(CIP)数据

C语言程序设计/时景荣,李鑫主编. —3版. —

北京:中国铁道出版社,2015.2

高等教育“十二五”规划教材

ISBN 978-7-113-19917-3

I. ①C… II. ①时… ②李… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第026725号

书 名: C语言程序设计(第三版)

作 者: 时景荣 李 鑫 主编

策 划: 滕 云

读者热线: 400-668-0820

责任编辑: 周 欣

编辑助理: 刘丽丽

封面设计: 大象设计·小威

封面制作: 白 雪

责任校对: 冯彩茹

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.51eds.com>

印 刷: 三河市宏盛印务有限公司

版 次: 2007年12月第1版 2009年12月第2版 2015年2月第3版 2015年2月第1次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 16.75 字数: 403千

印 数: 1~3 300册

书 号: ISBN 978-7-113-19917-3

定 价: 33.00元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

第三版前言

C 语言程序设计是大、中专院校普遍开设的计算机基础课程。C 语言具有功能强大、编程限制少、灵活性大等优点,这也意味着它不好把握,易出错,难检错,调试困难。所以对使用者的要求较高,尤其是初学者会感到 C 语言很难学。本书的编写就是针对这些问题,根据编者多年的教学实践经验,借鉴和吸取了多部已有 C 语言教材的优点,力求做到概念叙述简明清晰、通俗易懂,知识介绍由浅入深、循序渐进,例题选择实用丰富、针对性强,突出基础知识、基本方法、基本技能的训练,以培养学生程序设计的能力。

全书共分 12 章,包括程序设计概述,数据、数据类型与数据运算,顺序结构程序设计,选择结构程序设计,循环结构程序设计,编译预处理命令,函数,数组,指针,结构体与共用体,数据文件,C 语言综合应用等内容,并用实例阐述了应用程序的设计过程和可控菜单的设计方法。

本书具有以下特点:

- (1) 不追求 C 语言语法的全面性,突出常用、实用的重点概念,尽量做到少而精。
- (2) 在内容组织方面,集中讲授概念,用实例阐述方法。例如,在“循环结构程序设计”一章中,用一个例题集中讲解循环控制语句,用实例归纳总结循环控制方法,通过介绍常用的算法设计方法强化循环结构程序设计的训练。
- (3) 在章节安排方面,注意分散教学难点。“函数”是 C 语言的重点,函数之间的数据传递关系是一个难点。本书将“函数”安排在“数组”之前,先解决函数参数的“数值传递”问题,然后讲“数组”,解决函数参数的“地址传递”问题,从而分解了难点,并且增加了函数设计的机会,强化了“C 程序由函数组成”这个重点。
- (4) 加强程序设计风格的训练,提高程序的可读性。本书在程序设计前有算法分析,程序中有注释,程序后有测试,培养学习者严谨的程序设计作风。
- (5) 加强 C 语言综合应用的训练,用实例阐述应用程序的设计过程和可控菜单的设计方法,以开拓学生思维。

为了满足教学和实验要求,编者还编写了《C 语言程序设计同步训练与上机指导(第三版)》与本书配套使用。

本书由时景荣、李鑫任主编，罗时光、林琳任副主编，季玉茹、吴雪莉、孙维福参与了编写。其中，第1、5章由吴雪莉编写，第2、3章由季玉茹编写，第4章和附录由孙维福编写，第6、7章由林琳编写，第8、9、10章由罗时光编写，第11、12章由李鑫编写。全书由时景荣统稿，王立国教授主审。

在本书的编写过程中，有许多老师和同学提出了宝贵的意见和建议，有的还参加了书中部分程序的调试，在此表示衷心的感谢。

本书既可作为高等学校相关课程的教材，也可作为高职高专、培训机构的教学用书。

由于作者水平有限，书中难免有错误和不足之处，恳请读者不吝赐教，给予指正。

编者
2014年12月

目 录

第 1 章 程序设计概述	1
1.1 程序设计的概念	1
1.1.1 程序与程序设计	1
1.1.2 程序设计的过程	2
1.2 数据结构与算法	3
1.2.1 数据结构	3
1.2.2 算法	3
1.3 程序设计方法	5
1.3.1 结构化程序设计方法	5
1.3.2 面向对象的程序设计方法	6
1.4 程序设计语言	7
1.4.1 程序设计语言分类	7
1.4.2 C 语言概述	7
习题 1	11
第 2 章 数据、数据类型与数据运算	12
2.1 数据与数据类型	12
2.1.1 进制	12
2.1.2 进制转换	14
2.1.3 原码、反码、补码	16
2.1.4 数据类型	17
2.2 常量	17
2.2.1 整型常量	17
2.2.2 实型常量	18
2.2.3 字符型常量	18
2.2.4 字符串常量	20
2.2.5 符号常量	20
2.3 变量	21
2.3.1 变量的概念	21
2.3.2 变量的定义与初始化	21
2.3.3 变量的存储与变量的取值范围	22
2.4 数据运算	24
2.4.1 赋值运算	24

2.4.2 算术运算	26
2.4.3 关系运算	27
2.4.4 逻辑运算	27
2.4.5 位运算	28
2.4.6 其他运算	30
习题 2	33
第 3 章 顺序结构程序设计	35
3.1 C 语句概述	35
3.1.1 控制语句	35
3.1.2 顺序结构的语句	36
3.2 输入/输出和头文件的概念	36
3.2.1 C 语言的输入与输出	37
3.2.2 C 程序的头文件	37
3.3 字符数据的输出和输入	37
3.3.1 字符输出函数 putchar()	38
3.3.2 字符输入函数 getchar()	38
3.4 格式输出和格式输入	39
3.4.1 格式输出函数 printf()	39
3.4.2 格式输入函数 scanf()	42
3.5 顺序结构程序举例	45
习题 3	48
第 4 章 选择结构程序设计	51
4.1 if 语句	51
4.1.1 单选择结构	51
4.1.2 双选择结构	52
4.1.3 选择结构的嵌套	53
4.2 switch 语句	56
4.3 选择结构程序举例	58
习题 4	62
第 5 章 循环结构程序设计	65
5.1 循环控制语句	65
5.1.1 goto 语句及其构成的循环	65
5.1.2 while 语句	67
5.1.3 do...while 语句	67
5.1.4 for 语句	68
5.1.5 break 语句和 continue 语句	70
5.2 循环的控制方法	71

5.2.1 计数控制	71
5.2.2 条件控制	73
5.2.3 结束标志控制	74
5.3 算法设计的常用方法	74
5.3.1 穷举法	75
5.3.2 迭代法	75
5.3.3 递推法	76
5.4 循环嵌套结构	77
5.5 循环结构程序举例	78
习题 5	83
第 6 章 编译预处理	86
6.1 宏定义	86
6.1.1 不带参数的宏定义	86
6.1.2 带参数的宏定义	88
6.2 文件包含	92
6.3 条件编译	92
习题 6	94
第 7 章 函数	96
7.1 函数定义与函数调用	96
7.1.1 函数定义	96
7.1.2 函数调用	99
7.1.3 函数与带参宏的区别	104
7.2 函数的嵌套调用与递归调用	105
7.2.1 函数的嵌套调用	105
7.2.2 函数的递归调用	106
7.3 变量的存储类型与作用域	108
7.3.1 变量的存储类型	108
7.3.2 变量的作用域	109
7.4 内部函数与外部函数	115
7.4.1 内部函数	115
7.4.2 外部函数	116
7.5 函数设计举例	116
习题 7	119
第 8 章 数组	123
8.1 一维数组	123
8.1.1 一维数组的定义和初始化	123
8.1.2 数组元素的引用和基本操作	125

8.1.3	数组名作为函数的参数	126
8.1.4	一维数组应用举例	128
8.2	二维数组	133
8.2.1	二维数组的定义与初始化	133
8.2.2	二维数组元素的引用	134
8.2.3	二维数组应用举例	134
8.3	字符数组	137
8.3.1	字符数组的存储和初始化	137
8.3.2	字符数组的输入与输出	138
8.3.3	常用的字符串处理函数	139
8.3.4	字符数组应用举例	142
习题 8		145
第 9 章	指针	149
9.1	指针和指针变量的概念	149
9.1.1	变量的存取	149
9.1.2	变量的指针和指针变量	150
9.2	指向变量的指针变量	150
9.2.1	指针变量的定义	151
9.2.2	指针变量的引用	151
9.2.3	指针变量的初始化	152
9.2.4	指针变量作为函数参数	154
9.3	一维数组与指针变量	157
9.3.1	指向一维数组的指针变量	157
9.3.2	通过指针变量引用数组元素	157
9.3.3	指向数组元素的指针变量允许的运算	160
9.3.4	数组名作为函数参数	161
9.3.5	字符串与指针变量	161
9.4	多级指针与指针数组	163
9.4.1	多级指针	163
9.4.2	指针数组	164
9.4.3	main()函数的命令行参数	165
9.5	二维数组与指针变量	166
9.5.1	二维数组元素及其地址的表示方法	166
9.5.2	二维数组元素的引用	168
9.6	指向函数的指针变量	171
9.6.1	指向函数的指针变量的定义	171
9.6.2	指向函数的指针变量的引用	172

9.6.3	指向函数的指针变量作为函数参数	172
9.6.4	返回指针值的函数	173
9.7	存储空间的动态分配与释放	174
9.7.1	动态分配存储空间的概念	174
9.7.2	用于动态分配存储单元的函数	174
9.8	指针应用举例	176
习题 9		179
第 10 章	结构体与共用体	183
10.1	结构体类型	183
10.1.1	结构体类型的定义	183
10.1.2	结构体类型的变量	185
10.1.3	结构体类型的数组	188
10.1.4	结构体类型的指针变量	189
10.1.5	位段结构体	192
10.1.6	链表	193
10.2	共用体类型	199
10.2.1	共用体类型的定义	199
10.2.2	共用体类型的变量	200
10.3	枚举类型	203
10.3.1	枚举类型的定义	203
10.3.2	枚举类型的变量	204
10.4	用 typedef 定义类型标识符	206
习题 10		207
第 11 章	数据文件	210
11.1	文件的概念	210
11.2	文件类型的指针变量	211
11.3	文件的打开与关闭	211
11.3.1	打开文件函数 fopen()	211
11.3.2	关闭文件函数 fclose()	212
11.4	文件的读/写	213
11.4.1	字符读/写函数	213
11.4.2	格式化读/写函数	215
11.4.3	字符串读/写函数	217
11.4.4	数据块读/写函数	218
11.5	文件的定位	218
11.5.1	rewind()函数	218
11.5.2	fseek()函数	219

11.5.3	ftell()函数	219
习题 11		219
第 12 章	C 语言综合应用	222
12.1	字符屏幕与图形功能函数	222
12.1.1	常用的字符屏幕函数	222
12.1.2	常用的图形处理函数	226
12.2	程序开发的实用技术	235
12.2.1	菜单的设计方法	235
12.2.2	运行一个多文件的程序	236
12.2.3	文件之间的调用	238
12.3	综合应用举例	239
12.3.1	系统功能设计	239
12.3.2	程序模块划分与设计	239
12.3.3	程序连接与运行	250
附录 A	ASCII 字符编码表	251
附录 B	运算符的优先级和结合方向	252
附录 C	常用库函数	253
参考文献		258

计算机是硬件和软件的结合体，硬件是“躯体”，软件是“灵魂”，软件的主体是程序，计算机是在程序的控制下自动工作的。在实际应用中，有些任务可以使用现成的应用软件来完成，有些任务可能没有合适的软件供使用，这时就需要使用一种编程语言来编写程序，以完成特定的任务，这就是程序设计。学习程序设计是训练逻辑思维能力的好的方法。

本章主要介绍程序设计的基本概念和初步知识，包括数据结构与算法、程序设计方法、程序设计语言的简要介绍。

1.1 程序设计的概念

1.1.1 程序与程序设计

程序的概念是很普遍的。例如，要完成一项复杂的任务，需要事先做好计划，即将任务分解成一系列具体的工作。按照一定的顺序安排的工作序列，就是程序。例如，亲手缝制一条裤子的程序如下：

- (1) 购买布料、拉锁等物品。
- (2) 按身材量好尺寸。
- (3) 按尺寸裁剪布料。
- (4) 缝制成裤子。
- (5) 将裤子熨烫平整。

在这个程序中，涉及布料、拉锁、尺寸、裤子等对象（或说是具有一定关系的数据），以及购买、量、裁剪、缝制、熨烫等行为（或说是对数据的操作），这些行为是施加于对象之上的，有一定的顺序，要遵守一定的规则。所以说，一个程序主要描述两部分内容：对数据的描述和对操作的描述。

一个计算机程序，就是用某种计算机能识别和执行的语言，描述解决问题的方法和步骤。所以程序是计算机为完成某一任务所必须执行的一个指令序列。

例如，要将某班 40 名同学某门课程的成绩按分数从高到低的顺序输出。

分析：

(1) 数据及数据之间的关系: 每一个学号、姓名、成绩是一一对应的关系。

(2) 对数据的操作有: 输入数据、用某种方法排序、输出排序结果。

问题中的数据与数据之间的关系以及对数据的操作, 就是该问题的数据结构。对数据操作的实现步骤, 也就是为解决问题而采取的方法和步骤, 称为算法。著名计算机科学家沃思 (Niklaus Wirth) 提出一个公式:

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

这充分表明了数据结构和算法在程序中的重要地位。

程序设计的任务就是分析问题, 弄清数据与数据之间的逻辑关系以及对数据的操作 (即明确数据结构); 确定解决问题的方法和步骤 (即设计算法); 选择一种计算机语言来描述算法 (即编程); 调试并运行程序, 使之满足任务的要求, 输出正确的结果。所以说, 程序设计是利用计算机解决问题的过程。

1.1.2 程序设计的过程

程序设计通常需要如下步骤:

1. 任务分析

首先要对任务进行详细的分析, 弄清楚任务中数据与数据之间的逻辑关系, 以及具体的操作要求 (如需要输入哪些数据, 要对数据进行哪些处理, 要求输出哪些数据等), 即明确数据结构, 也就是弄清楚要计算机 “做什么”。

2. 算法设计

算法是对要解决问题的方法和步骤的描述, 是对问题处理过程的进一步细化。解决一个问题, 可能有多种算法, 不同的算法可能效率不同, 应该通过分析和比较挑选其中最优的算法。算法设计就是明确要计算机 “怎么做”。

3. 程序编制

首先根据算法, 选择一种程序设计语言, 写出源程序, 这个过程称为编码。然后将编写好的源程序通过编辑器 (可以用语言自带的编辑器, 也可以使用 Windows 的记事本) 输入计算机并以源程序的形式保存, 这个过程称为编辑。源程序必须是纯文本文件, 不能带有格式, 若用字处理软件 (如 Word) 编辑, 则必须另存为纯文本文件。

4. 调试运行

计算机不能直接执行源程序, 需要通过编译和连接, 生成计算机能够执行的可执行文件。

(1) 编译: 通过编译器将源程序翻译成目标程序。翻译时编译器对源程序进行语法检查, 给出编译信息。若有错, 通过编辑器修改后再编译, 直到编译成功, 生成目标程序。

(2) 连接: 将目标程序和程序中所需的目标程序模块 (如调用的标准函数、执行的输入/输出模块等) 连接后生成可执行文件。只有可执行文件才可以在操作系统上运行。

(3) 运行并分析结果: 如需要输入数据, 应设计能涵盖各种情况的测试数据, 然后运行程序, 检查结果是否符合问题要求, 是否正确。另外, 即使程序能正常运行并得到了运行结果, 也可能存在逻辑错误。这类错误可能是设计算法时的错误, 也可能是编写程序时出现疏忽所致, 或者是测试数据不合理、不全面, 而计算机无法检查出这些错误。存在逻辑错误时, 如果是算法有错,

应先修改算法，再修改程序；如果算法正确而程序编写错误，则应修改程序。

此过程称为调试程序，常常需要多次反复，需要有耐心、有信心、有决心，而且还要细心。这个过程也是不断积累程序调试经验的过程。

5. 编写程序文档

程序文档就是程序的使用说明书和技术说明书，它记录了程序设计的全过程，可以保证程序的可读性和可维护性。对于较小的程序，文档显得并不重要，在程序中加些注释，让读程序的人理解就足够了。但对于一个需要多人合作，并且开发和维护时间较长的软件来说，程序文档是至关重要的。尤其是软件的二次开发，就更离不开文档了。所以有人说：

软件=程序+文档

通常情况下，程序文档主要分为两部分：程序技术说明书和程序使用说明书。

程序设计的过程环环相扣，每一步都必须认真去做。然而，初学者往往是问题还没有分析清楚就急于编程，常常是思路混乱，甚至一见运行有结果就以为完成任务，对结果不分析和验证，所以很难得到预想的结果。因此，养成良好的程序设计习惯和严谨的科学作风是非常重要的。

1.2 数据结构与算法

计算机存取和处理的对象是数据，且数据之间都具有一定的逻辑关系。要编写一个程序，必须分析清楚待处理数据之间的关系，以及对各处理数据所进行的操作。程序是采用一种计算机语言依据算法编写出来的代码，而算法是依据数据结构设计的。

1.2.1 数据结构

什么是数据结构？在任何问题中，各处理对象都不是孤立存在的，它们之间存在着一定的关系，并要进行一些处理操作。在程序设计中，要处理的对象及其相互关系被抽象为数据。所以简单地说，数据结构是研究非数值计算的程序设计问题中数据和数据之间的关系以及对数据的操作的一门学科。它包括以下3个方面的内容：

- (1) 数据的逻辑结构：描述数据与数据之间的逻辑关系，是独立于计算机的。
- (2) 数据的存储结构：描述数据和数据之间的关系在计算机中存储的方式。
- (3) 数据的运算集合：即对数据进行的所有操作。

数据运算的集合是定义在数据的逻辑结构之上的，即逻辑结构一旦确定，运算的集合就明确了。而数据运算必须在数据的存储结构确定之后才能实现，只有确定了如何存储数据，才能设计算法。

1.2.2 算法

算法是对操作的描述，是为了解决一个问题而采取的方法和步骤。

1. 算法的基本结构

算法定义一个操作序列，描述怎样从给定的数据经过有限步骤的处理后产生所求的输出结果。其基本结构一般包括以下3个部分：

- (1) 初始化（包括输入原始数据和为数据处理所做的准备）。

(2) 数据处理 (实现具体的功能)。

(3) 输出处理结果。

【例 1.1】输入两个整数, 输出其中较大的数。

算法如下:

(1) 输入整型数据 a 和 b;

(2) 将 a 和 b 中较大的数放入 max 中;

(3) 输出 max。

【例 1.2】求 $\text{sum} = \sum_{i=1}^{100} i$ 。

算法如下:

(1) 初始化: $\text{sum} \leftarrow 0$; $i \leftarrow 1$;

(2) 当 $i \leq 100$ 时重复做 $\text{sum} \leftarrow \text{sum} + i$; $i \leftarrow i + 1$;

(3) 输出 sum。

2. 算法的特性

(1) 输入: 有 0 个或多个输入。

(2) 有穷性: 具有有限的操作步骤, 不能是死循环。

(3) 确定性: 每一个步骤必须精确定义, 不能有歧义。

(4) 可行性: 每一个步骤都能在有限的时间内完成。

(5) 输出: 有一个或多个输出。

3. 算法的表示

描述算法有多种方法。常用的有自然语言、流程图、N-S 图、伪代码、计算机语言等。

(1) 自然语言: 用人们交流时使用的语言描述算法。这种描述方法简单通俗, 但不够严格, 容易产生歧义性, 在表达算法的逻辑流程时不够直观。

(2) 流程图: 又称框图。它是用一些图框、线条以及文字说明来形象、直观地表达算法。美国国家标准化学会 (American National Standard Institute, ANSI) 规定了一些流程图符号。

- 开始/结束框: 用椭圆形表示。
- 输入/输出框: 用平行四边形表示。
- 判断框: 用菱形表示。
- 处理框: 用矩形表示。
- 流向线: 用带有箭头的线段表示。
- 连接点: 用圆形表示。

流程图形象、直观, 便于交流, 使用广泛。如描述例 1.1、例 1.2 算法的流程图如图 1-1 所示。

(3) N-S 图: 为简化控制流向, 美国学者 I.Nassi 和 B.Shneiderman 提出了新的流程图形式, 并以他俩姓名的第一个字母命名。在 N-S 图中去掉了流程图中带箭头的流向线, 用一个大矩形框表示全部算法, 大矩形可分隔成若干个水平矩形区, 每个矩形区表示一个基本结构, 适于结构化程序设计。如描述例 1.1、例 1.2 算法的 N-S 图如图 1-2 所示。

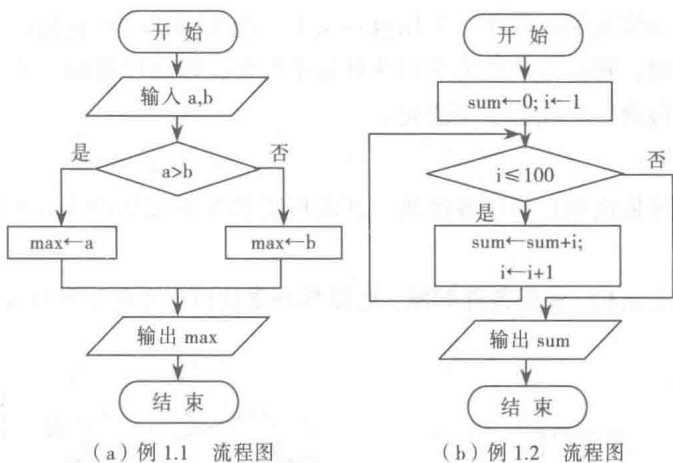


图 1-1 流程图

(4) 伪代码：伪代码介于自然语言和计算机语言之间，用文字和符号来描述算法。克服了用自然语言描述算法容易产生歧义性，表达逻辑流程不够直观和绘制流程图较费时的缺陷。伪代码不能被计算机识别，但接近于某种编程语言(如类 C、类 Pascal 等类语言)，便于转换成程序。

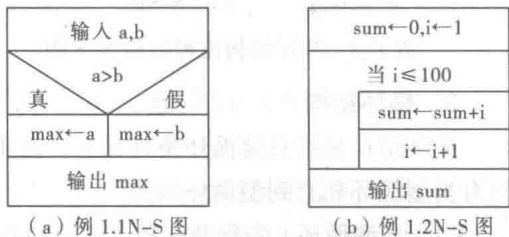


图 1-2 N-S 图

(5) 计算机语言：直接用计算机语言编写程序。计算机无法识别用自然语言、流程图、N-S 图、伪代码描述的算法。用这些方法描述算法只是为了帮助人们表示和理解算法，最终都必须转换成用某种计算机语言编写的程序才能让计算机运行。

对于较为复杂的问题，或是程序设计的初学者，最好先用自然语言、流程图、N-S 图或伪代码中的一种方法描述算法，可从粗到细，逐渐精确。最后依据算法的描述用计算机语言编写程序。只有比较简单的问题，才适合直接用计算机语言编写程序。

1.3 程序设计方法

早期的程序设计以运行速度快、占用内存少为主要标准，由程序设计人员随意设计，但存在程序调试困难、不易维护等问题，限制了代码的优化。随着计算机运算速度大大提高和存储容量不断扩大，结构清晰、易于阅读和理解、便于验证其正确性成为良好程序的评价标准，这也推动了程序设计方法的进步。最常用的程序设计方法有结构化程序设计方法和面向对象的程序设计方法。

1.3.1 结构化程序设计方法

结构化程序设计也称为面向过程的程序设计。结构化程序设计语言有 FORTRAN、BASIC、Pascal、C 语言等。

结构化程序设计方法的基本思想是，把一个复杂的问题分解成若干个功能独立的模块，分而治之。具体地说就是：

(1) 在软件设计和实现的过程中，采用自顶向下、逐步细化的模块化设计原则。

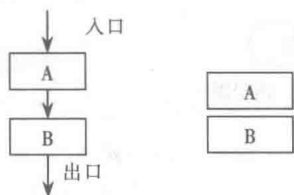
(2) 在代码编写时，每一个模块内采用 3 种基本结构，即顺序结构、选择结构和循环结构。每个基本结构内可以包含一个或若干个语句。

1. 顺序结构

顺序结构指按顺序依次执行程序各模块，其流程图和 N-S 图如图 1-3 所示。

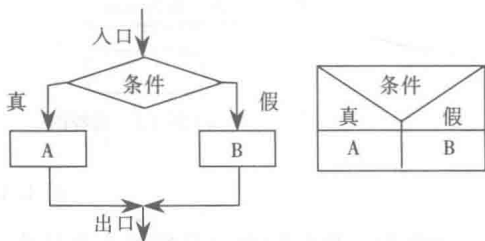
2. 选择结构

选择结构又称分支结构。根据条件判断，选择某分支执行，其流程图和 N-S 图如图 1-4 所示。



(a) 流程图 (b) N-S 图

图 1-3 顺序结构流程图和 N-S 图



(a) 流程图 (b) N-S 图

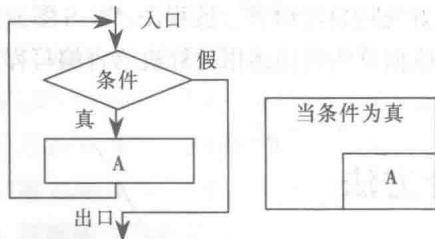
图 1-4 选择结构流程图和 N-S 图

3. 循环结构

循环结构是指只要循环条件成立，就重复执行模块 A（模块 A 通常被称作循环体）。循环结构有当型循环和直到型循环两种。

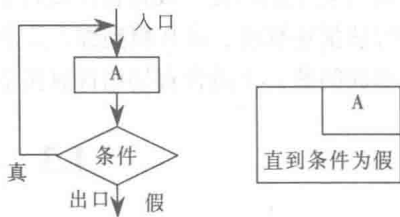
(1) 当型循环（先判断循环条件后执行）：当条件为真时重复执行模块 A，条件为假时循环结束，其流程图和 N-S 图如图 1-5 所示。

(2) 直到型循环（先执行后判断循环条件）：重复执行模块 A，直到条件为假时，循环结束，其流程图和 N-S 图如图 1-6 所示。



(a) 流程图 (b) N-S 图

图 1-5 当型循环结构流程图和 N-S 图



(a) 流程图 (b) N-S 图

图 1-6 直到型循环结构流程图和 N-S 图

3 种基本结构的共同特点是：单入口，单出口；结构中没有永远执行不到的语句；不构成死循环。

1.3.2 面向对象的程序设计方法

由于结构化程序设计方法是面向过程的，过分强调求解过程的细节，编写的程序不易重复使用，从而诞生了面向对象的程序设计方法和面向对象的程序设计语言。面向对象的程序设计语言有 C++、Java 等。

面向对象的程序设计语言引入了对象、消息、类、继承、封装、抽象、多态性等概念和机制。面向对象的程序设计不是将问题分解成过程，而是分解成对象。对象是现实世界中独立存在、可