

高等学校21世纪教材

GAODENG XUOXIAO 21 SHIJI JIAOCAI

# C语言 程序设计 教程

(修订本)

● 徐士良 编著



人民邮电出版社  
POSTS & TELECOM PRESS

高等学校 21 世纪教材

# C 语言程序设计教程

(修订本)

徐士良 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 / 徐士良编著. —北京: 人民邮电出版社, 2003.8  
高等学校 21 世纪教材  
ISBN 7-115-11357-2

I. C... II. 徐... 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 (2003) 第 048148 号

### 内 容 提 要

本书是作者通过长期教学实践编写成的。全书由浅入深, 介绍 C 语言中的基本概念和语法, 使读者全面地、系统地理解和掌握用 C 语言进行程序设计的方法。主要内容包括: 程序设计基本概念, C 程序的基本组成, 选择结构, 循环结构, 模块设计, 数组, 指针, 结构体与联合体, 文件, 位运算等。本次修订后, 例题和习题更丰富, 叙述更详细, 更有利于读者自学。

本书可作为高等院校各专业的学生学习 C 语言程序设计的教材。

### 高等学校 21 世纪教材 C 语言程序设计教程 (修订本)

◆ 编 著 徐士良  
策划编辑 滑 玉  
责任编辑 须春美

◆ 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>  
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: 17.75  
字数: 427 千字 2003 年 8 月第 2 版  
印数: 30 001 - 32 000 册 2006 年 6 月北京第 10 次印刷

ISBN 7-115-11357-2/TP · 3507

定价: 23.50 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223

## 丛书前言

当今世界，科学技术突飞猛进，知识经济已见端倪，国际竞争日趋激烈。教育在综合国力的形成中处于基础地位，国力的强弱将越来越取决于劳动者的素质，取决于各类人才的质量和数量，这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初，我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体，是进行教学的基本工具，也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此，高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代，计算机教学内容也随之更新，尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施，对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授，精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略，组织编写好这套教材，我们在国内高校做了系统、详细的调查，对教育部制订的教育计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中，我们多次组织了由专家及高校一线教师参加的研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

本套教材有如下特点：

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况，在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。

2. 紧密结合当前技术的新发展，在阐述理论知识的同时侧重实用性。

3. 力求在概念和原理的讲述上严格、准确、精练，理论适中，实例丰富，写作风格上深入浅出，图文并茂，便于学生学习。

4. 为适应当前高校课程种类多、课时数要压缩的数学特点，教材不仅篇幅有很大的压缩，而且均配有电子教案，以满足现代教学新特点的需要，做到易教易学。

5. 所选作者均是国内有丰富教学实践经验的知名专家、教授，所编教材具有较高的权威性。

教育的改革将不会停止，教材也将会不断推陈出新。目前本套教材即将推出，将接受广大教学第一线教师的检验。

由于我们的水平和经验有限，这批教材在编审、出版工作中还存在不少缺点和不足，希望使用本套教材的学校师生和广大读者提出批评和建议，以便改进我们的工作，使教材质量不断提高。

# 编者的话

程序设计是每个科技工作者使用计算机的基本功。C语言是目前使用最广泛的一种程序设计语言，它具有丰富的数据类型，它所提供的数据结构和控制结构适合于进行结构化程序设计，并且，利用C语言可以实现汇编语言的大部分功能，使用灵活，可移植性好。

本书虽然也介绍C语言的语法规则，并对一些细节进行了说明，但本书着重强调的是C语言的基本概念，通过大量的例题分析和算法实例，使读者理解和掌握利用C语言进行程序设计的方法。

全书共分10章，每章后面都有足够的习题供读者练习。

第1章主要介绍了程序设计的基本概念，程序设计的基本过程，以及程序设计的基本方法。通过这一章的学习，使读者对程序设计有一个全面、正确的理解。

第2章主要介绍C程序的基本组成。这一章是从程序的基本功能出发，结合C语言，着重介绍程序中的说明和定义成分，数据的输入与输出功能，以及进行数据处理所涉及到的基本运算。通过这一章的学习，使读者了解和掌握C语言中的基本数据类型，基本掌握了实现顺序程序设计的基本运算和相应的基本语句，初步具备了处理简单问题的程序设计能力。

第3章主要介绍选择结构的程序设计，利用C语言所提供的语句，实现两路分支选择结构和多路分支选择结构。通过这一章的学习，使读者熟练掌握选择结构程序设计的方法。

第4章主要介绍循环结构的程序设计，利用C语言所提供的语句，实现各种形式的循环结构。通过这一章的学习，使读者熟练掌握循环结构程序设计的方法。

第5章主要介绍如何利用C语言中的函数实现模块设计。通过这一章的学习，掌握C程序中各函数之间数据传递的基本途径以及数据的存储类别，掌握函数调用的正确方法。

第6章主要介绍数组在程序设计中的应用。通过这一章的学习，掌握数组在计算机中的存储方式以及用数组作为函数参数时的程序设计方法。

第7章主要介绍C语言中的指针运算。通过这一章的学习，学会正确使用指针进行程序设计，特别是掌握用指针作为函数参数实现数据的双向传递。

第8章主要介绍结构体、联合体以及枚举类型变量的定义和使用。通过这一章的学习，主要掌握用结构体类型实现表格设计的基本方法。

第9章主要介绍C语言缓冲文件系统中对文件的基本操作。通过这一章的学习，主要掌握对文件的打开与关闭，对文件的读写，以及文件的定位。

第10章主要介绍C语言中对二进制位的运算以及对位段操作。通过这一章的学习，主要掌握各种位运算符的正确使用以及对位段的基本操作。

本书的特点是简明扼要，通俗易懂，例题丰富。与本书配套的还有《C程序设计题解与实验指导》一书。

由于作者水平有限，书中难免存在错误和不妥之处，恳请读者批评指正。

编 者  
2001年4月

# 修订本前言

《C 语言程序设计教程》出版后，经过一年多的试用，得到了各院校广大教师和读者的肯定，并提出了不少宝贵的意见，在此向他们表示感谢。

根据大多数教师和读者的意见，在该书第一版的基础上，本修订本作了如下几个方面的改进。

(1) 部分内容作了适当调整，同时也增加了部分内容。例如，增加了“编译预处理”一节，将它放到“模块设计”这一章中。又如，增加了有关链表的基本运算等。

(2) 在第一版中，有些内容的叙述过于简练，读者自学有一定困难，在这次修订中，对这方面的内容作了详细的讲解。

(3) 适当增加了一些例题。

(4) 适当增加了一些习题。

总之，通过这次修订，使本书例题和习题更丰富，叙述更详细，更有利于读者自学。由于作者水平有限，书中难免还会存在错误和不妥之处，再次恳请读者批评指正。

编 者  
2003 年 3 月

# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 程序设计语言的发展 .....	1
1.2 程序设计的基本过程 .....	4
1.3 程序设计的基本方法 .....	13
1.3.1 结构化程序设计 .....	13
1.3.2 模块化程序设计 .....	15
1.3.3 自顶向下、逐步细化的设计过程 .....	16
习题 1 .....	18
<b>第 2 章 C 程序的基本组成</b> .....	19
2.1 简单的 C 语言程序 .....	19
2.2 基本数据类型变量的定义与宏定义 .....	22
2.2.1 整型常量及其变量的定义 .....	22
2.2.2 实型常量及其变量的定义 .....	27
2.2.3 字符型常量及其变量的定义 .....	28
2.3 数据的输入与输出 .....	31
2.3.1 格式输出函数 .....	31
2.3.2 格式输入函数 .....	33
2.3.3 字符输出函数 .....	36
2.3.4 字符输入函数 .....	36
2.4 数据的处理 .....	37
2.4.1 赋值运算 .....	37
2.4.2 算术运算 .....	37
2.4.3 关系运算 .....	39
2.4.4 逻辑运算 .....	40
2.4.5 增 1 与减 1 运算符 .....	43
2.4.6 sizeof 运算符 .....	43
2.4.7 逗号运算符 .....	44
2.4.8 标准函数 .....	45
2.5 宏定义 .....	45
2.5.1 符号常量定义 .....	46
2.5.2 带参数的宏定义 .....	47
习题 2 .....	51



<b>第 3 章 选择结构</b> .....	55
3.1 语句与复合语句 .....	55
3.2 if 语句 .....	58
3.3 if...else 结构 .....	61
3.4 条件运算符 .....	66
3.5 switch 结构 .....	68
3.6 程序举例 .....	74
习题 3 .....	81
<b>第 4 章 循环结构</b> .....	83
4.1 当型循环与直到型循环 .....	83
4.1.1 当型循环结构 .....	83
4.1.2 直到型循环结构 .....	86
4.1.3 当型循环结构与直到型循环结构的区别与联系 .....	87
4.2 for 循环 .....	89
4.3 循环的嵌套与其他有关语句 .....	92
4.3.1 循环的嵌套 .....	92
4.3.2 break 语句 .....	95
4.3.3 continue 语句 .....	97
4.4 程序举例 .....	99
4.4.1 列举算法 .....	99
4.4.2 密码问题 .....	103
4.4.3 对分法求方程实根 .....	105
4.4.4 迭代法求方程实根 .....	107
4.4.5 牛顿法求方程实根 .....	108
4.4.6 梯形法求定积分 .....	110
习题 4 .....	112
<b>第 5 章 模块设计</b> .....	114
5.1 模块的实现——函数 .....	114
5.1.1 函数的概念 .....	114
5.1.2 函数的定义 .....	116
5.1.3 函数的调用 .....	118
5.2 模块间的参数传递 .....	120
5.2.1 形参与实参的结合方式 .....	120
5.2.2 局部变量与全局变量 .....	122
5.2.3 动态存储变量与静态存储变量 .....	123
5.2.4 内部函数与外部函数 .....	129
5.3 模块的递归调用 .....	131
5.4 程序举例 .....	133
5.5 编译预处理 .....	138

5.5.1	文件包含命令 .....	138
5.5.2	条件编译命令 .....	140
习题 5	.....	146
<b>第 6 章</b>	<b>数组</b> .....	<b>147</b>
6.1	一维数组 .....	147
6.1.1	一维数组的定义与引用 .....	147
6.1.2	一维数组的初始化 .....	150
6.2	二维数组 .....	153
6.2.1	二维数组的定义与引用 .....	153
6.2.2	二维数组的初始化 .....	153
6.3	字符数组与字符串 .....	155
6.3.1	字符数组的定义与初始化 .....	155
6.3.2	字符串 .....	156
6.3.3	字符数组与字符串的输入与输出 .....	157
6.3.4	字符串处理函数 .....	160
6.4	数组作为函数参数 .....	162
6.4.1	形参数组与实参数组的结合 .....	162
6.4.2	二维数组作为函数参数 .....	164
6.5	程序举例 .....	166
习题 6	.....	171
<b>第 7 章</b>	<b>指针</b> .....	<b>172</b>
7.1	指针的基本概念 .....	172
7.2	指针变量 .....	173
7.2.1	指针变量的定义与引用 .....	173
7.2.2	指针变量作为函数参数 .....	175
7.3	数组与指针 .....	177
7.3.1	数组的指针与数组元素的指针 .....	177
7.3.2	数组指针作为函数参数 .....	179
7.3.3	多维数组与指针 .....	181
7.4	字符串与指针 .....	183
7.4.1	字符串指针 .....	183
7.4.2	字符串指针作为函数参数 .....	187
7.5	指针数组与指向指针的指针 .....	189
7.5.1	指针数组的概念 .....	189
7.5.2	指向指针的指针 .....	191
7.5.3	main 函数的形参 .....	192
7.6	函数与指针 .....	193
7.6.1	用函数指针变量调用函数 .....	193
7.6.2	函数指针变量作为函数参数 .....	195

7.7 程序举例 .....	196
习题 7 .....	200
<b>第 8 章 结构体与联合体 .....</b>	<b>202</b>
8.1 结构体类型变量 .....	202
8.1.1 结构体类型变量的定义 .....	202
8.1.2 结构体类型变量的引用 .....	204
8.1.3 结构体的嵌套 .....	205
8.1.4 结构体类型变量的初始化 .....	205
8.1.5 结构体与函数 .....	207
8.2 结构体数组 .....	210
8.2.1 结构体数组的定义与引用 .....	210
8.2.2 结构体数组作为函数参数 .....	213
8.3 结构体与指针 .....	215
8.3.1 结构体类型指针变量的定义与引用 .....	215
8.3.2 结构体类型指针作为函数参数 .....	216
8.4 链表 .....	220
8.4.1 链表的基本概念 .....	220
8.4.2 链表的基本运算 .....	222
8.5 联合体 .....	226
8.6 枚举类型与自定义类型名 .....	227
8.6.1 枚举类型 .....	227
8.6.2 自定义类型名 .....	230
8.7 程序举例 .....	231
习题 8 .....	239
<b>第 9 章 文件 .....</b>	<b>240</b>
9.1 文件的概念 .....	240
9.1.1 文本文件与二进制文件 .....	240
9.1.2 缓冲文件系统 .....	240
9.1.3 文件类型指针 .....	241
9.2 文件的打开与关闭 .....	241
9.2.1 文件的打开 .....	241
9.2.2 文件的关闭 .....	242
9.3 文件的读写 .....	242
9.3.1 文件读函数 .....	242
9.3.2 文件写函数 .....	244
9.4 文件的定位 .....	246
9.5 程序举例 .....	247
习题 9 .....	249
<b>第 10 章 位运算 .....</b>	<b>251</b>

10.1 二进制位运算 .....	251
10.2 位段 .....	256
10.3 程序举例 .....	258
习题 10 .....	260
<b>附录 1 基本 ASCII 码表 .....</b>	<b>261</b>
<b>附录 2 Turbo C 常用库函数 .....</b>	<b>264</b>
<b>参考文献 .....</b>	<b>270</b>

# 第 1 章 绪 论

## 1.1 程序设计语言的发展

人们要利用计算机解决实际问题，特别是对于数值计算问题，一般总是要编制程序。所谓程序，是指用某种程序设计语言为工具编制出来的动作序列，它表达了人们解决问题的思路，也反映了需要计算机所作的一系列操作。程序设计语言就是用户用来编写程序的语言，它是人与计算机之间交换信息的工具。

程序设计语言一般分为机器语言、汇编语言和高级语言三大类。

### 1. 机器语言

对于计算机来说，一组机器指令就是程序，称为机器语言程序。

机器语言是最底层的计算机语言。用机器语言编写的程序，计算机硬件可以直接识别。在用机器语言编写的程序中，每一条机器指令都是二进制形式的指令代码。在指令代码中一般包括操作码和地址码，其中操作码告诉计算机作何种操作，地址码则指出被操作的对象。对于不同的计算机硬件（主要是 CPU），其指令系统是不同的，因此，针对一种计算机所编写的机器语言程序不能在另一种计算机上运行。由于机器语言程序是直接针对计算机硬件的，因此它的执行效率比较高，能充分发挥计算机的速度性能。但是，用机器语言编写程序的难度比较大，容易出错，而且程序的直观性比较差，也不容易移植。

### 2. 汇编语言

为了便于理解与记忆，人们采用能帮助记忆的英文缩写符号（称为指令助记符）来代替机器语言指令代码中的操作码，用地址符号来代替地址码。用指令助记符及地址符号书写的指令称为汇编指令（也称符号指令），而用汇编指令编写的程序称为汇编语言源程序。汇编语言又称符号语言。

汇编语言与机器语言一般是一一对应的，因此，汇编语言也是与具体使用的计算机有关的。由于汇编语言采用了助记符，因此，它比机器语言直观，容易理解和记忆，用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。

例如，为了计算表达式“3+2”的值，用汇编语言编写的程序与用机器语言（8086CPU 的指令系统）编写的程序如下：

PUSH BP	01010101	
MOVE BP, SP	10001011	11101100
DEC SP	01001100	
DEC SP	01001100	
PUSH SI	01010110	
PUSH DI	01010111	
MOVE DI, 0003	10111111	00000011 00000000
MOVE SI, 0002	10111110	00000010 00000000
MOVE AX, DI	10001011	11000111
ADD AX, SI	00000011	11000110
MOVE [BP-02], AX	10001001	01000110 11111110
POP DI	01011111	
POP SI	01011110	
MOVE SP, BP	10001011	11100101
POP BP	01011110	
RET	11000011	

其中每一行的前半部分为汇编语言指令，后半部分（二进制形式的指令代码）为对应的机器语言指令。

需要指出的是，计算机不能直接识别用汇编语言编写的程序，必须由一种专门的翻译程序将汇编语言源程序翻译成机器语言程序后，计算机才能识别并执行。这种翻译的过程称为“汇编”，负责翻译的程序称为汇编程序。

### 3. 高级语言

机器语言和汇编语言都是面向机器的语言，一般称为低级语言。低级语言对机器的依赖性太大，用它们开发的程序通用性很差，普通的计算机用户也很难胜任这一工作。

在保证程序正确的前提下，程序设计的主要目标是程序的可读性、易维护性和可移植性。为使程序的可读性好，不仅要求编程者具有良好的编程风格，还要求所使用的编程语言具有易懂的语句。所谓程序易维护，是指当程序的功能需要修改或增强时，所需要的开销应尽量的小。一个可移植性好的程序，应该在各种计算机和操作环境中都能运行，并得到同样的结果。显然，前面所提到的机器语言程序与汇编语言程序很难达到这样的目标。

随着计算机技术的发展以及计算机应用领域的不断扩大，计算机用户的队伍也在不断壮大。为了使广大的计算机用户也能胜任程序的开发工作，从 20 世纪 50 年代中期开始逐步发展了面向问题的程序设计语言，称为高级语言。高级语言与具体的计算机硬件无关，其表达方式接近于被描述的问题，易为人们接受和掌握。用高级语言编写程序要比低级语言容易得多，并大大简化了程序的编制和调试，使编程效率得到大幅度的提高。高级语言的显著特点是独立于具体的计算机硬件，通用性和可移植性好。

目前，计算机高级语言已有上百种之多，得到广泛应用的有十几种，并且，几乎每一种高级语言都有其最适用的领域。表 1.1 列出了几种最常用的高级语言及其最适用的领域。

表 1.1

语言名称	适用范围
Basic	教学和小型应用程序的开发
Fortran	科学及工程计算程序的开发
Pascal	专业教学和应用程序的开发
C	中、小型系统程序的开发
COBOL	商业与管理应用程序的开发
dBASE	数据库管理程序的开发
FoxBASE	数据库管理程序的开发
C++	面向对象程序的开发
LISP	人工智能程序的开发
PROLOG	人工智能程序的开发
Java	面向对象程序的开发

为了计算表达式“3+2”的值，如果使用高级语言来编程就简单得多。

例如，用 Basic 语言编写的程序为

```
10 I=3
20 J=2
30 K=I+J
```

又如，用 Pascal 语言编写的程序为

```
Program Addit
var
  i, j, k: integer;
begin
  i:=3;
  j:=2;
  k:=i+j;
end
```

再如，用 C 语言编写的程序为

```
main()
{ int i, j, k;
  i=3;
  j=2;
  k=i+j;
}
```

由上述例子可以看出，程序设计语言越低级，就越靠近计算机硬件，其描述的程序就越复杂，其中的每一条指令（或语句）也就越难懂。反之，程序设计语言越高级，就越靠近人的表达与思维，其描述的程序就越简单，其中的每一条语句也就越容易理解。

必须指出，用任何一种高级语言编写的程序（称为源程序）都要通过编译程序翻译成机器语言程序（称为目标程序）后计算机才能执行，或者通过解释程序边解释边执行。

从程序设计语言的发展过程可以看出，程序设计语言将越来越接近人的自然语言。

## 1.2 程序设计的基本过程

什么是程序设计？对于初学计算机的人来说，往往简单地把它理解为编制一个程序。其实这是不对的，至少是不全面的。实际上，程序设计包括多方面的内容，而具体编制程序只是其中的一个方面。有人将程序设计描述成如下的一个公式：

程序设计 = 算法 + 数据结构 + 方法 + 工具

由此可以看出，在整个程序设计的过程中，要涉及到算法的设计、数据结构的设计、方法的设计和选择工具的选择等诸多方面。从这个概念出发，一般来说，可以将程序设计的过程分为以下 5 个基本步骤：

- ① 问题的分析；
- ② 结构特性的设计；
- ③ 算法的设计；
- ④ 流程的描述；
- ⑤ 调试与运行。

下面分别对这 5 个步骤作简要的说明。

### 1. 问题分析

问题分析是进行程序设计的基础。如果在没有把所要解决的问题分析清楚之前就想着手编制程序，是很难得到预想结果的，这只能起到事倍功半的效果。根据所要解决的问题性质与类型，需要分析的内容可能是不同的，但作为最基本的分析内容主要有以下几个方面。

#### (1) 问题的性质

人们所要解决的问题是各种各样的，而对于不同性质的问题，所用的方法、工具以及输入输出的形式一般也是不同的。通过对问题性质的分析，进一步确定在解决这个问题过程中要做什么？怎么做？例如，你所要解决的问题是属于数值型还是非数值型的问题；如果是数值型的问题，则要求确定一个合理的精度要求；不管是数值型问题还是非数值型问题，都需要明确最终的结果是什么。又如，对于一元二次方程求根的问题，需要明确是只求实根还是实根与复根都要求。

#### (2) 输入/输出数据

数据处理是计算机应用中最广泛的一个领域。在用计算机解决问题时，一般总要有一些输入数据，计算的结果也要以某种方式进行输出。因此，在进行程序设计的过程中，需要考虑对输入/输出数据的处理。一般来说，对于输入/输出数据主要应考虑以下几个方面：

- 数据的类型是什么？如整型、实型、双精度型、字符型等。
- 在何种设备上输入或输出？



- 采用什么样的格式进行数据的输入或输出？

### (3) 数学模型或常用的方法

对于数值型问题，一般要考虑数学模型的设计，或者要对常用的一些方法进行分析与比较，从而根据问题的性质选择一种合理的解决方案。对于非数值型问题，通常也需要从众多的方法中选择一种最合理的方法。

例如，为了求一元二次方程  $Ax^2+Bx+C=0$  的两个实根  $x_1$  和  $x_2$ ，通常有以下3种方法。

#### ① 求根公式：

$$x_{1,2} = (-B \pm \sqrt{B^2 - 4AC}) / (2A)$$

#### ② 韦达定理：

$$x_1 + x_2 = -B/A$$

$$x_1 x_2 = C/A$$

#### ③ 迭代法。

对于上述3种方法，虽然从理论上讲都可以使用，但与实际应用之间还是有一定的差距。例如，因为计算工具的有效数字位数是有限的，在运算过程中不可避免地会出现误差，因此，利用理论上精确成立的求根公式计算得到的结果也不一定可靠；韦达定理虽然指出了一元二次方程两个实根之间的关系，但没有指明如何实现的具体步骤；迭代法一般一次只能求一个实根，并且还存在着收敛性的问题。因此，在实际解决问题时，必须要对各种方法进行分析比较，不能随便使用某种方法。

## 2. 结构特性的设计

结构特性设计得好坏，直接影响到程序设计的效率，乃至程序执行的效率。结构特性的设计主要包括控制结构和数据结构的设计。

### (1) 控制结构

一个程序的功能不仅取决于所选用的操作，而且还取决于各操作之间的执行顺序，即程序的控制结构。程序的控制结构实际给出了程序的框架，决定了程序中各操作的执行顺序。在程序设计过程中，通常用流程图形象地表示程序的控制结构。常用的流程图有两种：传统流程图与NS结构化流程图（简称NS图）。

1966年，Bohm和Jacopini证明了任何复杂的程序都可以用顺序、选择和循环3种基本结构组合而成。其中选择结构中包括一般的选择结构和多情况选择结构，循环结构又可以分为当型循环和直到型循环两种。这几种基本控制结构的传统流程图如图1.1所示。其中：

- 顺序结构反映了若干个模块之间连续执行的顺序；
- 在选择结构中，由某个条件P的取值来决定执行两个模块之间的哪一个；
- 在当型循环结构中，只有当某个条件成立时才重复执行特定的模块（称为循环体）；
- 在直到型循环结构中，重复执行一个特定的模块，直到某个条件成立时才退出该模块的重复执行；
- 在多情况选择结构中，根据某控制变量的取值来决定选择多个模块中的哪一个。