

21世纪高等学校计算机基础教育系列教材

21 SHIJI GAODENG XUOXIAO JISUANJI JICHU JIAOYU XILIE JIAOCAI

# C语言程序设计

## 教程

■ 李玲 桂玮珍  
刘莲英 编著



人民邮电出版社  
POSTS & TELECOM PRESS

21 世纪高等学校计算机基础教育系列教材

# C 语言程序设计教程

李 玲 桂玮珍 刘莲英 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

C 语言程序设计教程 / 李玲, 桂玮珍, 刘莲英编著. —北京: 人民邮电出版社, 2005.2  
(21 世纪高等学校计算机基础教育系列教材)

ISBN 7-115-12838-3

I. C... II. ①李... ②桂... ③刘... 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 (2005) 第 005803 号

## 内 容 提 要

“C 语言程序设计”是高等学校普遍开设的一门计算机基础课程。本书根据 C 语言课程教学大纲的要求, 在对 C 语言的数据类型、运算符与表达式等基本知识进行讨论的基础上, 介绍结构化程序设计的思想和方法, 同时对构造类型数据的应用及变量的存储类型和指针等重点和难点知识也进行详细的介绍。每章后均附有小结和多种类型的习题, 能帮助读者更好地消化、理解有关的知识。为了配合本书的学习, 作者还编写了与本书配套的《C 语言程序设计教程习题解答与实验指导》, 供读者参考使用。

本书可作为高等学校本科生“C 语言程序设计”课程的教材, 也可供 C 语言自学者或参加各种 C 语言考试的读者学习使用。

21 世纪高等学校计算机基础教育系列教材

### C 语言程序设计教程

◆ 编 著 李 玲 桂玮珍 刘莲英  
责任编辑 郭 晶

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线: 010-67129259  
北京隆昌伟业印刷有限公司印刷  
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16  
印张: 19.5  
字数: 499 千字 2005 年 2 月第 1 版  
印数: 1-5 000 册 2005 年 2 月北京第 1 次印刷

ISBN 7-115-12838-3/TP · 4325

定价: 25.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

## 编者的话

随着计算机技术的不断发展，计算机应用的迅速普及推广，各种软件像雨后春笋般大量涌现出来。不论是计算机上必不可少的操作系统软件还是为解决某个领域中的具体问题而设计的应用软件，都是使用某种计算机语言编制的程序。计算机语言有很多种，例如 BASIC、FORTRAN、Pascal、COBOL 和 C 语言等。由于 C 语言功能丰富、应用面广、目标程序效率高、可移植性好，既能编制系统软件也能编制应用软件，因此近年来 C 语言已经成为国内外广泛流行的一种通用的计算机程序设计语言。

目前大部分高等学校的计算机基础课都选择了 C 语言作为本科生的程序设计必修课程。《C 语言程序设计教程》及其配套的《C 语言程序设计教程习题解答与实验指导》是作者根据多年从事 C 语言课程教学的经验和体会而精心编写的。本书按照 C 语言课程教学大纲的要求，在内容安排上遵循深入浅出、循序渐进的原则，通过全面、系统地讨论 C 语言基本语法和组成，介绍程序设计的基本思想和方法，培养读者的计算机编程能力。全书结构紧凑、概念准确、重点突出、例题丰富、叙述流畅，阐述问题思路清晰。本书重视对 C 语言重点和难点的阐述，通过典型例题，对常用算法和学习中的重点、难点进行分析归纳，便于读者理解掌握。各章后配有多种形式的习题，并在配套的《C 语言程序设计教程习题解答与实验指导》教材中给出习题参考答案及编程题的算法和参考程序。通过练习题和上机实验，使读者在掌握基本算法的基础上，能够逐步提高阅读程序和编写程序的能力，掌握调试程序的基本方法和技能，为将来自学其他计算机语言及编制大型程序打下良好的基础。

全书共分为 10 章，主要内容包括：预备知识、C 语言基础知识、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数与预处理命令、指针、结构体与共用体和文件等，每章后附有小结及习题。本书以大学“C 语言程序设计”课程教学大纲为基础，参考了《全国计算机等级考试（二级）C 程序设计考试大纲》和《北京地区普通高等学校非计算机专业学生计算机应用水平测试考试大纲》中有关 C 语言的要求。书中所有程序均在 Turbo C 2.0 环境下调试通过。

本书第 1~5 章由李玲编写，第 6~7 章由刘莲英编写，第 8~10 章由桂玮珍编写。在本书编写过程中，北京科技大学计算机基础课程教学组的各位任课老师给予了多方面的支持和帮助，在此表示深切的感谢。

由于时间仓促，作者水平有限，书中难免有错误或不严谨之处，欢迎同行和读者批评、指正。

编者

2005 年 1 月

# 目 录

第 1 章 预备知识	1
1.1 程序设计与算法概述	1
1.1.1 计算机语言与程序设计	1
1.1.2 算法	4
1.1.3 结构化程序设计	6
1.2 C 语言概述	8
1.2.1 C 语言的发展	8
1.2.2 C 语言的特点	8
1.2.3 C 程序的基本结构	10
1.2.4 C 程序的执行过程	13
1.3 C 程序的符号系统	13
1.3.1 基本字符	14
1.3.2 标识符	14
1.4 小结	15
习题	16
第 2 章 C 语言基础	18
2.1 基本数据类型	18
2.1.1 整型数据	19
2.1.2 实型数据	21
2.1.3 字符型数据	22
2.1.4 空类型	23
2.2 常量、变量与标准库函数	23
2.2.1 常量	23
2.2.2 变量	27
2.2.3 标准库函数	29
2.3 运算符和表达式	31
2.3.1 算术运算符和算术表达式	31
2.3.2 赋值运算符和赋值表达式	35
2.3.3 关系运算符和关系表达式	36
2.3.4 逻辑运算符和逻辑表达式	37
2.3.5 位运算符和位运算表达式	39
2.3.6 条件运算符和条件表达式	42

2.3.7 逗号运算符和逗号表达式	43
2.3.8 其他运算符及其运算	44
2.4 各种运算中数据类型的转换	44
2.4.1 数据类型的自动转换	44
2.4.2 数据类型的强制转换	46
2.5 小结	47
习题	48
<b>第 3 章 顺序结构程序设计</b>	<b>51</b>
3.1 C 语言的语句类型	51
3.1.1 说明性语句	51
3.1.2 表达式语句	51
3.1.3 控制语句	52
3.1.4 复合语句	53
3.2 数据输出	53
3.2.1 字符输出函数	53
3.2.2 格式输出函数	54
3.3 数据输入	60
3.3.1 字符输入函数	60
3.3.2 格式输入函数	61
3.4 顺序结构程序设计	66
3.4.1 顺序结构程序的一般算法描述	66
3.4.2 顺序结构程序设计举例	66
3.5 小结	68
习题	69
<b>第 4 章 选择结构程序设计</b>	<b>73</b>
4.1 if 语句	73
4.1.1 if 语句的一般形式	73
4.1.2 if 语句的嵌套	77
4.2 switch 语句	80
4.2.1 switch 语句的一般形式	80
4.2.2 break 语句	81
4.3 选择结构程序设计举例	82
4.4 小结	87
习题	88
<b>第 5 章 循环结构程序设计</b>	<b>93</b>
5.1 while 循环控制	93

5.1.1	while 语句	93
5.1.2	while 语句的简单应用	95
5.2	do-while 循环控制	97
5.2.1	do-while 语句	97
5.2.2	do-while 语句的简单应用	98
5.3	for 循环控制	99
5.3.1	for 语句	99
5.3.2	for 语句的简单应用	101
5.4	用 goto 语句实现循环控制	102
5.4.1	goto 语句	102
5.4.2	goto 语句的简单应用	103
5.5	循环结构中的跳转语句	104
5.5.1	continue 语句及应用	104
5.5.2	循环结构中 break 语句的应用	106
5.6	循环的嵌套	106
5.6.1	循环嵌套的基本形式	106
5.6.2	多重循环程序举例	107
5.7	循环结构程序设计举例	109
5.8	小结	113
	习题	114
<b>第 6 章</b>	<b>数组</b>	<b>121</b>
6.1	一维数组	121
6.1.1	一维数组的定义	121
6.1.2	一维数组元素的引用	122
6.1.3	一维数组的初始化	123
6.1.4	一维数组应用举例	123
6.2	二维数组	129
6.2.1	二维数组的定义	129
6.2.2	二维数组元素的引用	129
6.2.3	二维数组的初始化	130
6.2.4	二维数组应用举例	131
6.3	字符数组与字符串	132
6.3.1	基本概念	132
6.3.2	字符数组的定义	133
6.3.3	字符数组的初始化	133
6.3.4	字符数组的引用	134
6.3.5	字符串处理函数	135
6.3.6	字符数组应用举例	139

6.4 小结	141
习题	142
<b>第 7 章 函数与预处理命令</b>	<b>147</b>
7.1 概述	147
7.2 函数的定义与调用	149
7.2.1 函数的定义	149
7.2.2 函数的调用	151
7.2.3 函数参数与函数的返回值	152
7.2.4 对被调函数的声明和函数原型	155
7.3 数组作函数参数	157
7.3.1 一维数组元素作函数参数	157
7.3.2 一维数组名作函数参数	158
7.3.3 多维数组作函数参数	160
7.4 函数的嵌套调用和递归调用	162
7.4.1 函数的嵌套调用	162
7.4.2 函数的递归调用	162
7.5 局部变量和全局变量及其作用域	167
7.5.1 变量的作用域	167
7.5.2 局部变量及其作用域	167
7.5.3 全局变量及其作用域	168
7.6 变量的存储类别及变量的生存期	169
7.6.1 变量的生存期与变量的存储分类	169
7.6.2 变量的存储类别	169
7.6.3 归纳变量的分类	175
7.7 函数的存储分类	176
7.7.1 外部函数	176
7.7.2 静态函数	176
7.8 编译预处理	177
7.8.1 不带参数的宏定义	178
7.8.2 带参数的宏定义	179
7.8.3 文件包含	180
7.9 多文件程序的调试方法	181
7.10 小结	183
习题	184
<b>第 8 章 指针</b>	<b>190</b>
8.1 指针和指针变量	190
8.1.1 指针和指针变量的概念	190



8.1.2 指针变量的定义	191
8.1.3 指针的基本运算	192
8.2 指针与数组	196
8.2.1 指向数组的指针变量	196
8.2.2 指向字符串的指针变量	202
8.2.3 指针数组	205
8.2.4 多级指针	208
8.3 指针与函数	210
8.3.1 指针变量作函数参数	210
8.3.2 带参数的主函数	217
8.3.3 指针型函数	219
8.3.4 指向函数的指针变量	220
8.4 小结	225
习题	226
<b>第9章 结构体与共用体</b>	<b>231</b>
9.1 结构体	231
9.1.1 结构体与结构体类型的定义	231
9.1.2 结构体变量的定义与初始化	233
9.1.3 结构体的指针	237
9.1.4 函数间结构体数据的传递	239
9.1.5 结构体的应用——链表	241
9.2 共用体	252
9.2.1 共用体与共用体类型定义	252
9.2.2 共用体变量的定义与初始化	253
9.3 枚举类型与类型命名	257
9.3.1 枚举类型	257
9.3.2 类型的重新命名	260
9.4 小结	262
习题	263
<b>第10章 文件</b>	<b>271</b>
10.1 文件的概念	271
10.1.1 文件和文件指针	271
10.1.2 文件的打开与关闭	273
10.2 文本文件的读写	274
10.2.1 文件的字符输入输出函数	275
10.2.2 文件结束测试函数 feof()	276
10.2.3 文件的字符串输入输出函数	277

10.2.4 文件的格式输入输出函数.....	279
10.3 二进制文件的读写.....	280
10.3.1 文件的字输入输出函数.....	280
10.3.2 文件的数据块输入输出函数.....	282
10.3.3 文件读写指针定位函数.....	284
10.4 小结.....	286
习题.....	287
附录 A C 语言常用库函数.....	293
附录 B C 语言中运算符的优先级和结合性.....	297
附录 C 基本 ASCII 表.....	298
参考文献.....	299

# 第 1 章

## 预备知识

随着科学技术的发展和人类社会的进步，计算机得到了越来越普遍的应用。众所周知，计算机是一种能够按照事先编排好的操作命令自动地、高速地进行运算处理和逻辑判断的电子设备。要使用计算机解决某一应用问题，首先必须将复杂的问题分解为计算机可以执行的若干简单的基本运算和基本操作，然后再使用某种计算机语言的语句描述这些基本运算和基本操作，得到的语句序列就是我们所说的程序，而书写程序的计算机语言就称为程序设计语言。C 语言是目前广泛应用的一种计算机程序设计语言。

本章首先概要介绍计算机语言、程序设计和算法的有关概念，然后着重讨论分析 C 语言程序的基本结构和 C 语言使用的基本字符集，为进一步学习 C 语言程序设计打下基础。

### 1.1 程序设计与算法概述

#### 1.1.1 计算机语言与程序设计

##### 1. 计算机语言

语言是人与人之间进行交流的工具，而计算机语言则用于人与计算机之间进行对话。按照计算机语言的发展和特点，可以将其分为机器语言、汇编语言和高级语言。

##### (1) 机器语言

计算机只能识别和执行由“0”和“1”构成的二进制形式的操作命令，这种操作命令称为计算机指令。一条指令通常由操作码和操作对象两部分组成，其中操作码指出要进行的操作类型，操作对象可以是一个具体的操作数，也可以是存放操作数的存储单元地址。

某一种类型的计算机能够识别和执行的指令集合称为该计算机的指令系统，或者称为它的机器语言。用这种语言编写的程序，称为机器语言程序，计算机可以直接识别并执行。

机器语言是设计计算机硬件时所确定的，由一串“0”和“1”数码构

成，它与人们日常使用的自然语言和数学语言有很大的差别，对计算机硬件不熟悉的人几乎不可能直接使用它来编写程序。此外，不同类型计算机的机器语言是不通用的，在一台计算机上编写的机器语言程序通常不能直接在另一种类型的计算机上使用。这样就给计算机的应用推广带来很大的不便。

### (2) 汇编语言

用特定的助记符号来表示二进制形式的计算机指令，例如，用符号 ADD 表示加法，用 SUB 表示减法，用 AX、BX 表示操作的寄存器，用 BUFFER、DATAS 等符号表示数据存储单元地址等，这样就构成了一种与机器语言对应的符号语言，这种符号化的语言就称为汇编语言。

汇编语言的操作命令与机器语言指令一一对应，即用一条汇编语言指令代替一条机器指令。例如，ADD AX, BX 表示将 AX 和 BX 寄存器中的数相加，结果存放到 AX 寄存器中。显然这种表示方法比二进制机器语言要容易理解，易于编程。

用汇编语言编写的程序称为汇编语言源程序。这种程序计算机是不能直接识别和执行的，必须通过一个专门的程序将它翻译成机器语言程序后才能执行。这个翻译过程称为汇编，能够完成翻译任务的程序称为汇编语言处理程序，简称汇编程序。经过汇编后得到的机器语言程序称为目标程序。这种程序一般还需要通过一个称为连接程序的软件进行连接，最后生成可执行程序。整个汇编过程如图 1.1 所示。

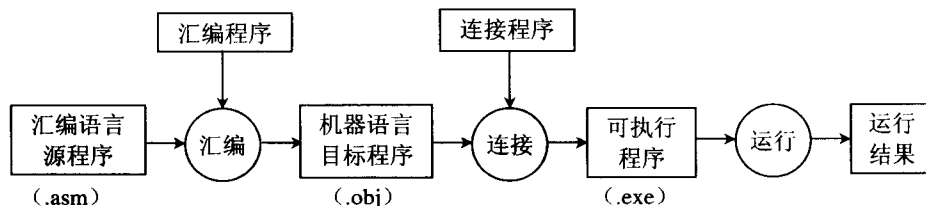


图 1.1 汇编过程

汇编语言保留了机器语言执行速度快、占存储容量小的优点，在一定程度上克服了机器语言难于阅读和编码的不足。但是它和机器语言一样都是属于面向机器的“低级语言”，编程时要依赖于具体的计算机，因此汇编语言也不具有通用性。

### (3) 高级语言

高级语言是一种接近于自然语言和数学语言的计算机语言，常用的有 C 语言、BASIC 语言、FORTRAN 语言和 Pascal 语言等。每种高级语言均规定了专门的词法规则和语法规则，使用特定的英文单词和数学符号编写程序。例如，C 语言中用 float 表示单精度实型数据类型，用 continue 表示终止本次循环，用 a=3+b 表示变量 b 的值加 3 后赋给变量 a 等。

用高级语言编写的程序称为高级语言源程序。计算机也不能直接识别和执行这种程序，必须经过翻译，才能将其转换成机器语言程序执行。翻译的方法有两种，一种是编译方式，另一种是解释方式。

编译的方法与汇编过程类似，使用一种称为编译程序的软件，将高级语言源程序整个翻译成机器语言目标程序，再经过连接生成可执行的程序。编译过程如图 1.2 所示。

解释的方法是通过一种称为解释程序的软件对源程序逐句翻译，逐句执行，即采用边解释边执行的方式运行程序，解释过程中不生成目标程序。解释过程如图 1.3 所示。

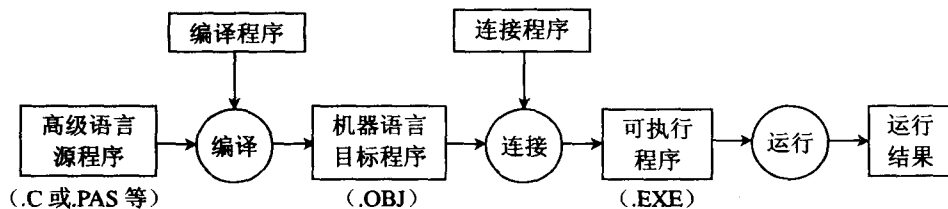


图 1.2 编译过程

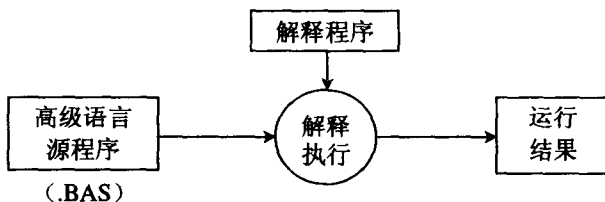


图 1.3 解释过程

高级语言是一种面向过程的计算机语言。编写程序时，使用语句描述解决实际应用问题的过程。一个语句往往相当于许多条计算机指令，描述方式也与我们日常处理问题的逻辑思维习惯相接近，因此高级语言易于学习、易于阅读理解。此外由于不涉及计算机指令，使得高级语言摆脱了对计算机硬件的依赖，编程时不需要考虑机器的内部结构和存储单元的分配，只要安装了某一种高级语言的编译程序（或解释程序），其源程序就可以不经修改，或只需要很少量的修改就可以在其他类型的计算机上运行，因此高级语言具有良好的可移植性。

随着计算机技术的发展，目前又出现了各种面向对象的计算机语言，也称为第四代语言。其特点是将实际问题中处理的事物抽象成对象和对象之间的关系，而与该事物有关的数据及对数据的具体操作都与对象封装在一起。编程时只要考虑如何认识问题中的对象和描述对象，而不必具体说明对象中的数据操作，只要根据对对象的描述，计算机就可以自动完成相应的数据处理。这种方法简化了程序设计过程，更加符合人类对客观事物的认识过程及思维方式。典型的面向对象的计算机语言包括 SQL 语言、Visual Basic、Visual C++ 和 Delphi 等。

## 2. 程序设计

使用计算机解决一个实际应用问题时，通常需要经过这样的处理步骤：首先要明确需要解决的问题是什么，即提出问题；其次要分析问题中涉及了哪些数据，如何在计算机中进行表示，即描述数据结构；同时还要将复杂的问题分解为计算机可以完成的若干操作步骤，即确定算法；然后用选定的某种计算机语言描述数据结构，并根据算法编写程序；编好的源程序输入计算机后，往往需要进行反复调试，修正其中的语法错误和逻辑错误，直至得到正确的运算结果。

如图 1.4 所示，从提出问题、确定数据结构和算法，并据此编写程序，直到程序调试通过的整个过程就称为程序设计。

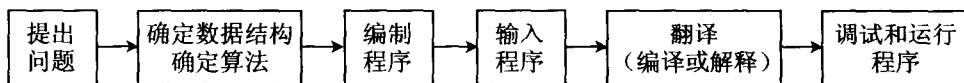


图 1.4 程序设计过程

从上述过程中可以看出，使用计算机解题必须先编写程序，而编程的依据是算法。在学习程序设计时，首先要理清思路，确定算法，再用某种计算机语言去实现算法。语言是编程的工具，只懂得语言的规则和语法格式，并不能保证一定就可以编制出正确、有效和高质量的程序。而如果算法正确，则将它用任何一种高级语言实现并不困难。因此有必要了解有关算法的一些基本知识。

## 1.1.2 算法

### 1. 算法的概念

为解决一个特定的问题而采取的方法和步骤称为算法。做任何事情实际上都有确定算法的问题。例如外出旅游，要先确定旅游地点、线路，然后预定车票、旅店，最后乘车前往游览。这些步骤是按照一定顺序进行的，一步步实施，就可以达到旅游的目的。对同一个问题，可以采用不同的算法。例如去大连旅游，既可以乘火车，也可以乘飞机或轮船前往，最终都能够到达大连，但是不同的方式所需要的时间、费用和途经线路都是不一样的。

同样，使用计算机解决某个问题，也必须先确定解题的具体步骤，即确定算法，将问题分解为若干个计算机可以顺序执行的基本操作，然后才能编写程序。对同一个问题也可以采用不同的算法。例如计算 3 个半径分别为  $R_1$ 、 $R_2$ 、 $R_3$  的圆面积，可以按公式  $R_1^2 \times \pi + R_2^2 \times \pi + R_3^2 \times \pi$  计算；也可以按  $(R_1^2 + R_2^2 + R_3^2) \times \pi$  计算，显然两种算法的操作步骤是不同的。一般来说用计算机解题时应该选择步骤较少，相对简单的算法。

一个正确的算法应该具备以下主要特点。

#### (1) 有穷性

有穷性即算法必须由有限个操作步骤组成，不能是无限的。如果一个算法能够解决问题，但是所需要的时间超出合理的限度，也不能称其为算法。例如，可以用算法描述计算 100 个、1 000 个自然数的和，但是却不能描述计算所有自然数的和。

#### (2) 确定性

确定性即算法中每一个步骤的含义都应该明确无误，不能存在歧异性。例如“输出  $x$  加  $y$  乘以 3 的值”就是一个二义性的表述，因为无法确定到底是要输出  $(x+y) \times 3$  的值，还是要输出  $x + (y \times 3)$  的值。在算法中必须避免这种模棱两可的描述。

#### (3) 有效性

有效性即算法的每个步骤都应该能够有效地执行，并能够在执行算法后得到确定的结果。例如要求解  $x$  的平方根，当  $x$  的值是一个负数时，运算就不能有效进行。算法中必须考虑到这种情况，只有当  $x$  是一个大于或等于 0 的数时才能够进行相应的处理。

#### (4) 有 0 个到多个输入，有 1 个或多个输出

输入是指算法执行过程中需要用户提供的信息。这些数据信息既可以在程序运行过程中输入，也可以直接在程序中指定。此外，算法中必须至少有 1 个以上的输出，没有输出结果的算法没有任何意义。

### 2. 算法的描述方式

算法可以有多种描述方法，包括自然语言、流程图和 N-S 结构图等。无论用什么方式，

都需要将操作步骤和操作简单内容明确无误地描述出来。

### (1) 用自然语言描述算法

这是一种采用日常人们交流用的语言来描述解题步骤的方法。例如要判断输入的两个数据的大小，并输出其中的大数，可以用如下的自然语言描述算法。

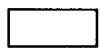
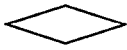
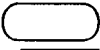
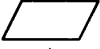

第一步，输入  $x$  和  $y$  的值。

第二步，比较  $x$  和  $y$  的值，如果  $x$  大于  $y$ ，就输出  $x$  的值，否则就输出  $y$  的值。

这种描述方式易于理解，但在描述复杂问题的解题步骤时算法冗长，不够精确，难于描述复杂算法。

### (2) 用流程图描述算法

这是一种采用框图符号来描述解题步骤的方法，得到的算法流程图又称为程序框图。在流程图中使用的主要符号标识包括以下几种。

- |   |   |                    |
|---|---|--------------------|
| ① |  | 处理框：表示基本操作处理。      |
| ② |  | 判断框：表示根据条件进行操作处理。  |
| ③ |  | 起止框：表示程序的开始或结束。    |
| ④ |  | 输入输出框：表示输入数据或输出结果。 |
| ⑤ |  | 流程线：表示程序的执行流向。     |

例如，判断两个输入数据的大小，并输出其中的大数，其算法可以用如图 1.5 所示的流程图描述。图中 Y 表示判断框中条件成立时程序的执行流向，N 表示条件不成立时的程序流向。

用流程图描述算法比用自然语言描述更直观清晰，易于理解。但是流程图中使用的符号标识不够严格、规范，对流程线的使用也没有严格的限制，使得其流向具有任意性。当处理的问题比较复杂时，流程图难以阅读，难以实现，使程序质量受到一定影响。

要解决这些问题，除了采用规范的图形符号描述外，通常对流程线的流向也应该进行严格控制，使得算法中各个步骤或步骤组合都是按一定的顺序进行的。也就是说要求每个步骤或步骤组合都只能有一个入口和一个出口，并从顶部进入，从底部流出。按照这样的设计思想构造的算法就比较规范，易于阅读交流，也易于实现。

### (3) 用 N-S 结构图描述算法

目前还可以使用一种称为 N-S 结构图的方法进行算法描述。这种方法取消了流程线，使用的基本元素是方框，因此也称为方块图或者盒图。

图 1.6 给出了不同形式的基本元素框，分别用来表示算法中顺序、选择和循环等不同的算法结构。框中的 P 为判断条件，T 表示条件成立，F 表示条件不成立。每一种框都是从顶

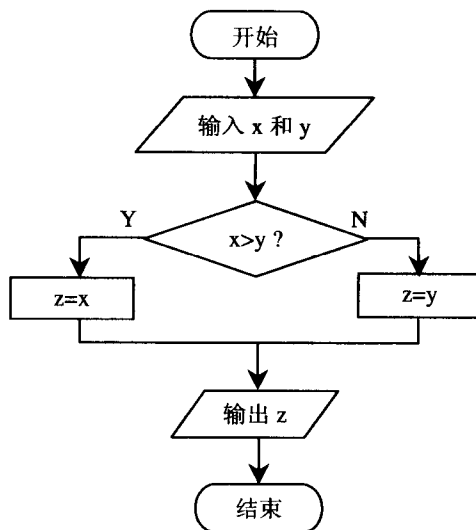


图 1.5 用流程图描述算法

部进入，从底部流出。

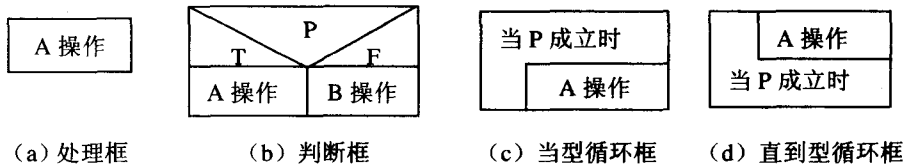


图 1.6 N-S 结构图中使用的标识

构造算法时，将一个方框的底和另一个方框的顶连接起来，就构成它的 N-S 结构图。例如判断两个输入数据的大小，并输出其中的大数，用 N-S 结构图描述的算法如图 1.7 所示。

用 N-S 结构图描述算法直观简捷，结构化程度高，但是开始接触时会感觉有些特别，不太容易辨认。应该在学习中多看多练习，尽快熟悉并学会使用。

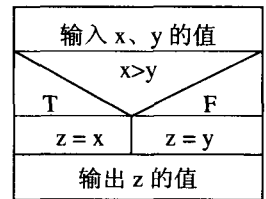


图 1.7 用 N-S 结构图描述的算法

### 1.1.3 结构化程序设计

#### 1. 程序的三种基本结构

在流程图或者 N-S 结构图的基础上，就可以使用选定的计算机语言来描述算法，即编写程序。已经证明，任何复杂的问题都可以用顺序、选择和循环三种基本算法结构来描述，因此用计算机语句编写的程序也包含三种基本结构。

##### (1) 顺序结构

顺序结构程序是按照语句的书写顺序依次执行各语句序列。图 1.8 给出顺序结构程序的算法流程图和 N-S 结构图。

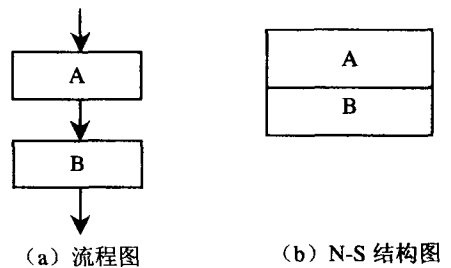


图 1.8 顺序结构程序算法

图中 A 框和 B 框表示基本的操作处理，可以是一条语言也可以是多条语句。此含义可推广到下面两种结构中。程序运行时会在执行完 A 框操作后，顺序执行 B 框操作，严格按照语句的书写顺序执行。这是一种最基本的程序结构。

##### (2) 选择结构

选择结构程序是按照条件判断选择执行某段语句序列。它的算法流程图和 N-S 结构图如图 1.9 所示。

图 1.9 中 P 为判断条件，A 框和 B 框为两个选择执行的语句序列。程序运行时究竟会选择 A 框和 B 框中的哪一个分支，则取决于条件 P 的测试结果。当条件 P 成立时执行 A 框操作，反之则执行 B 框操作。

需要指出的是，在选择结构程序中 A 框和 B 框的操作只能二选一。即执行了 A 框操作，就不会再执行 B 框操作；而执行了 B 框操作，也就不会再执行 A 框操作。无论是执行了 A 框操作还是执行了 B 框操作，之后都会向下顺序执行后续的操作。



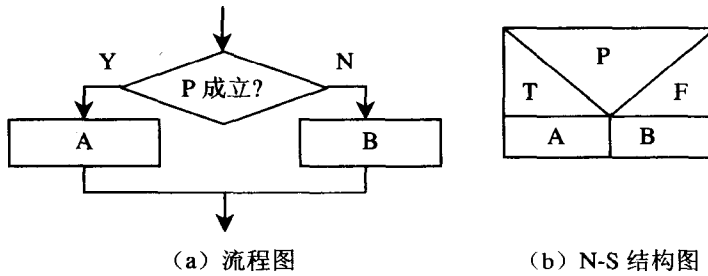


图 1.9 选择结构程序算法

(3) 循环结构

循环结构程序能够通过条件判断控制循环执行某段语句序列。按照条件和循环执行的语句段之间的关系，可以分为当型循环结构和直到型循环结构。

① 当型循环

当型循环结构的流程图和 N-S 结构图如图 1.10 所示。

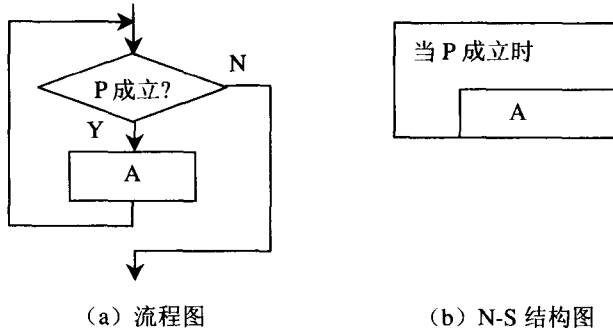


图 1.10 当型循环结构的算法

图 1.10 中 P 是循环控制的判断条件，A 框是循环执行的语句序列。当条件 P 成立时，循环执行 A 框操作。每执行一次 A 就判断一次条件 P。一旦 P 不成立就退出循环，执行后续语句。在当型循环结构中，需要先判断条件 P，然后执行 A 框操作。若一开始 P 就不成立，则 A 框操作一次也不执行。

② 直到型循环

直到型循环结构的流程图和 N-S 结构图如图 1.11 所示。

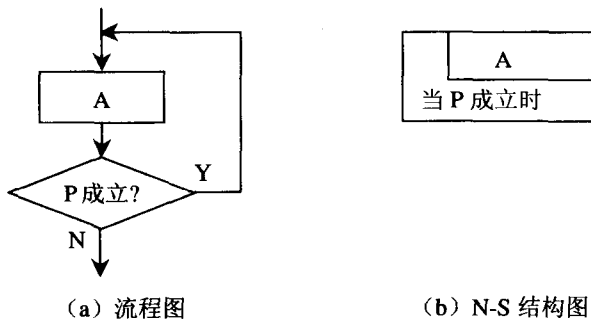


图 1.11 直到型循环结构的算法